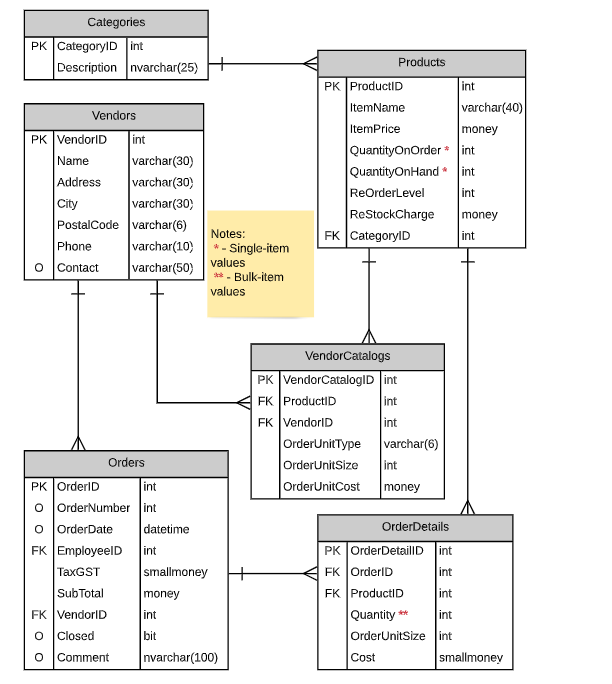
Purchase Order

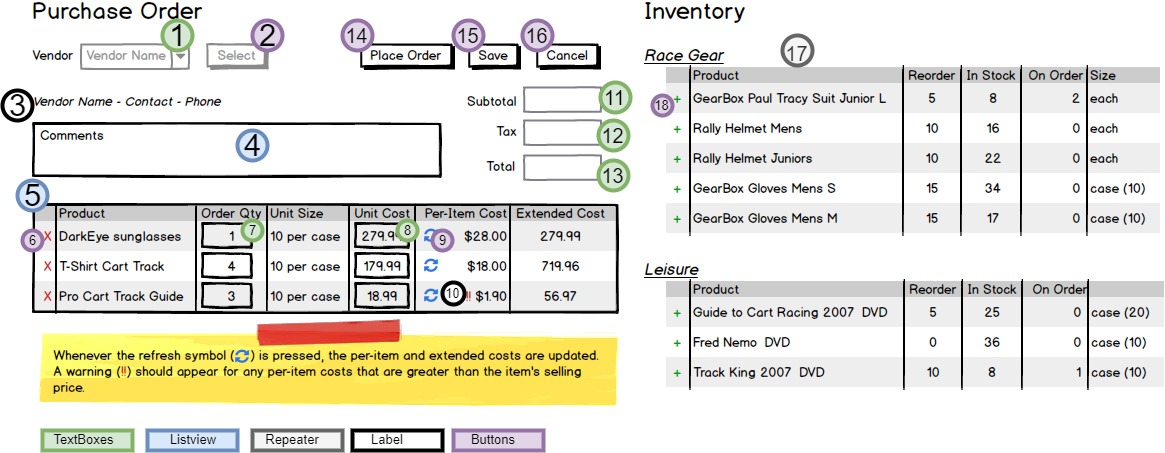
# ERD for UX process

Place your ERD diagram here.



# Possible UX Interface

Place your screen design(s) here



|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| VendorDDL (**1**) | | None/ Wired via ODS |
| **BBL Class(es) and Method (s)** | | |
| VendorsController | List<Vendors> List\_Vendors()  Retrieve a list of vendors names & vendorID for DDL | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Vendors(R) | | Vendors |

# Event and Wiring Summations

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Fetch (OrderDetails) (**2**), Refresh Button (**9**), RemoveItem (**6**), PlaceOrder(**14**), SaveOrder(**15**), Cancel(**16**) | | OnClick / wired via ObjectDataSource   * Validate active order * Use DDL VendorID to retrieve order * Refresh the Extended cost & Per-Item Cost * Validate a Per-Item Cost and show a warning if Per-Item Cost > Selling price * Collect ProductID and OrderID * Send to BLL for removing Product from OrderDetails * Bulk Update current displayed order but not place the order on Save * Bulk Update and set Order Date and new Purchase Order Number for current Order update the QtyOnorder in inventory on Place order * Refresh Order Details * Bind data to Order Details * Clear the webpage with current purchase order and reset the vendor list on Cancel |
| **BBL Class(es) and Method (s)** | | |
| OrderDetailsController | * List<OrderDetailsList> List\_OrderDetialsByVendor (int VendorID)   Retrieve an Active order for supplied selection type and argument id. Return Vendor Name, Contact, Phone, Comments, Listview(Product, OrderQty, UnitCost)   * Calculate\_ExtendedCost (Double UnitCost, Double OrderQty)   This function will calculate the Extended Cost using the unitCost & OrderQty   * Calculate\_PerItemCost (Double UnitCost,Double OrderQty, Double UnitSize)   This function will calculate the perItemCost using the unitCost divide by OrderQty multiply by UnitSize   * CalculateSubtotal(Double ExtendedCosts)   This function will calculate the sum of all the extended cost on this orderDetail   * CalculateTax(Double Subtotal)   This function will calculate the sum of all the tax on this orderDetail   * CalculateTotal(Double ExtendededCost, Double Subtotal)   This function will calculate the grand total after tax on this order | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| OrderDetails (R) | | OrderDetailsList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Inventory: Plus, Sign (**18**) | | OnCommand / wired via ObjectDataSource   * Collected Product Information (ProducID) and OrderID and validate * Sedn to BLL for Adding Item to OrderDetails * Refresh OrderDetails |
| **BBL Class(es) and Method (s)** | | |
| OrderDetailsController | Void Add\_ProductToOrderDetails (ProductID, OrderID)   * Verify if desired productid does not already exist, if so reject * Add new product to the orderDetails | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| OrderDetails (C) | |  |