13-12-2019

Group 24

Smart-Shelf Inventory Management System

IS52018C Software Projects: Project Proposal

Christopher Mayes	cmaye001@gold.ac.uk
Elhassania Ghazoini	eghaz001@gold.ac.uk
Francis Aquino	faqui001@gold.ac.uk
Jignesh Devji	jdevj001@gold.ac.uk
Jasper Robinson	jrobi001@gold.ac.uk
Marisa Doyne	mdoyn010@gold.ac.uk

Contents

Concept Introduction:

Inventory management is performed everywhere, from individual shoppers wondering if they need more kitchen roll to aerospace companies keeping track of thousands of high-grade components.

In the home managing inventory can feel extraneous, the consequences of poor systems are low. In large institutions however inventory management is often of paramount importance. Poor management causes inefficiencies and leads to losses. Companies like Amazon and Boeing have developed incredibly complex and effective systems to keep track of items.

In the middle ground between the inventory needs of homes and large companies stock management is often viewed as both essential and an inconvenience, one that keeps managers from the other work they want to achieve.

We are proposing an all in one inventory system targeted towards medium sized stock environments. Our goal is to reduce the time and stress associated with inventory management by providing stock information instantly to the user on request. This should eliminate the need for stock checks and the physical updating of stock logs.

Our system is generally comprised of a ‘smart-shelf’ which collects stock data and an application which displays that information to the stock manager. The shelf and application are connected via a database over the internet.

Users can set up shelf spaces to keep track of different items. The application will display the current and past stock levels of the items on this shelf space. Warnings will be sent to the inventory manager if any specific item falls below a set threshold, meaning if set up correctly even stock checks within the app need only be done when prompted.

These features combined should hopefully make our system a compelling solution to the inventory needs of medium sized stock environments: preventing stock shortages and taking many of the pressures of inventory management tasks off the manager.

Stakeholder & User Needs:

Stakeholders:

Consultations with potential home users, via a short questionnaire were performed and market research into existing solutions for large businesses done. Following this it was decided that the primary stakeholders for this concept should be laboratory technicians and managers working in medium sized stock environments, such as: small shops, online businesses, schools and universities.

Consultations with university-based stakeholders and the manager of a small Londis convenience store were arranged and performed using a semi-structured interview methodology. At least two interviewers were present at consultations to ensure responses were recorded accurately. The interviews conducted with the laboratory managers of the Department of Computing Hatch Labs and the Department of Education Science Labs were the most influential in shaping our concept.
Their responses revealed that most inventory management they performed was reactive rather than proactive. Stock checks were generally performed by eye and items only re-ordered when close to, or completely depleted. Records of past and present stock levels were minimal; however, those consulted did have extensive systems for storing re-order and safety information.

The problems identified by stakeholders with current methods were as follows:

- **Risk of items being out of stock:** For shops this can translate to, lost revenue. For labs it can prove disruptive if experiments (and teaching objectives) are dependent on the use of specific items.
- **Time and stress:** Lab managers and business owners have many responsibilities, managing stock and re-ordering items only adds to them. The time spent checking inventory and ensuring levels are high enough for upcoming demand, is time they would rather spend elsewhere.
- **Accurate information on items used or sold was lacking:** Especially important for small businesses, or university labs that both provide and sell certain items. An easy way to gauge revenue from specific items would be useful.

Stakeholder needs:

The key needs and requirements outlined by are relevant stakeholders were as follows:

- System must accurately display stock information.
- Using the shelf must save time over current systems. Setting up items and modifying existing ones must be quick processes.
- Re-ordering information must be available or integrated with their current systems.
- Stock warnings when items run low would be very useful.
- A stock history should be available, showing item usage over time.
- Shelf should accommodate a wide range of item types.
- Statistics that predict when items will run out would be useful.

(See appendix I for expanded details of the interview methodology and personas.)

Prior Knowledge & Market Survey:

Information Gathering:

Initial information gathering was performed through google searches. Trade specific websites were identified and searched for competing solutions. Patent libraries were searched, and academic articles were briefly surveyed to discover competitors and interest in the field.

Competitors and Market Landscape:

Our current focus towards small labs and businesses has few direct competitors.

The closest competing solution found was provided by 'Mettler Toledo'. They sell all in one smart shelf standard racking systems as well as weighing pads that can be integrated into existing racking systems (Meter Toledo, 2019). Mettler Toledo's solutions are similar in concept to ours, but their target customers are large companies. They specifically target the automotive industries, as well as large research labs. Mettler Toledo provide little information on how stock information is presented to their customers, which is a key area of our focus.

Market Research and Product development:

Our original concept was a smart shelf for home use. The shelf would allow users to keep track of essential items and set up automated systems to re-order them before they ran out. Despite the current interest in smart-devices searches brought up no competitors in this space. This combined with hesitant enthusiasm from the shoppers we surveyed *(see appendix I)* dissuaded us from pursuing this avenue further.

Research into supermarket applications was then performed. There is significant interest in this field (Intel Labs and The Store WPP, n.d.), for three main reasons:

1. Significant losses made by supermarkets due to items being out of stock.
2. Reducing the need for staff to perform inventory management will increase profits.
3. Data on what time and days items sell is valuable both for predictive restocking and to advertisers.

Many competitors were found in the supermarket space. Few used pressure sensors alone to gather stock information. Most used a combination of load cells and Image recognition (Shekelbrainweigh.com, 2019) which required many cameras throughout the shop (potentially raising privacy concerns). Some used light sensors on the shelves to give estimates of shelf occupancy (Wiseshelf.com, 2019). Many store shelf inventory patents also existed, complicating development (Wistron NeWeb Corp, DJB GROUP LLC, 2013) (CLEAR-VIEW-TECHNOLOGIES Inc, 2009).

While supermarkets were a valid application for our concept it was decided the competitive landscape was too crowded. The technical requirements for integration with current supermarket shelving combined with the need to be scalable to hundreds of shelves across a store were also beyond our feasibility estimates for the product development period.

Design:

Before the design process began, stakeholders were consulted to narrow down the large set of suggested features collected so far. Only the most essential features were modelled. Useful, but non-essential features were kept in a master list for future reference.

Six key interactions were identified from the essential features: setting up items, removing items, modifying items, displaying stock levels, searching for items and receiving stock warnings.

After conceiving a basic system configuration (see figure 15 page 17 for an updated version) the key interactions were planned out as use cases and scenarios.

These use cases and scenarios were then modelled into a series of UML Use case, sequence and flow diagrams. These diagrams model both the user logic while performing tasks and a brief overview of how the system responds *(see figures 1-8 below and on the following pages).*

Smart Inventory System Use Cases

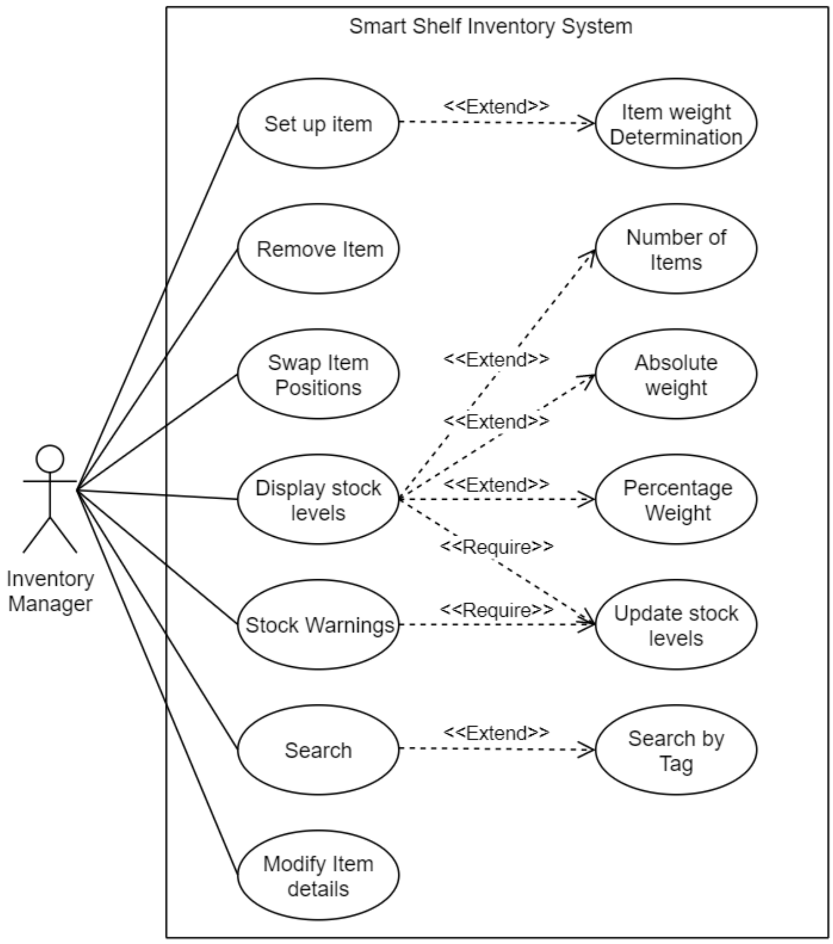


Figure 1: UML Use case diagram for our proposed smart shelf inventory system. Displays interactions required by the system. <<Extend>> items are non-requisite extended functions. <<Require>> items are functionalities required for the linked use case to function

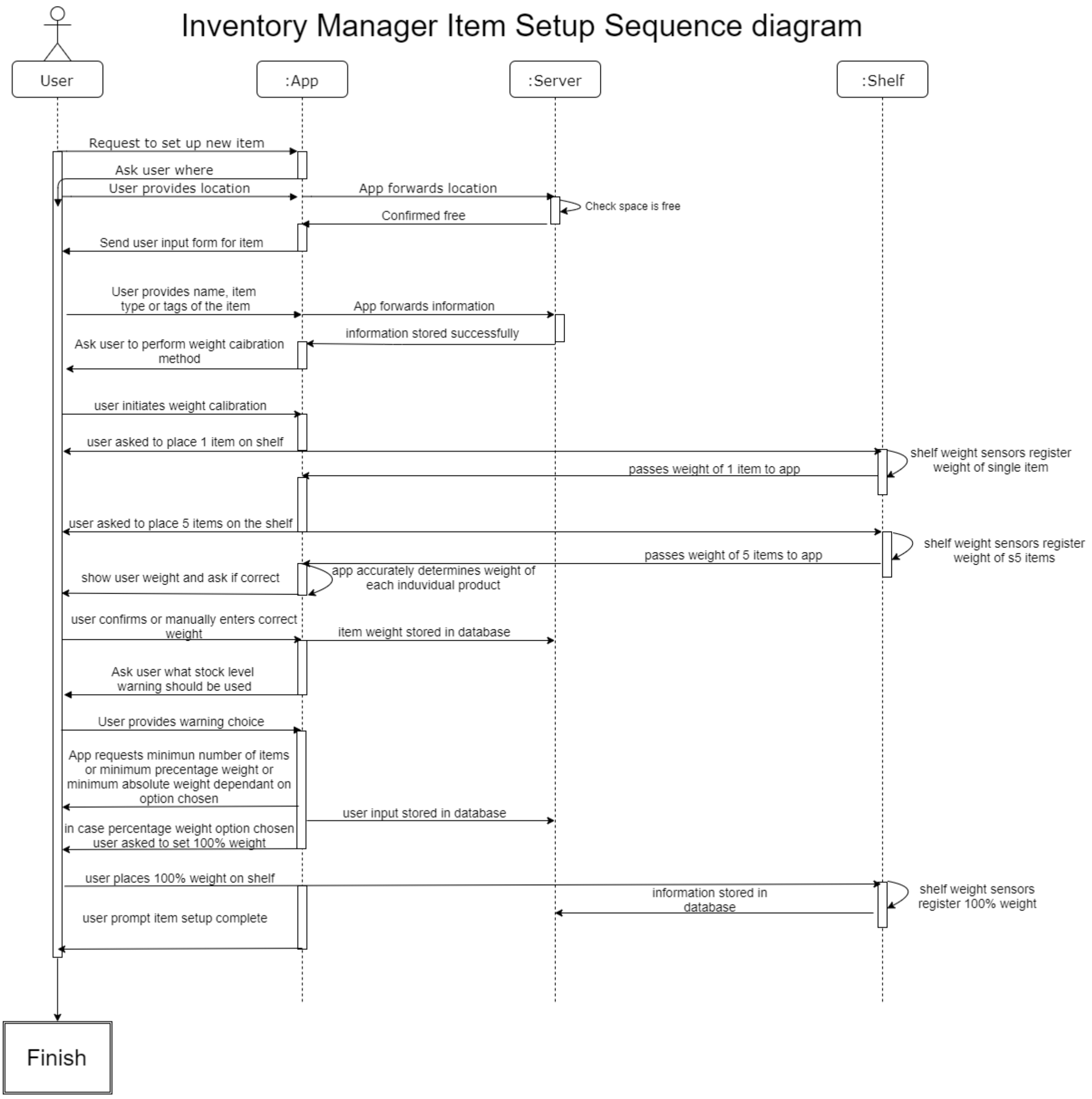


Figure 2: Sequence diagram showing the interactions of a shelf owner with the system when setting up an item. Also shown are the paths information takes through the major system components in the same process.

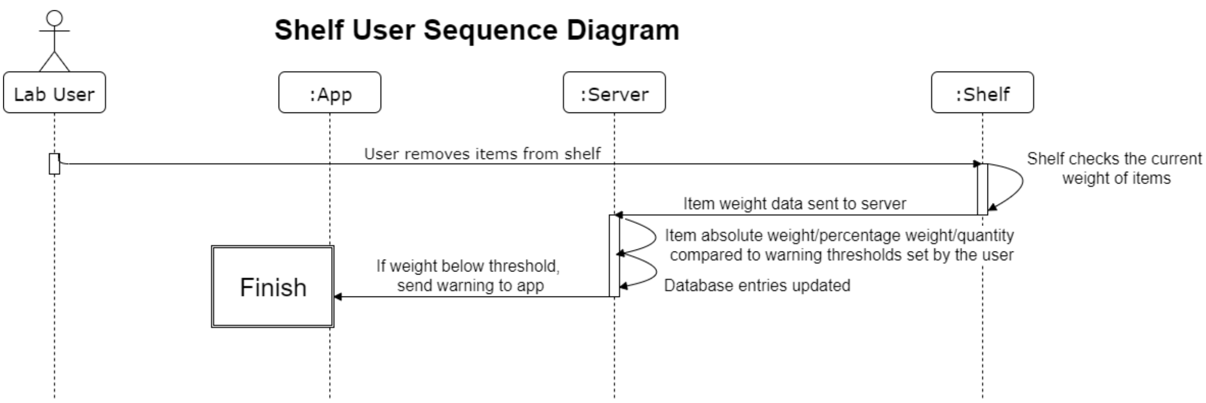


Figure 3: Sequence diagram showing the interactions of a normal user with the shelf, their interaction will generally be limited to taking items off the shelf. This diagram shows how the system responds.

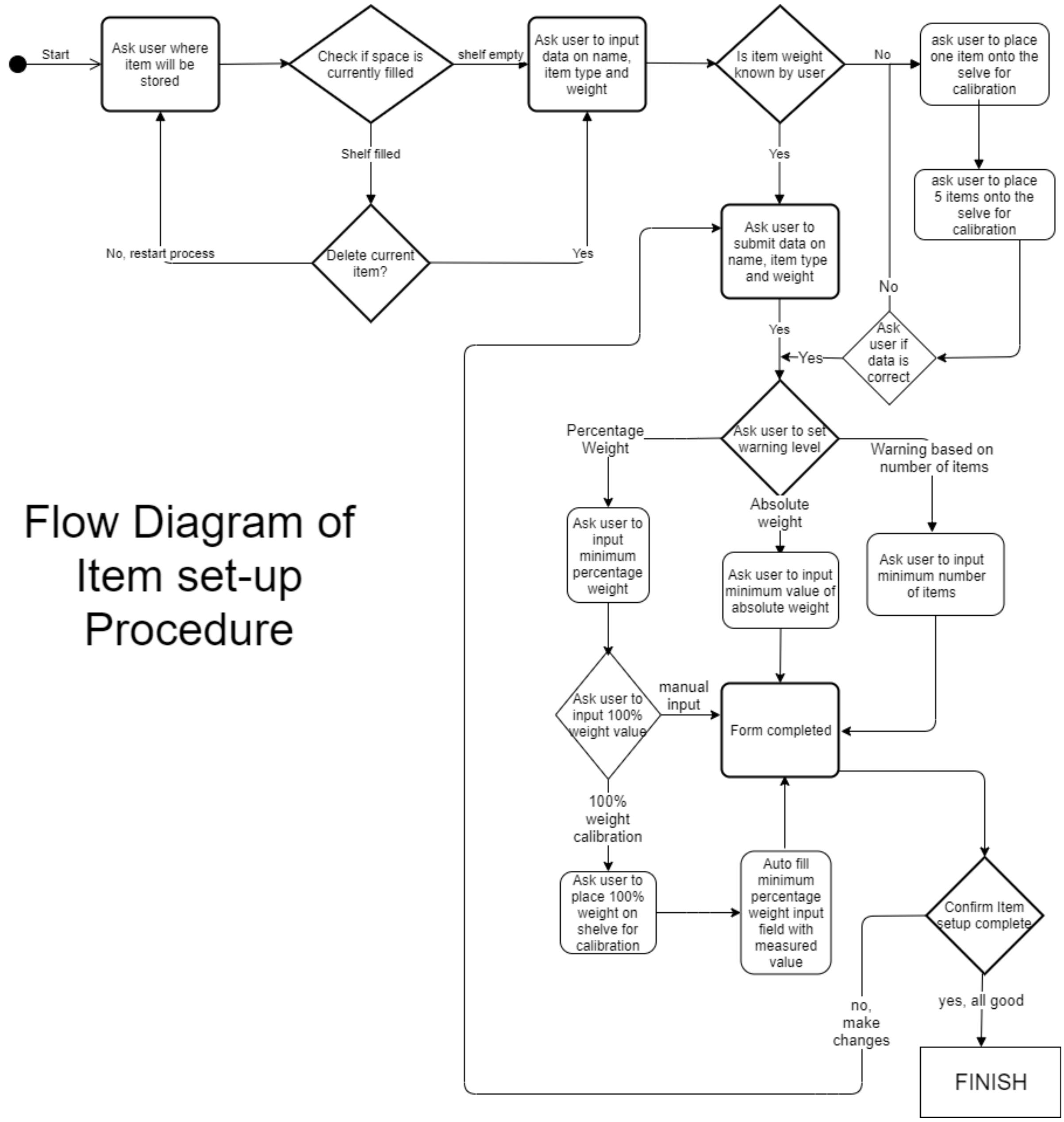


Figure 4: Flow diagram showing the various paths a user can take during the item setup process. Decisions are presented in diamonds, with system prompts and responses in squares.

Flow Diagram: Shelf 100% automatic weight calibration process

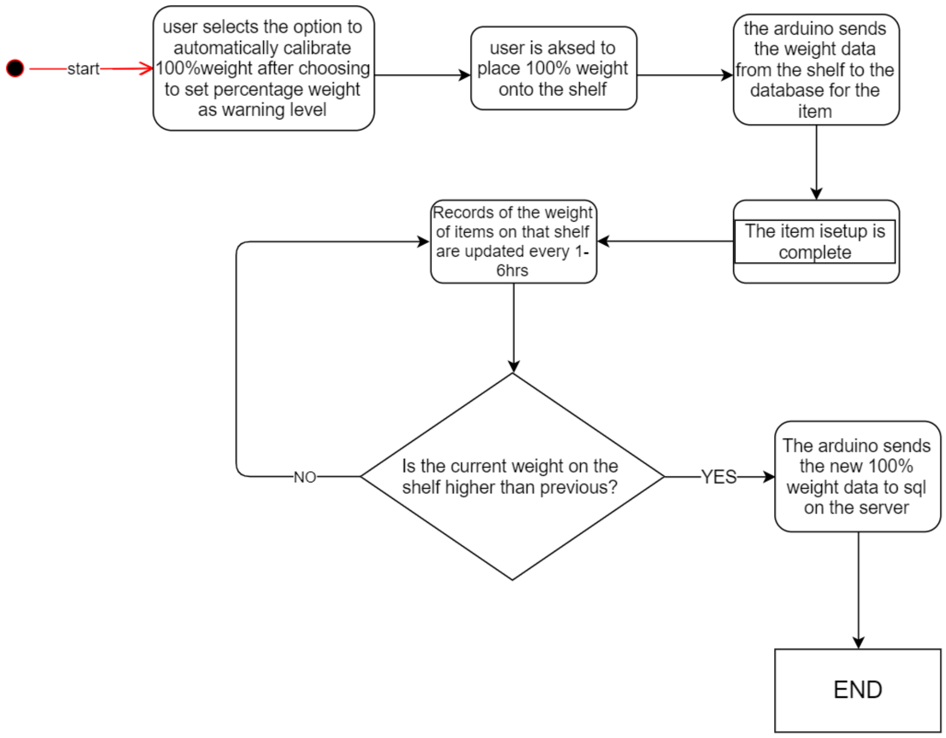


Figure 5: Flow diagram showing the system to automatically recalibrate a shelf to 100% following a higher weight being placed on the shelf than has appeared in recent history.

Flow diagram: Editing Item Details

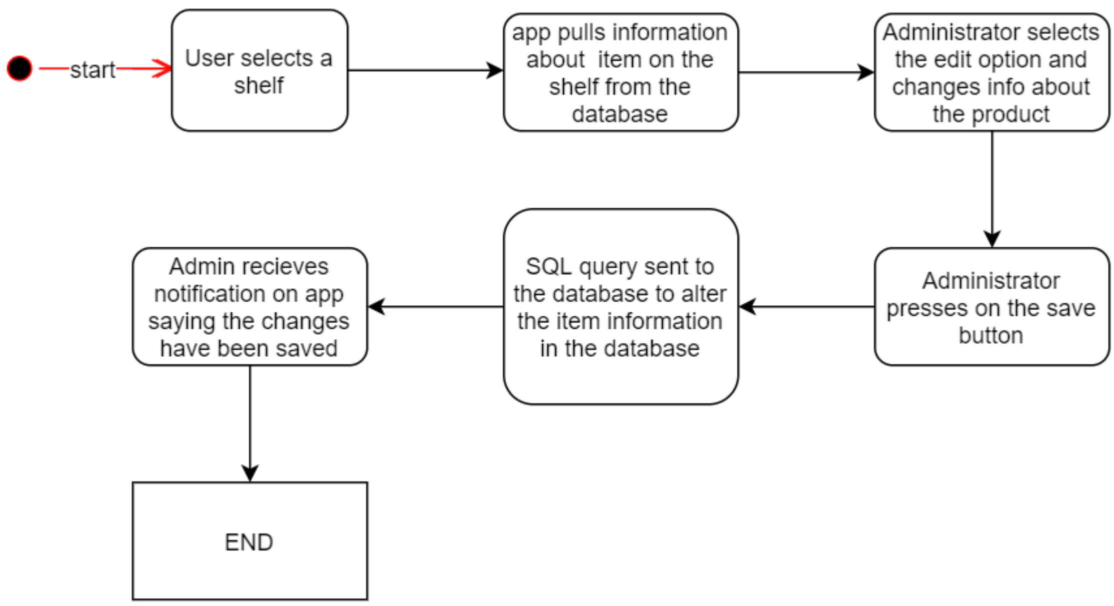


Figure 6: Flow diagram showing the processes occurring when editing of item data after an item has been set up.

Flow Diagram of Swapping Shelf Positions

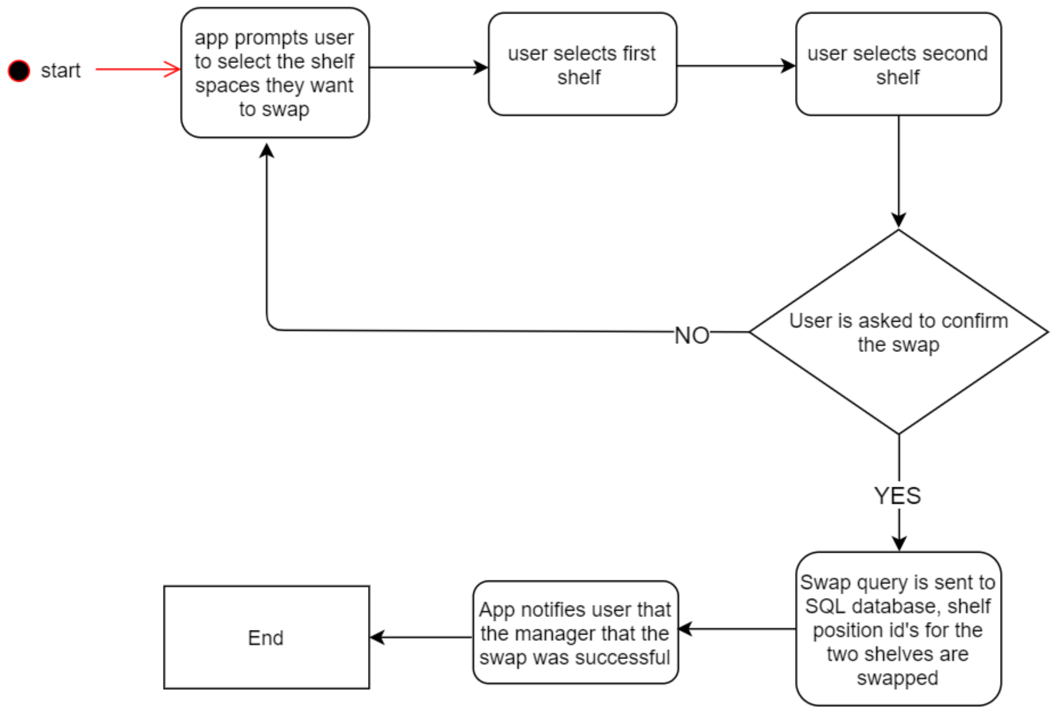


Figure 7: Flow diagram showing the process of swapping two shelf positions around. Used when a user wishes to re-organise their shelves without losing data.

Flow Diagram: Search by Name or Tag

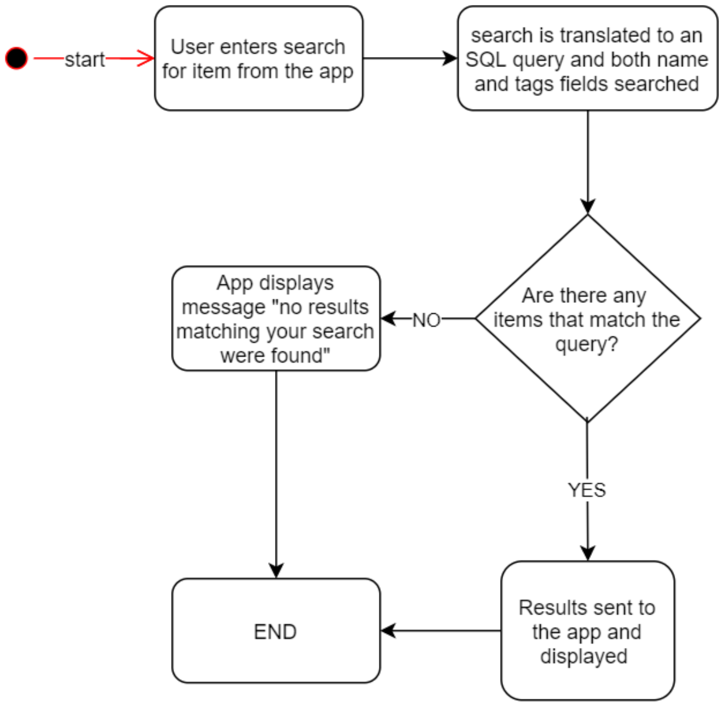


Figure 8: Flow diagram showing the processes related to searching items either by name, or tags. Tags can be applied to many items of a similar type e.g. resistors.

Prototyping:

Prototypes were presented to stakeholders in semi-structured interviews in which we explained their context before allowing the stakeholder to interpret them and provide feedback.

Lo-fi Prototyping:

A storyboard version of the item setup process was created based on the UML flow diagram *(figure 4)*, showing the steps of the item setup process *(figure 9 below)*.

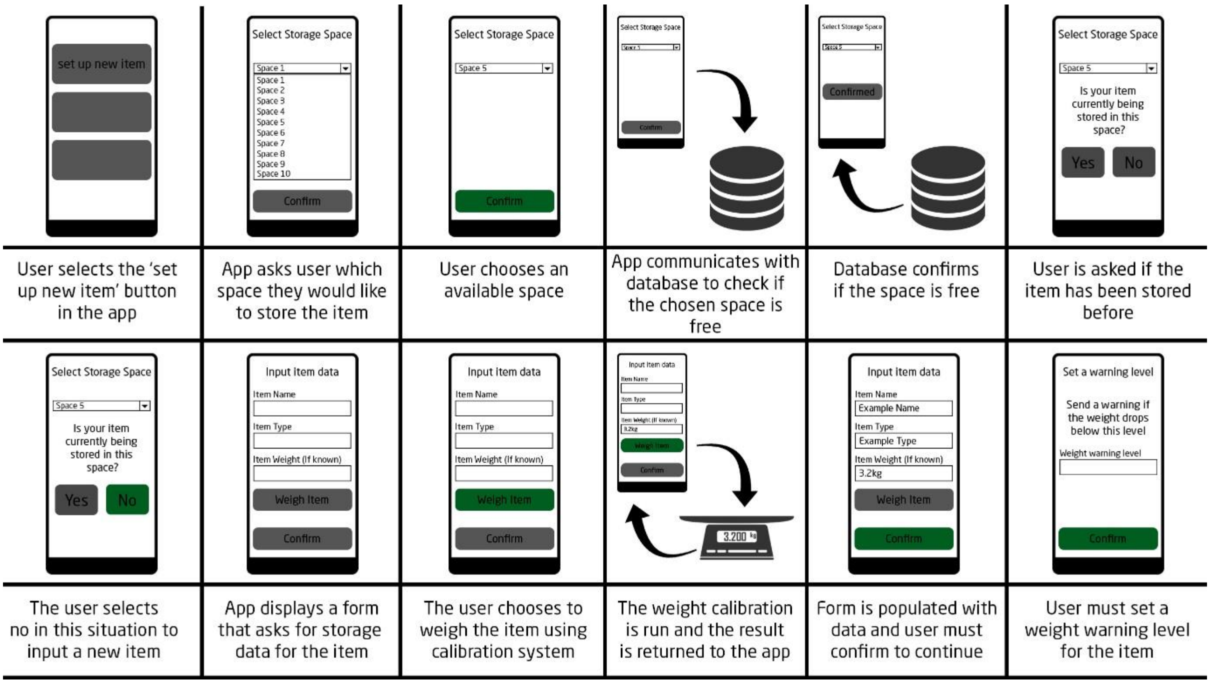


Figure 9: A storyboard version of the Item setup process visually presenting the process mapped out in figure 4.

Feedback on the storyboard was positive. Questions were raised about how the weight calibration system and weight warning systems would work.

Lo-fi of mock-ups of the home screen were made. Processes for the weight calibration and weight warning systems were drafted *(see figures 10, 11, 12 on the following pages)*.

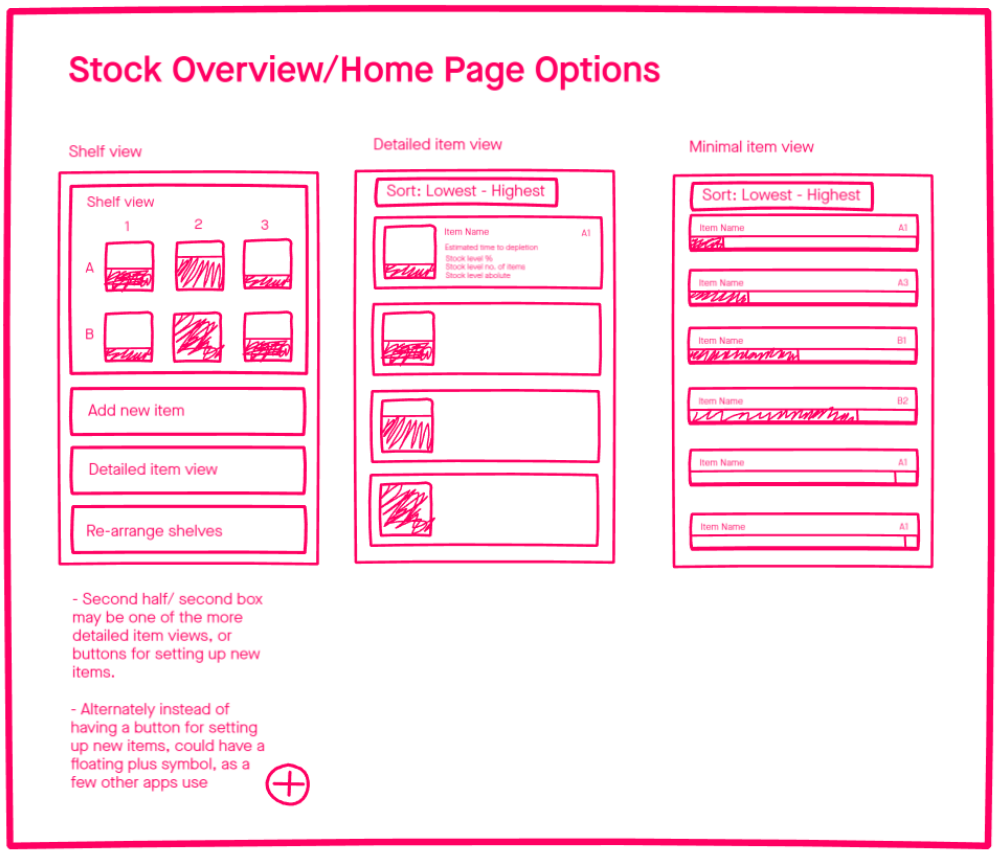


Figure 10: A lo-fi mockup of three alternative ways to present stock information on the home screen.

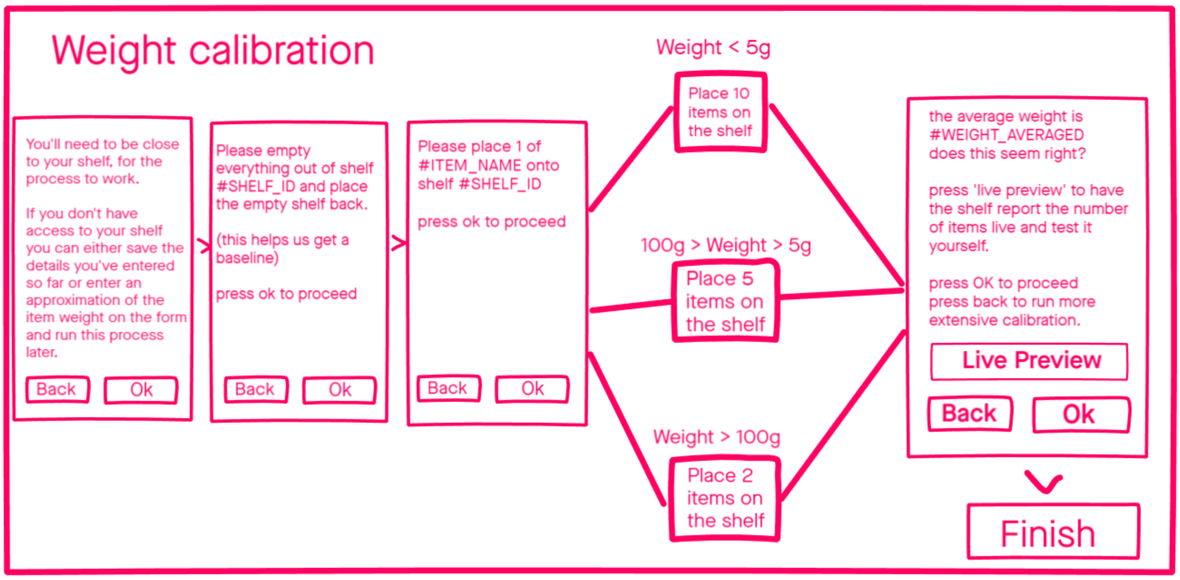


Figure 11: A lo-fi mock-up of the weight calibration process. The system should weigh the initial item placed, then request a sensible number of items be placed to ensure an accurate average can be obtained in as few steps as possible.

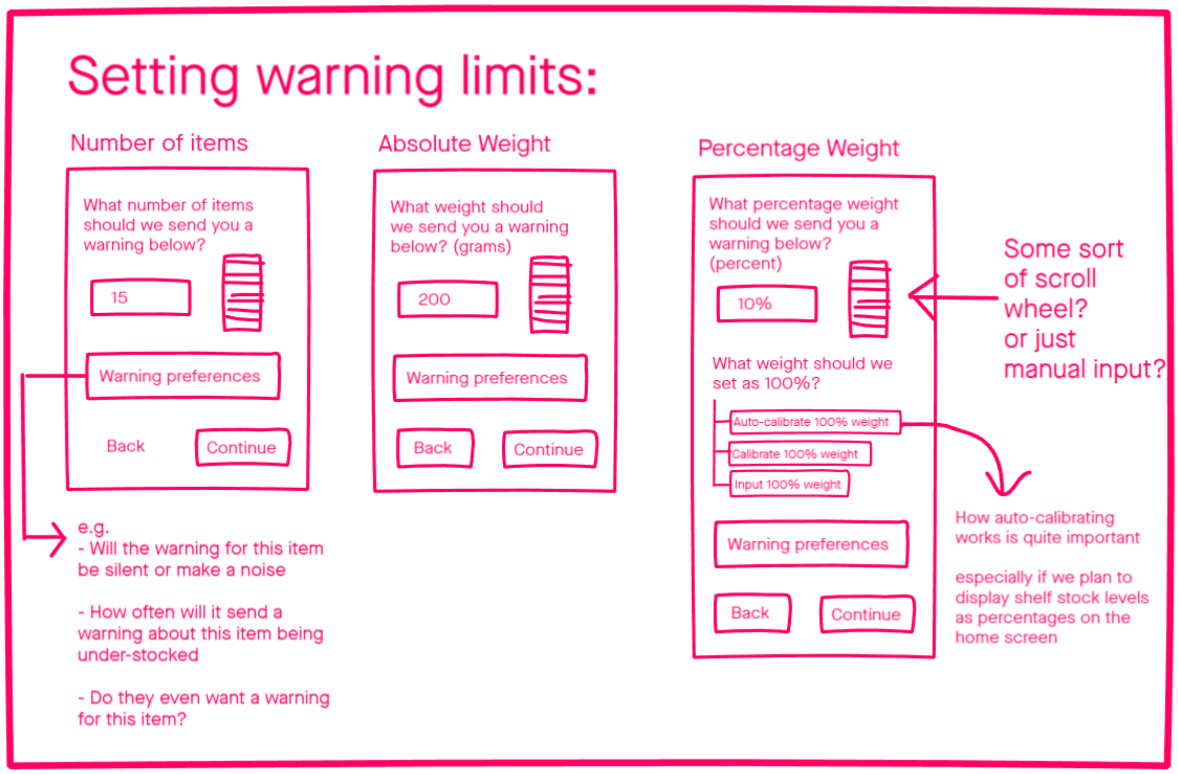


Figure 12: Lo-fi mock-up of the three screens for setting up warning limits: Number of items, Absolute weight and percentage weight.

Feedback was positive. Stakeholders liked all three home screen layouts and requested that there should be some way to change the display of stock information in the final product. There was a preference for item warnings to be on a percentage weight basis, but generally they liked the choice of all three options (see figure 12).

Functional Hi-fi Prototype

A functional prototype which simulated the item setup process was created in Invision studio, so that stakeholders could identify any missing steps and suggest changes (see figures 13, 14).

Feedback was positive with stakeholders saying the item setup process was well thought through. Presenting the prototype in a simulated way brought useful suggestions that had not come up before. The most major suggestions were:

- Preference for a web or desktop application over a mobile one, stakeholders see themselves using the application mainly when re-ordering items from suppliers on the computer.
- Form Improvements, with optional fields for price information and a large field to store re-stocking information and links to vendors.
- Ability to arrange the shelves in custom layouts.
- Different form layouts for storing different types of items.
    - Example: Hazardous chemicals should have large fields for storing safety information.

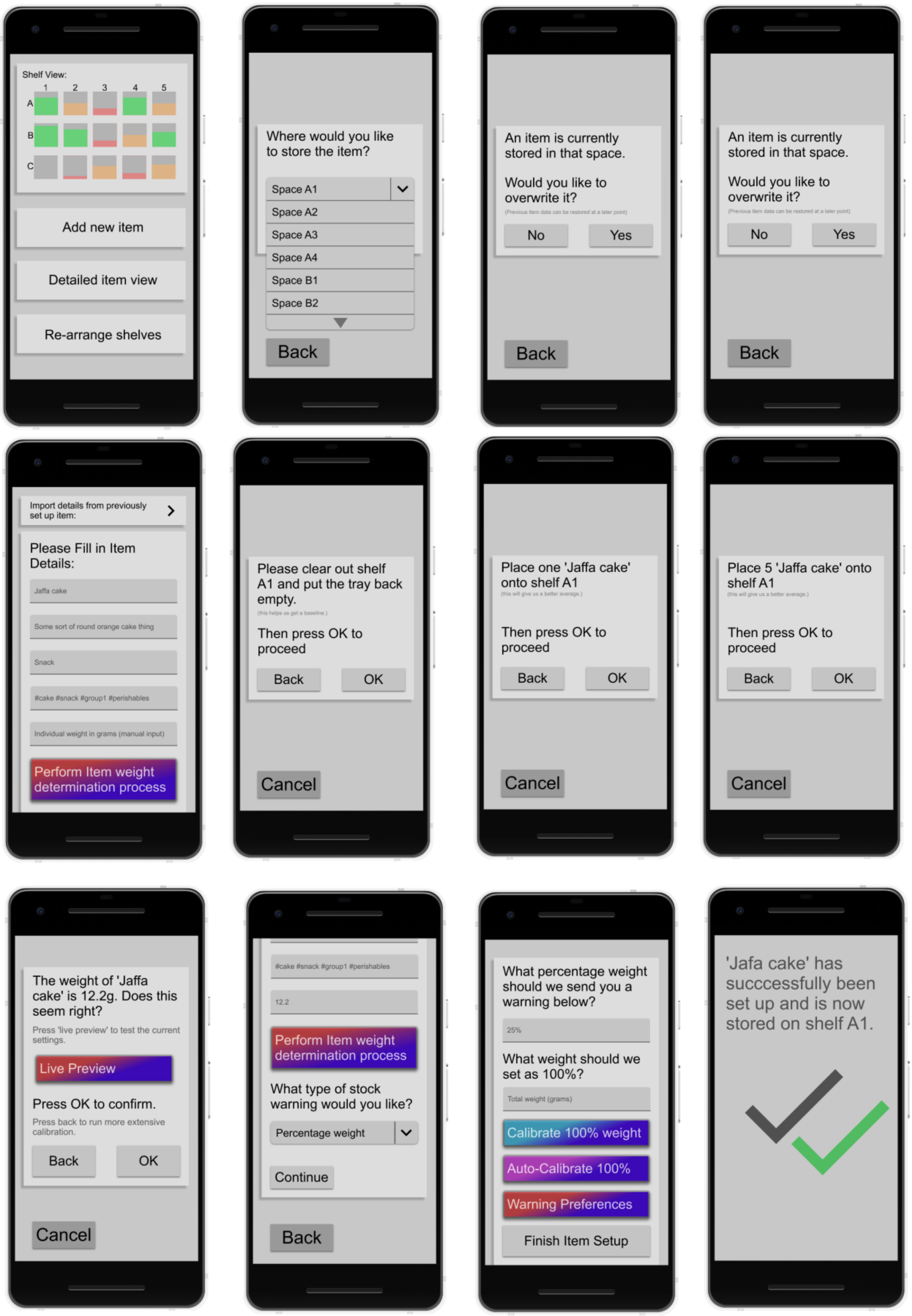


Figure 13: Several screens from the h-fi functional prototype made in Invision studio. Screens were linked using appropriate animations and form fields simulated entering data when tapped.

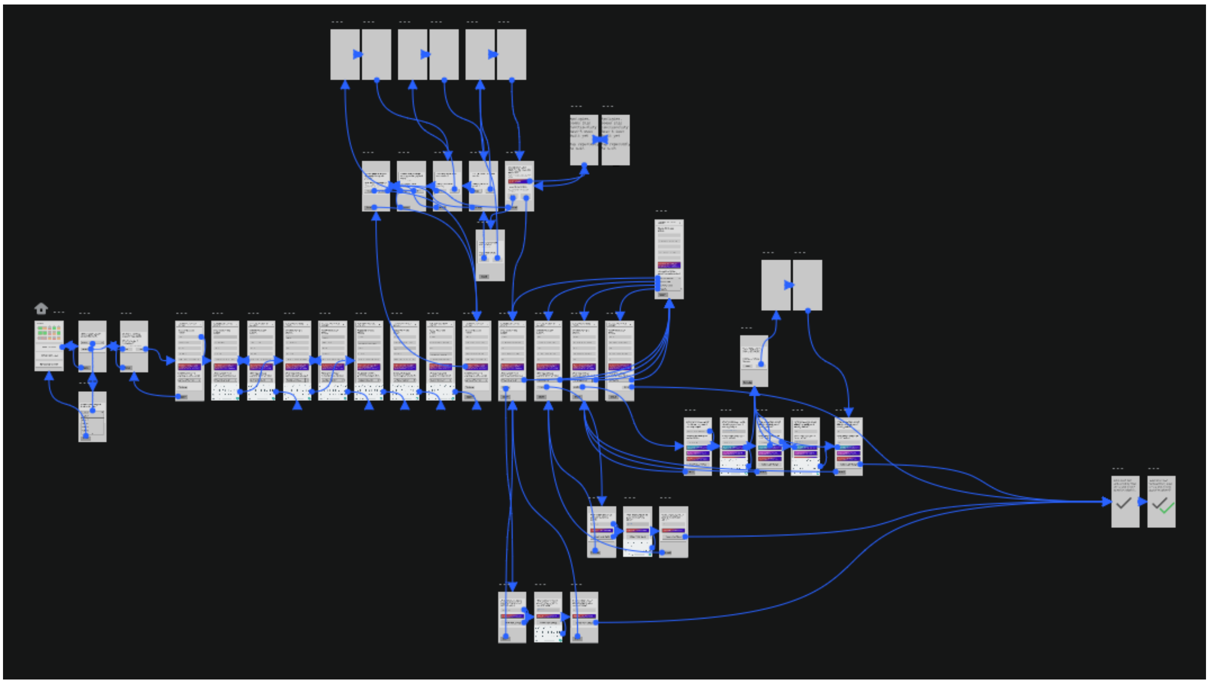


Figure 14: Zoomed out view of every screen created for the functional prototype in Invision studio. Blue lines represent the transitions between screens, based on user interaction.

Technical Prototype:

Prototyping was done to assess the technical feasibility of our designs. This was approached by researching each of the components that make up our smart shelf inventory system *(see figure 15 in technical specification page 17).*

The aim of this research was to determine if there were any open technical questions. The technical architecture of our system was planned, and no open technical questions remained.

A copy of this research can be found in appendix 2.

Functional Specification:

There are five key functional elements in our smart shelf system.

1. Load cells – to weigh items.
2. Signal amplifiers – to amplify load cell signals to useful levels.
3. Microcontroller – to process load cell data and send it to the server.
4. Server - with a database for storing weight data.
5. Application – to display processed data to the user.

The shelf hardware, load cells, signal amplifiers and microcontroller will be connected by wires. The microcontroller will send formatted shelf information to our server over the internet. The server will manage connections to microcontrollers and clients allowing them access to a database of shelf item information. The client application will connect to the database (via the server) over the internet.

Technical Architecture:

Shelf Hardware:

The only requirement for the load cell is that it has a high resolution, the specific model does not matter. Load cells are metal instruments with strain gauges glued to them. Strain gauges are resistors that change their resistance when bent. A signal amplifier is needed to increase the strength of this resistance signal so it can be read by a microcontroller. We plan to use the HX711 amplifier which has been specifically made for amplifying load cell signals to microcontrollers. We plan to use an Arduino as our microcontroller as there is more extensive documentation on how to use it for our purpose over other controllers (DegrawSt, 2017).

Server and database:

We will be using a MySQL database system. The Arduino and MySQL database will be connected via wireless technology through the MySQL Connector Arduino library built by Dr Charles Bell (Bell, 2016). This library has functionality which allows the Arduino to connect directly to the MySQL database without using an intermediate computer or service. It has been specifically built for projects like ours and it allows for set up directly over Wi-Fi.

Application:

Our client application will be a desktop Java application. We chose a Java application over a web based one because it should allow for easier separation of functional code.

Our Java application will connect to the MySQL database using inbuilt functions. The Java SQL package has functions to query and display information to the user (Javatpoint.com, 2011).

Any calculations and functions not achievable via MySQL queries will be coded in the desktop Java application. This will reduce the number of systems on which code we write will be running, which should make debugging and management easier.

System Requirements Specification:

1. 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 the ability to check stock levels immediately should reduce the time spent over manual inventory systems.

2. Scope: What users can expect to be able to do

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 indicated that they would prefer the app to 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.

3. System Overview:

The system components and reasons for usage were run through in the technical specification *(page 15)*. Below is a that specification presented as a system overview *(figure 15)*.

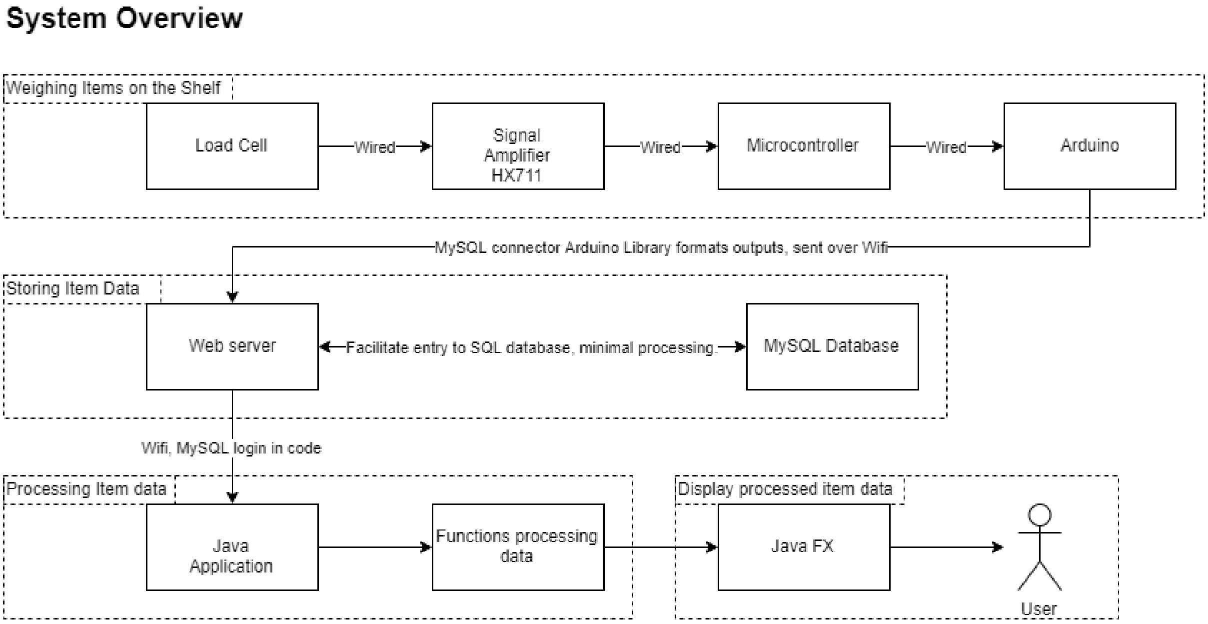


Figure 15: Overview of the System based on the technical specification, with data paths labelled.

Ethical audit:

At this point there are no plans to share shelf usage data. A possible identified feature was a database of items set up by other shelf users that would allow the item setup form to be populated quickly. If implemented data sharing would be both optional and anonymised.

Our target stakeholders are those already in charge of inventory management, our system should not present any difficulties over current systems.

Steps will be taken to ensure the colour scheme of the application has sufficient contrast for those with mild visual impairments and a range of colours suitable to those with colour blindness.

No minors or vulnerable adults will need to be consulted through the development of this product.

Evaluation Plan:

Testing will be performed to ensure the product is built correctly with reference to the design specifications (verification) and to ensure that the product provides the functionality required by the user (validation).

There are four levels of product testing: Unit testing, Integration testing, System testing and Acceptance testing (Hambling et al., 2019). Ideally all four levels will be performed however as the timetable for development, testing and delivering the product is short. It may be necessary to prioritise the most important tests at each level.

For details on how the different levels of testing will be performed, see appendix 3.

With reference to the product documents, the most important features to test are:

- Item weights are correctly measured, stored and monitored by the system
- User can add new items and view, update or delete existing items
- User can navigate through the user interface successfully
- User receives accurate alerts and information from the system

Example test condition: user receives accurate stock level warning alert
Example test case:

- Preconditions: a test item has been added to the system with weight above stock level warning trigger
- Inputs: reduce weight of test item to below stock level warning trigger
- Expected result: system displays/sends stock level warning alert for the item

Project management:

The project so far has been managed on a milestone basis with a project manager keeping track of tasks and setting internal deadlines. The project has been coordinated and planned through meetings, group messaging, GANTT charts, a Trello board for tasks and the centralisation of information into a Wiki (see Appendix 4).

During the development phase the role of project manager will be diminished. Agile development techniques will be employed with frequent scrum meetings in which team members all decide which tasks to do. A Kanban board will be set up on Trello, to ensure tasks are tracked. The group manager will still be responsible for setting non-development milestones and for ensuring records of progress are kept.

Conclusion:

It is our belief that our smart shelf solution can help medium sized labs or businesses to reduce the amount of time and stress organising and collecting stock information.

Market research allowed us to find a niche in the market that we feel is currently underserved. From there, stakeholder interaction has helped us refine our ideas to meet the specific needs and requirements that medium size stock environments face.

As a group, we took a fast-iterative approach to working on the designs and prototypes of our concept. This allowed for validation of our designs and refinement of processes. The result being that after displaying the hi-fi prototype stakeholders were impressed and somewhat excited to see the system working properly.

The user and system logic of the product has been planned out, technical questions answered, the Kanban backlog has been filled and development is ready to begin.

Bibliography:

Bell, C. (2016). MySQL Connector/Arduino.[online] GitHub. Available at: https://github.com/ChuckBell/MySQL_Connector_Arduino

CLEAR-VIEW-TECHNOLOGIES Inc (2009). *System for wirelessly monitoring inventory in the dispensing of items*. US20090261974A1

DegrawSt (2017). *Arduino Bathroom Scale With 50 Kg Load Cells and HX711 Amplifier*. [online] Instructables. Available at: https://www.instructables.com/id/Arduino-Bathroom-Scale-With-50-Kg-Load-Cells-and-H

Hambling, B., Morgan, P., Samaroo, A., Thompson, G. and Williams, P. (2019). *Software testing: An ISTQB-BCS Certified Tester Foundation guide.* 4th edition. BCS, The Chartered Institute for IT, Swindon

Intel Labs and The Store WPP (n.d.). *The Future of Retail: Shopping and the Smart Shelf*. futurecasting report. [online] Intel Corporation. Available at: https://www.intel.com/content/dam/www/public/us/en/documents/reports/futurecasting-report-june-4.pdf [Accessed 12 Dec. 2019].

Javatpoint.com. (2011). *Java Database Connectivity with MySQL - javatpoint*. [online] Available at: https://www.javatpoint.com/example-to-connect-to-the-mysql-database

Meter Toledo. (2019). *Manage Inventory Remotely with Smart Weighing*. [online] Available at: https://www.mt.com/hk/en/home/supportive_content/news/po/ind/ind-smartshelf.html [Accessed 12 Dec. 2019].

Shekelbrainweigh.com. (2019). *Shekel - Retail Innovations*. [online] Available at: https://shekelbrainweigh.com/index.php/products-solutions-retail/retail-innovations-retail [Accessed 12 Dec. 2019].

Wistron NeWeb Corp, DJB GROUP LLC (2013). *Shelf-monitoring system*. US8695878B2

Wiseshelf.com. (2019). *WiseShelf | Retail IoT smart shelf solution*. [online] Available at: https://www.wiseshelf.com/ [Accessed 12 Dec. 2019].

Appendices:

Appendix I: Stakeholder and user needs:

Methodology: Requirements gathering from the stakeholders was carried out using a semi-structured interview methodology. Key themes were discussed and agreed by the team in advance which could be used as a guide when formulating questions during each interview. This approach allowed for greater flexibility regarding the depth to which themes could be explored during each interview and also allowed for exploration of novel themes that had not been anticipated. As the interviews were conducted in the spaces where the concept would be applied, it was also possible to make observations about these environments relevant to the research. The duration of each interview was approximately one hour.

Summary of Shopper Survey: A survey of approximately 15 shoppers was performed. They were asked about their shopping habits and whether they would be interested in a shelf/product that measures your stock of certain items and either warns or re-orders them when they run low.

There was some interest, however most consumers had concerns over cost and practicality. Especially over whether shelves could accommodate more than one type of item.

Summary of Londis Manager Interview: Stock records that were kept, tied to the till, however they still had issues keeping track of stock. Knowing which items were most popular and needed reordering more frequently was identified as an important feature that would be especially well suited to their drinks section.

Persona: A typical user of this concept is a graduate in a STEM subject working in a technical academic support role such as a laboratory technician or manager. They are confident using technology both in their personal and professional lives. Analytical and practical they enjoy the challenge of finding solutions to problems as they arise, so any new product requiring additional set up efforts and adherence to proactive processes rather than a reactive or individual approach to inventory management must add significant value in order to persuade them to adopt it. This user however is also creative and apart from pursuing personal research projects of their own is highly motivated to maximise work time spent assisting users with their own research. Therefore, any product which can be seen to reduce time spent on maintenance and management tasks will be of interest. They are keen to provide an environment and services that are perceived by users as being safe, well stocked and efficiently run, with wastage and delay avoided as much as possible.

Appendix 2: Prototyping:

High-Fidelity Technical Prototype Research

Database:

The database system we will be using is MySQL due to its scalability, exhibiting the ability to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. This is valuable to our concept of an inventory system because it has unlimited potential in terms of how much may be stored using it. We will also be able to set up a MySQL database with either a web app or mobile app, depending on which route we go down for the final build. We will all have experience with this database system when the building phase of the project takes place, so it makes sense for us to use it over other options.

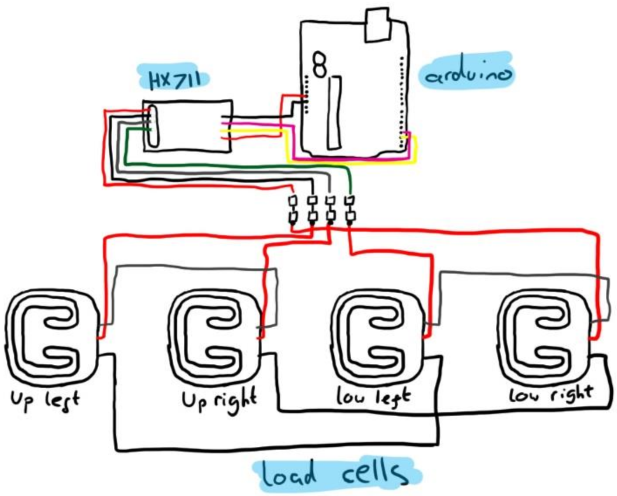
Web Vs Mobile App:

We are currently planning on building a mobile app to pair with our inventory system that allows the user to add, check and keep track of stock levels within the physical inventory system. It is understood that this process may prove difficult, so we have opted for a Web App as a backup if we're unable to create a mobile app in the given time frame. We will be using Java in JavaFX to build this app. JavaFX builds on top of JDK 13 and is a standalone component. There are 2 different options for developing JavaFX applications, using the JavaFX SDK or a build system like maven or gradle. We plan on using the JavaFX SDK from Eclipse.

Hardware:

For the hardware it is likely that we will use a variation of load cells, an HX711 amplifier and an Arduino to link everything and send data to our database. The HX711 is specially made for amplifying the signals from load cells and reporting them to another microcontroller. The load cells plug into this board, and this board tells the Arduino what the load cells measure. Load cells are specially shaped metal parts that have strain gauges glue to them. The strain gauges are resistors that change their resistance when bent. When the metal part bends, the resistance of the load cell changes which is what the HX711 measures accurately.

There are a lot of examples online of setups like this or similar with code snippets for calibration and setup which we will refer to when setting ours up. To the right is a rough sketch for the setup of the weighing system.

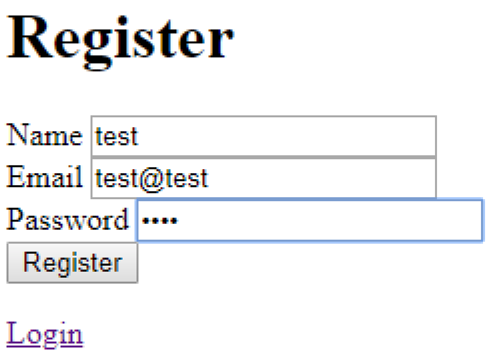


Login System:

If we choose to build a login system for the app or web app, it will most likely be done using MySQL to store the user information. A prototype has been set up for saving user input details to a database and hashing the password for security using bcrypt. This prototype has been setup using node and express for web use; however the processes are similar and this understanding will translate over into an app scenario. Below, a code snippet was taken from the register section which shows how the user inputs are saved and input to a database.

```
app.post('/register', checkNotAuthenticated, async (req, res) => {
  try {
    const hashedPassword = await bcrypt.hash(req.body.password, 10)
    con.getConnection(function(err) {
      if (err) throw err;
      var sql = `INSERT INTO accounts (username, password, email)VALUES("${req.body.name}", "${hashedPassword}", "${req.body.email}")`;
      con.query(sql, function (err, result) {
        if (err) throw err;
        console.log("1 record inserted");
      });
    });

    res.redirect('/login')
  } catch {
    res.redirect('/register')
  }
})
```




```
mysql> SELECT * FROM accounts;
+----+-------------+--------------------------------------------------------------+-----------------+
| id | username    | password                                                     | email           |
+----+-------------+--------------------------------------------------------------+-----------------+
| 13 | Chris Mayes | $2b$10$RBQUbjZYGjSBRH3g5PYhP.TN6b3s/xHCDpnMMzEpa8zjlBux7ZA4e | chris@mayes.me  |
| 15 | Chris Mayes | $2b$10$49.AvQpZJom7/ZA4frmkU.9TVbiUDvg/nx2E6DtqiFIyaB7GCVYhe | email@test      |
| 16 | test        | $2b$10$lRvzQqM22CqcTCCDRSbY9OYYnnU5.ziefbUlIWliZR2bqhxMlJyIi | test@test       |
+----+-------------+--------------------------------------------------------------+-----------------+
3 rows in set (0.00 sec)
```

Appendix 3: Testing

Testing Methodologies:
Unit and integration testing can be achieved by a combination of static and dynamic techniques. For static testing, the unit requirements and design documents can be reviewed prior to coding in order to detect any errors or ambiguities. For dynamic testing, the J-Unit testing framework available in the Eclipse IDE and other Eclipse debugging and testing tools will be used to ensure that the unit and interface code is functioning as intended.

For system and acceptance testing, the main test case design technique used will be specification based, drawing on the specification and requirements documents of the product to identify the core functional and non-functional elements that require testing. Test conditions can then be identified and used to generate test cases and test scripts, which will verify the test condition and execute the test.

Appendix 4: Project Management

Images of current practices: maintained GitLab wiki, GANTT chart and Trello:

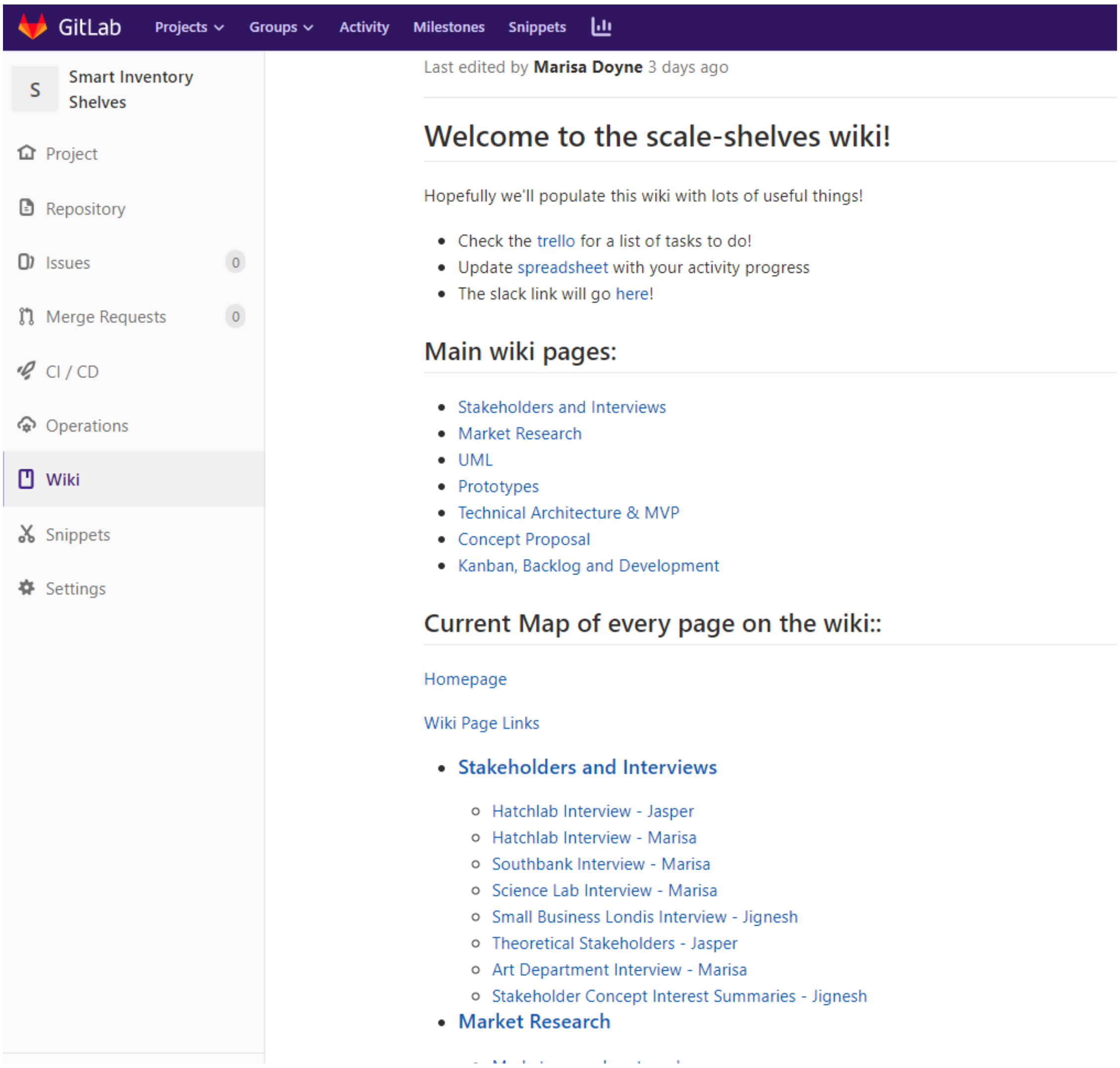


Figure 16: GitLab wiki, currently stands at approximately 30 pages. It acts as a store for both rough and neat notes. It is more easily accessible and navigable than folders of individual text files.

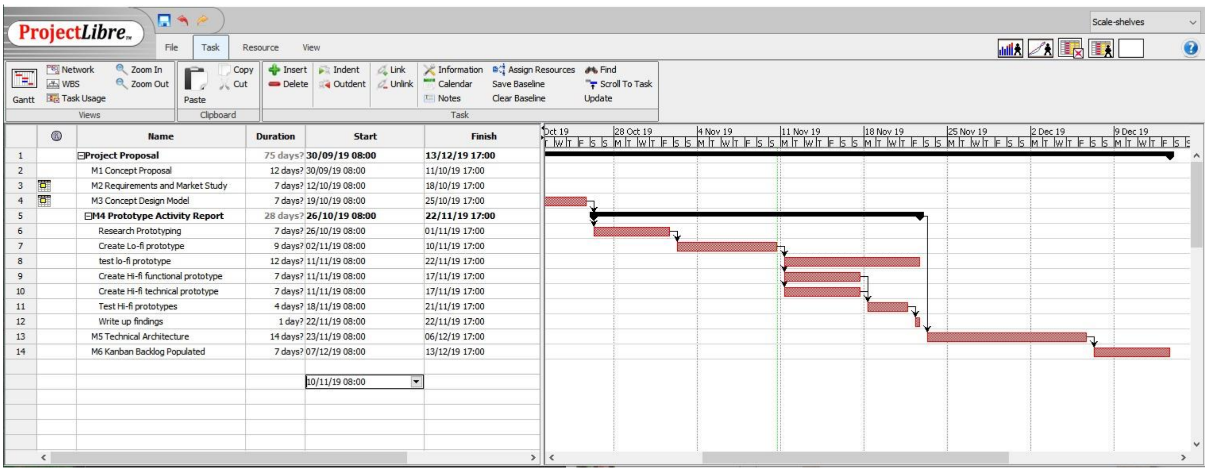


Figure 17: The project GANTT chart as it was during the Prototyping phase, showing both tasks following a pre-requisite order and tasks that can be performed simultaneously by different members.

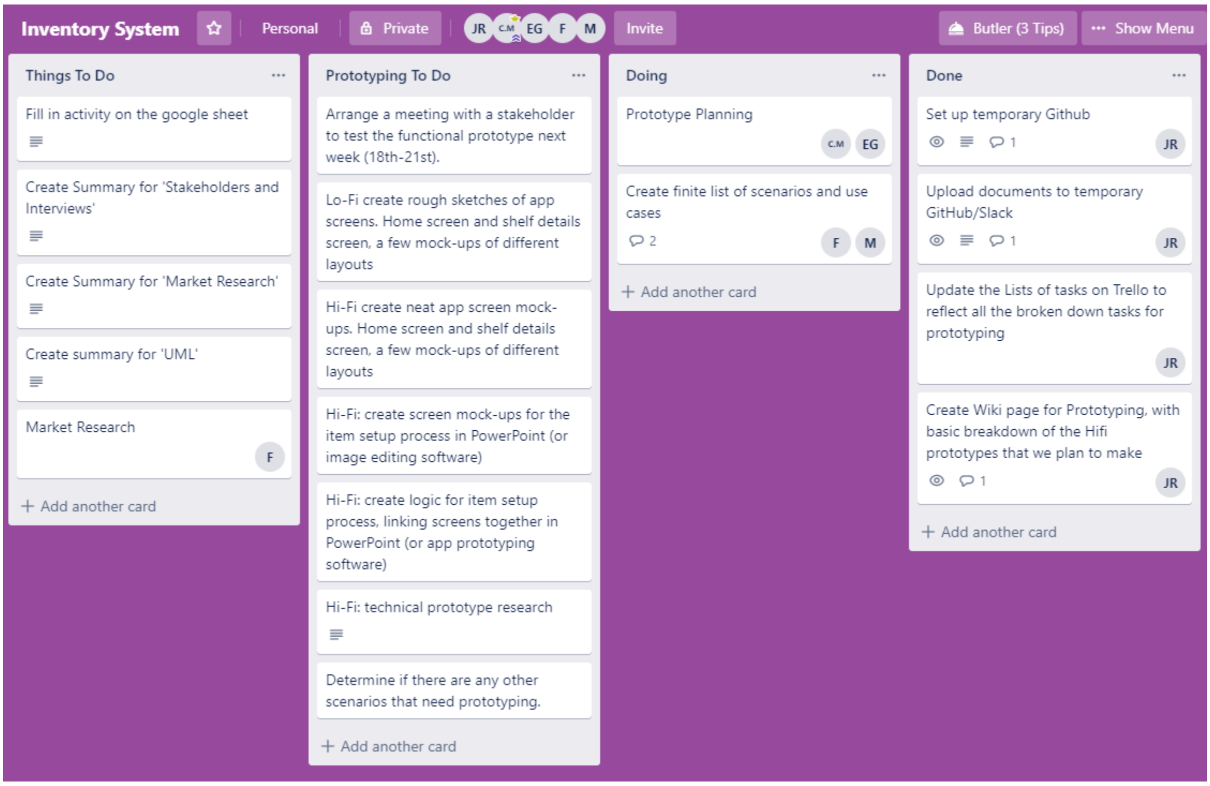


Figure 18: Project Trello board as it was during the prototyping phase. Tasks are split up amongst members with a list of currently unassigned 'to do' tasks. The Trello board has since been updated to work as a Kanban board for our development phase.