## Develop an order management system

### Scenario

XYZ & Co. has requested your services for developing a multi-layer data-driven desktop application. They need an Order Management System (OMS) so their employees can record and manage custom orders of their available stock items.

XYZ & Co. has already been working with a consultant and provided you with the client requirements for the product as listed below.

Download and unzip the resource folder (Cl\_Programming\_2\_AE\_Pro\_1of3\_SR1.zip) to access files referred to within the assessment.

#### Product description

The OMS will be used for recording and managing customer orders and tracking stock item levels.

##### Adding a new order

A typical scenario of use starts when a customer contacts XYZ & Co and places an order. At this point an employee of XYZ & Co will start to enter the details of the order into the system. When a new order is created, a unique identifier is generated and assigned to the order. The current date and time is also captured and associated with the order. If at any time the employeecancels entering the order, any data relating to the Order that has already been saved to the database must be deleted.

##### Adding order line items

The employee will then add the order line items by selecting a stock item, entering the quantity and clicking the ‘Add Item’ button. The employee will follow this process for every order line item.

If the quantity entered for a stock item is higher than what is currently available in stock a warning must be displayed. Validation is also required to ensure a value greater than zero is entered for the quantity.

When an order is first created it is flagged as ‘New’ indicating order line items are still being added. After all of the items have been added the employee will ‘Submit’ the order, this will result in the order being flagged as ‘Pending’. An order can be pending for any period of time but most orders are processed within 2 – 3 business days.

It is also required that order items can be deleted from an order whilst it is flagged as ‘New’.

##### Viewing and processing orders

When the application is launched, the employee will be presented with a view that lists all of the orders in a data grid.

Each row will display the unique identifier (Id), date/time, number of line items, total and the current state of the order (New, Pending, Complete or Rejected).

The employee will then be able to select one of the orders and navigate to the ‘Order Details’ view, which will display the order details including the line items (SKU, Name, Price, Quantity and Total).

If the order is flagged as ‘Pending’ a ‘Process’ button will be visible so the order can be processed. When an order is processed the quantity of each item ordered is checked against the quantity currently in stock. If there is enough items in stock for the quantity ordered of all line items, the quantity ordered is decremented from the stock available and the order it is flagged as ‘Complete’. If there is not enough stock available at the order is flagged as ‘Rejected’, see **Figure 1 - Order State Flow Diagram**.

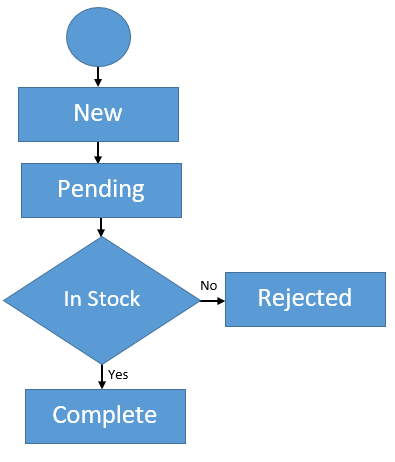


Figure 1 - Order State Flow Diagram

##### Stock items

Each stock item has a unique identifier (SKU), name, price and current stock level (number of items currently in stock). When an order is processed the number of items in stock is decremented accordingly.

##### Navigation

A diagram depicting the navigation between the different views in the application is shown in **Figure 2 - Application Navigation**.

Diagram

Description automatically generated

Figure 2 - Application Navigation

##### 5.1 - Launch Application

* The application automatically navigates to the **[Orders View].**
* All **Orders** in the system are displayed as rows in a data grid.
* Each row has a link for navigating to the **[Order Details View]** for the selected row.

##### 5.2 - Add Order (New)

* The application navigates to the **[Add Order View]**.
* Initially there are no **Order Line Items**.
* A **Unique Identifier** is generated, assigned to the Order and displayed.
* The current **Date and Time** is captured, recorded and displayed.
* The **Total** for the **Order** is displayed, initially it is **$0.00** as there are no **Order Line Items**
* When an **Order** is first created it is flagged as **New**

##### 5.3 - Add Order Line Item

* The application navigates to the **[Add Order Item View]**
* A list of **Stock Items** are displayed as rows in a data grid.
* The employee is able to select a **Stock Item** by clicking on the related row in the data grid.
* A Textbox is displayed for entering the desired order **Quantity** of the selected **Stock Item**
* A button with the value **[Add Item]** is displayed. When a **Stock Item** has been selected and a valid **Quantity** for the **Order Line Item** has been entered clicking the **[Add Item]** button will create a new **Order Line Item** and add it to the **Order**.
* If a Stock Item was selected and a valid **Quantity** entered the application navigates back to the **[Add Order View]** after the **[Add Item]** button has been clicked:
* The new **Order Line Item** is displayed in a data grid as rows with the columns Stock Item’s Unique Identifier (SKU), Stock Item’s Name (Item), Quantity, Price, Total. See **Table 3 - Order Item Data Grid [Add Order View]**.
* The **Order Total** is updated to include the **Total** for the new **Order Line Item**
* If the **Quantity** is greater than the current Stock Level, a warning is displayed but the new Order Line Item is still created.
* Info: There are currently not enough items in stock. Requested **X**, In stock: **X**. This order might be rejected if there is not enough stock on hand when the order is being processed.
* **Note**: **X** represents the quantity ordered and quantity currently in stock
* If no Stock Item was selected, no quantity or an invalid quantity was entered, a message will be displayed:
  + Info: Please select a Stock Item
  + Info: Please enter the Quantity
  + Info: The Quantity must be greater than zero

Table 3 - Order Item Data Grid [Add Order View]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SKU | Item | Price | Quantity | Total |
| 3 | Chair | $50.00 | 3 | $150.00 |

##### 5.4 – Submit Order (New > Pending)

* After all the required **Order Line Items** have beenadded to the **Order**, the **Order** is **submitted**. This is done by clicking the **[Submit]** button located on the **[Add Order View]**.
* After the **[Submit]** button has been clicked the state of the **Order** changes from **New** to **Pending** and the application navigates back to the **[Orders View]**.
* The details of the **Pending Order** will be displayed as a row in the data grid along with a **[Details]** button for navigating to the **[Order Details View]** of the selected **Order**, see **Table 4 - Order Row (Orders View)**.
* **Note**: At any time an **Order** can be **Cancelled;** if this is done any data related to the order that has been saved in the database must be deleted.

Table 4 - Order Row (Orders View)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | DateTime | Items | Total | State |  |
| 52 | 10/7/2019 10:21:33 PM | 3 | $640.00 | Pending | Details |

##### 5.5 –Order Details View

* On clicking the **[Details]** button for a selected **Order** the application navigates to the **[Order Details View]**.
* The **[Order Details View]** is similar to the **[Add Order View]** with one exception, there is no option for adding new **Order Items**. The Order Number, Date & Time, Total, State and Order Line Items (SKU, Description, Quantity, Price and Total) are visible.
* If the **Order** is **Pending** a **[Process]** button is displayed
* If the **Order** is **Complete** or **Rejected** no button is displayed
* After the **[Process]** button has been clicked the following process occurs:
* The **Quantity** of each **Order Line Item** is checked against the amount **In Stock** for the related **Stock Item**.
* If there is enough items **In Stock** for **ALL** items ordered, the **Order State** is changed to **Complete**.
* If there is not enough **Stock** for **ANY** of the items, the **Order State** is changed to **Rejected**.
* It is also possible to navigate back to the **[Orders View]** from the **[Order Details View]** by clicking the **[Orders]** navigation button available in the **[Order Details View]**.

### Technical specifications

* OrderManagementSystem (Solution)
  + Domain (Class library)
  + DataAccess (Class library)
  + Controllers (Class library)
  + UI (WPF).

### Development principles and practices

* All coding must comply with the coding standards as described at [C# Coding Standards and Naming Conventions](https://www.dofactory.com/reference/csharp-coding-standards).
* All methods must have internal XML documentation as illustrated below in **Figure 3: C# Internal XML comments (example)**; further information is available at [Recommended Tags for Documentation Comments (C# Programming Guide)](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/xmldoc/recommended-tags-for-documentation-comments).

Graphical user interface, text

Description automatically generated

Figure 3: C# Internal XML comments (example)