A-Level Course Work

Contents

**List of Figures .................................................................................................3**

**Analysis..........................................................................................................4**  
Problem Identification...................................................................................................4  
Stakeholders ..................................................................................................................4  
Existing Solutions...........................................................................................................6  
Requirements ................................................................................................................8  
Success Criteria............................................................................................................10

**Design ..........................................................................................................11**  
Problem Decomposition ..............................................................................................11  
User Interface Design...................................................................................................11

List of Figures

Figure 1 Unleashed’s Sales Orders Interface.................................................................6

Figure 2 Unleashed’s Add Product Interface.................................................................7

Figure 3 Fishbowl’s Sales Orders Interface....................................................................8

Figure 4. Decomposition of the main window ........................................................................11

Figure 5. UI mock-up for the Stock Tab ...................................................................................12

Figure 6. UI mock-up for the Customers Tab ..........................................................................12

Figure 7. UI mock-up for the Orders Tab.................................................................................13

Figure 8. UI mock-up for the Help button ...............................................................................13

Figure 9. Decomposition of the add dialog .............................................................................14

Analysis

# Problem Identification

A company is struggling to keep track of its current stock and the orders that need to be processed. Being able to store all the stock, customers and orders electronically would help the organisation of the business . It could help identify what orders need to be processed, provide contact information if there is any need to contact the customers and warn the company when stock is getting low.

Storing the information on a computer is vastly better than trying to keep track of physical copies of the data. This is because it would be extremely hard to search, copy or analyse the records if they are not stored electronically. It also increases chances of data being lost if the only copy of the data is a physical copy.

## Computational Approach

The problem lends itself to various methods such as problem decomposition when designing this solution. This allows the problem to be broken down into smaller tasks such as calculating order prices, adding or updating records, searching for information and notifying the business when the stock is low. The solution could use object-oriented programming so windows can inherit features and functions and use classes control behaviour of dialog. It could also use N-Tier architecture with a front-end GUI and an abstracted interaction with a database using a data layer.

## Main Solution Features

The database should have a customer, stock and orders table to keep track of the most essential information. The customer table will allow the company to record customer information and apply discounts to certain customers. Keeping track of stock, without having to do stock takes, helps notify the company when they need to order new stock and keep track of which items are the most popular. The orders table will help show all orders that need to be or have been processed and calculate the amount that would need to be paid. It must have an interface for the users to interact with it.

# Stakeholders

The clients for this project will be a small bookstore that sells books to distributers. The users will not necessarily have a in depth knowledge about computers and the use of databases so it must be very user friendly. This will allow the customers to use the database with little to no training making it very accessible. The company would also like to keep track of how many orders have been placed by specific customers so they can use that information when deciding if they should give that company a discount on their prices.

## Main Users

The main users of the system would be warehouse pickers and the sales team. This is because in the warehouse they need to check what orders they need to pick and what orders have already been processed. The sales team will need to use the database to add new orders and customers to the database and check or update discounts.

I decided to ask 10 of the main users questions the following questions to find out how they want the interface to look like and what they want to be able to view.

1. Do you have any experience using databases, if so what?
2. What information do you want to be able to see?
3. How do you want to access or input data into the database?
4. Is there anything else you want to add?

Question 1 allowed me to find out how confident the people using the database are so I can ensure it is easily accessible. I discovered 8/10 employees have some experience with databases but need the interface that is simple to use. This would also mean the employees who are not familiar with using databases would need less time to be able to use it.

The second question informed me on how to design the interface. The users said they want sperate pages or tabs for each category of information so it’s clear and less cluttered. This should also make the interface simpler to interpret.

For question 3, I was told they want buttons and text boxes that would allow you to enter the specific information to find data from the table or all the information to add it to the database. This informed me on how I should set up my interface.

In the last question, most employees didn’t have anything else to add but one user said that the design should be very spacious so it doesn’t seem too cluttered which I will try to incorporate into my design.

## Senior Users

I also decided to ask the purchasing manager, operations manager and sales director some questions so that I could get a better idea of what features the stakeholders wanted out of my solution.

1. Would you like the system to calculate order prices?
2. Do certain customers have discounts that need to be applied?
3. How would you want to sort the data?
4. What else do you want to database to be able to do?
5. Are there any other features you want the solution to have?

For question 1 all the employees said that calculating order prices would be a useful feature as it can be used to track how much a customer has spent and how big an order is. Hence, I will include it in my final solution.

In question 2 I was told that there are certain discounts for some customers so if I calculate order prices, I need to incorporate this into my database.

This question told me that the stakeholders want to be able to sort the orders based of oldest order date. I was also told that they want to be able to sort customers based on biggest discount prices. For the stock they wanted to be able to sort it based of how much stock as left so they could see which items need to be purchased next.

In the fifth question I was asked by the purchasing manager if I could include a feature that would notify them when the stock got below 10 for any item. I was also asked if I could keep track of how much each customer has spent by the sales director. I will therefore try and add these features into my final solution.

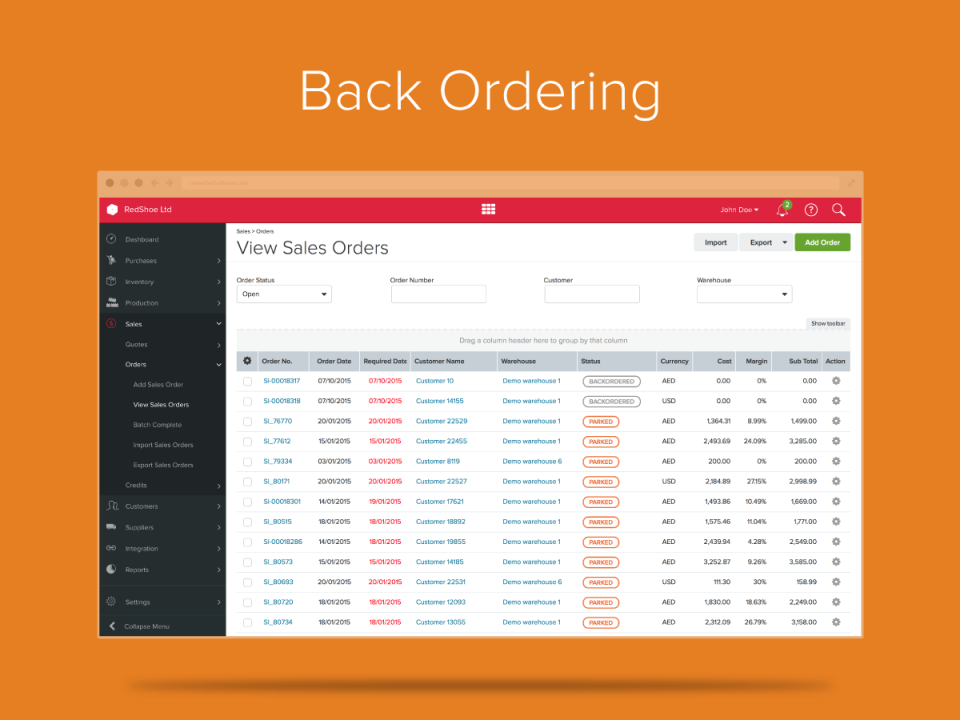
# Existing Solutions

A lot of small businesses like the bookstore, use Microsoft Excel and Google spreadsheets but they are not nearly as powerful or useful as databases. This is because with large volumes of information, it is hard to transfer data in or out of spreadsheets and sometimes require you to manually import and export the data into different programs. For example, keeping purchases and prices on a database instead of a spreadsheet allows the company to transfer the information directly to the accounting system without having to find the data and input it manually.

## Order Management Systems

I decided to look at existing order management systems and their interface so I could decide what features I would need to include in my solution.

The first order management system I looked at was called Unleashed. This system uses the cloud to store the data and can be accessed on the web and on mobile devices. Unleashed does inventory management, billing and invoicing, order management, distribution management and production tracking. As it is a high cost solution and has a monthly cost of $135.00 the product has extra features such as sales forecasting and returns management. This solution also contains warehouse management which is not necessary for my solution as the company only operates out of one location.



Clear title so user is aware what table they are looking at.

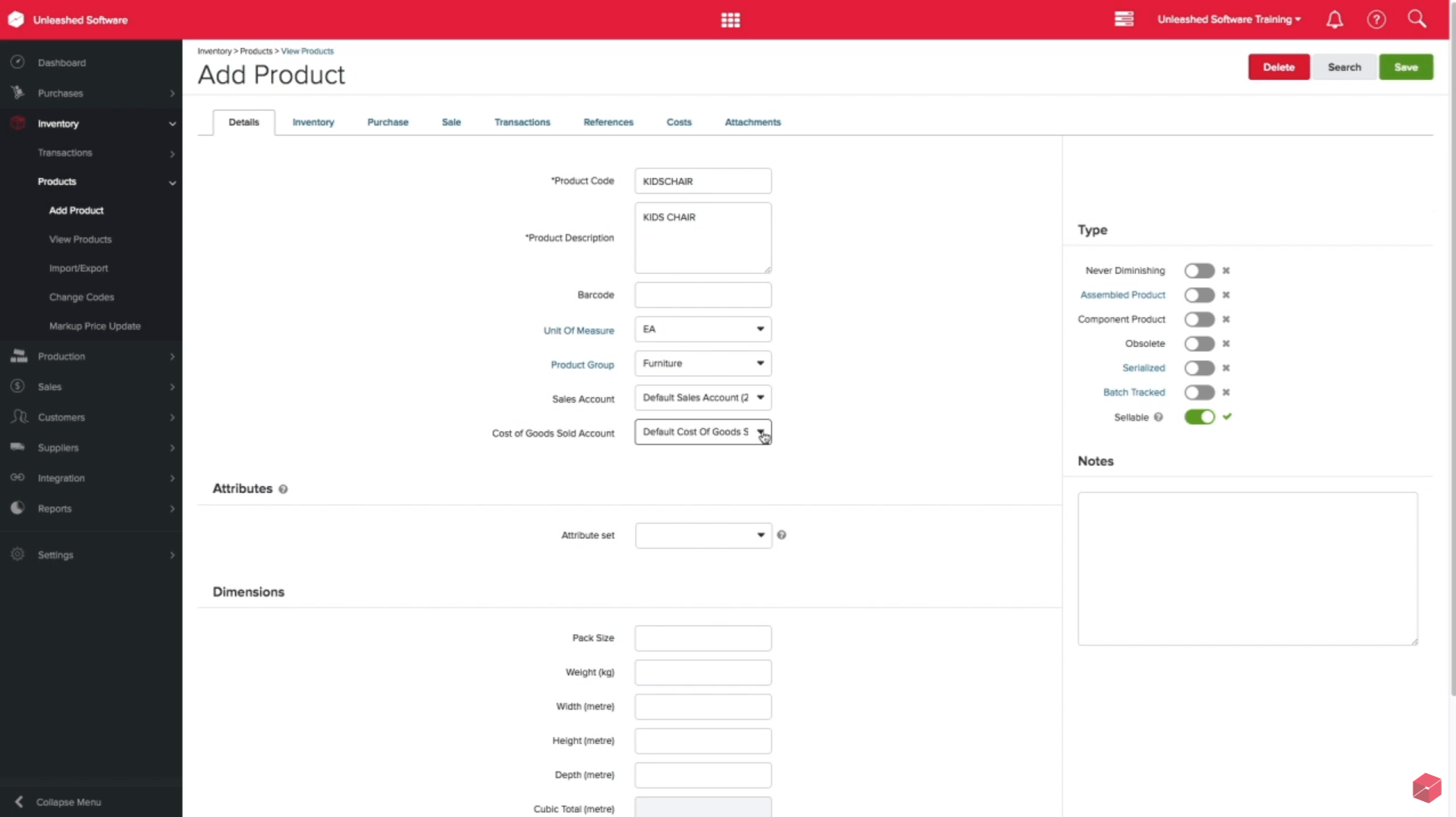
Easy options to search data using 4 pieces od data.

Bold colours make the design aesthetically pleasing.

Menu bar so the database is easy to navigate.

Colour Codes to make the interface easier to interpret.

Figure 1 Unleashed’s Sales Orders Interface



Uses clearly named text boxes to input unique information and drop-down boxes for when there is a set number of options so easy to input.

Help option to make the product more accessible with no training.

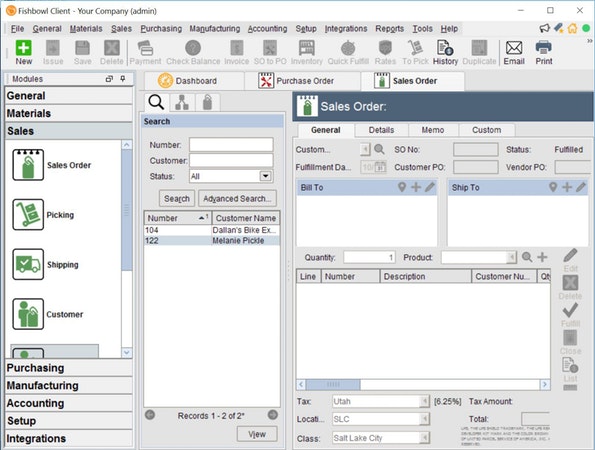
Switches so it is easy to input binary information.

Symbols used on the menu, so it is clearer to navigate.

Figure 2. Unleashed’s Add Product Interface

Unleashed has a great interface as the use of colours codes, clear options and titles, symbols and help button makes it easy to understand how to use it. The software is also accessible through both computers and mobile devices making it easier for the stockholders to have access. These key features will improve my solution so I will try to incorporate them into my final design.

Another existing solution is Fishbowl which has one upfront cost. It works with QuickBooks so small businesses can stay on the platform for accounting and incorporate inventory features into their database. Like Unleashed, this solution contains features such as inventory management, billing and invoicing, order management, distribution management and production tracking. It also contains a feature that allows back orders. Fishbowl is still expensive as it costs $4,395.00 which allows it to have additional features such as barcode generation and scanning.



Symbols used with subheadings to make them easily identifiable.

Help option to make it easier for inexperienced users.

The delete and accept options for adding lines have symbols to make it more user friendly.

Uses order headers to group orders by one customer and reduce repeated information.

Figure 3. Fishbowl’s Sales Orders Interface

After looking at existing solutions, I have found out that colours and having the interface be very spacious are very useful as the Unleashed interface looks a lot more user friendly due to this. Both designs split orders into order headers and order lines so information such as customer name, delivery address and order date don’t have to be repeated for stock item they have ordered which I will include in my design. The designs also include symbols and a help option to make the interface easier to navigate. Therefore, I will try to include these features in my interface. I will also try to include a help feature like they have in both these solutions to make it more user friendly. These designs also have a forecasting feature but due to the time constraints on my project I won’t be able to incorporate that into my design. Also, my solution will not be accessible on mobile devices as they don’t meet my software requirements.

# Requirements

My solution must meet the following requirements:

|  |  |
| --- | --- |
| Requirement | Justification |
| The solution must be able to store customer, stock and order information in a database. | This is necessary to keep track of all the data the business needs so they can make more informed decisions. |
| There must be a feature to add new information into all the tables. | This allows to company to process new orders, grow when they receive new customers and adapt when they expand their product range. |
| You must be able to search the data stored in the database. | This is required so the user can find information stored especially when there is a large amount of data. |
| The solution must include a user interface that is clear and easy to interpret. | The database must be accessible so the employees can easily use it as and from my interview I know some users have no experience with databases. |
| There must be a way to change existing data in the tables. | This is necessary as the company might want to change information such as the list prices of books or the amount of discount a customer is getting. |
| It must be able to sort the data based on certain categories. | In the interview these features were asked to be included in the solution so they can find the biggest orders or what product has the least stock. |
| The system must be able to automatically reduce stock quantity for each item. | From my interview the stakeholders said this would be very useful as tracking stock quantity allows the company to see when they need to buy more stock. |
| The interface must have a simple design and not be cluttered. | This makes it quicker for the users to become familiar with the system and was requested by the employees during the interview. |
| The GUI should have separate tabs or pages for each table and use buttons textboxes to add data. | During the interview they said this would be the easiest way to navigate the system. |
| The solution should calculate price orders and track customer spending. | In the interview with senior users they said these features would be useful as it allows the company to make more informed decisions. |
| It must allow you to apply discounts to certain customers’ orders. | This allows the sales team to negotiate and is essential if I am calculating prices. |
| The solution must be able to sort the data based off stock quantity, order date and discount rates. | The stock quantity lets the company see which stock is running low, order dates show most pressing orders and discount rates can be used to compare companies. |
| A notification should be sent when stock is below 10. | This was requested by the purchasing manager, so they are reminded when they need to order stock. |
| It should calculate the total amount of money a company has spent. | During interview this was asked to be incorporated so they can analyse the data and use it when deciding companies’ discounts. |
| The orders will be split into order headers, that will contain main information such a customer, and order lines for each stock item. | I looked at two existing solutions and both systems used this so there is no need to repeat information such a customer and delivery address for each type of stock the order includes making it more efficient. |
| The interface should use bold colours or colour codes. | When looking at Unleashed I found this makes the design look more aesthetically pleasing. |
| There should be a menu bar with symbols. | Both existing solutions I looked at had this feature to make it easier to navigate. |

The solution also has the following software requirements:

* PyQt5 module to create the user interface
* SQLite module to connect to the SQLite database
* uic module to load the user interface

# Success Criteria

|  |  |
| --- | --- |
| Specification Points | Testing |
| The solution must be able to store customer, stock and order information in a database. | I will use test data to check if the database stores the information. |
| There must be a feature to add new information into all the tables. | I will try to add test data into the each of the tables in the database. |
| You must be able to search the data stored in the database. | I will check if the search feature works correctly using test data. |
| The solution must include a user interface that is clear and easy to interpret. | I will do a survey to investigate if people find the interface easy to use and clear. |
| There must be a way to change existing data in the tables. | I will try to change some of the test data and check if it stores the new data correctly. |
| It must be able to sort the data based on certain categories. | I will check if you can change information stored using test data. |
| The system must be able to automatically reduce stock quantity for each item. | I will find out if my stock automatically decreases when orders are made. |
| The interface must have a simple design and not be cluttered. | I will do a survey to investigate if people think the interface is simple or too cluttered. |
| The GUI should have separate tabs or pages for each table and use buttons textboxes to add data. | I will check if my interface uses tabs or pages and how you add data using the interface. |
| The solution should calculate price orders and track customer spending. | I will use test data to see if it calculates the order prices correctly. |
| It must allow you to apply discounts to certain customers’ orders. | I will investigate if different order costs apply discounts appropriately to find prices. |
| The solution must be able to sort the data based off stock quantity, order date and discount rates. | I will try to sort the test data on stock quantity, order date and discount rates and will check if they are sorted correctly. |
| A notification should be sent when stock quantity is below 10. | I will use test data to find out if a notification is sent if an order is placed or a direct edit is made to the stock table causing an item to have less the 10 copies. |
| It should calculate the total amount of money a company has spent. | I will check if the system records the total amount a company had spent using test data. |
| The orders will be split into order headers, that will contain main information such a customer, and order lines for each stock item. | I will look at the database and interface to see if uses order headers and order lines or just one table for orders. |
| The interface should use bold colours or colour codes. | I will ask my stakeholders if they think the interface makes good use of colours. |
| There should be a menu bar with symbols. | I will and check if my interface uses symbols to help navigate the menu. |

Design

# Problem Decomposition

Since the problem is made up of smaller tasks, we can break down the problem to make solving each smaller problem easier and quicker. For this problem we can break down the databases into separate tables: stock, customer and orders. The orders table will be further broken down into order headers and order lines. The separate tables can use the same or similar methods for retrieving, adding and searching data.

# User Interface Design

The interface will use a variety of windows, so each window is clear and easy to navigate and avoid the interface looking cluttered.

## Main Window

The main window will be split into 3 tabs, so the interface is easy to navigate and a help button to explain how to use all the features. Each tab will contain the necessary tables as well as buttons to refresh, add and search stock.

Figure 4. Decomposition of the main window

Customer

**?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Title | Author | List Price | Quantity |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |

Figure 5. UI mock-up for the Stock Tab

Orders

Stock

Add

Refresh

Search

X

Cus kjkn

Order Management System

Customer

**?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Name | Email | Tel | Address |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |

Figure 6. UI mock-up for the Customer Tab

Orders

Stock

Add

Refresh

Search

X

Cus kjkn

Order Management System

Search

Customer

**?**

Order Headers:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Customer | Delivery Address | Delivery Charge | Order Date |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

Order Lines:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Book | Quantity | Line Price |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

Figure 7. UI mock-up for the Orders Tab

Orders

Stock

**?**

Add Header

Refresh

Add Line

X

Cus kjkn

Order Management System

**X**

Help Information

Customer

**?**

Figure 8. UI mock-up for the Help button

Orders

Stock

**?**

X

Cus kjkn

Order Management System

## Adding Data

When you push the Add button a dialog will appear to add the data. The dialog will be specific to the current tab as each table requires different information. It will include textboxes to for the user to type in the information and buttons to add the data to the database or close the dialog button.

Figure 9. Decomposition of the add dialog

X

Add Customer

Cancel

OK

Textbox for discount

Textbox for address

Tel:

Email:

Textbox for email

Name:

Textbox for telephone

Textbox for name

Address:

Discount:

Figure 10. UI mock-up for add customer dialog

X

Add Stock

List Price:

Title:

OK

Textbox for standard price

Textbox for book title

Cancel

Quantity:

Author:

Textbox for quantity in stock

Text box for book’s author

Figure 10. UI mock-up for add stock dialog