Receiving

# ERD for UX process

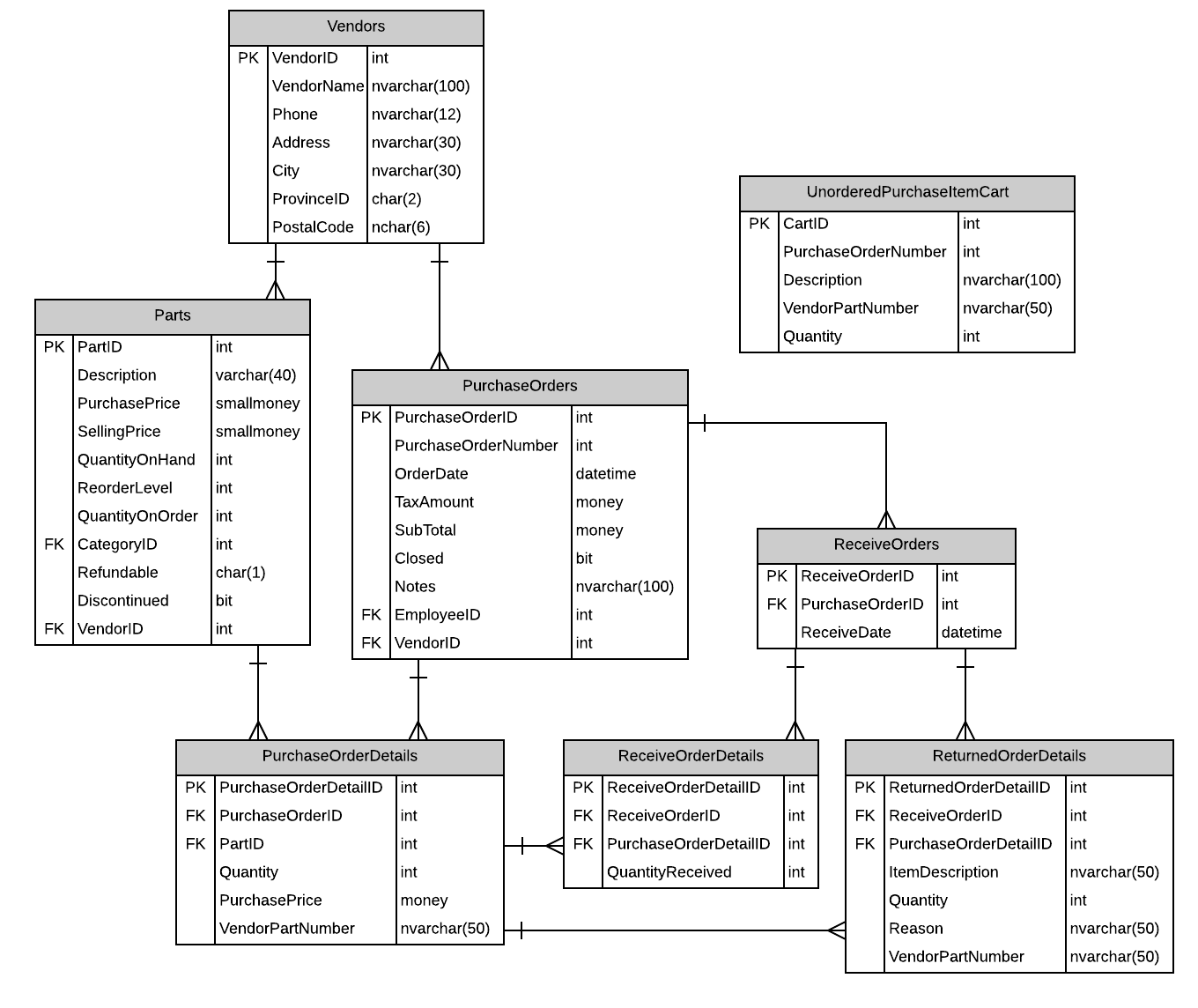


Figure ERD for Receiving Database

# Possible UX Interface

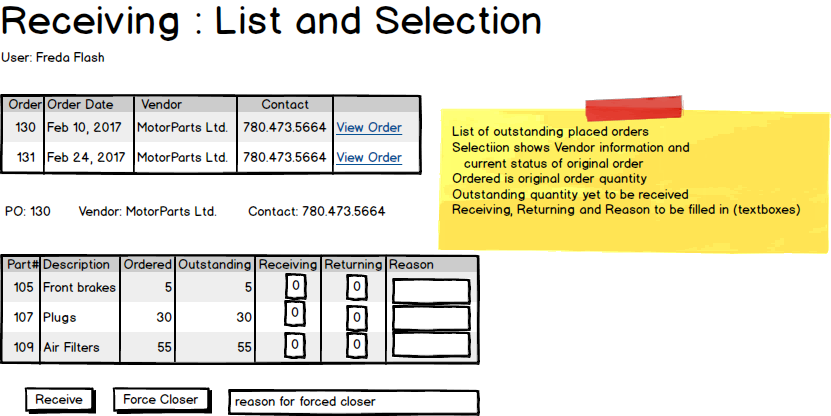


Figure Open Order Selection List with a selected order

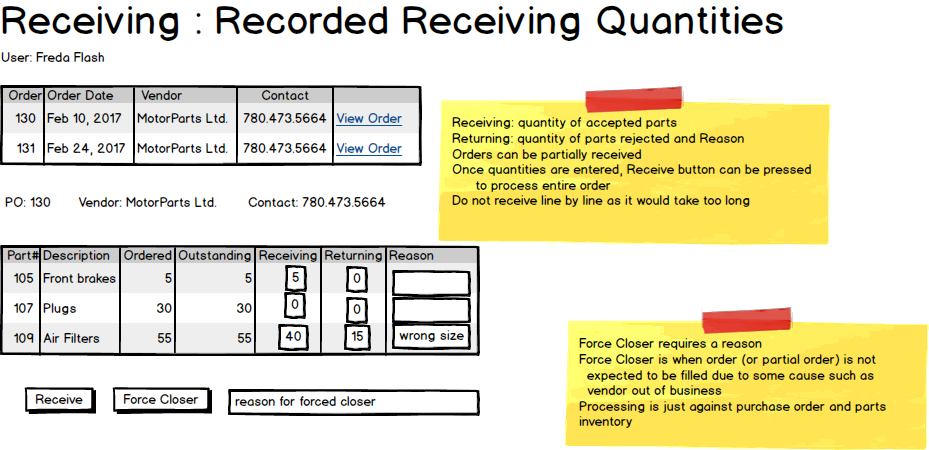


Figure Receiving a partial order

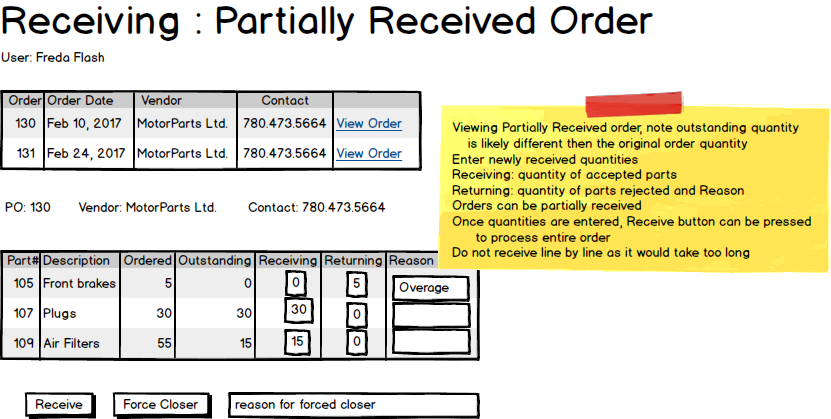


Figure Receiving additional parts to an order already partially received

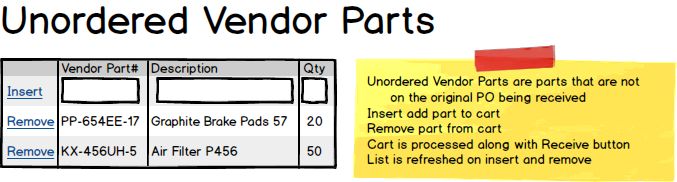


Figure Tracking unordered vendor parts shipped with an order which will be returned

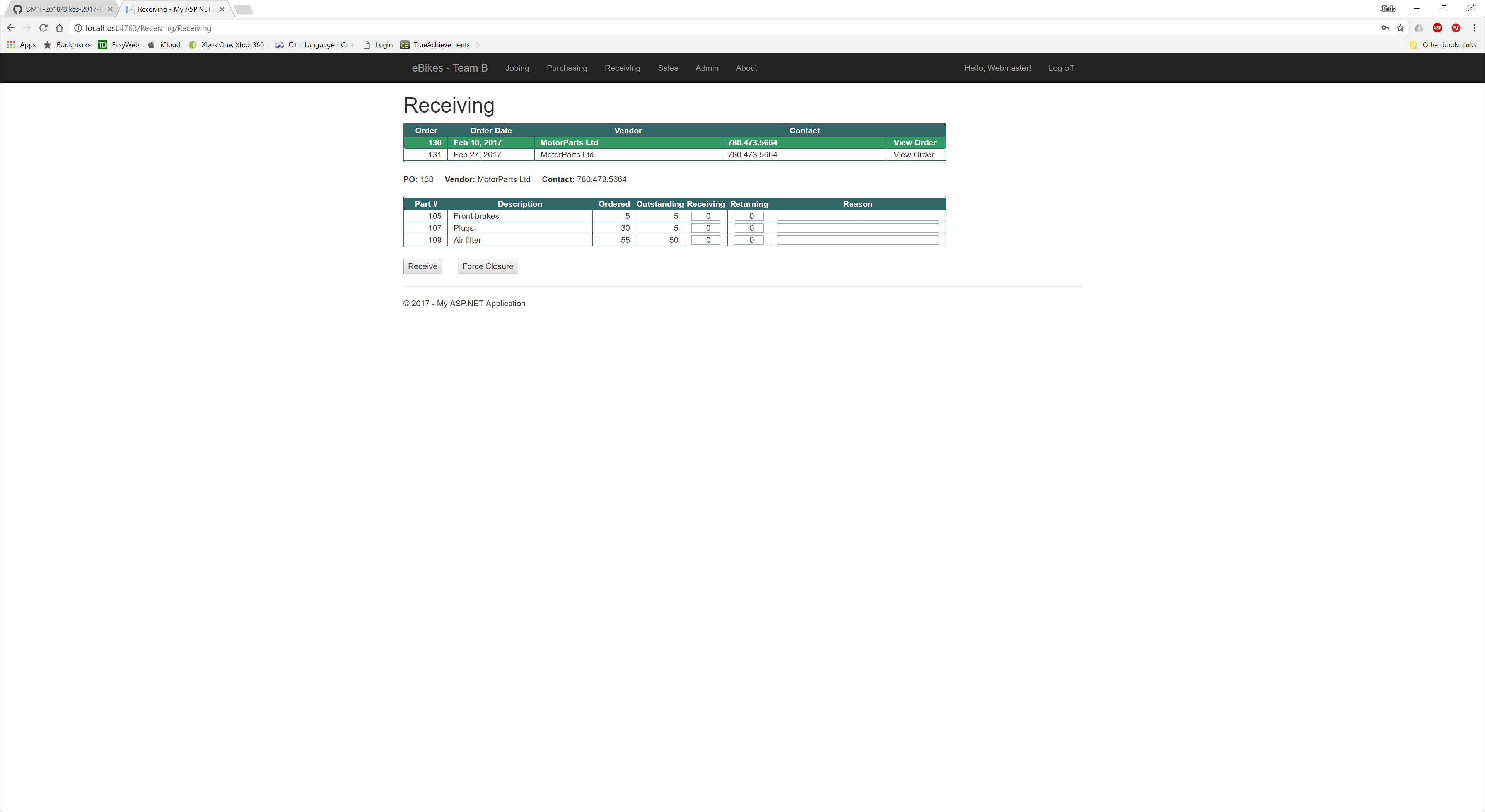


Figure - Receiving Screen Mock-up Minus Cart

# Event and Wiring Summations

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| OutstandingOrdersGridView | | * Wired via ODS * Makes OutstandingOrderDetailsGridView appear in code-behind * CommandField Makes ReceiveBtn, ForceCloseBtn, SelectedOrderFormView and OutstandingOrderDetailsListView appear |
| **BBL Class(es) and Method (s)** | | |
| PurchaseOrderController | List<PurchaseOrder> PurchaseOrders\_ListOutstandingOrders  Gets a list of orders that are not closed, has a order date and a Purchase Order Number | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PurchaseOrders(R) | | OutstandingPurchaseOrders(P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| SelectedOrderFormView | | * None * Wired Via ODS |
| **BBL Class(es) and Method (s)** | | |
| PurchaseOrderDetailsController | List<OutstandingPurchaseOrderDetails> PurchaseOrderDetails\_GetOutstandingOrdersByPurchaseOrder  Retrieve a list of outstanding items on the order id supplied | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PurchaseOrderDetails (R) | | OutstandingPurchaseOrderDetails (D) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| OutstandingDetailsGridView | | * None * Wired via ODS |
| **BBL Class(es) and Method (s)** | | |
| PurchaseOrderDetailsController | List<OutstandingPurchaseOrderDetails> PurchaseOrderDetails\_GetOutstandingOrdersByPurchaseOrder  Retrieve a list of outstanding items on the order id supplied | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PurchaseOrderDetails (R) | | OutstandingPurchaseOrders(P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| ReceiveBtn | | OnCommand()   * Validate data * Create a new Receiving Order * Create new ReceivingOrderDetail for each part received * Increase the QuantityOnHand of Part * Decrease the QuantityOnOrder of Part * Make a returnOrderDetail for each return part * Add anything in the cart to returnorderdetails * Close the order if completed |
| **BBL Class(es) and Method (s)** | | |
| ReceiveOrderController | Int ReceiveOrder\_ReceiveNewOrder (ReceiveOrder newReceiveOrder)  Adds a new Receive Order | |
| ReceiveOrderDetailController | Int ReceiveOrderDetail\_ReceiveNewOrderItem (ReceiveOrderDetail newReceiveOrderDetail)  Adds a new Receive Order item | |
| PartController | Part Part\_Get (int partID)  Get part based off partID  int Part\_Update(Part item)  Update Part to parameter item | |
| ReturnOrderDetailController | int ReturnOrderDetail\_AddNewReturnOrderDetail(ReturnedOrderDetail newReturnOrderDetail)  Adds a new Return Order Detail for each part being returned | |
| UnorderedPurchaseItemCartController | List<UnorderedPurchaseItemCart> UnorderedCartItems\_List()  Gets all Items in the Cart  void UnorderedCartItems\_EmptyCart()  Removes all items from the cart | |
| PurchaseOrderController (U) | int PurchaseOrder\_CloseOrder(int purchaseOrderID)  Sets the order to status of Closed | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PurchaseOrders(U), ReceiveOrders(C), ReceiveOrderDetails(C), Parts(R,U), ReturnOrderDetails(C), UnorderedPurchaseItemCart (C,R,D) | | OutstandingPurchaseOrderDetails(D),  ReceiveOrder(E), ReceiveOrderDetail (E), Part (E), ReturnedOrderDetail (E) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| ForceCloseBtn | | OnClick()   * Validate data * Call BLL * Bind data to OutstandingOrdersListView |
| **BBL Class(es) and Method (s)** | | |
| PurchaseOrderController | int PurchaseOrder\_ForceClose(int purchaseOrderID, string notes)  Sets the desired PO to closed and adds the notes explaining the force close. | |
| PartController | int Part\_Update(Part item)  Update the Part QtyOnOrder to reflect the closure of the Purchase Order | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PurchaseOrders(U), Parts(U), PurchaseOrderDetails(R) | | PurchaseOrder(E), Part(E) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| InsertCartItemBtn | | OnClick()   * Validate data * Call BLL * Bind data to UnorderedCartListView |
| **BBL Class(es) and Method (s)** | | |
| UnorderedPurchaseItemCartController | int UnorderedPurchaseItemCart\_AddItem(UnorderedPurchaseItemCart item)  Adds an Unordered Item to the Cart | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| UnorderedPurchaseItemCart(C) | | UnorderedPurchaseItemCart (E) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| RemoveCartItemBtn | | OnClick()   * Validate data * Call BLL * Bind data to UnorderedCartListView |
| **BBL Class(es) and Method (s)** | | |
| UnorderedPurchaseItemCartController | int UnorderedPurchaseItemCart\_RemoveItem(int id)  Deletes an item from the unordered cart | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| UnorderedPurchaseItemCart(D) | | UnorderedPurchaseItemCart (E) |

You seem to know what to do in the system however Receive and Force are transactions in their respective method (each are one method, one transaction, one controller)