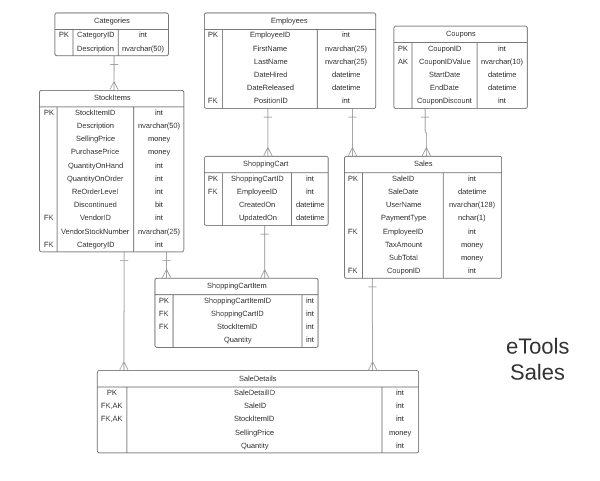
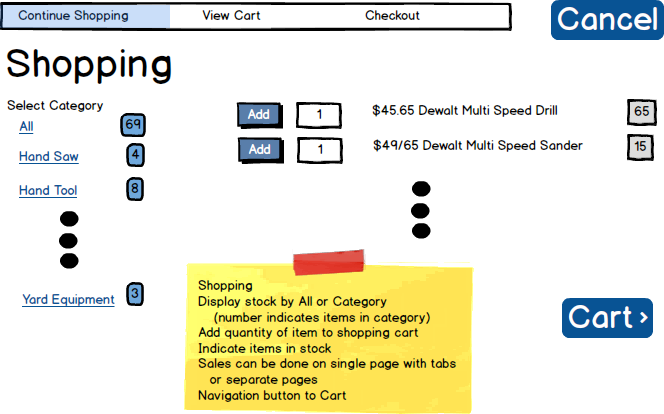
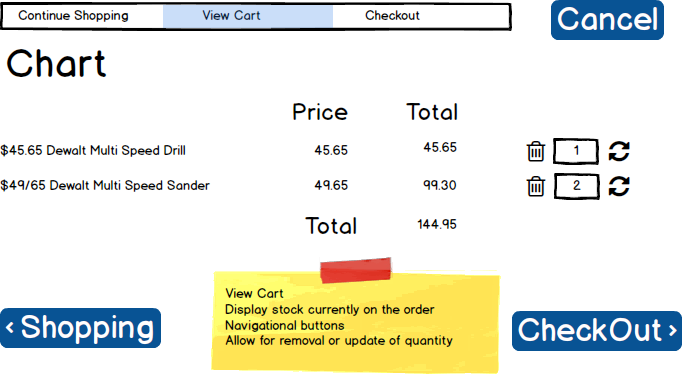
Sales

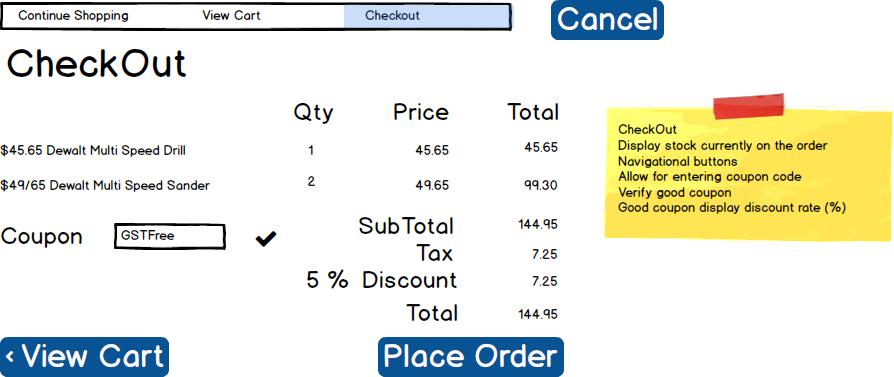
# ERD for UX process



# Possible UX Interface







|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CategoryRepeater | | None / wired via ObjectDataSource |
| **BBL Class(es) and Method (s)** | | |
| CategoryController | List<CategoryList> List\_Categories  Retrieve a list of Category Descriptions, CategoryID and count of Items in each Category. | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Categories(R) | | CategoryList(P) |

# Event and Wiring Summations

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CategoryButton  (CategoryID) | | - On Click / move CommandArgument to ItemListListView label  -Set label to ”-1” if All categories was selected  -Refresh (ListView) itemSelectionList |
| **BBL Class(es) and Method (s)** | | |
| StockItemController | List<StockItemList> List\_ItemsForCartSelection(int categoryID)  -Retrieve a list of StockItems from the value of label “Category”  -Return StockItemID, PurchasePrice, Description, QuantityOnHand  -Display in (ListView) ItemSelectionList  -Databind() | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| StockItems (R) | | StockItemList(P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CartButton, ShoppingButton, CheckoutButton | | On Click / Directed to the proper tab  Done with javascript |
| **BBL Class(es) and Method (s)** | | |
| none | none | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| none | | none |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CancelButton | | On Click  -bring to Sales Page Tab1 |
| **BBL Class(es) and Method (s)** | | |
| ShoppingCartController | Empty\_Cart(string username)  -Find cart for username  -if no cart return null  If there is a cart  -Delete Shopping cart items  -Delete Shopping Cart  Return true | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ShoppingCart(D), ShoppingCartItem(D) | | ShoppingCartItem(E), ShoppingCart(E) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| AddButton | | On Click  -Validate value in Quantity text box  -Send to BLL for adding cart and item  -Refresh CartListView, CheckoutListView |
| **BBL Class(es) and Method (s)** | | |
| SalesController | -test to see if Username has a Shopping cart already  -if so, test to see if item exists in the cart already  AddToCart(string userName, int productid, int quantity)  - If no cart exists create a cart and add the item | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ShoppingCart(C), ShoppingCartItem(C) | |  |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CartGarbageButton | | On Click  -Refresh Cartlistview, CheckoutListview |
| **BBL Class(es) and Method (s)** | | |
| ShoppingCartItemController | Delete\_CartItem(int ShoppingCartItemID) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ShoppingCartItem(D) | | ShoppingCartItem(E) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CartUpdateButton | | On Click  -validate and get value from textbox  -send to BLL for update  -refresh cartListView |
| **BBL Class(es) and Method (s)** | | |
| ShoppingCartItemController | Update\_ShoppingCartItemQuantity(int ShoppingCartItemID, int quantity) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ShoppingCartItem(U) | | ShoppingCartItem(E) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| PlaceOrderButton | | On Click  -Send All data to BLL  -Add success message |
| **BBL Class(es) and Method (s)** | | |
| SalesController | Add\_Sale()  TRX(-update QOH  -Add Sale  -Add sale Items  -Delete cart items  -Delete cart) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| StockItem(U)  ShoppingCart(D)  ShoppingCartItemController(D)  Sales(C)  SaleDetails(C) | | StockItemID(E)  ShoppingCartID(E) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| VerifyCouponButton | | OnClick  -Refresh checkout page |
| **BBL Class(es) and Method (s)** | | |
| CouponController | Coupon\_Find(int CouponID)  Returns couponID  Returns null coupon doesn’t exist  -if it returns a coupon then make coupon label visible  -if it returns null show error in message user controls | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Coupons(R) | | CouponID(E) |

Comments

You seem to have a good handle on your subsystem. You might wish to review some of your plan and indicate how certain data is moving or not moving between projects.