# Final Version Specification

# API Design

The API design is critical in order to produce a well maintainable site that other developers can come in and use with minimal learning time. With this site’s API it has been designed to keep the various types of needed files separate while keeping similar files grouped together. As the file structure tree shows I have separated the files into 3 main directories, navigation to store all the sidebars for each main page, database to store all the PHP responders and any file that interacts with the database and structure to store files that are used for the default page structure (for example the footer and header).

The other 3 folders include files that are no longer used or have been scrapped from the design. They have been kept in the API as a point of reference for not only my benefit but also for the benefit of developers who come after me so they can see the evolution of the site.

With this API design and structure should not only allow anybody who comes after myself to easily find what files they are looking for but to also allow any develop who is working on the product to easily maintain and add to the API without having to restructure it.

# Client <–> Server Interaction

This product was designed with reducing server load in mind. The majority of website creation is achieved at the client side through JavaScript and a series of AJAX functions in order to retrieve page elements and data from the server.

AJAX is used extensively in this product to interact between the client and the server sides. It is used to not only retrieve page elements that give the illusion that the page has changed but it is used to send data in the form of structured URLs to the server and send data back using JSON. The response from the AJAX calls (those involving data) is handled by the JavaScript typically by inserting said data into local arrays or are handled directly by the functions.

By reducing server load in theory (there is currently no way of testing this theory with available resources as far as I am aware) it should ensure that customers have the lowest amount of wait time when loading the site and individual pages thereby allowing for the best possible experience for them. The JavaScript does come with some inherent load time but this is unaffected by the amount of customers using the site, if the work was done through PHP then the wait time would steadily increase according to the amount of people using the site.

# Existing Functionality

## Customer

### Basic

* Customer is able to view products according to product type.
* Customer can navigate through the site through a combination of a top navigation bar and navigation sidebar.
* Customer is able to add products to the basket through the use of ‘Add to Basket’ button that is present on both the individual product pages and the product tables.
* Customer can search for products by either name or code in the search bar and be presented with a table showing the results.
* Customer can view the contents of their basket, remove individual items or clear the entire basket.
* Customer can purchase the contents of their basket, input recipient details and submit said order.

### Higher

* Customer can drag product image over basket image to add product to basket.

## CMS

### Basic

- Admin can add new product to the system.

- Admin can edit existing products that are in the system by entering the product code in order to retrieve the information.

- Admin can change stock levels of individual products.

- Admin can delete product from system according to product code.

- Admin can edit site details such as size of navigation links or colour schemes.

- Admin can add new product types into the system.

- Admin can delete product types from the system. This action deletes products associated with this type.

### Higher

- Admin can set the order in which the product sidebar appears by clicking and dragging a series of dynamically created images into their desired order. This order is saved to the database and appears as the default order from then on.

- Admin can delete a product by entering a valid product code and then dragging the generated product image over the bin image.

-Admin can disable or enable images on the site through the use of a checkbox.

- Admin can edit most text elements on the page. Said elements can be found by hovering over them (which produces a red border) and element can be edited right on the page.

- Admin can show editable elements by holding down Shift + E.

- Admin can save changes to text elements with Shift + S.

## Admin

### Basic

* Admin can view stock levels of all products or sort by product type.
* Admin can view details of orders including their status (shipped or not shipped). Admin can then change the status of the order or report a problem in which a input text area is generated, a message is typed and then sent to the recipient’s email.

### Higher

* Admin can generate a stock report in the form of a .txt file which is created in /reports. The name of the file is stockReport(date of creation).txt. The file includes the title of the report the author and the date of creation as well as the details for each individual product (product name, code and quantity).

# Rationale

As per the project mark scheme and defined objectives I have added the basic functionality expected from a system such as this while keeping usability in mind the in the implementation of these features. The higher functionality was added for two reasons, one to improve how the website and it’s features are used for the user’s benefits and to also make the product stand out among other similar products being produced for this unit.

One consistent feature throughout the higher functionalities is the ability to drag and drop. With the release of HTML 5 came its drag and drop API making it relatively simple (once the developer has a basic understanding of it) to implement such features. This feature was added in order to make the process of completing tasks such as deleting products faster and more interactive, streamlining the entire process. Due to time constraints detailed within my ‘End of Project’ report I was unable to implement further interactive functionality which in my own personal opinion would improve not only the customer’s use of the website but also the admin’s. One such feature that I would like to have implemented (for the CMS) is the ability to view products in the same as the customer (in a product table) but have to ability to click and drag and table row to a specific spot on the page to delete the product from the system. This would also apply to the product types that exist on the product sidebar.

If given the opportunity to continue working on the product I would have also liked to developed higher functionality for the admin section of the website as in its current state it is somewhat lacking. On the other hand if someone else was to continue the product in an attempt to expand upon what has already been achieved there would be two aspects of the product that I would just be attempted to begin with. The first is the design of the product, due to more marks being available for functionality compared to visual design I have focussed primarily on functionality and therefore the product isn’t visually impressive in the slightest. This is a factor that customers taken into account when deciding to use or reuse a website therefore if this is improved I feel that the appeal of the website and its effectiveness as an ecommerce site would be improved.

The second is improving user interactive through other methods than just inputting text. I have attempted this with the delete functionality but I believe that it can be continued further in order to provide both the customers and admins with a more streamlined and faster experience. In the same way that a more attractive site design would encourage customers to use the site, a site with streamlined tasks would make the customer’s time at the site easier and therefore make it more likely that they would return.