Functional Requirements

* Store inventory of current stock

*This is currently implemented through Quickbooks. System should update stock in real time. (Current system only updates stock at then end of the month via invoice)*

* Create invoice

*This is currently implemented through Sage. The invoice should be created from data on what products have been taken, the quantity and the department they have been taken from.*

* Create log of products being taken

*Logging the products being taken, and who has taken them should allow to create an end of month invoice*

* Display currently available stock
* Unit Conversion (Boxes to Items)

*The system should convert the codes used by the supplier into an amount that is used by the system. E.g. a supplier code AGH43579 Could mean 1 box of 100 packs of individual pens.*

* Register products received from shipments

*The relevant staff members should be able to input the code used by the supplier which then updates the stores current inventory*

* Database to cross reference codes used by the store and the supplier

*The store uses alpha codes to identify specific products. A different code is used by the supplier to identify the products. The system should allow the admin/relevant staff member to input the suppliers code and have the inventory update with the relevant code used in store.*

* Provide directions through the store to a desired product

*Optional feature. Would be useful in particular when the store is busy. However is not a crucial part of the system.*

* System could remind relevant staff of when the next delivery is/details about what is being delivered.

*Optional feature. Not desired but would be useful for when customers require something out of stock and want to know when it will be back in*

* Return feature for products to be returned

*The feature should allow products to be returned which will then update the inventory and the corresponding invoice for the department that returned it.*

*Must be exclusive to staff members to avoid consumers returning damaged goods*

* VAT Option

*Must allow admin to select whether products being added to the catalogue have VAT or not*

* Allow admin to update the catalogue of items being sold

*The admin could have the ability to add/remove products that are being sold in the event new products are desired or old products need to be removed*

*(Ask Nigel)*

* Low quantity reminder

*The system should notify the relevant staff members that a products stock is low so that more can be ordered in*

* Log in

*Facilitate the login of an admin, staff member and regular consumer. Staff may update the inventory and receive shipments. Admin has more control over things such as the catalogue of items. Regular consumer may only take out and return items to the store. Apprentice’s cannot*

* Inspection Stock

*The system must store products in an inspection database first and can only be added as general stock after a staff member has inspected. Staff member can have ability to add products to the general stock after inspecting an item.*

* Bar Code Scanning

*To improve ease of use, a barcode scanner could be implemented that removed the need to input product data into the system manually.*

*Optional feature. Not necessary for system completion but would improve ease of use.*

* Product Tags/Categories

*The system should display important information for products that could be deemed dangerous such as toxic chemicals/explosive gases.*

* Basket System

*The consumer should be able to see their current items they plan to take out with pictures of the items, quantity, unit. They should be able to remove items and add new ones to their basket before they finalize their removal and “Checkout”. (System is very similar to the scan and go used at Tesco)*

*If barcode scanner is used this could update the basket by scanning multiple times for multiple items.*

* Track items that expire

*Certain items cannot remain on the shelf for certain periods of time and as such should be monitored and kept track of how long an item has left on the shelf.*

*(Ask Nigel)*

* Track Shipments Received

*Must include:*

*Order Head*

*Supplier Name*

*Supplier Site Name*

*Supplier Remit to address*

*Order Number*

*Order Date*

*Requested Date*

*Promised Date*

*Goods and Services Total*

*VAT*

*Invoice Total*

*Order Line*

*Supplier item number and/or universal product code*

*Item description*

*Quantity*

*Unit of Measure*

*Unit of Price*

*Extended Price*

*E5 will provide this information which the staff member must then input into the system to keep a log of the shipment.*

Non Functional Requirements

* Simple User Interface with minimal user input
* Simple process to manage stock

*The sequence of steps needed to complete a process must be simple and not complex so that new staff members can be walked through and training in the system with ease.*

* Catalogue with Pictures

*Without a barcode scanner, a browsable catalogue with pictures and names of products could be displayed to the user which can then be selected. Similar to an online stores catalogue.*

* Data Encryption

*Personal information being stored for logins should ideally be encrypted so that a potential hacker is less likely to obtain an individual’s personal details.*

* Backup Data

*Data should be backed up to an external/cloud based storage to prevent data loss which is a problem with the current system.*

* Borrow Stock between Stores

*In the eventuality of another store being opened, the ability for stores to trade stock between each other when ones stock is low could be a useful feature.*

* Languages Used

*C# for backend, (Potentially using .NET for interface) MySQL for database.*

* Universally friendly Interface Theme

*The interface should be usable by individuals with impaired sight. For instance colour blind people should be able to utilise it. It would be a poor interface if the background colour and button colour were identical and a colour blind individual could not distinguish the two.*

* Responsive Interface Display

*The systems interface should resize depending on the screen size of the display being used.*

*Optional feature.*

**3.0 Interfaces**

**3.1 User interfaces**

Design

The user interface is to be designed in a simplistic manner in order reduce the learning curve of the system. Thanks to controls in the UI following a similar layout to similar existing systems, staff should have little difficulties transferring their knowledge of previously used systems.

Controls such as buttons will have black outlines to provide clarity in their purpose and location as well as allow colour blind users to distinguish them from other controls. Any of these controls which are used on multiple pages will be situated in the same location for each page to be consistent, reducing cognitive load for the user.

Each page of the system will have a distinguishable help button. Clicking on this button will open a popup describing the purpose of the page and of any ambiguous elements it may contain.

Colour Palette

Surface & background colours



Error, message & action colours



The chosen surface & background colours have been selected as weak toned colours as to not distract the user from more important UI elements. Two variants of the surface & background colours allow contrast between UI elements, such as distinguishing a background of a UI element and the system background, while keeping the same overall theme consistent.

In contrast, error, message and action colour tones have been chosen as bold colours, which draw users’ attention to important information.

GUI Mock-ups

Below are some example mark-up designs for how the system will look and act.

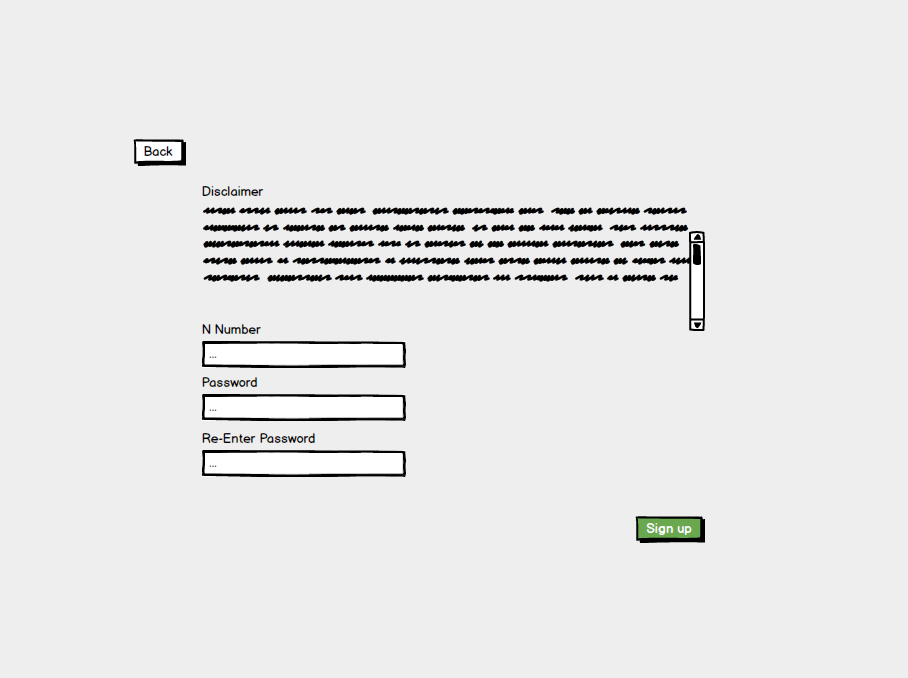
**Login page**

This page will be used by users to login to the system. An option for new users to sign up will also be available.



**Sign up page**

The sign-up page will be used by new users to sign up. Any disclaimer texts for users signing up will be displayed here. Users will be required to enter their N number so their account information can be linked with their NTU account.



**Products page**

Once signed in users will be navigated to this page. From here products can be selected/searched for. A menu bar at the bottom of the page consistent across many screens will be available for navigating around the system.

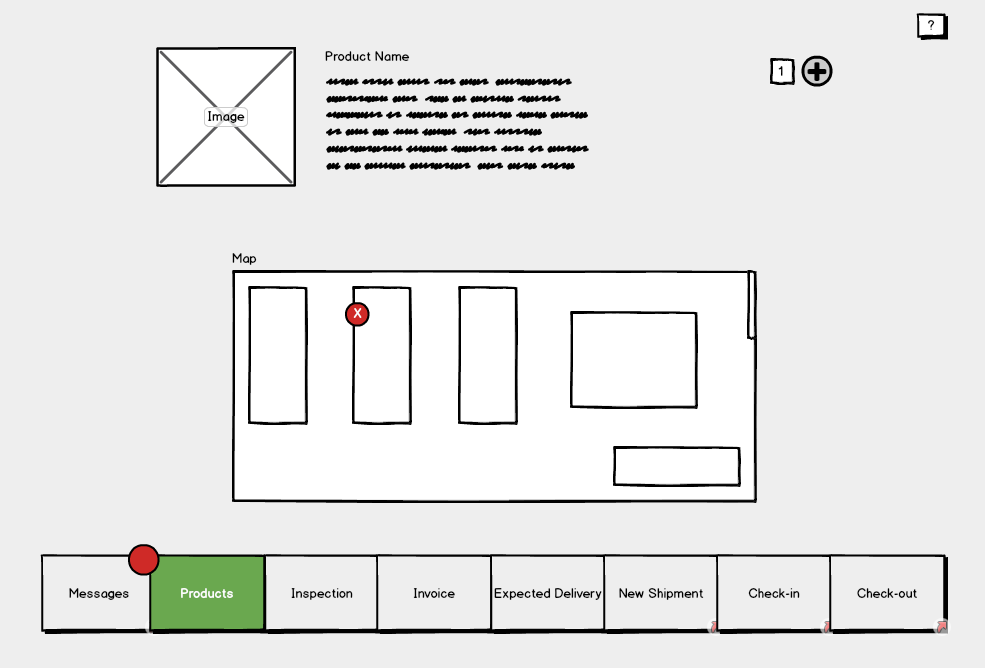


Users, as opposed to staff and admins, have less options and therefore their menu bar will be more simplistic.



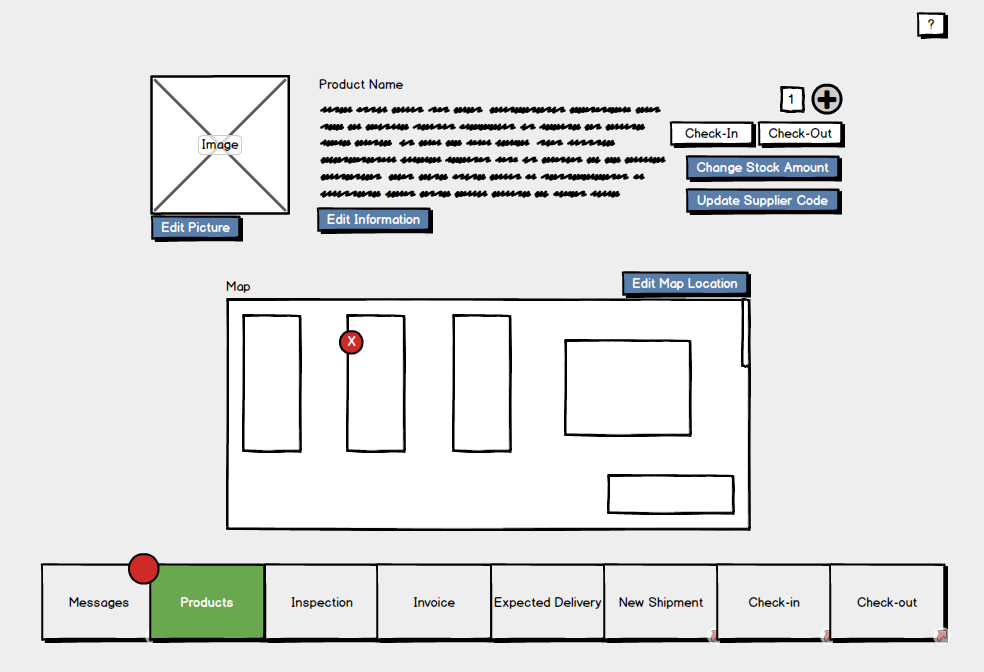
**Single product page**

This page will display the necessary information of a product. From here a user can add a product to their basket to checkout. A map of the location of the product within the store will also be available. If no items of the displayed product type are available users will be notified here as to when, if known, the next shipment of this product is expected.



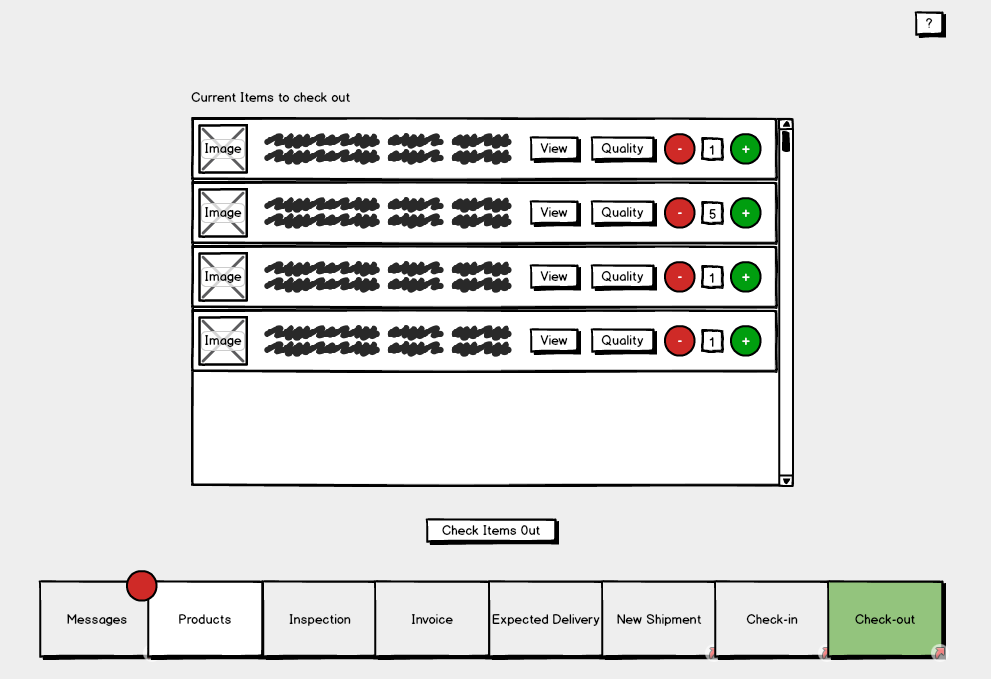
**Single product page (For staff & admins)**

Admins and staff will have access to more features on this page such as the ones shown below. Blue controls represent ones which only admins will be able to use.



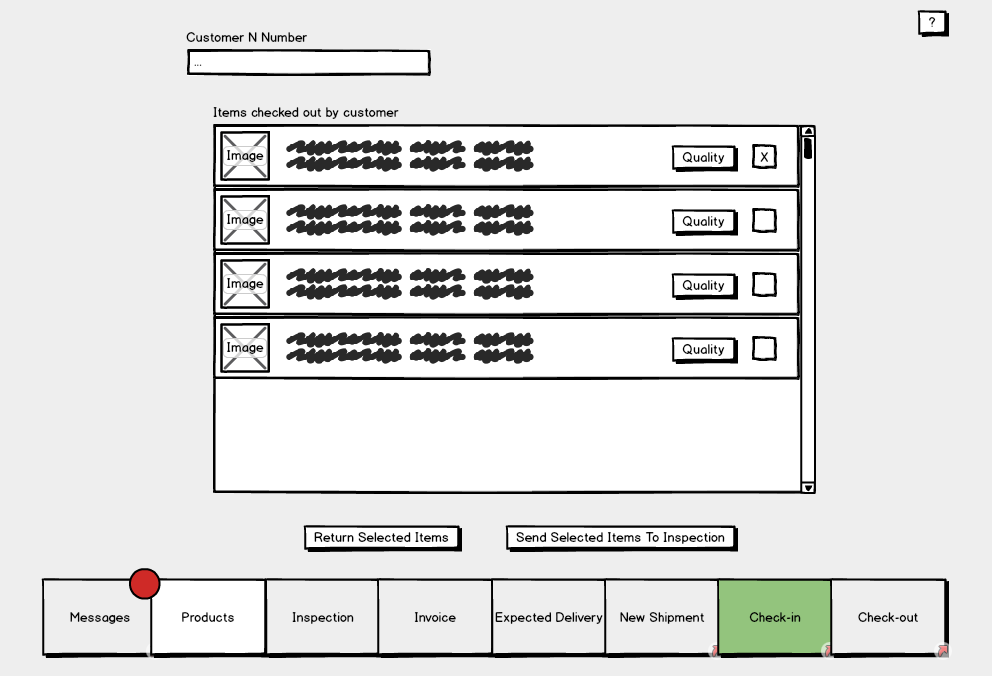
**Check out page**

This will be like a basket page seen on various shopping sites displaying a list of the products which the user wishes to checkout. Users will be able to remove items from this list, check the quality of the items and view their descriptions.



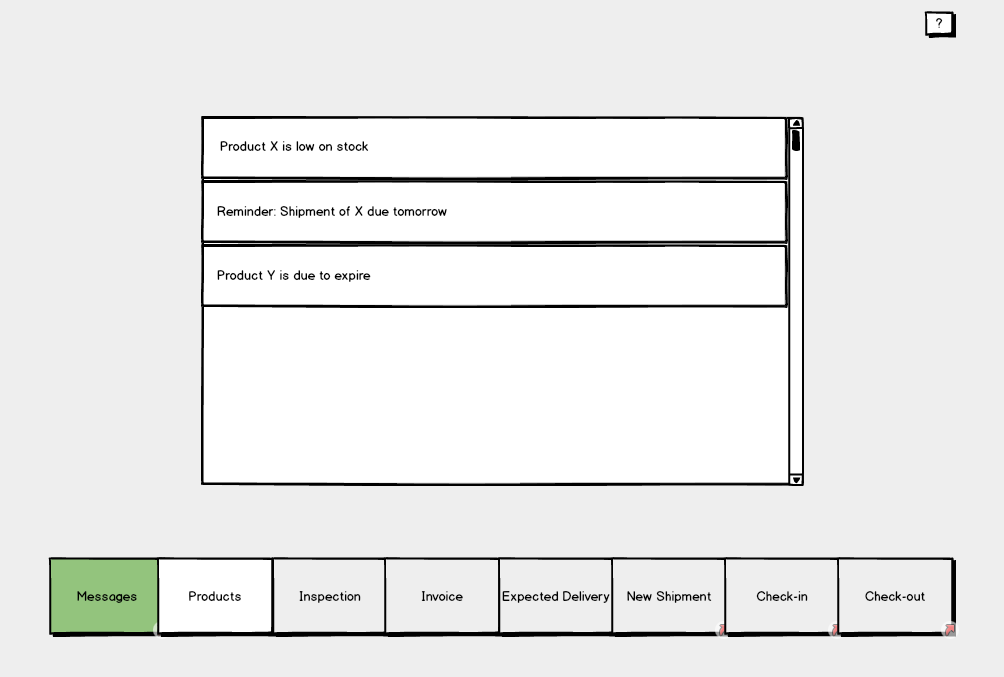
**Check-in page**

On this page staff will be able to search for a customer to see the items they have currently checked out. These items can be selected and checked back in. (This will send them either to the inspection database if it’s developed in time, or flag them as returned in stores)



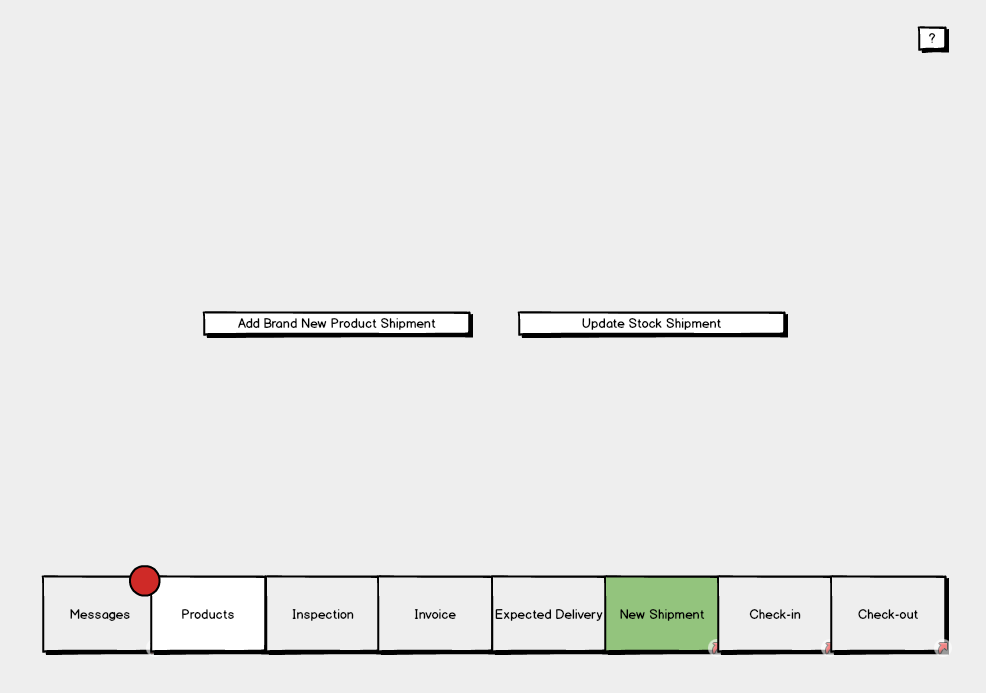
**Messages page**

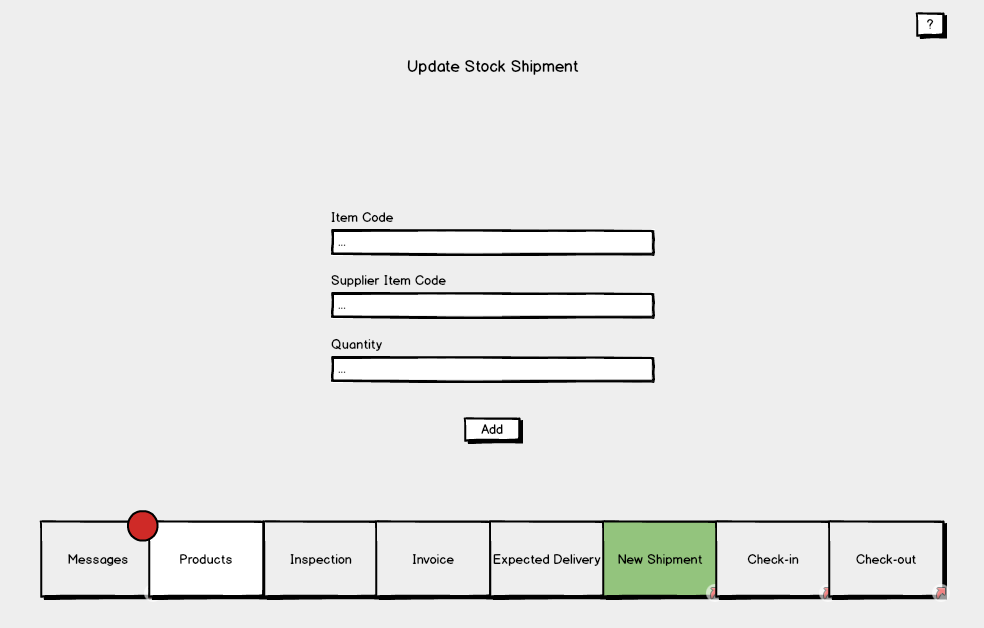
The messages page will display messages for the logged in user. Categories of messages include; Overdue returns, products low on stock, expected shipments and expiring products.

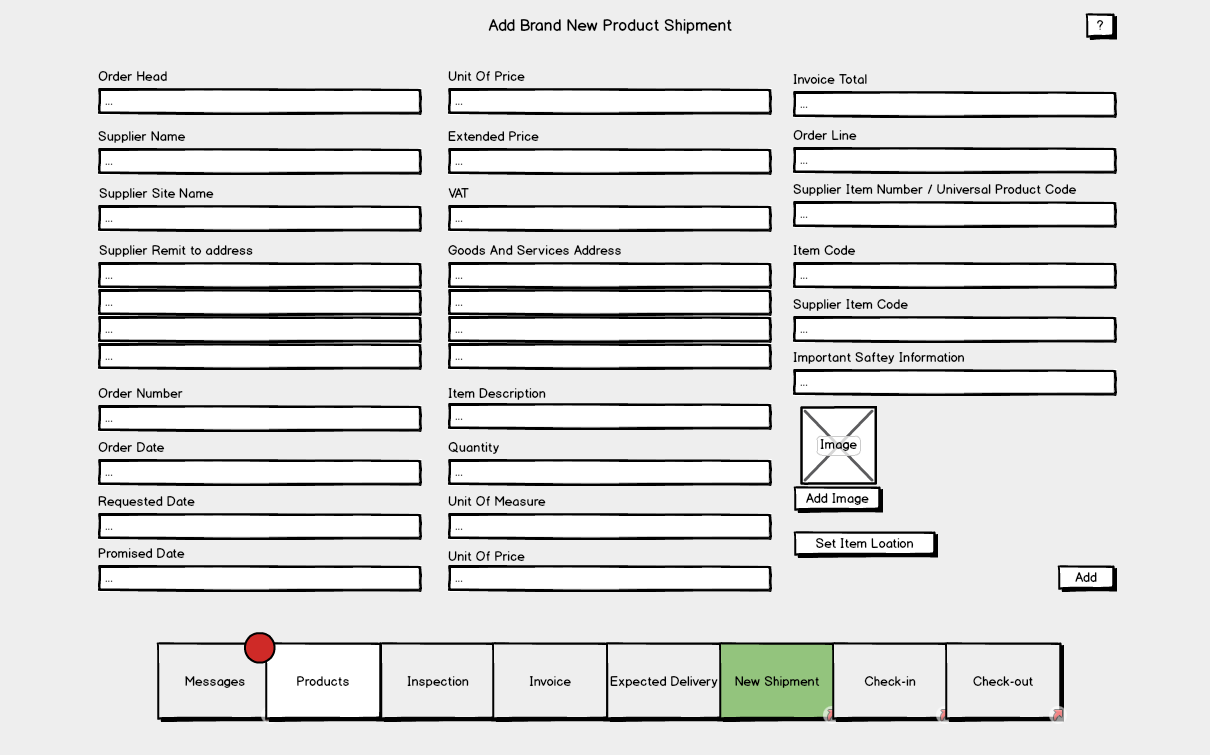


**New shipment pages**

This page, used by staff, is to enter information about an arrived shipment. Depending on weather the shipment is a repeat shipment the user will navigate to either the “Add brand new product” page or “Update stock shipment” page.







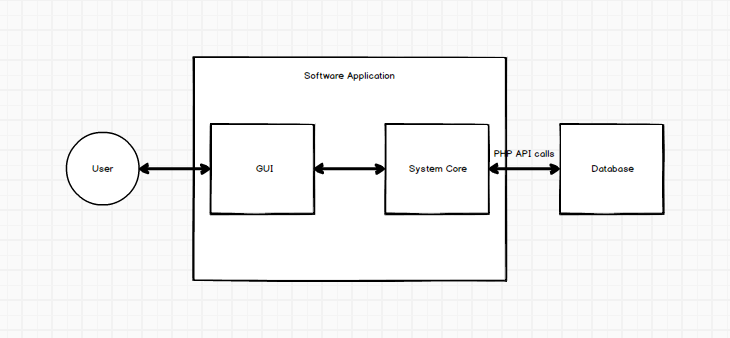
TODO: INVOICES PAGE

**3.2 Hardware interfaces**

Thanks to the system being software based there are only two hardware interfaces to be considered. Firstly, the systems interface with the barcode scanner, and secondly the interface between the system and computer on which it will run. Both interfaces will be used by staff, admins and customers.

**3.3 Software interfaces**

The diagram below illustrates the systems software interfaces.



The systems interface with the database will be used to transfer data between the database and the system. This will include the retrieval and uploading of; products, user data, transactions and invoices. In order to achieve its functionality, the interface will make use of the internet via PHP API requests.

The systems backend will be linked via an interface to the GUI. This link will provide the user with a method of accessing the systems features.

Project Planning

If I had to be in charge of a team for building a complex system, I would start by clearly writing job descriptions for each team member and making them available to every member of the team. Everyone has to know their areas of responsibility. To make a group of people feel like they belong to a large community, I would introduce a number of team building activities so that everyone would get a chance to know each other in an informal way. I would not care about placing all team members in one location - telecommuting is perfectly fine in the contemporary world.

To foster collaboration, I would implement a specific type of organizational culture - clan culture - that would foster mentorship and team accountability. The teamwork combines individual commitment with group results, which makes it different from regular working groups, such as a function in a corporation. I would select people into the team based on their transferable skills and skill potential, not for their personality (Principles of Management 499).

The team will work their ground rules by discussing expectations of collaboration, discipline, confidentiality matters, project approach, conflict resolution, to name a few. To set tasks and track performance, I would equip the team members with a methodological framework and versatile tools. The framework would be scrum, one of the implementations of agile methodology. This framework supposes conducting everyday meetings mandatory for all team members where each member reads their tasks for the day, and the status of processed tasks is established. What is more important, everyone thinks of the ways they can help their colleagues succeed in fulfilling the daily tasks. In case the timeline gets challenged, the brainstorm for solutions is conducted. This way, I would ensure the smooth work process of each employee and know whether they were available for new jobs.

To track working time on each task, as well as their status, I would introduce Jira or Trello. New assignments would be in backlog; then they would be moved to in progress, then approval, later on - revision if necessary, and completion stages. These digital logs provide managers and employees with statistics regarding the productive time one works and allow predicting timeframes for upcoming projects.

Communication within the team would be conducted via emails, messengers, and video conferencing. Emails would be mainly for formal approvals and distributing minutes of meetings, messengers, such as WhatsApp, would serve as daily formal and informal communication channels; and video conferencing would be conducted in applications, such as Microsoft teams or Zoom. Storing files and regulating access to information is one of the paramount issues for a team manager. Since everyone has to have equal rights to files, secure storage has to be selected. Onedrive is one of the file hostings that allows uploading files, synchronizing the work of several people on a number of files in real-life mode.

Regarding the tools for building a system, GitHub proves to be an indispensable software development partner within the developers’ community. It encompasses code creation, integration, review, and overall team management. This way, the working processes are transparent, and each team member is easily replaceable. In case a new team member arrives, they will be able to seamlessly continue where the previous employee left.

Overall, team management is a challenging task. Leaders have to unite team members, supply them with measurable job descriptions, and equip with necessary tools for work. To remain productive, agile methodologies have to be welcomed. The crucial aspect is helping coworkers to reach mutual success.

Work Cited

*Principles of Management*. Openstax. Roice University.