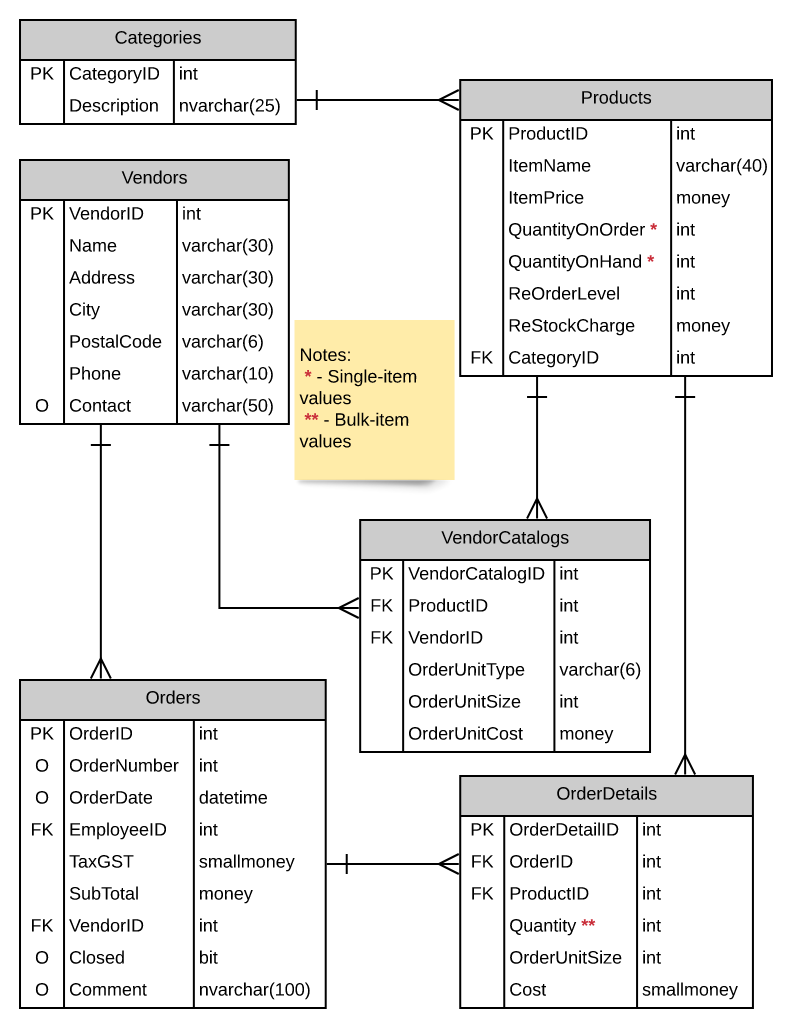
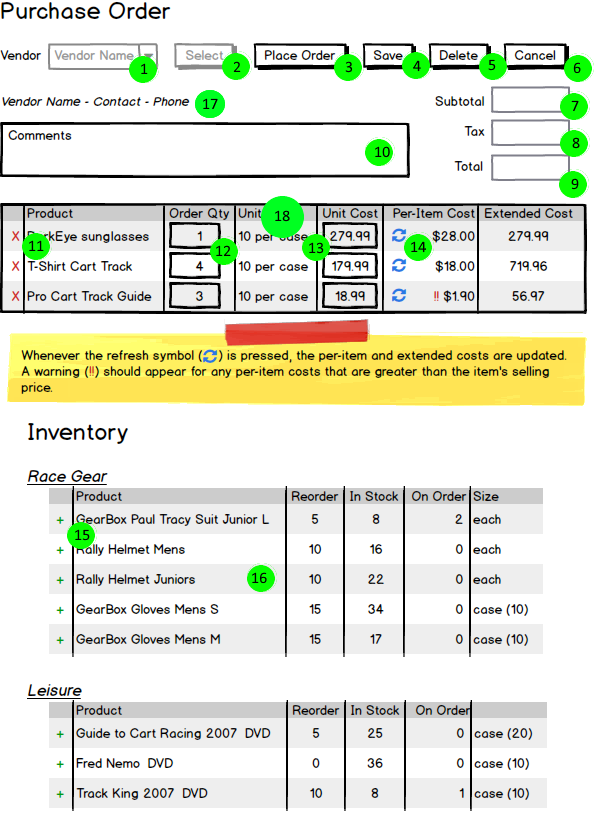
# Purchasing SubSystem - Ben

## ERD for UX process



## Possible UX Interface

Note: Changed from side to side to vertical for ease of reading.



## Event and Wiring Summations

## Activities

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| VendorDDL(1) | | None / ODS |
| **BLL Class(es) and Method (s)** | | |
| VendorController | List <SelectionList> List\_Vendors()  Retrieve and populate the DDL with the full list of Vendor IDs and Names | |
| **Other** | Default displayed selection value of “Select a Vendor” and a value of 0 using AppendDataBoundItems="True" and a new listitem in the webpage DDL | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
| Vendor (R) | | SelectionList |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| VendorSelect(2),  VendorDDL(1),  VendorInfoTxt(17), SubTotalTxt (7), TaxTxt(8), TotalTxt(9), CommentTxt (10), OrderProductList(18), InventoryList(16) | | OnClick / ODS   * Verify selected Vendor is selected if not display error message * Send selected id to BLL, check if the “currently open” order exists for the Vendor   + If exists and is “currently open” (edit)     - Display Order info in TaxTxt, TotalTxt, SubTotalTxt, CommentTxt (Tax, Total, SubTotal, Comment)     - Bind OrderDetail info in OrderProductList (with hidden OrderID column)     - Display Vendor info in VendorInfoTxt     - Display products from VendorCatalogues in InventoryList   + If not exists (new order)     - Reset TaxTxt, TotalTxt, SubTotalTxt to zero     - Revert CommentTxt and VendorInfoTxt to default values     - Clear OrderProductList and keep the headings     - Display all products from VendorCatalogues in InventoryList |
| **BLL Class(es) and Method (s)** | | |
| OrderController  VendorController  OrderDetailController  VendorCataloguesController | OrderInfo Orders\_CheckOpen(int vendorid) TRX  Check if an open order exists (NULL value for OrderDate and OrderNumber), and if so gets order data, else create a new Order  VendorInfo Vendors\_Get(int vendorid)  Get Vendor info  List <OrderDetailListItem> OrderDetail\_GetByVendorId(int orderid, int vendorid)  Get the list of items in the order  List <VendorCataloguesListItem> VendorCatalogues\_GetItems(int vendorid, int orderid, int searchmode)  Get list of all items in the vendor’s catalogue if searchmode is not negative, and get items in vendor’s catalogue that aren’t in the order already if searchmode is negative | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
| Order (C opt, R opt, U opt), Vendor(R), OrderDetail(C, R opt, U, D), VendorCatalogues(R), Products (R) | | OrderInfo, VendorInfo, OrderDetailListItem, VendorCataloguesListItem |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| PlaceOrder(3),  OrderProductList(18)  QtyField(12), UnitCostField(13),  SubTotalTxt (7), TaxTxt(8), CommentTxt (10), | | OnClick / ODS   * Validate all required fields (QtyField, UnitCostField) * QtyField validation for proper amount * UnitCostField validation for proper cost and no !! warnings * Send all items in OrderProductList to BLL for update and processing * Update Order information for Placing the Order * Clear page of selected order |
| **BLL Class(es) and Method (s)** | | |
| OrderDetailController  OrderController | void Update\_ProductFromOrder (List <OrderDetailListItem> OrderedProductsList, string comment) TRX   * Validate all data requirements * Validate all quantities are valid (not negative) * Verify cost fields are all valid * Updates all OrderDetail for each item   void Update\_Order (double subtotal, double tax, string comment, int orderid, int savetype) TRX   * savetype will be used for placing orders - parameter of 1 for placing * Validate both subtotal and tax amounts to be correct * Validate comment to not be past the max length * Set OrderDate to current date and assign a new OrderNumber * Update the Order with the relevant information | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
| OrderDetail (U), Order(U) | | OrderDetailListItem |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| SaveOrder(4),  OrderProductList(18),  QtyField(12), UnitCostField(13),  SubTotalTxt (7), TaxTxt(8), CommentTxt (10), | | OnClick / ODS   * Validate all required fields (QtyField, UnitCostField) * QtyField validation for proper amount * UnitCostField validation for proper cost and no !! warnings * Send all items in OrderProductList to BLL for update and processing * Update Order information |
| **BLL Class(es) and Method (s)** | | |
| OrderDetailController  OrderController | void Update\_ProductFromOrder (List <OrderDetailListItem> OrderedProductsList) TRX   * Validate all data requirements * Validate all quantities are valid (not negative) * Verify cost fields are all valid (Unit cost matches up with Per Item Cost and Extended Cost) * Updates all OrderDetail for each item   void Update\_Order (double subtotal, double tax, string comment, int orderid, int savetype) TRX   * savetype will be used for placing orders - parameter of 2 for saving * Validate both subtotal and tax amounts to be correct * Validate comment to not be past the max length * Update the Order with the relevant information | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
| OrderDetail (U), Order(U) | | OrderDetailListItem |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Delete(5),  VendorDDL(1), VendorInfoTxt(17), SubTotalTxt (7), TaxTxt(8), TotalTxt(9), CommentTxt (10), OrderProductList(18), InventoryList(16) | | OnClick / ODS   * Clear all list/gridviews and relevant fields * Reset VendorDDL to default value * Send Order info and OrderDetail to BLL for deletion, with OrderDetail first |
| **BLL Class(es) and Method (s)** | | |
| OrderDetailController  OrderController | void Delete\_OrderDetail (List <OrderDetailListItem> OrderedProductsList)   * Delete all items in the OrderDetailList (if saved prior)   void Delete\_Order (int OrderID)   * Delete the order using the OrderID (if saved prior) | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
| Order (D), OrderDetail(D) | | OrderDetailListItem |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Cancel(6),  VendorDDL(1), VendorInfoTxt(17), SubTotalTxt (7), TaxTxt(8), TotalTxt(9), CommentTxt (10), OrderProductList(18), InventoryList(16) | | OnClick   * Clear all list/gridviews and relevant fields * Reset VendorDDL to default value |
| **BLL Class(es) and Method (s)** | | |
|  |  | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
|  | |  |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| AddProduct(15),  InventoryList(16),  OrderProductList(18),  SubTotalTxt (7), TaxTxt(8), TotalTxt(9) | | OnCommand   * Collect product information using productid and orderid and send to BLL to add to the OrderProductList * Refresh the page, so the OrderProductList should have the new item * Recalculate SubTotal, Tax and Total fields |
| **BLL Class(es) and Method (s)** | | |
| OrderDetailController | void Add\_ProductToOrder(int orderid, int productid) TRX   * Verify the product isn’t already on the order (which it shouldn’t) but output an error if so * Add new product to the orderlist | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
| OrderDetail (U) | | OrderDetailListItem |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| RemoveProduct(11),  InventoryList(16),  OrderProductList(18),  SubTotalTxt (7), TaxTxt(8), TotalTxt(9) | | OnCommand   * Collect product information using productid and orderid and send to the BLL to remove from the order * Refresh page, so the InventoryList should have the item back from OrderProductList * Recalculate SubTotal, Tax and Total fields |
| **BLL Class(es) and Method (s)** | | |
| OrderDetailController | void Delete\_ProductFromOrder(int orderid, int productid) TRX   * Check if the product was saved onto the order, if so, delete the record, if not, do nothing | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
| OrderDetail (D) | |  |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Refresh(14),  OrderProductList(18), UnitCostTxt(13), OrderQtyTxt(12),  SubTotalTxt (7), TaxTxt(8), TotalTxt(9) | | OnCommand / ODS   * Collect info from fields (Unit Cost, OrderQty, UnitSize) * Verify and calculate Per-Item Cost and Extended Cost are valid (Per-Item Cost = UnitCost / Unit Size), (Extended Cost = OrderQty \* UnitCost) * Retrieve original UnitCost from Product table * Verify UnitCost text field is not larger than the original retrieved UnitCost, if so add !! symbol * Recalculate SubTotal, Tax and Total fields |
| **BLL Class(es) and Method (s)** | | |
| ProductController | double Product\_GetItemPrice(int productid)   * Returns the ItemPrice for the specified product | |
| **SQL Table(s): (C,R,U,D)** | | **ViewModels** |
| Product(R) | |  |

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## ViewModels:

**SelectionList**

* int ValueID
* string DisplayText

**OrderInfo**

* int OrderID
* int OrderNumber
* datetime OrderDate
* decimal TaxGST
* decimal SubTotal
* int EmployeeID
* int VendorID
* bool Closed
* string Comment

**VendorInfo**

* int VendorID
* string Name
* string Contact
* string Phone

**OrderDetailListItem**

* int OrderDetailID
* int OrderID
* int ProductID
* int Quantity
* int OrderUnitSize
* decimal Cost

**VendorCataloguesListItem**

* int VendorCatalogID
* int ProductID
* int VendorID
* int OrderUnitSize
* string ItemName
* int QuantityOnOrder
* int QuantityOnHand
* intReOrderLevel