# CS190 – Stock Manager Version 1.0

# Specifications

By: Taha Mohamedali (tmohamed)

## 1. Introduction

### 1.1 Motivation

Many small business owners lack the technology and education to integrate an efficient stock control and management system into their business. They do however often have a computer that is capable of word processing and checking email. Motivation for this project comes from the fact that often, business owners can't keep track of what stock items they available, which stock items need re-ordering and the average lifetime of perishable stock items. The resultant scenario is one characterized by the ordering of stock after it is completely exhausted and the throwing away of perishable stock due to reduced demand or overstocking. This on-the-spur-of-the-moment approach is often the culprit of reductions in net profit.

Businesses of this caliber include ones that sell ready-made products, sell or fix parts of a product, or a combination thereof. Examples include a bicycle shop, a picture framing shop, a grocery shop and a restaurant. In implementing this project a stock control system can be designed to improve the management and optimization of stock usage and as a result improve net profit.

There are numerous stock control programs available,but the good ones are generally too complicated, have unnecessary features and are too expensive for this consumer market. There are also companies that build custom systems for businesses, but these too are out of budget in most cases.

The ideal solution would be to create a stock control system that is simple enough to allow the end user to configure it himself/herself and powerful enough to fulfill the user's needs.

### 1.2 Description

Stock Manager ver 1.0 will contain a feature set that is geared towards this consumer market, specifically addressing the needs pertaining to the stocking of various “raw material” items and “finished product” items. It will also contain adaptable features that can be customized to fit the consumer's needs. A more thorough description of this flexibility will follow shortly.

The features will be divided into the four following categories: Management, Financial, Customer, Optional

*Management:*

- Customizable order forms that can be tailored to allow the input of consumer specific data items. This customization will also allow the consumer to change the type of data items that can be entered in a customer order if the need arises.
- Electronic submission of orders will replace the traditional paper-based methods of writing sales receipts, and manual updating of stock records or the lack thereof. As soon as an order is entered and confirmed, the stock database will be updated to reflect the changes.
- Graphical representations of past and current stock usage to gage at what rate different stock items are used up.
- Tracking of supplier information including a list of suppliers, payments made and stock received over an extended period of time.
- Alerts for users when stock is soon to run out
- User accounts with different permission levels to safeguard sensitive information.
- Client/Server architecture to allow multiple clients to access the centralized server which hosts the databases

*Financial:*

- Graphical and tabular representation of sales/profitdata
- Supplier comparisons of similar and different stock items so as to enable ordering from the cheapest / best supplier
- Dynamic balance sheet creation

*Customer:*

- Tracking of sales to all customers
- Functionality to enable the searching of previous and pending transactions

*Optional:*

- Printing capability for: current and past stock usage, financial analyses as well as faxable order forms.
- Stock location tracking and reshuffling for optimal space usage
- Secure online ordering with credit cards

### 1.3 Users

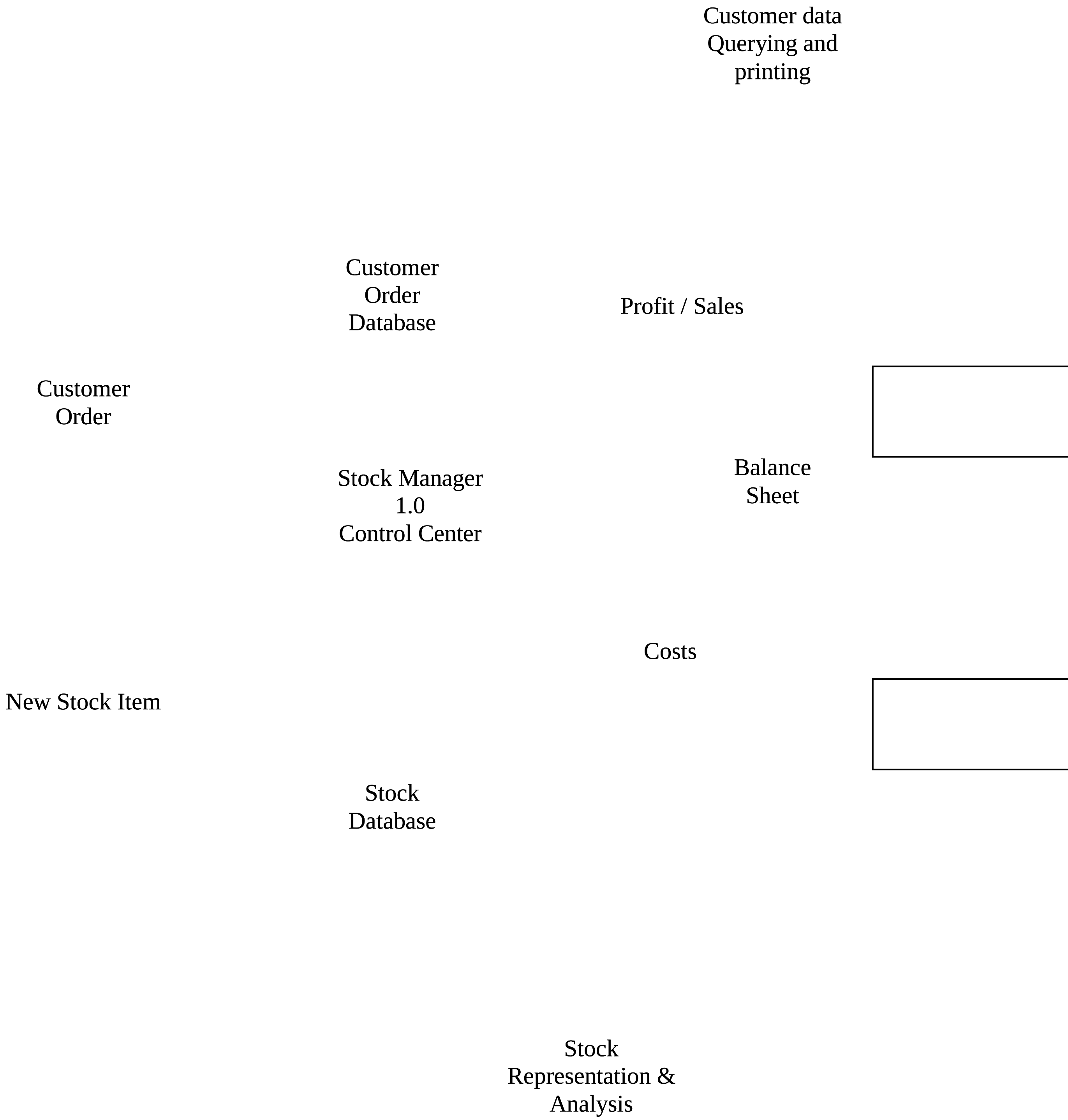
The target users for Stock Manager 1.0 are small business owners. Local area users include but are not limited to: Esta's bike shop, Campus fine wines, East Side Framery, numerous small computer stores in RI that undertake assembly and repair or machines as well as the sale of spare parts.

Taha Mohamedali – CS190

Stock Manager 1.0 would also be successful if used in a restaurant to track which menu items are the most popular, which foods need to re-ordered, etc…
The program is intended to replace or supplement an employee who assists the business owner.

Taha Mohamedali – CS190

# 2. Data Flow Diagram


Customer data Querying and printing
Customer Order Database
Profit / Sales
Customer Order
Balance Sheet
Stock Manager 1.0 Control Center
Costs
New Stock Item
Stock Database
Stock Representation & Analysis

## 3. Component Explanation

The data flow diagram shows how information is transferred within the stock control system. The rectangles represent modules, or processes and the ovals represent information repositories. The arrows point in the direction of the flow of data. There will be two back-end databases, one to store the customer order information and the other to store the stock details. The stock database will be filled using records of stock items, each record will have its fields specified by the user. When entering stock in the database, even if the item being entered already exists, a new record will be created in case of variation in prices, supplier or expiration date. The old stock record will however be linked to the new stock item and the total quantity will be the sum of the two records.

The records in the stock database can be queried based on any of the fields or criteria and there will be options to allow for the graphical display of characteristics like the amount of a specific stock item that has been ordered every month for the last six months. Information about the total amount of revenue spent on purchasing stock can be extracted from the stock database and can then be combined with the profit/sales analysis to generate a balance sheet.

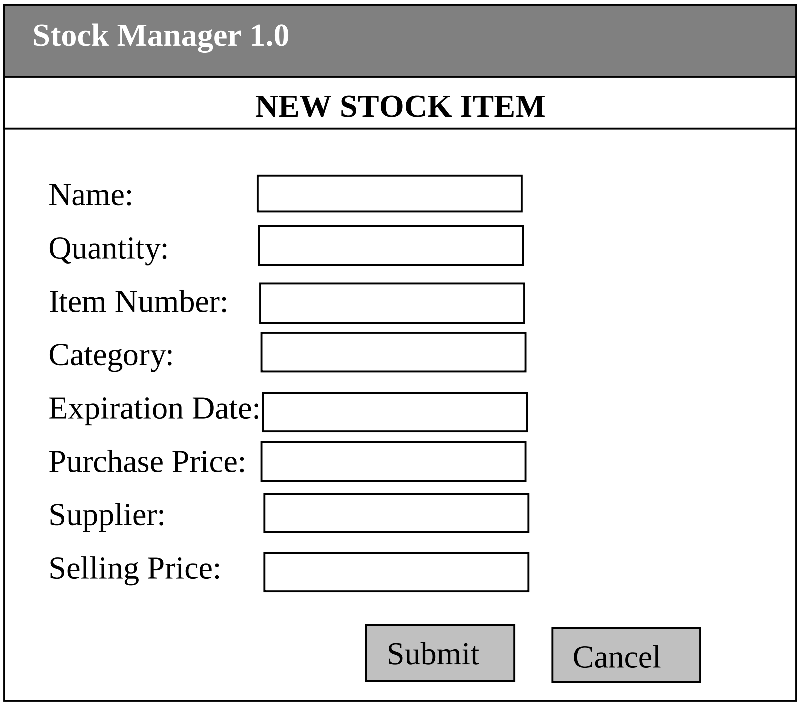
The Control Center component serves to represent the logic of the program, it interacts directly with the user to allow customer and stock input and also retrieves information from the databases to produce financial and stock analyses. All information will pass through the control center before reaching its final destination; records for the databases for example will be entered by the user on a form that will submit the information to the control center, which in turn writes the information to the database. The user accounts, permissions and possible printing capability will also be stored and configured in the Control Center.

The customer orders would be submitted in a similar way to the new stock and when the order is submitted, the stock items specified in the order would be decremented from the stock database. Queries can also be submitted to the customer database to retrieve information on past and pending customer orders.

The processes in the rectangles would be carried out by the client side and the information would then be transmitted to the Control Center that would reside on the server and host the databases. The idea behind this architecture is that in the even that the business expands and has two terminals from which to serve customers and enter stock, on machine would host a client and the server while the other one would simply run a client that sends information and requests to the remote server.

## 4. User interface

The key behind implementing a successful user interface is simplicity. Below is diagram of the “new stock item” window that will pop up when the user decided to order new stock or add a new stock item to the existing database:



**Stock Manager 1.0**

**NEW STOCK ITEM**

Name:

Quantity:

Item Number:

Category:

Expiration Date:

Purchase Price:

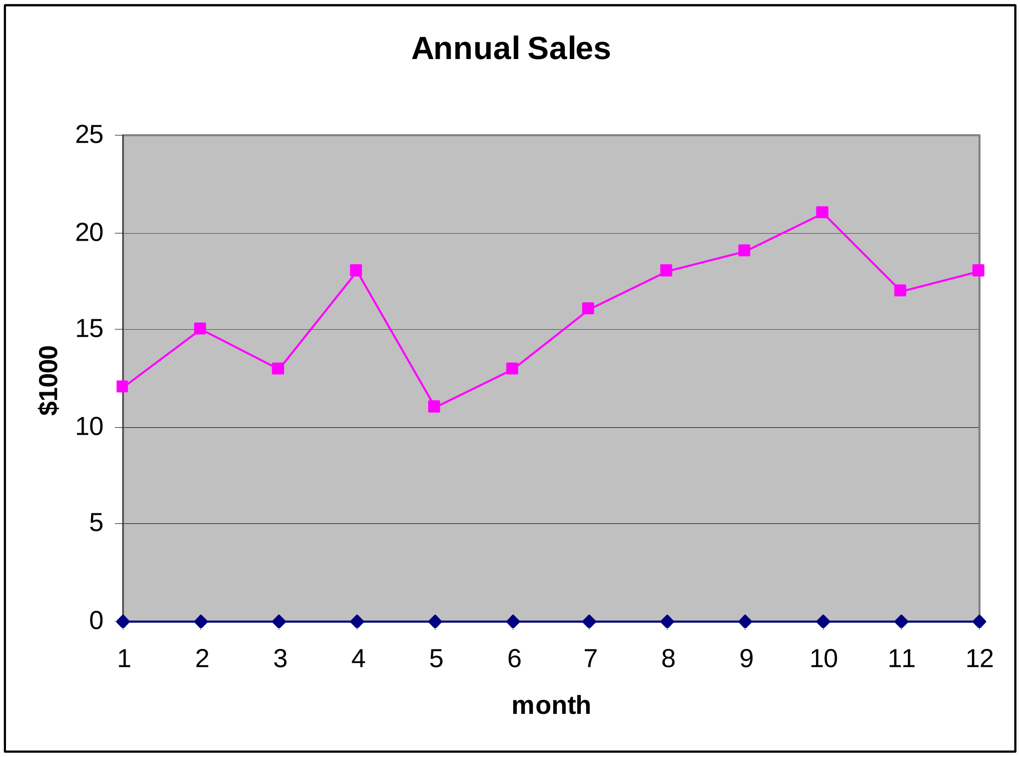
Supplier:

Selling Price:

The customer order form will look similar except that the fields would hold different information. For every order, a new customer order record is created for tracking and querying purposes.

Taha Mohamedali – CS190

The financial module will be able to analyze records in the database and create graphs showing the trends of a specific set of variables over time. It will contain a graphing wizard that will determine exactly what kind of data the user wants to see and finally generate a graph based on the user choices and information in the database:



Taha Mohamedali – CS190

## 5. Non-functional Requirements

### Performance:

It is imperative that the program is designed to be as responsive as possible. All data entry and retrieval from the databases should occur instantaneously and eliminate all possibility of data corruption. The graph generation should be fast and memory efficient to allow users to plot many graphs simultaneously in and not have to wait for them to be generated. All network communication must be as fast as possible

### Testing:

As outlined in the requirements document, the testing portion of the project will be extensive. There will be continuous testing and code reviews as functions and objects are written, by the programmers themselves as well as other members of the team. A comprehensive set of drivers will be created for testing purposes and this will prove useful as a testing tool after bug fixes and the implementation of new features. After separate components have been thoroughly tested, they will be integrated with the other components and there will be a plan for intensive integration testing.
Lastly after this is complete, rigorous user-level testing will be carried out to ensure that the program will not crash, that there are no memory leaks and so on.

### Reliability, Portability and ease of use:

Reliability will be a major concern in all phases of the development cycle and the main concerns will be that the information that is entered through the clients is valid and complete and that this data is stored properly in the databases. There should be no data loss during transmission and manipulation of the contents of the database should not cause existing data any harm.
The program should be portable and the significance of the portability of the program arises from the fact that a large proportion of the target users will be using PC's and Macintoshes. The program seems simple enough that building in portability should not be a problem.
The design of the GUI will be such that the interface will be easy to understand and use, there will be no hidden functionality three mouse clicks deep. The customizability of the program to the user specifications will eliminate the occurrence of ambiguous fields when entering stock and customer information.

### Dependencies:

The main dependencies in the development process would be between the client and server modules. If the database is not implemented, then the records that can be entered through the client can't be tested effectively. There will also be other dependencies, for example the graph generating widgets will need the database implemented for ideal testing.

**Division of Labor:**

This project can be scaled to an appropriate size as follows:

Control Center Logic and Database implementation – 3
Graphing panels – 2
Graphing engines - 2
Client side user interface – 3
Networking components – 1 or 2

**Risks:**

I think that the specifications for this project are pretty straightforward, it contains some database work, some networking, and some graphing. The only risks I can forsee is that we may have to forfeit some features depending on the progress of the development.

**Requirements:**

The requirements are the same except that I have moved the printing requirements into the optional section and I have added the networking component in the Management section.