Requirements Engineering

DIY Shop System

Submitted By: T00225039

Computing with Software Development

Date Submitted: 08/12/2023

**Table of Contents**

[1. Introduction/overview 4](#_Toc152659649)

[2. Functional Components 5](#_Toc152659650)

[3. User Requirements 6](#_Toc152659651)

[3.1 DIY Shop System will Manage Suppliers 6](#_Toc152659652)

[3.2 DIY Shop System will Manage Type 6](#_Toc152659653)

[3.3 DIY Shop System will Manage items 6](#_Toc152659654)

[3.4 DIY Shop System will Perform administrative reporting 6](#_Toc152659655)

[4. System Requirements 7](#_Toc152659656)

[4.1. System Level Use Case Diagram 7](#_Toc152659657)

[4.2. Module 1 (Manage Supplier) 8](#_Toc152659658)

[4.2.1. Functional Requirement 1 (Add Supplier) 9](#_Toc152659659)

[4.2.2. Functional Requirement 2 (Update Details) 10](#_Toc152659660)

[4.2.3. Functional Requirement 3 (Manage Supplier Status) 12](#_Toc152659661)

[4.3. Module 2 (Manage Types) 13](#_Toc152659662)

[4.3.1. Functional Requirement 4 (Add Type) 14](#_Toc152659663)

[4.3.2. Functional Requirement 5 (Update Type Details) 15](#_Toc152659664)

[4.3.3. Functional Requirement 6 (Manage Type Status) 17](#_Toc152659665)

[4.4. Module 3 (Manage Items) 18](#_Toc152659666)

[4.4.1. Functional Requirement 7 (Add Item) 19](#_Toc152659667)

[4.4.2. Functional Requirement 8 (Update Item) 21](#_Toc152659668)

[4.4.3. Functional Requirement 9 (Manage Item Status) 23](#_Toc152659669)

[4.4.4. Functional Requirement 10 (Query Items) 25](#_Toc152659670)

[4.4.5. Functional Requirement 11 (Order Items) 26](#_Toc152659671)

[4.4.6. Functional Requirement 12 (Sell items) 28](#_Toc152659672)

[4.5. Module 4 (Preform Admin) 30](#_Toc152659673)

[4.5.1. Functional Requirement 13 (Yearly Revenue Analysis) 31](#_Toc152659674)

[4.5.2. Functional Requirement 14 (Yearly Item Analysis) 32](#_Toc152659675)

[5. System Model 33](#_Toc152659676)

[5.1. Level-0 DFD 33](#_Toc152659677)

[5.2. Level-1 DFD 34](#_Toc152659678)

[5.3. Level-2 DFD (Process P1: Manage Supplier) 35](#_Toc152659679)

[5.4. Level-2 DFD (Process P2: Manage Types) 36](#_Toc152659680)

[5.5. Level-2 DFD (Process P2: Manage Item) 37](#_Toc152659681)

[5.6. Level-2 DFD (Process P4: Preform Admin) 38](#_Toc152659682)

[6. Data Model (Class Diagram) 39](#_Toc152659683)

[6.1. Class Diagram 39](#_Toc152659684)

[6.2. Relational Schema 40](#_Toc152659685)

[6.3. Database Schema 40](#_Toc152659686)

[7. Conclusion 42](#_Toc152659687)

# Introduction/overview

This project is a DIY Shop System, It Will allow you to add, update or retire shop items and add what type of item you want to sell. You can update this type or manage the status of the item type.

You can order items from a supplier and sell items with this System, the results of either of those actions can be viewed using charts in the software. You can also view a chart on items sold during the year. You can also add and manage supplier data in this system.

# Functional Components

# User Requirements

## DIY Shop System will Manage Suppliers

* + 1. DIY Shop System will Add Supplier
    2. DIY Shop System will Update Suppliers Details

## DIY Shop System will Manage Type

* + 1. DIY Shop System will Add Type
    2. DIY Shop System will Update Type Details
    3. DIY Shop System will Manage Type Status

## DIY Shop System will Manage items

* + 1. DIY Shop System will Add Item
    2. DIY Shop System will Update Item
    3. DIY Shop System will Remove Item
    4. DIY Shop System will Query Items
    5. DIY Shop System will Restock Items
    6. DIY Shop System will Sell to Customer

## DIY Shop System will Perform administrative reporting

* + 1. DIY Shop System will produce a yearly revenue analysis
    2. DIY Shop System will produce a yearly Item analysis

# System Requirements

## System Level Use Case Diagram

The following system level use case diagram illustrates the high-level system requirements.

Supplier

Manager

Cashier

Customer

## Module 1 (Manage Supplier)

Manager

Supplier

<<includes>>

### Functional Requirement 1 (Add Supplier)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Add Supplier | |
| **Use Case Id** | 01 | |
| **Priority** | 1 | |
| **Source** | supplier | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | supplier | |
| **Description** | This function will add supplier Details to Supplier file | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Adds Supplier Details Successfully;** | **Step 1:** invoke add Supplier  **Step 3:** input data   * Name * Phone number * Email * Delivery fee   **Step 9:** continue | **Step 2:** Display UI  **Step 4:** validate data   * name * phone number * email * Delivery fee   **Step 5:** data is valid  **Step 6:** Display confirmation message  **Step 7:** save data onto supplier file  **Step 8:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| Data not valid |  | **Step 5:** data not valid  **Step 6:** display appropriate error message  **Step 7:** return to step 3 |
| **Conclusions** | Supplier details are added to supplier file | |
| **Post conditions** | Item From Supplier Can Be Added to Items file | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 2 (Update Details)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Details | |
| **Use Case Id** | 02 | |
| **Priority** | 2 | |
| **Source** | supplier | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | supplier | |
| **Description** | This function will update the details of the supplier | |
| **Preconditions** | Add supplier must be completed to have a supplier in the supplier file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager updates supplier Details Successfully** | **Step 1:** invoke update supplier details  **Step 4:** select supplier to be updated  **Step 5:** input data that requires updating   * Name * Phone number * email * Delivery fee   **Step 11:** Continue | **Step 2:** retrieve data of all suppliers from Supplier file   * supplier id * name * phone number * email * Delivery fee * status   **Step 3:** Display UI  **Step 6:** validate data   * Phone number * email * Delivery fee   **Step 7:** data valid  **Step 8:** Display confirmation message  **Step 9:** update data in supplier file  **Step 10:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| **Data not valid** | **Step 9:** Continue | **Step 7:** data not valid  **Step 8:** send appropriate error message |
| **File not found** |  | **Step 2:** supplier file has not been found  **Step 3:** display error message- “supplier file does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Supplier details are updated | |
| **Post conditions** | Updated details will be seen in query items | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 3 (Manage Supplier Status)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Remove Supplier | |
| **Use Case Id** | 03 | |
| **Priority** | 3 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function is to remove Supplier details from supplier file | |
| **Preconditions** | Add supplier must be completed to have a supplier in the supplier file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Remove Supplier from System** | **Step 1:** invoke Remove Suppliers  **Step 4:** select supplier  **Step 5:** Press Button to change supplier status  **Step 9:** Continue | **Step 2:** retrieve Supplier data from Supplier File   * Supplier id * Name * Phone number * Email * Delivery fee * Status   **Step 3:** display Ui  **Step 6:** send confirm message  **Step 7:** Update Supplier Status in supplier file  **Step 8:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| **File not found** |  | **Step 2:** supplier file has not been found  **Step 3:** display error message- “supplier file does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Supplier status has changed | |
| **Post conditions** | Supplier cannot be picked when adding item if status is changed to deactivated while it can be picked when activated | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Module 2 (Manage Types)

Manager

<<includes>>

### Functional Requirement 4 (Add Type)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Add Type | |
| **Use Case Id** | 04 | |
| **Priority** | 4 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will add Type Details to Type file | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Adds Type Details Successfully;** | **Step 1:** invoke add Type  **Step 3:** input data   * Name   **Step 9:** continue | **Step 2:** Display UI  **Step 4:** validate data   * name   **Step 5:** data is valid  **Step 6:** Display confirmation message  **Step 7:** save data onto supplier file  **Step 8:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| Data not valid |  | **Step 5:** data not valid  **Step 6:** display appropriate error message  **Step 7:** return to step 3 |
| **Conclusions** | Supplier details are added to supplier file | |
| **Post conditions** | Item of type added Can Be Inserted to Items file | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 5 (Update Type Details)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Details | |
| **Use Case Id** | 05 | |
| **Priority** | 5 | |
| **Source** | supplier | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | supplier | |
| **Description** | This function will update the details of the Type | |
| **Preconditions** | Add Type must be completed to have a Type in the Type file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager updates Type Details Successfully** | **Step 1:** invoke update Type details  **Step 4:** select Type to be updated  **Step 5:** input data that requires updating   * Name   **Step 11:** Continue | **Step 2:** retrieve data of all suppliers from Type file   * type id * name * status   **Step 3:** Display UI  **Step 6:** validate data   * name   **Step 7:** data valid  **Step 8:** Display confirmation message  **Step 9:** update data in Type file  **Step 10:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| **Data not valid** | **Step 9:** Continue | **Step 7:** data not valid  **Step 8:** send appropriate error message |
| **File not found** |  | **Step 2:** supplier file has not been found  **Step 3:** display error message- “supplier file does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Type details are updated | |
| **Post conditions** | Updated details will be viewed and used when managing items | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 6 (Manage Type Status)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Remove Supplier | |
| **Use Case Id** | 06 | |
| **Priority** | 6 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Supplier | |
| **Description** | This function is to remove Supplier details from supplier file | |
| **Preconditions** | Add supplier must be completed to have a supplier in the supplier file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Remove Supplier from System** | **Step 1:** invoke Remove Suppliers  **Step 4:** select Type  **Step 5:** Press Button to change Type status  **Step 9:** Continue | **Step 2:** retrieve Supplier data from Type File   * Type id * Name * status   **Step 3:** display Ui  **Step 6:** send confirm message  **Step 7:** Update Type Status in type file  **Step 8:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| **File not found** |  | **Step 2:** supplier file has not been found  **Step 3:** display error message- “supplier file does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Type status has changed | |
| **Post conditions** | Type cannot be picked when adding item if status is changed to deactivated while it can be picked when activated | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Module 3 (Manage Items)

<<includes>>

Manager

Cashier

Customer

### Functional Requirement 7 (Add Item)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Yearly Item Analysis | |
| **Use Case Id** | 07 | |
| **Priority** | 7 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will show the item types sold during the year | |
| **Preconditions** | Add supplier must be completed to have a supplier in the system | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Adds Item to System** | **Step 1:** invoke add Item  **Step 5:** input data   * name * Type id * Price * Cost * Supplier id * Quantity   **Step 11:** Continue | **Step 2:** retrieve data from supplier file   * Supplier id * Name   **Step 3:** retrieve data from types file   * Type id * Type name   **Step 4:** Display UI  **Step 6:** validate data   * name * Price * Cost * Quantity   **Step 7:** data is valid  **Step 8:** send confirm message  **Step 9:** input data to stock file  **Step 10:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| **Invalid data** | **Step 7:** Continue | **Step 5:** Data not valid  **Step 6:** send appropriate error message |
| **Conclusions** | Item is added to item file | |
| **Post conditions** | Item can be sold | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 8 (Update Item)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update item | |
| **Use Case Id** | 08 | |
| **Priority** | 8 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function is used to update an item | |
| **Preconditions** | Add item must be completed to have an item in the stock file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager updates item Data** | **Step 1:** invoke update item  **Step 6:** select item to be updated  **Step 7:** input data that requires updating   * Name * Type id * Price * Cost * Supplier id   **Step 12:** Continue | **Step 2:** retrieve data from stock file   * stock id * name * price * Cost * status   **Step 3:** retrieve data from type file   * Type id * Type name   **Step 4:** retrieve data from supplier file   * supplier id * name   **Step 5:** Display UI  **Step 8:** validate data   * name * Price * Cost   **Step 9:** is valid  **Step 10:** Send Confirmation Message  **Step 11:** update data in stock file |
| **Alternate Scenarios** | **Actor** | **System** |
| **Invalid data** | **Step 11:** Continue | **Step 9:** is not valid  **Step 10:** Display appropriate error message |
| **File not found** |  | **Step 2:** file has not been found  **Step 3:** display error message- “stock file does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Item is updated | |
| **Post conditions** | Updated item values will show in shop | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 9 (Manage Item Status)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Manage Item Status | |
| **Use Case Id** | 09 | |
| **Priority** | 9 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function is used to removed items | |
| **Preconditions** | Add item must be completed to have an item in the stock file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Removes item** | **Step 1:** invoke Remove item  **Step 6:** select item  **Step 7:** press button to change item status  **Step 11:** continue | **Step 2:** retrieve item data from stock file   * Stock id * Name * Price * Cost * status   **Step 3:** retrieve data from type file   * Type id * Type name   **Step 4:** retrieve data from supplier file   * Supplier id * Name   **Step 5:** display Ui  **Step 8:** send confirm message  **Step 9:** update item status in stock file  **Step 10:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| **File not found** |  | **Step 2:** file has not been found  **Step 3:** display error message- “stock file does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Item status has changed | |
| **Post conditions** | Item will not be available in shop | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 10 (Query Items)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Yearly Item Analysis | |
| **Use Case Id** | 10 | |
| **Priority** | 10 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function is used to query items | |
| **Preconditions** | Add item must be completed to have an item in the stock file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Query’s item** | **Step 1:** invoke query items  **Step 6:** search using name  **Step 8:** continue | **Step 2:** retrieve data from stock file   * Stock id * Name * Price * Cost * quantity   **Step 3:** retrieve data from type file   * Type id * Type name   **Step 4:** retrieve data from supplier file   * Name * Phone number * email   **Step 5:** Display Ui  **Step 6:** select data by name |
| **Alternate Scenarios** | **Actor** | **System** |
| **File not found** |  | **Step 2:** file has not been found  **Step 3:** display error message- “stock file does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Manager has queried data | |
| **Post conditions** | Manager can decide to contact supplier or make business decisions | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 11 (Order Items)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Yearly Item Analysis | |
| **Use Case Id** | 11 | |
| **Priority** | 11 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will show the item types sold during the year | |
| **Preconditions** | Add item must be completed to have an item in the stock file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
|  | **Step 1:** invoke order items  **Step 6: select** supplier  **Step 7:** select item to put into order list  **Step 8(optional):** remove item from order list  **Step 9:** press order button  **Step 13:** continue | **Step2:** retrieve data from stock file   * stock id * Name * quantity * Cost   **Step 3:** retrieve data from type file   * Type id * Type name   **Step 4:** retrieve data from supplier file   * Supplier id * name * Delivery fee   **Step 5:** Display UI  **Step 10:** send confirmation message  **Step 11:** update data in stock file   * quantity   **Step 12:** add data to order file   * order id * order date * total * supplier id   **step 13:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| **File not found** |  | **Step 2:** file has not been found  **Step 3:** Send Error message- “Files Required for Analysis does not exist, Contact an administrator”  **Step 4:** Return to Home |
| **Conclusions** | Items quantities are updated | |
| **Post conditions** | Items can be bought | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Functional Requirement 12 (Sell items)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Sell Items | |
| **Use Case Id** | 12 | |
| **Priority** | 12 | |
| **Source** | customer | |
| **Primary Business Actor** | Cashier | |
| **Other Participating Actors** | Customer | |
| **Description** | This function will sell the item by first putting the item into a basket and then manage the basket, after that you sell the item. | |
| **Preconditions** | Add item must be completed to have an item in the stock file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Customer buys item from shop** | **Step 1:** invoke order items  **Step 5:** select item to put into basket  **Step 6(optional):** choose to remove item from basket  **Step 12**: continue | **Step 2:** retrieve data from stock file   * Stock id * Name * Price * Quantity   **Step 3:** retrieve data from types file   * Type name   **Step 4:** Display UI  **Step 7:** send confirmation message  **Step 8:** add data to sales file   * sale id * Sale date * total   **Step 9:** add data to basket file   * Stock id * Sale id * Quantity bought   **Step 10:** update data in stock file  Quantity  **Step 11:** reset Ui |
| **Alternate Scenarios** | **Actor** | **System** |
| **Data failed to save** |  | **Step 8:** data cannot be saved to files  **Step 9:** Send error message- “Data failed to save, Contact your administrator” |
| **File not found** |  | **Step 2:** supplier file has not been found  **Step 3:** display error message- “supplier file does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Items are sold | |
| **Post conditions** | Profit per month will be viewed in yearly revenue analysis function and items sold per year will be viewed in yearly item analysis function | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Module 4 (Preform Admin)

Manager

### Functional Requirement 13 (Yearly Revenue Analysis)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Yearly Revenue Analysis | |
| **Use Case Id** | 13 | |
| **Priority** | 13 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This will show the revenue made during the year month by month | |
| **Preconditions** | Sell item must be completed to have sales data in the sales file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Reviews Yearly Revenue** | **Step 1:** invoke add item to basket  **Step 4:** Select year | **Step 2:** retrieve data from sales file   * Total * Date Sold   **Step 3:** Display UI  **Step 5:** Display Data in Bar chart format |
| **Alternate Scenarios** | **Actor** | **System** |
| **File not found** |  | **Step 2: Sale** file has not been found  **Step 3:** Send Error message- “Sales File does not exist, Contact an administrator”  **Step 4:** Return to Home |
| **Conclusions** | The data will be shown in bar chart format comparing the profit during the months of the year | |
| **Post conditions** | Owner can compare the profits made compared to other years | |
| **Business Rules** |  | |
| **Implementation Constraints** | The sales file must exist | |

### Functional Requirement 14 (Yearly Item Analysis)

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Yearly Item Analysis | |
| **Use Case Id** | 14 | |
| **Priority** | 14 | |
| **Source** | manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will show the item types sold during the year | |
| **Preconditions** | Sell item must be completed to have data in sales file and basket file | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Name** | **System** |
| **Manager Reviews Item Types Sold During the year** | **Step 1:** invoke add item to basket  **Step 6:** Select year | **Step 2:** retrieve data from stock file   * Type   **Step 3:** retrieve data from sales file   * Date Sold   **Step 4:** retrieve data from basket file   * Amount sold   **Step 5:** Display UI  **Step 7:** Display Data in pie chart Format |
| **Alternate Scenarios** | **Actor** | **System** |
| **File not found** |  | **Step 2:** stock file has not been found  **Step 3:** display error message- “files required for function does not exist, contact an administrator”  **Step 4:** return to home |
| **Conclusions** | Manager can view data about item popularity | |
| **Post conditions** | Manager is informed on which item types are selling then other item types | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

Buys items

customer

Live auction system

Receives Order

Gives information

supplier

## Level-1 DFD

Send item details

supplier

Manage Types

P2

Gives information

supplier

P1

Retrieve supplier details

Retrieve type details

Saves supplier values

save type details

supplier file

D1

Retrieve supplier details

Type file

D2

Retrieve type details

Order file

D4

Save order details

Save item values

Manage items

P3

stock file

D3

Buys items

customer

retrieve item details

Save basket values

Saves sales values

Sales file

D5

Retrieve sales values

basket file

D6

admin

P4

Retrieve basket values

## Level-2 DFD (Process P1: Manage Supplier)

Add supplier

P 1.1

Save supplier values

Gives information

save supplier values

Update supplier details

P 1.2

Retrieve supplier values

Gives information

supplier

supplier file

D1

update supplier values

Remove supplier values

could stop supplying

Retrieve supplier values

supplier

P 1.3

save supplier values

## Level-2 DFD (Process P2: Manage Types)

Save type values

Add type

P 2.1

Retrieve type values

Type file

D2

Update Type

P 2.2

Save type values

Manage Type Status

P 2.3

Save type values

Retrieve type values

## Level-2 DFD (Process P2: Manage Item)

Save basket values

Buys item

customer

Basket file

D6

Retrieve supplier values

Sell Items

p 2.6

supplier file

D1

Save sales values

Retrieve type values

Retrieve supplier values

Save item values

Sales file

D5

type file

D2

Retrieve type values

retrieve item values

Add Item

P 2.1

Save item values

Item file

D3

Retrieve supplier values

Retrieve supplier values

supplier file

D1

Retrieve type values

Retrieve type values

Manage Item Status

P 2.3

Update Item

P 2.2

type file

D2

Retrieve item values

Retrieve item values

Item file

D3

Save item values

Save item values

Retrieve supplier values

Query item

P 2.4

supplier file

D1

Retrieve supplier values

Retrieve type values

Retrieve item values

type file

D2

Retrieve type values

Retrieve item values

Item file

D3

Order item

P 2.5

Receives order

supplier

Add Order values

Orders file

D4

## Level-2 DFD (Process P4: Preform Admin)

Retrieve order values

Orders file

D4

basket file

D4

Sales file

D3

Stock file

D2

Retrieves amount

Retrieves stock type

Retrieves sale dates

Retrieves sale values

Yearly Revenue Analysis

P 4.1

Yearly item analysis

P 4.2

# Data Model (Class Diagram)

This will be the data model of DIY Shop System. The tables included will be suppliers for the supplier values, stock for the items in stock, sales for the sales made and basket for connecting the item with the sale and will store the amount bought

## Class Diagram

Supplier id

Name

Phone number

Email

Delivery fee

Status

Supplier

Type id

Name

status

type

1

Stock id

Name

Price

Cost

Quantity

status

Stock

1

0..\*

0..\*

Order id

order date

total

Order

1

0..\*

Quantity Sold

price

Basket

Sale id

Sale date

total

Sales

1..\*

1

## Relational Schema

Supplier (Supplier id, name, phone number, delivery fee, email, status)

Type (type id, name, status)

Stock (Stock id, name, type id, price, cost, Supplier id quantity, status)

Order (order id, order date, total)

Sales (Sale id, Sale date, total)

Basket (Stock id, Sale id, Quantity Sold, price)

## Database Schema

Relation Suppliers

Supplier id numeric (3) not null, unique

Name varchar2(20) not null,

Phone number char (10) not null,

Email varchar2 (20) not null,

Delivery fee numeric (7,2) not null,

Status char (1) default ‘a’

Primary key Supplier id

Relation Types

Type id numeric (3) not null, unique

Name varchar2(20) not null

Status char (1) default ‘a’

Primary key Type id

Relation stock

Stock id numeric (3) not null unique

Name varchar2 (20) not null

Type id numeric (3) not null,

Price numeric (7,2) not null,

Cost numeric (7,2) not null,

Supplier id numeric (3) not null,

Quantity numeric (3) not null,

Primary key Stock id,

Status char (1) default ‘a’

Foreign key Supplier id references Suppliers

Foreign key type id reference types

Relation Order

Order id numeric (3) not null, unique

Order date Date not null

Total numeric (7,2) not null

Primary key Order id

Relation Sales

Sale id numeric (4) not null unique

Sale date Date not null

Total numeric (7,2) not null

Primary key Sale id

Relation Basket

Stock id numeric (3) not null

Sale id numeric (4) not null

Quantity Sold not null

Foreign key Stock id references Stock

Foreign key Sale id references Sales

Primary key Stock id, Sale id

# Conclusion

This project will be used to Add, Manage, Order and Sell items in a DIY shop and allow data generated to be viewed using charts.