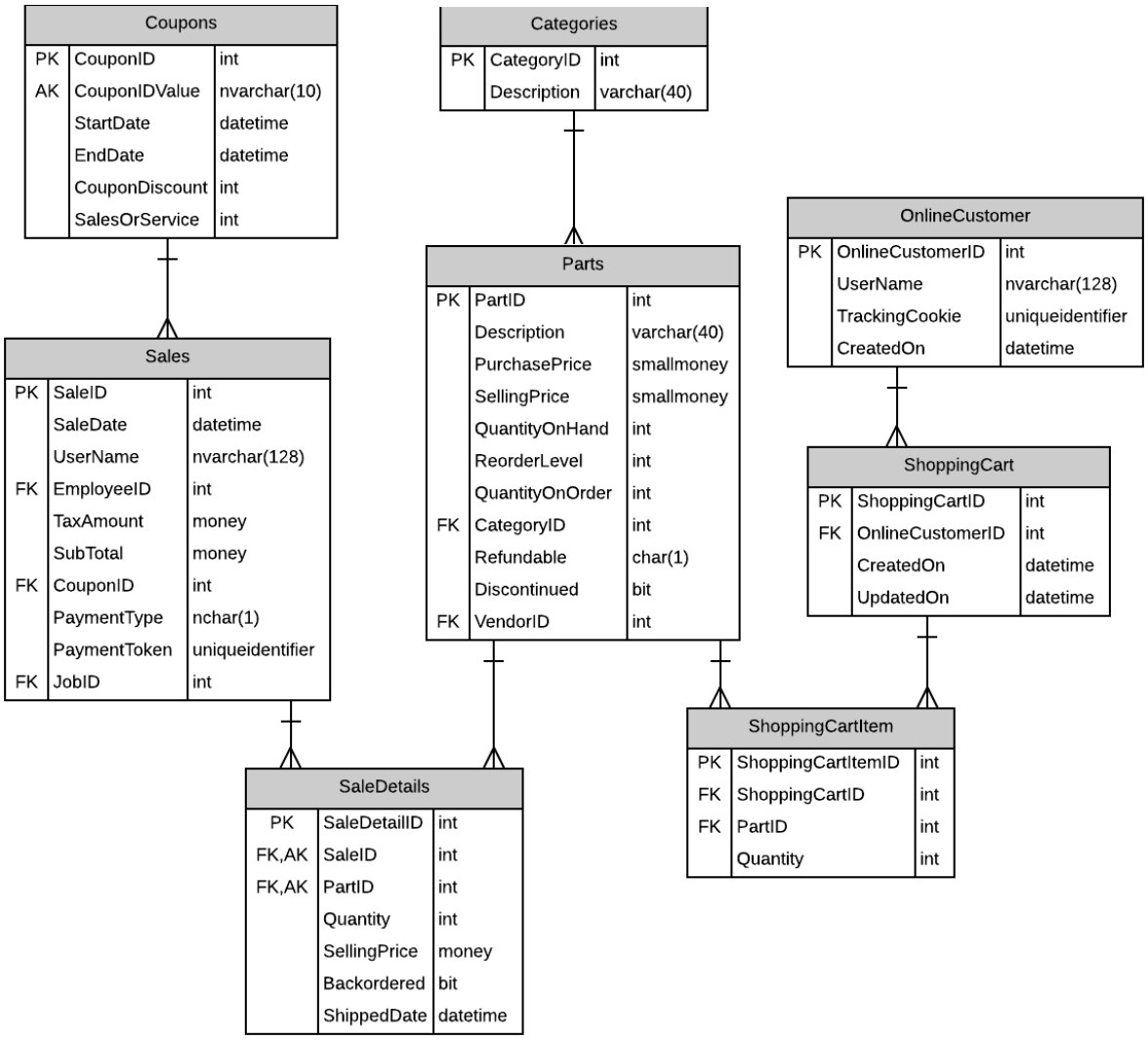
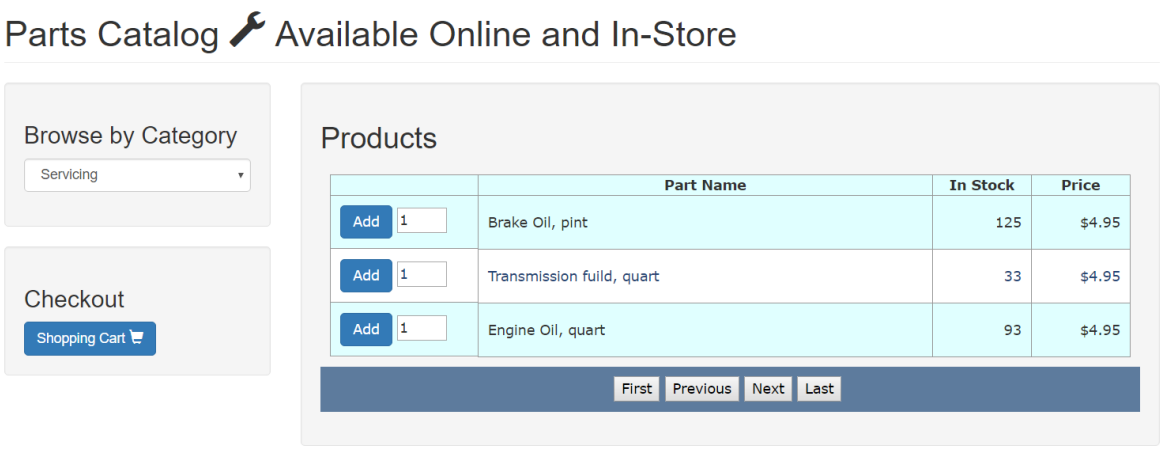
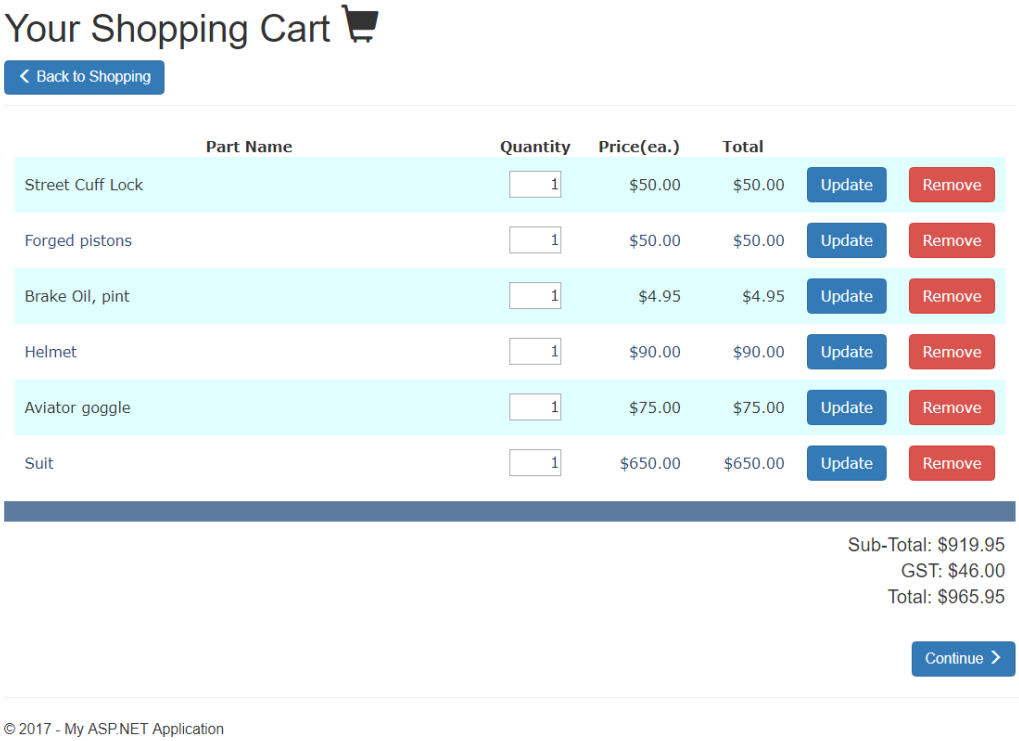
e Bike Sales by Michael Bateman

ERD for UX process



Possible UX Interface





Purchase Details

Back to Shopping

Enter your information for billing and shipping.

Billing Details

Name:

Email:

Address:

Phone:

Shipping Details

Name:

Email:

Address:

Phone:

Back

Continue

© 2017 - My ASP.NET Application

Place Your Order

Back to Shopping

Part Name	Quantity	Price(ea.)	Total
Street Cuff Lock	1	$50.00	$50.00
Forged pistons	1	$50.00	$50.00
Brake Oil, pint	1	$4.95	$4.95
Helmet	1	$90.00	$90.00
Aviator goggle	1	$75.00	$75.00
Suit	1	$650.00	$650.00

Select a coupon code...

Payment Method

Credit Debit

Sub-Total: $919.95
Discount: $0.00
GST: $46.00
Total: $965.95

Back

Place Order

© 2017 - My ASP.NET Application

Event and Wiring Summations

Controls	Events
CategoryListDDL	None / wired via ObjectDataSource
BLL Class(es) and Method (s)	
PartsController	List<CategoryListPOCO> CategoryList() Retrieve a list of Category titles/item count and id for DDL
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
Categories(R), Parts(R)	CategoryListPOCO(P)

Controls	Events
PartsByCategory ListView	None / wired via ObjectDataSource
BBL Class(es) and Method (s)	
PartsController	List<Part> Parts_byCategoryList(int categoryid) Retrieve a list of parts for a category for List View
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
Parts(R)	Part(E)

Controls	Events
Button command AddToCartBtn	OnClick • Send a part id and quantity to be added to a shopping cart • Check if registered user, if not throw error • Create online customer profile if first time shopper • Check if customer has a shopping cart, if not create one • Check if item is already in cart, if it is throw exception
BBL Class(es) and Method (s)	
ShoppingCartController	Void Add_ItemToCart (string username, int partid, int quantity) Validate registered user Validate online customer Validate shopping cart Validate part Add a quantity of a part to a user's shopping cart Update quantity in Parts table Use MessageUser to return success message
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
OnineCustomers(R, C) ShoppingCarts(R, C) Parts(R, U), ShoppingCartItems(C)	OnlineCustomer(E), ShoppingCart(E), Part(E), ShoppingCartItem(E)

Controls	Events
ShoppingCartList ListView	None / wired via ObjectDataSource
BBL Class(es) and Method (s)	
ShoppingCartController	List<ShoppingCartItem> ShoppingCartList(string username) Retrieves a list of shopping cart items from a users cart for List View
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
ShoppingCartItems(R)	ShoppingCartItem(E)

Controls	Events
TotalsList GridView	None / wired via ObjectDataSource
BBL Class(es) and Method (s)	
ShoppingCartController	List<TotalsPOCO> Totals_List(int cartid) Find the items in a cart by id and calculate the subtotal, coupon, gst and total using selling price from parts table (don't display coupon discount in shopping cart view, only place order view)
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
ShoppingCarts(R) ShoppingCartItems(R) Parts(R)	TotalsPOCO(P)

Controls	Events
RemoveItem Button	OnClick • Send the shopping cart item id for deleting • Bind data to current ShoppingCart • Bind data to totals
BBL Class(es) and Method (s)	
ShoppingCartController	List<ShoppingCartItem> Remove_Item(int itemid) Find the item by id and delete it from ShoppingCartItems
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
ShoppingCartItems(R, D)	ShoppingCartItem(E)

Controls	Events
UpdateQty Button	OnClick • Send the shopping cart item id and quantity for updating • Bind data to current ShoppingCart • Bind data to totals
BBL Class(es) and Method (s)	
ShoppingCartController	List<ShoppingCartItem> Update_ItemQty(int itemid, int quantity) Find the item by id and update the quantity field in ShoppingCartItems
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
ShoppingCartItems(R, U)	ShoppingCartItem(E)

Controls	Events
CouponList DDL	None / wired via ObjectDataSource
BBL Class(es) and Method (s)	
SalesController	List<Coupon> Coupon_List() Retrieve a list of coupons that apply to sales (using SalesOrService int)
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
Coupons(R)	Coupon(E)

Controls	Events
CouponListRefresh Button	OnClick • Send selected coupon id • Bind totals list
BBL Class(es) and Method (s)	
SalesController	List<TotalsPOCO> Update_Totals(int couponid) Find the coupon by id and add the discount to the totals list
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
Coupons(R)	Coupon(E) TotalsPOCO(P)

Controls	Events
PlaceOrder Button	OnClick • Send username, couponid and payment type to BLL for processing • Bind data to Shopping Cart • Bind data totals list
BBL Class(es) and Method (s)	
SalesController Also uses ShoppingCartController	void PlaceOrder(string username, int couponid, char paymenttype) Create a new sale with coupon, totals and payment type info Create SaleDetail for each cart item in cart Update quantities in Parts table Delete ShoppingCartItems from cart Delete ShoppingCart
SQL Table(s): (C,R,U,D)	**Entities/DTOs/POCOs**
Coupons(R) Sales(C) SaleDetails(C) Parts(U) ShoppingCartItems(D) ShoppingCarts(D)	Coupon(E) TotalsPOCO(P) Sale(E) SaleDetail(E) Part(E) ShoppingCartItem(E) ShoppingCart(E)

You seem to have a good handle on this subsystem. Remember, this is a transaction oriented system so changes to the database is done in the bll as transactions.