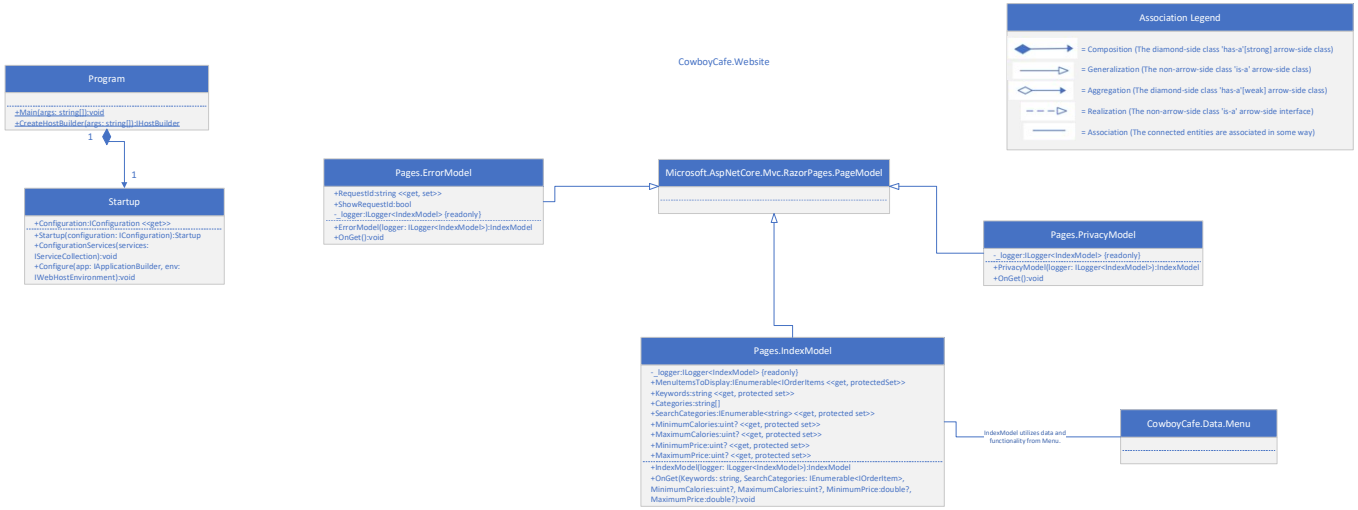
CowboyCafe.Data
TrailBurger
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+Bun:bool <<get, set>> = true
+Ketchup:bool <<get, set>> = true
+Mustard:bool <<get, set>> = true
+Pickle:bool <<get, set>> = true
+Cheese:bool <<get, set>> = true
+SpecialInstructions:List<string> <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
<<abstract>>
Entree
+Price:double <<get>> {virtual}
+Calories:uint <<get>> {virtual}
+SpecialInstructions:List<string> <<get>> {virtual}
+<<event>>PropertyChanged:PropertyChangedEventHandler {abstract}
AngryChicken
+Bread:bool <<get, set>>
+Pickle:bool <<get, set>> = true
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+SpecialInstructions:List<string> <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
RustlersRibs
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+SpecialInstructions:List<string> <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
Association Legend
= Composition (The diamond-side class 'has-a'[strong] arrow-side class)
= Generalization (The non-arrow-side class 'is-a' arrow-side class)
= Aggregation (The diamond-side class 'has-a'[weak] arrow-side class)
= Realization (The non-arrow-side class 'is-a' arrow-side interface)
= Association (The connected entities are associated in some way)
Menu
+Entrees:IEnumerable<IOrderItem>
+Sides:IEnumerable<IOrderItem>
+Drinks:IEnumerable<IOrderItem>
+All:IEnumerable<IOrderItem>
+FilterMenuItemsByKeywords(items: IEnumerable<IOrderItem, keywords: string):IEnumerable<IOrderItem>
+FilterMenuItemsByCategory(items: IEnumerable<IOrderItem>, categoriesToRemove: IEnumerable<string>):IEnumerable<IOrderItem>
+FilterMenuItemsByCategory(items: IEnumerable<IOrderItem, minimumCalories: uint?, maximumCalories: uint?):IEnumerable<IOrderItem>
+FilterMenuItemsByPrice(items: IEnumerable<IOrderItems>, minimumPrice: double?, maximumPrice: double?):IEnumerable<IOrderItem>
DakotaDoubleBurger
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+Bun:bool <<get, set>> = true
+Ketchup:bool <<get, set>> = true
+Mustard:bool <<get, set>> = true
+Pickle:bool <<get, set>> = true
+Cheese:bool <<get, set>> = true
+Tomato:bool <<get, set>> = true
+Lettuce:bool <<get, set>> = true
+Mayo:bool <<get, set>> = true
+SpecialInstructions:List<string> <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
TexasTripleBurger
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+Bun:bool <<get, set>> = true
+Ketchup:bool <<get, set>> = true
+Mustard:bool <<get, set>> = true
+Pickle:bool <<get, set>> = true
+Cheese:bool <<get, set>> = true
+Tomato:bool <<get, set>> = true
+Lettuce:bool <<get, set>> = true
+Mayo:bool <<get, set>> = true
+Bacon:bool <<get, set>> = true
+Egg:bool <<get, set>> = true
+SpecialInstructions:List<string> <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
CowpokeChili
+Cheese:bool <<get, set>>
+SourCream:bool <<get, set>>
+GreenOnions:bool <<get, set>>
+TortillaStrips:bool <<get, set>>
+Price:double <<get, set>> {override}
+Calories:uint <<get, set>> {override}
+SpecialInstructions:List<string> <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
PecosPulledPork
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+Bread:bool <<get, set>> = true
+Pickle:bool <<get, set>> = true
+SpecialInstructions:List<string> <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
Order
-lastOrderNumber:uint
-items:List<OrderItem>
+Items:IEnumerable<IOrderItem> <<get>>
+Subtotal:Double <<get>>
+OrderNumber:uint <<get>>
+<<event>>PropertyChanged:PropertyChangedEventHandler
+Add(Item:IOrderItem)
+Remove(item:IOrderitem)
-<<event>>OnItemChanged(sender:object, e: PropertyChangedEventArgs)
1
0..*
<<Interface>>
IOrderItem
+Price:Double <<get>>
+SpecialInstructions:List<string>() <<get>>
<<Interface>>
System.ComponentModel.INotifyPropertyChanged
+<<event>>PropertyChanged:PropertyChangedEventHandler
<<enum>>
Size
Small
Medium
Large
Water
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+Size:Size <<get, set>>{override}
+Ice:bool <<get, set>>{override}
+Lemon:bool <<get, set>> = false
+SpecialInstructions:List<string> <<get>>
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
<<abstract>>
Drink
+Size:Size <<get, set>> {abstract}
+Price:double <<get>> {abstract}
+Calories:uint <<get>> {abstract}
+Ice:bool <<get, set>> = true; {abstract}
+SpecialInstructions:List<string> <<get>> {abstract}
+<<event>>PropertyChanged:PropertyChangedEventHandler {abstract}
1
1
1
1
ChiliCheeseFries
+Size:Size <<get, set>> {override}
+Calories:uint <<get>> {override}
+Price:double <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
<<abstract>>
Side
+Size:Size <<get, set>> {abstract}
+Price:double <<get>> {abstract}
+Calories:uint <<get>> {abstract}
+SpecialInstructions:List<string>
+<<event>>PropertyChanged:PropertyChangedEventHandler {abstract}
TexasTea
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+Size:Size <<get, set>> {override}
+Ice:bool <<get, set>>{override}
+Sweet:bool <<get, set>> = true
+Lemon:bool <<get, set>> = false
+SpecialInstructions:List<string> <<get>>
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
<<enum>>
SodaFlavor
CreamSoda
OrangeSoda
Sarsaparilla
BirchBeer
RootBeer
1
1
CowboyCoffee
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+Size:Size <<get, set>>{override}
+Ice:bool <<get, set>>{override}
+Decaf:bool <<get, set>>
+RoomForCream:bool <<get, set>> = false
+SpecialInstructions:List<string> <<get>>
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
JerkedSoda
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+Flavor:SodaFlavor <<get, set>>
+Size:Size <<get, set>> {override}
+Ice:bool <<get, set>>{override}
+SpecialInstructions:List<string> <<get>>
+<<event>>PropertyChanged:PropertyChangedEventHandler {override}
+ToString():string {override}
CornDodgers
+Size:Size <<get, set>> {override}
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+<<event>>PropetyChanged:PropertyChangedEventArgs {override}
+ToString():string {override}
PanDeCampo
+Size:Size <<get, set>> {override}
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+<<event>>PropertyChanged:PropertyChangedEventArgs {override}
+ToString():string {override}
BakedBeans
+Size:Size <<get, set>> {override}
+Price:double <<get>> {override}
+Calories:uint <<get>> {override}
+<<event>>PropertyChanged"PropertyChangedEventArgs {override}
+ToString():string {override}

ExtensionMethods
+<<extension>>FindAncestor<DependencyObject>(element: this DependencyObject):DependencyObject
CowboyCafe.PointOfSale
Association Legend
= Composition (The diamond-side class 'has-a'[strong] arrow-side class)
= Generalization (The non-arrow-side class 'is-a' arrow-side class)
= Aggregation (The diamond-side class 'has-a'[weak] arrow-side class)
= Realization (The non-arrow-side class 'is-a' arrow-side interface)
= Association (The connected entities are associated in some way)
<<Enumeration>>
CashRegister.Bills
One
Two
Five
Ten
Twenty
Fifty
Hundred
<<Enumeration>>
CashRegister.Coins
Dollar
HalfDollar
Quarter
Dime
Nickel
Penny
<<Enumeration>>
CashRegister.ResultCode
Success
InsufficientFunds
CancelledCard
ReadError
UnknownError
System.Windows.Window
MainWindow
+MainOrderControl:OrderControl
+MainOrderTransactionControl:OrderTransactionControl
+MainCollectCashFromCustomerControl:CollectCashFromCustomerControl
+MainModelViewCashRegister:ModelViewCashRegister
+MainGiveChangeToCustomerControl:GiveChangeToCustomerControl
+MainSwapBillsForCoinsControl:SwapBillsForCoinsControl
+MainManageRegisterDrawerControl:ManageRegisterDrawerControl
/+DataContext:Object
+MainWindow():MainWindow
+SwapOrderControlAndOrderTransactionControl()
+SwapTransactionControlAndCollectCashFromCustomerControl()
+SwapCollectCashFromCustomerControlAndGiveChangeToCustomerControl()
+SwapGiveChangeToCustomerControlForOrderControl()
+SwapGiveChangeToCustomerControlAndBillToCoinSwapControl()
+SwapGiveChangeToCustomerAndManageRegisterDrawerControl()
ManageRegisterDrawerControl
/+DataContext:Object
+ManageRegisterDrawerControl():ManageRegisterDrawerControl
+<<event>>OnDoneManaging(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
CollectCashFromCustomerControl
/+DataContext:Object
+CollectCashFromCustomerControl():CollectCashFromCustomerControl
+<<event>>OnNextStep(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+<<event>>OnBackStep(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
OrderControl
-Container:Border
/+DataContext:Object
+OrderControl():OrderControl
<<event>>ItemSelection_Click(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
<<event>>CancelOrder_Click(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
<<event>>CompleteOrder_Click(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+SwapScreen(element: FrameworkElement)
CashRegister.ReceiptPrinter
+Print(receipt: string)
CashRegister.CardTerminal
+ProcessTransaction(amount: double):ResultCode
OrderSummaryControl
/+DataContext:Object
+OrderSummaryControl():OrderSummaryControl
+<<event>>OnSelectionChanged(sender: object, e: SelectionChangedEventArgs):EventHandler<SelectionChangedEventArgs>
+<<event>>OnDeleteItem(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
OrderTransactionControl
/+DataContext:Object
+OrderTransactionControl():OrderTransactionControl
+<<event>>OnPaymentByCredit(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+<<event>>OnPaymentByCash(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+<<event>>OnCancelTransaction(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
CoinPaymentControl
+CoinTypeProperty:DependencyProperty {readonly}
+CoinType:Coin <<get, set>>
+CoinCountProperty:DependencyProperty {readonly}
+CoinCount:int <<get, set>>
/+DataContext:Object
+CoinPaymentControl():CoinPaymentControl
+<<event>>OnAddCoinToPayment(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+<<event>>OnRemoveCoinToPayment(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
BakedBeansCustomization
/+DataContext:Object
+BakedBeansCustomization():BakedBeansCustomization
-<<event>>BakedBeansSizeSelect(sender: object, e: RoutedEventArgs)
MenuItemSelectionControl
/+DataContext:Object
+MenuItemSelectionControl():MenuItemSelectionControl
+<<event>>OnAddOrderItemButtonClicked(sender: object, e: RoutedEventArgs)
GiveChangeToCustomerControl
/+DataContext:Object
+GiveChangeToCustomerControl():GiveChangeToCustomerControl
+<<event>>OnBackStep(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+<<event>>OnCompleteTransaction(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+<<event>>OnManageMoneyinDrawer(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+<<event>>OnSwapBillsAndCoins(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
BillPaymentControl
+CoinTypeProperty:DependencyProperty {readonly}
+CoinType:Coins <<get, set>>
+CoinCountProperty:DependencyProperty {readonly}
+CoinCount:int <<get, set>>
/+DataContext:Object
+CoinPaymentControl():CoinPaymentControl
+<<event>>OnAddCoinToPayment(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
+<<event>>OnRemoveCoinToPayment(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
AngryChickenCustomization
/+DataContext:Object
+AngryChickenCustomization():AngryChickenCustomization
CowboyCoffeeCustomization
/+DataContext:Object
+CowboyCoffeeCustomization():CowboyCoffeeCustomization
-<<event>>CowboyCoffeeSizeSelect(sender: object, e: RoutedEventArgs)
System.Windows.Controls.UserControl
SwapBillsForCoinsControl
/+DataContext:Object
+SwapBillsForCoinsControl():SwapBillsForCoinsControl
+<<event>>OnDoneSwapping(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
ChiliCheeseFriesCustomization
/+DataContext:Object
+ChiliCheeseFriesCustomization():ChiliCheeseFriesCustomization
-<<event>>ChiliCheeseFriesSizeSelect(sender: object, e: RoutedEventArgs)
CowpokeChiliCustomization
/+DataContext:Object
+CowpokeChiliCustomization():CowpokeChiliCustomization
BillToCoinSwapControl
+BillTypeProperty:DependencyProperty {readonly}
+BillType:Bills <<get, set>>
+CoinTypeProperty:DependencyProperty {readonly}
+CoinType:Coins <<get, set>>
+NumOfBillProperty:DependencyProperty {readonly}
+NumOfBill:int <<get, set>>
+NumOfCoinProperty:DependencyProperty {readonly}
+NumOfCoin:int <<get, set>>
/+DataContext:Object
+BillToCoinSwapControl():BillToCoinSwapControl
+<<event>>OnSwapBillForCoins(sender: object, e: RoutedEventArgs):EventHandler<RoutedEventArgs>
ModelViewCashRegister
+<<event>>PropertyChanged:PropertyChangedEventHandler {abstract}
+CashDrawer:CashRegister.CashDrawer
+RegisterDollars:int <<get, set>>
+RegisterHalfDollars:int <<get, set>>
+RegisterQuarters:int <<get, set>>
+RegisterDimes:int <<get, set>>
+RegisterNickels:int <<get, set>>
+RegisterPennies:int <<get, set>>
+RegistersOnes:int <<get, set>>
+RegisterTwos:int <<get, set>>
+RegisterFives:int <<get, set>>
+RegisterTens:int <<get, set>>
+RegisterTwenties:int <<get, set>>
+RegisterFifties:int <<get, set>>
+RegisterHundreds:int <<get, set>>
+UserDollars:int <<get, set>>
+UserHalfDollars:int <<get, set>>
+UserQuarters:int <<get, set>>
+UserDimes:int <<get, set>>
+UserNickels:int <<get, set>>
+UserPennies:int <<get, set>>
+UserOnes:int <<get, set>>
+UserTwos:int <<get, set>>
+UserFives:int <<get, set>>
+UserTens:int <<get, set>>
+UserTwenties:int <<get, set>>
+UserFifties:int <<get, set>>
+UserHundreds:int <<get, set>>
+OrderCharge:double <<get, set>>
+OrderPayment:double <<get, private set>>
+SufficientFundsIndicator:string <<get, private set>>
+SufficientFundsColor:SolidColorBrush <<get, set>>
+SufficientChangeIndicator:string <<get, set>>
+SufficientChangeColor:SolidColorBrush <<get, set>>
+AppropriateChange:ObservableCollection<string> <<get>>
+ModelViewCashRegister(cashDrawer: CashDrawer):ModelViewCashRegister
+RemoveCoin(coinType: Coins, amount: int):void
+RemoveBill(billType: Bills, amount: int):void
CashRegister.CashDrawer
+Pennies:int {readonly}
+Nickels:int {readonly}
+Dimes:int {readonly}
+Quarters:int {readonly}
+HalfDollars:int {readonly}
+Dollars:int {readonly}
+Ones:int {readonly}
+Twos:int {readonly}
+Fives:int {readonly}
+Tens:int {readonly}
+Twenties:int {readonly}
+Fifties:int {readonly}
+Hundreds:ing {readonly}
+TotalValue {readonly}
+AddCoin(type: Coins, quantity: int):void
+AddBill(type: Bills, quantity: int):void
+RemoveCoin(type: Coins, quantity: int):void
+RemoveBill(type: Bill, quantity: int):void
Throws
<<exception>>
CashRegister.DrawerOverdrawnException
CornDodgersCustomization
/+DataContext:Object
+CornDodgersCustomization():CornDodgersCustomization
-<<event>>CornDodgersSizeSelect(sender: object, e: RoutedEventArgs)
DakotaDoubleBurgerCustmoization
/+DataContext:Object
+DakotaDoubleBurgerCustomization():DakotaDoubleBurgerCustomization
JerkedSodaCustomization
/+DataContext:Object
+JerkedSodaCustomization():JerkedSodaCustomization
-<<event>>JerkedSodaSizeSelect(sender: object, e: RoutedEventArgs)
-<<event>>JerkedSodaFlavorSelect(sender: object, e: RoutedEventArgs)
PanDeCampoCustomization
/+DataContext:Object
+PanDeCampoCustomization():PanDeCampoCustomization
-<<event>>PanDeCampoSizeSelect(sender: object, e: RoutedEventArgs)
PecosPulledPorkCustomization
/+DataContext:Object
+PecosPulledPorkCustomization():PecosPulledPorkCustomization
TexasTeaCustomization
/+DataContext:Object
+TexasTeaCustomization():TexasTeaCustomization
-<<event>>TexasTeaSizeSelect(sender: object, e: RoutedEventArgs)
RustlersRibsCustomization
/+DataContext:Object
+RustlersRibsCustomization():RustlersRibsCustomization
WaterCustomization
/+DataContext:Object
+WaterCustomization():WaterCustomization
-<<event>>WaterSizeSelect(sender: object, e: RoutedEventArgs)
<<Interface>>
System.ComponentModel.INotifyPropertyChanged
+<<event>>PropertyChanged:PropertyChangedEventHandler
TexasTripleBurgerCustomization
/+DataContext:Object
+TexasTripleBurgerCustomization():TexasTripleBurgerCustomization
TrailBurgerCustomization
/+DataContext:Object
+TrailBurgerCustomization():TrailBurgerCustomization


CowboyCafe.Website
Association Legend
= Composition (The diamond-side class 'has-a'[strong] arrow-side class)
= Generalization (The non-arrow-side class 'is-a' arrow-side class)
= Aggregation (The diamond-side class 'has-a'[weak] arrow-side class)
= Realization (The non-arrow-side class 'is-a' arrow-side interface)
= Association (The connected entities are associated in some way)
Program
+Main(args: string[]):void
+CreateHostBuilder(args: string[]):IHostBuilder
1
1
Startup
+Configuration:IConfiguration <<get>>
+Startup(configuration: IConfiguration):Startup
+ConfigurationServices(services: IServiceCollection):void
+Configure(app: IApplicationBuilder, env: IWebHostEnvironment):void
Pages.ErrorModel
+RequestId:string <<get, set>>
+ShowRequestId:bool
- _logger:ILogger<IndexModel> {readonly}
+ErrorModel(logger: ILogger<IndexModel>):IndexModel
+OnGet():void
Microsoft.AspNetCore.Mvc.RazorPages.PageModel
Pages.PrivacyModel
- _logger:ILogger<IndexModel> {readonly}
+PrivacyModel(logger: ILogger<IndexModel>):IndexModel
+OnGet():void
Pages.IndexModel
- _logger:ILogger<IndexModel> {readonly}
+MenuItemsToDisplay:IEnumerable<IOrderItems <<get, protectedSet>>
+Keywords:string <<get, protected set>>
+Categories:string[]
+SearchCategories:IEnumerable<string> <<get, protected set>>
+MinimumCalories:uint? <<get, protected set>>
+MaximumCalories:uint? <<get, protected set>>
+MinimumPrice:uint? <<get, protected set>>
+MaximumPrice:uint? <<get, protected set>>
+IndexModel(logger: ILogger<IndexModel>):IndexModel
+OnGet(Keywords: string, SearchCategories: IEnumerable<IOrderItem>, MinimumCalories:uint?, MaximumCalories:uint?, MinimumPrice:double?, MaximumPrice:double?):void
IndexModel utilizes data and functionality from Menu.
CowboyCafe.Data.Menu