Note: This document serves as a high-level overview and may not cover all rules or key features. Its purpose is to offer additional clarity for you and your team. If you have any questions, please see your instructor.

Receiving Purchase Orders Rules

Access & Authentication:

- Employees must log onto the system.
- Access is restricted to authenticated users within the `Receiving Role`.
- Either an Associate or a Department Head manages receiving.
- The Employee's full name should appear on the form.

Order Verification:

- Receiving is done after the purchase order was placed.
- Receiving involves manual verification against the vendor's shipping sheet.
- For items not matching the order:
    - Note the received quantity.

    - Document damaged items.
    - Document items sent in error.

Order Processing:

- Display all outstanding orders (orders not closed with an order date) for selection.
- Order details are shown upon selection for verification.
- Receivers can:
    - Enter the number of items received.
    - Specify refused or returned items and provide a reason.
    - Document items shipped in error.

Receiving:

- Receive will do a bulk update processing of the current displayed purchase order
    - A ReceiveOrder Entry is made
    - A ReceiveOrderDetail entry is made for each stockitem received, indicating quantity received
    - The StockItem QuantityOnHand is adjusted (increased) by the amount received
    - The StockItem QunatityOnOrder is adjusted (decreased) by the amount received
    - A ReturnOrderDetail entry is made for each stockitem returned, indicating the quantity returned and the reason

    - A ReturnOrderDetail entry is made for each vendor item in the unordered purchase cart, indicating the description, quantity returned, reason (not ordered), and the vendor part number. After which, empty the cart.
    - The order is checked to see if it can be closed (no outstanding items). If the order is complete, close the order.

Force Close:

- An order can be forcibly closed based on management decisions.
- Reasons for a force close might include supplier issues or product unavailability.
- A reason must be provided when using `Force Close`.

Form Display & Entry:

- Only process one order at a time.
- Display:
    - Order PO id
    - Order Date
    - Vendor Name & Phone
    - StockItem details (ID, Description, Quantity On Order, Quantity Outstanding)
- Entry fields for:
    - Received Quantity

    - Returned Quantity & Reason

Unordered Purchase Items:

- Receivers identify and process items not on the original order.
- Entries include Description, VendorPartNumber, and Quantity.
- These entries are stored in a temporary cart, emptied after processing.

Command & Process Buttons:

- `View Order` : Retrieves and displays the order details.
- `Receive` : Updates the displayed PO, adjusts stock item quantities, processes returned items, and checks if the order can be closed.
- `Force Close` : Closes the order and adjusts stock item entries, providing a reason is mandatory.
- For unordered items:
    - `Insert` : Adds an item to the unordered item cart.
    - `Remove` : Deletes an item from the unordered item cart.

Images (Samples)

Receiving

> Welcome Hess Agonor!

Outstanding Orders

Select an outstanding order to receive...

	Order ID	Order Date	Vendor Name	Vendor Phone Number
View Order	358	Wednesday, November 29, 2017	HandTools Wholesale	780.421.1265

Receiving

> Welcome Hess Agonor!

Order Details

Purchase Order: 358

Purchase Order Date: Wednesday, November 29, 2017

Return to Outstanding Orders

Vendor: HandTools Wholesale

Vendor Phone Number: 780.421.1265

11206-106 ST. Edmonton

AB T5R6H7

Stock Item ID	Stock Item Description	Qty On Order	Qty Outstanding	Receive	Return	Reason
34	Dewalt Multi Speed Sander	10	10	0	0	Enter a reason for the return...
5566	Footprint Tools Wood Chisel Set 6	15	15	0	0	Enter a reason for the return...
5567	Makita 13 & 1 Racheting Screw Driver	144	144	0	0	Enter a reason for the return...
5572	Dewalt Folding JabSaw	71	71	0	0	Enter a reason for the return...

<First> <Prev> **1** <Next> <Last>

Receive | Force Close | Reason: Enter a reason for force closing the PO... | Clear

Unordered Returns

	CID	Description	VSN	Qty
Insert		Barbies	992145	17

View Models

View models and methods are used as examples, and you are welcome to modify them as needed in your coding example. Please include comments to explain any modifications or choices you make.

```csharp
public class OutstandingOrderView
{
    public int PurchaseOrderID { get; set; }
    public DateTime? OrderDate { get; set; }
    public string VendorName { get; set; }
    public string VendorPhone { get; set; }
}
```

```csharp
public class VendorView
{
    public string VendorName { get; set; }
    public string Address { get; set; }
    public string City { get; set; }
    public string Province { get; set; }
    public string PostalCode { get; set; }
    public string VendorPhone { get; set; }
}
```

// Nov 8 // Added public int ReceiveOrderID { get; set; } // Nov 9 // Remove ReceiveOrderID

```csharp
public class ReceivingView
{
    public int PurchaseOrderID { get; set; }
    public DateTime? OrderDate { get; set; }
    public VendorView Vendor { get; set; }
    public bool CanBeClosed { get; set; }
    public List<ItemDetailView> ItemDetails { get; set; }
    public List<UnorderedReturnItemView> UnorderedReturnItems { get; set; }
}
```

```csharp
public class ItemDetailView
{
    public int PurchaseOrderDetailId { get; set; }
    public int StockItemId { get; set; }
    public string StockItemDescription { get; set; }
    public int QtyOnOrder { get; set; }
    public int QtyOutstanding { get; set; }
    public int QtyReceive { get; set; }
    public int QtyReturn { get; set; }
    public string Reason { get; set; }
}
```

```csharp
public class ReturnedOrderDetailView
{
    public int ReturnedOrderDetailID { get; set; }
    public int ReceiveOrderID { get; set; }
    public int PurchaseOrderDetailID { get; set; }
    public int UnOrderedItemID { get; set; }
```

```
    public string ItemDescription { get; set; }
    public int Quantity { get; set; }
    public string Reason { get; set; }
    public string VendorStockNumber { get; set; }
    public bool RemoveFromViewFlag { get; set; }
}
```

// Nov 8 // Rename UnOrderedItem to UnOrderedItemID // Rename PurchaseOrderID to ReceiveOrderID

```csharp
public class UnorderedReturnItemView
{
    public int UnOrderedItemID { get; set; }
    public int ReceiveOrderID { get; set; }
    public int ItemID { get; set; }
    public string Description { get; set; }
    public string VSN { get; set; }
    public int Quantity { get; set; }
    public bool RemoveFromViewFlag { get; set; }
}
```

Methods

```csharp
public List<OutstandingOrderView> PurchaseOrders_FetchOutstandingOrders()
public ReceivingView PurchaseOrders_FetchOrderDetails(int purchaseOrder)
public List<UnorderedReturnItemView> UnOrderedItems_FetchUnOrderedItems()

public ReceivingView ReceiveOrders_ProcessReceivedPurchaseOrder(ReceivingView receiving)
public void or bool PurchaseOrders_ForceCloseOrder(string reason, ReceivingView receiving)
```

```
//  Within the UI
public void UnOrderedItems_AddUnOrderedItem(UnorderedReturnItemView itemForProcessing,
                                            int? poId)
public void UnOrderedItems_RemoveUnOrderedItem(int? itemId)
```