

Purchasing

View Models

```
public class VendorView
{
    public int VendorID { get; set; }
    public string VendorName { get; set; }
    public string HomePhone { get; set; }
    public string City { get; set; }
}
```

csharp 

```
public class PurchaseOrderView
{
    public int PurchaseOrderID { get; set; }
    public int PurchaseOrderNumber { get; set; }
    public int VendorID { get; set; }
    public decimal SubTotal { get; set; }
    public decimal GST { get; set; }
    public int EmployeeID { get; set; }
    public List<PurchaseOrderDetailView> PurchaseOrderDetails {get;
set;}}
```

csharp 

```
public class PurchaseOrderDetailView
{
    public int PurchaseOrderDetail { get; set; }
    public int PartID { get; set; }
    public string Description { get; set; }
    public int QOH { get; set; }
    public int ROL { get; set; }
    public int QOO { get; set; }
    public int QTO { get; set; }
    public decimal Price { get; set; }
}
```

csharp 

```
public class ItemView
{
    public int PurchaseOrderDetail { get; set; }
    public int ID { get; set; }
    public string Description { get; set; }
    public int QOH { get; set; }
    public int ROL { get; set; }
    public int QOO { get; set; }
    public int Buffer { get; set; }
    public decimal Price { get; set; }
}
```

Business Processing Requirements

Query Service Methods

List of Vendor

```
public list<VendorView> GetVendors()
{

}
```

Purchase Order

```
public PurchaseOrderView GetPurchaseOrder(int vendorID)
{

}
```

- check if there is a existing purchase order with no place order date (empty order date)
 - yes (existing purchase order)
 - return purchase order view

- return list of PurchaseOrderDetailView from database
 - no (no purchase order exist or no purchase order without empty order date)
 - return a empty purchase order view with VendorID set.
 - return a list of suggested items based on reorder level (ReorderLevel - (QuantityOnHand + QuantityOnOrder) > 0)
-

List of Inventory Items

```
public List<ItemView> GetInventory(int vendorID)
{

}
```

csharp 

- return list of ItemView for the vendor
 - Update the list by removing anything that is on the purchase order detail list
-

Command Methods

Save

```
PurchaseOrder_Save(PurchaseOrderView purchaseOrderView)
{

}
```

csharp 

- check if place order date is empty
 - no
 - throw NullArgumentException that this order is close
- check if there is purchase order details (greater than zero)
 - no
 - throw NullArgumentException that there needs to be purchase order details
- foreach each purchase order detail
 - Quantity greater than zero
 - no

- `errorList.Add(new Exception($"Quantity for PO item () must have a positive non zero value"));`
 - Price greater than zero
 - no
 - `errorList.Add(new Exception($"Price for PO item () must have a positive non zero value"));`
 - check for errors
 - Yes
 - throw the list of all collected exceptions
 - no
 - save all work to the database
 - if purchase order number is zero (new order), generate a new number
-

Place Order

csharp 

```
PurchaseOrder_Save(PurchaseOrderView purchaseOrderView)
{

}
```

- check if place order date is empty
 - no
 - throw `NullArgumentException` that this order is close
- check if there is purchase order details (greater than zero)
 - no
 - throw `NullArgumentException` that there needs to be purchase order details
- foreach each purchase order detail
 - Quantity greater than zero
 - no
 - `errorList.Add(new Exception($"Quantity for PO item () must have a positive non zero value"));`
 - Price greater than zero
 - no
 - `errorList.Add(new Exception($"Price for PO item () must have a positive non zero value"));`

- check for errors
 - Yes
 - throw the list of all collected exceptions
 - no
 - save all work to the database
 - if purchase order number is zero (new order), generate a new number
 - will set the OrderDate for the current order
 - update the QuantityOnOrder for each stock item on the purchase order
-

Delete Order

```
DeleteOrder (int purchaseOrderID)
{

}
```

csharp 

- check if place order date is empty
 - no
 - throw NullArgumentException that this order cannot be deleted
 - check for errors
 - Yes
 - throw the list of all collected exceptions
 - no
 - delete all purchase order detail records
 - delete purchase order record
-

Form Methods

Add Part

```
AddPart(int partId)
{

}
```

csharp 

- check if part exists on the purchase order detail table
 - yes
 - `errorList.Add(new Exception($"Part ID " already exist on the purchase order detail table and cannot be added again));`
 - no append the part to the purchase order detail list
-

Remove Purchase Order Detail

csharp 

```
RemovePurchaseOrderDetail(int partId)
{

}
```

- check if part exists on the parts table
 - yes
 - `errorList.Add(new Exception($"Part ID " already exist on the parts table and cannot be added again));`
 - no
 - append the part to the parts list
-

Refresh

csharp 

```
Refresh()
{

}
```

- Update the Subtotal and tax values
-

Clear

csharp 

```
Clear()
{

}
```

- Empties bound properties
-