Receiving Subsystem

# ERD for UX process



# Possible UX Interface

Graphical user interface, table

Description automatically generated



# Activities

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| EmployeeDLL(10) | | Authentic User Login |
| **BLL Classes and Methods** | | |
| EmployeeController | Login\_GetEmployeeName(Employee ID)  Return Employee Name | |
| **SQL Table :C, R, U, D** | | **ViewModels** |
| Employees(R) | | EmployeeLogin |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| PurchaseOrderDLL(1) | | None / wired via ODS get dropdown list |
| **BLL Classes and Methods** | | |
| PurchaseOrderController | List<PurchaseOrderList>List\_GetActivePurchaseOrder  Retrieve a list of active purchase order number with vendor name  return | |
| **SQL Table :C, R, U, D** | | **ViewModels** |
| Orders(R), Vendors(R) | | PurchaseOrderList |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Open(2), VendorDetails(3), StateofGoodReceived(9), UnorderedItems(8) | | OnClick   * Use selection value(OrderID) to Order\_GetVendorInfo and get vendor info display at (3) * Clear UnOrderedItem data for selected OrderID(8) * Use OrderID to List<OrderDetailList>List\_GetOrderDetail to display StateofGoodReceived(9) * Get Outstanding Qty and display at (9) |
| **BLL Classes and Methods** | | |
| VendorController  OrderDetailController  ReceiveOrderItemController  UnOrderedItemController | Vendor\_GetVendorInfo(OrderID)  Get vendor name, address, phone and contact  List<OrderDetailList>List\_GetOrderDetail(OrderID)  Retrieve a list of OrderDetailID, product name, quantity ordered, order unit size  Int Item\_ReceivedQty(OrderDetailID)  Return Receive Item Qty  Outstanding Qty = Qty Ordered\*Order unit size – Receive Item Qty  Clear\_RemoveItems(OrderID)  If item exists for selected OrderID in the UnOrderedItem table, remove the item | |
| **SQL Table: C, R, U, D** | | **ViewModels** |
| Orders(R), Vendors(R), OrderDetails(R), ReceiveOrdersItems(R), UnOrderedItem(D) | | VendorDetail  OrderDetailList |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Add(7), UnorderedItems(8) | | OnCommand   * Collect OrderID, itemName, VendorProductID and Qty to validate; * send to BLL for adding item to UnOrderedItem; * Refresh UnOrderedItems(8) |
| **BLL Classes and Methods** | | |
| UnorderedItemController | void Add\_ItemToUnOrderedItem(string itemName, string VendorProductID, int Qty, int OrderID)  TRX(transaction)   * Verify Qty is valid, OrderID exists, if not, reject * add new item to UnOrderedItem, generate ItemID * Bind data in UnOrderedItem table * Display UnOrderedItems(8) | |
| **SQL Table: C, R, U, D** | | **ViewModels** |
| UnOrderedItem(C), Orders(R) | |  |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Delete (6), UnOrderedItems(8) | | OnClick   * Collect ItemID * send to BLL for removing item from UnOrderedItem; * refresh UnOrderedItems(8) |
| **BLL Classes and Methods** | | |
| UnOrderedItemController | void Remove\_DeleteItem (int ItemID)  TRX   * Verify ItemID exists, if not, reject * remove that UnOrderedItem * Refresh UnOrderedItems(8) | |
| **SQL Table: C, R, U, D** | | **ViewModels** |
| UnOrderedItem(D) | |  |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Receive Shipment(4) | | On Click   * Verfiy the Order is active * Create new ReceiveOrder * Create new ReceiveOrderItem for each item * Adjust Qty on hand and Qty on order * Copy user input to ReturnOrderItems * Copy UnOrderedItems to ReturnOrderItems * Close the order if is complete * Clear and refresh the display |
| **BLL Classes and Methods** | | |
| ReceiveOrderController  ReceiveOrderItemController  ProductController  ReturnOrderItemController  PurchaseOrderController | TRX   * If OrderID is not active, reject TRX * If ItemQty (Received Unit \* OrderUnitSize + Salvaged Qty) – OutstandingQty > OrderUnitSize, reject TRX * Create new ReceiveOrders table with OrderID, current Datetime and logged in EmployeeID, return ReceivedOrderID * Create new ReceiveOrderItem table with ReceiveOrderID(generate from previous step), OrderDetailID, ItemQty(from input\*OrderUnitSize + Salvaged Qty) * Adjust Qty on hand = Qty on hand + Item Qty * Adjust Qty on order = Qty on Order – Item Qty * Create new ReturnOrderItems with ReceiveOrderID (generate in step 2), OrderDetailID, UnOrderedItem(from input), ItemQty(from input\*order unit size – Salvaged Qty), Comment (from input) * Copy UnOrderedItems to ReturnOrderItems. ReceiveOrderID (generate in step2), OrderDetailID=null, UnorderedItem=ItemName, ItemQty = Qty, Comment=” Not on original order”, VendorProductID = VendorProductID * If Outstanding Qty = 0 on all OrderDetails of selected OrderID, close the Purchase Order using   Void Update\_ClosePurchaseOrder(OrderID)  TRX   * Verify OrderID exists, if not, reject * Verify the Purchase Order is active, if not, reject * Verify Outstanding Qty=0 for all OrderDetails, if not, reject * Update Closed=1, Comment=”Order Complete” | |
| **SQL Table: C, R, U, D** | | **ViewModels** |
| ReceiveOrders(C), ReceiveOrderItems(C)  Products(U)  ReturnOrderItems(C)  Orders(U)  UnOrderedItems(R) | | OrderDetailList |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Force Close(5) | | On Click   * Prompt the user for close * Use OrderID to process the TRX * Adjust the Qty on Order * Clear and refresh the display |
| **BLL Classes and Methods** | | |
| PurchaseOrderController | Void Update\_ForceClosePurchaseOrder(OrderID, comment, List<OrderDetailList>ProductID, List<OrderDetailList>Outstanding Qty)  TRX   * Verify OrderID exists, if not, reject * Verify the Purchase Order is active, if not, reject * Verify comment is not null, if is null, reject * Verify All ProductID exists, if not, reject * If Outstanding Qty is 0, do not need adjust Qty on Order * If Outstanding Qty >0, Qty on Order=Qty on Order - Outstanding Qty * Update Closed=1, Comment=comment | |
| **SQL Table: C, R, U, D** | | **ViewModels** |
| Orders(U), Products(U) | | OrderDetailList |

# **ViewModels:**

### PurchaseOrderList

* Int OrderID
* Int OrderNumber
* Int VendorID
* String VendorName
* Bit Closed
* Datetime OrderDate

### VendorDetail

* Int OrderID
* String VendorName
* String Address
* String Phone
* String Contact

### OrderDetailList

* Int OrderDetailID
* Int OrderID
* Int ProductID
* String ProductName
* Int OrderQty
* Int OrderUnitSize
* Int ReceiveItemQty
* Int OutstandingQty=OrderQty\*OrderUnitSize-ReceiveItemQty

### EmployeeLogin

* Int EmployeeID
* String EmployeeFullName