akwindBuffet.PointOfSal urrencyPayment ırrencyControl +ScreenChangeEvent: EventHandler<ScreenChangeEventArgs> +CustomerMoney: DependencyProperty +NewOrderEvent: EventHandler<NewOrderEventArgs> +ChangeMoney: DependencyProperty - orderInstance: Order +ValueType: string <<get, set>> +Hundreds:CurrencyControl <<get, set>> +CustomerMoneyAmount: int <<get, set>> +Fifties:CurrencyControl <<get, set>> +ChangeMoneyAmount: int <<get, set>> +Twenties:CurrencyControl <<get, set>> -OnAddCustomerAmount(sender:object, e:RoutedEventArgs): void +Tens:CurrencyControl <<get, set>> -OnLowerCustomerAmount(sender:object, e:RoutedEventArgs): void +Fives:CurrencyControl <<get, set>> +Twos:CurrencyControl <<get, set>> +Ones:CurrencyControl <<get, set>> +DollarCoins:CurrencyControl <<get, set>> +HalfDollars:CurrencyControl <<get, set>> +Quarterts:CurrencyControl <<get, set>> +Dimes:CurrencyControl <<get, set>> +Nickels:CurrencyControl <<get, set>> +Pennies:CurrencyControl <<get, set>> +FromCustomerLabelBills: Label <<get, set>> +CustomerChangeLabelBills: Label <<get, set>> +FromCustomerLabelCoins: Label <<get, set>> +CustomerChangeLabelCoins: Label <<get, set>> +ReturnToOrderButton: Button <<get, set>> +TotalSaleText: TextBlock << get, set>> +TotalSaleAmount: TextBlock <<get, set>> +AmountDueText: TextBlock << get, set>> +AmountDue: TextBlock <<get, set>> +ChangeOwedText: TextBlock <<get, set>> +ChangeOwed: TextBlock <<get, set>> +FinalizeSale: Button << get, set>> -FinalizeSale_Click(sender:object, e:RoutedEventArgs): void -ReturnTo Order_Click(sender:object, e:Routed EventArgs): void mSelectionScreen +ScreenChangeEvent: EventHandler<ScreenChangeEventArgs>

CustomizationScreen

+MainPanel: StackPanel << get, set>> +TopPanel: TextBox <<get, set>> +MenuItemLabel: Textbox << get, set>> +ConfirmItem: Button <<get, set>> +CancelItem: Button <<get, set>> +ScreenChangeEvent: EventHandler<ScreenChangeEventArgs> -setControls: List<Control> -item: IOrderItem

-CreateFromBoolean(orderItem:IOrderItem, p:PropertyInfo,

-parentItem: IOrderItem

-Refresh(): void

addLast:List<CheckBox>): void -CreateFromEnum(orderItem:IOrderItem, p:PropertyInfo): void -CreateComboltem(combo:Comboltem): void -ConfirmItem_Click(sender:object, e:RoutedEventArgs): void -Cancelltem_Click(sender:object, e:RoutedEventArgs): void -CustomizeEntreeButton_Click(sender:object, e:RoutedEventArgs): void -CustomizeSideButton_Click(sender:object, e:RoutedEventArgs): void -CustomizeDrinkButton_Click(sender:object, e:RoutedEventArgs): void -GetItemsForComboBox(items:IEnumerable<IOrderItem>, obj:IOrderItem): void

+AddDoubleDraugrButton: Button << get, set>> +AddThalmorTripleButton: Button <<get, set>> +AddGardenOrcOmeletteButton: Button <<get, set>> +AddPhillyPoacherButton: Button << get, set>> +AddSmokehouseSkeletonButton: Button << get, set>> +AddThugsTBoneButton: Button <<get, set>> +AddAretinoAppleJuiceButton: Button <<get, set>> +AddCandlehearthCoffeeButton: Button << get, set>> +AddMarkathMilkButton: Button << get, set>> +AddSailorSodaButton: Button << get, set>> +AddWarriorWaterButton: Button << get, set>> +AddDragonbornWaffleFriesButton: Button <<get, set>> +AddFriedMiraakButton: Button << get, set>> +AddMadOtarGritsButton: Button <<get, set>> +AddVokunSaladButton: Button <<get, set>> +AddComboButton: Button << get, set>> +EntreeTxt: TextBox << get, set>>

-AddItem(sender:object, e:RoutedEventArgs): void

+SideTxt: TextBox <<get, set>>

+DrinkTxt: TextBox << get, set>>

+AddBriarhearBurgerButton: Button <<get, set>>

inWindow

-currentControl: UserControl -OrderPanelScreen: OrderPanel <<get, set>> -ltemSelectionScreen: ItemSelectionScreen <<get, set>> -NewOrder(sender:object, e:NewOrderEventArgs): void -ScreenChange(sender:object, e:ScreenChangeEventArgs): void

NewOrderEventArgs

+_newOrder : Order << get, private set>>

ScreenChangeEventArgs

+_newScreen : UserControl <<get, private set>>

Payment Options

+ScreenChangeEvent: EventHandler<ScreenChangeEventArgs> +NewOrderEvent: EventHandler<NewOrderEventArgs> +PaymentOptionsLabel: TextBox <<get, set>> +CashButton: Button<<get, set>> +CreditDebitButton: Button<<get, set>> -CashButton_Click(sender:object, e:RoutedEventArgs): void

-CreditDebitButton Click(sender:object, e:RoutedEventArgs): void

OrderPanel

+NewOrderEvent: EventHandler<NewOrderEventArgs> +OrderNumberLabel: TextBlock << get, set>> +OrderListView: ListBox << get, set>> +SubTotalLabel: TextBlock <<get, set>> +TaxLabel: TextBlock <<get, set>> +TotalLabel: TextBlock <<get, set>> +CompleteOrderButton: Button<<get, set>> +CanelOrderButton: Button<<get, set>> -CompleteOrder(sender:object, e:RoutedEventArgs): void -CancelOrder(sender:object, e:RoutedEventArgs): void -OrderItemSelectionChanged(sender:object, e:SelectionChangedEventArgs): void

+ScreenChangeEvent: EventHandler<ScreenChangeEventArgs>

leakwindBuffet.Website.Pages

ndexModel

+CaloriesMin: uint? < <get, set="">></get,>
+CaloriesMax: uint? < <get, set="">></get,>
+PriceMin: double? < <get, set="">></get,>
+PriceMax: double? < <get, set="">></get,>
+Terms: string < <get, set="">></get,>
+ShowEntrees: bool < <get, set="">></get,>
+ShowSides: bool < <get, set="">></get,>
+ShowDrinks: bool << get, set>>
+OnGet(Terms:string, CaloriesMin:uint?, CaloriesMax:uint?,
PriceMin:double?, PriceMax:double?, ShowEntrees:bool,
ShowDrinks:bool, ShowSides:bool): void
AboutModel

+Entrees: IEnumerable<IOrderItem> <<get, set>>

+Sides: Dictionary<string, List<IOrderItem>> <<get, set>>

+Drinks: Dictionary<string, List<IOrderItem>> <<get, set>>

+OnGet():void

PrivacyModel -_logger:ILogger<PrivacyModel>

rrorModel

+OnGet():void

+RequestID:string <<get,set>> +ShowRequestID:bool <<get>> +OnGet():void

OrderItem

Interface>>

+Price: double <<get>> +Calories: uint <<get>> +Description: string<<get>> +SpecialInstructions: List<string> <<get>>

<u>+Entrees():IEnumerable<IOrderItem></u> +Sides():IEnumerable<IOrderItem>

+Drinks():IEnumerable<IOrderItem>

+ItemName: string<<get>>

+FullMenu():IEnumerable<IOrderItem> +Search(baseItems:IEnumerable<IOrderItem>, terms:string):IEnumerable<IOrderItem>

+FilterByCalories(baseItems:IEnumerable<IOrderItem>, uint?:min, <u>uint?:max): IEnumerable<IOrderItem></u>

+FilterByPrice(baseItems:IEnumerable<IOrderItem>, double?:min, double?:max): IEnumerable<IOrderItem>

CashRegister

+Total: double << get,set>> +Due: double <<get>> +Owed: double <<get>> +AmountPaid: double <<get>> +HundredsCustomer: int <<get,set>> +FiftiesCustomer: int <<get,set>> +TwentiesCustomer: int <<get,set>> +TensCustomer: int <<get,set>>

+Fives Customer: int <<get, set>>

+TwosCustomer: int <<get,set>> +OnesCustomer: int <<get,set>> +DollarCoinsCustomer: int << get,set>> +HalfDollarCustomer: int <<get, set>> +QuartersCustomer: int <<get,set>> +DimesCustomer: int << get, set>>

+NickelsCustomer int <<get,set>> +PenniesCustomer: int <<get,set>>

+HundredsChange: int <<get>> +FiftiesChange: int <<get>> +TwentiesChange: int << get>> +TensChange: int <<get>>

+FivesChange: int <<get>> +TwosChange: int <<get>> +OnesChange: int <<get>> +DollarCoinsChange: int <<get>>

+HalfDollarsChange: int <<get>> +QuartersChange: int <<get>> +DimesChange: int <<get>> +NickelsChange: int <<get>> +PenniesChange: int <<get>>

+UpdateChangeValues(amount:double):void +FinalizeTransaction():void

-_items: ObservableCollection<IOrderItem> _salesTaxRate: double

-_orderNumber: int -nextOrderNumber: int +PropertyChanged: PropertyEventHandler<<get>>> +CollectionChanged: NotifyCollectionChangedEventHandler<<get>>

+Subtotal: Double <<get>> +Tax: Double <<get>> +Total: Double <<get>>

+Number: int <<get>> +OrderItems: IEnumerable<IOrderItem> << get>>

+Count: int <<get>> +IsReadOnly: bool <<get>>

+Add(item:IOrderItem):void +Remove(item:IOrderItem):bool +Contains(item:IOrderItem):bool

+LimitLine(it:string):List<string>

+SalesTaxRate: Double <<get, set>>

+CopyTo(array:IOrderItem[], arrayIndex:int):void +GetEnumerator():IEnumerator<IOrderItem> -ItemPropChanged(sender:object,e:PropertyChangedEventArgs):void +PrintReciept(type:PaymentType, changeOwed:double):void

omboltem

 \Diamond

+PropertyChanged: PropertyEventHandler<<get>>> +Entree: Entree <<get, set>> +Side: Side <<get, set>>

+Drink: Drink <<get, set>> +Price: double <<get>> +Calories: uint <<get>>

+SpecialInstructions: List<string> <<get>> +Description: string<<get>>

+ItemName: string<<get>>

+ToString(): string {override}

-EntreePropChanged(sender:object,e:PropertyChangedEventArgs):void -SidePropChanged(sender:object,e:PropertyChangedEventArgs):void -DrinkPropChanged(sender:object,e:PropertyChangedEventArgs):void

hillyPoacher +PropertyChanged: PropertyEventHandler<<get>>

+Price: double <<get>> +Calories: uint <<get>>

+Description: string<<get>> +Roll: bool <<get, set>> +SpecialInstructions: List<string> <<get>> +ItemName: string<<get>>

riarheartBurger

leakwindBuffet.Data.Entre

+PropertyChanged: PropertyEventHandler<<get>> {override} +Bun: bool <<get, set>> +Ketchup: bool <<get, set>> +Mustard: bool <<get, set>> +Pickle: bool <<get, set>> +Cheese: bool <<get, set>> +Price: double <<get>> {override} +Calories: uint <<get>> {override} +Description: string <<get>> {override} +SpecialInstructions: List<string> << get>> {override} +ToString(): string {override}

oubleDraugr

+PropertyChanged: PropertyEventHandler<<get>>> {override} +Bun: bool <<get, set>> +Ketchup: bool <<get, set>> +Mustard: bool <<get, set>> +Pickle: bool <<get, set>> +Cheese: bool << get, set>> +Tomato: bool <<get, set>> +Lettuce: bool <<get, set>> +Mayo: bool <<get, set>> +Price: double <<get>> {override} +Calories: uint <<get>> {override} +Description: string <<get>> {override} +SpecialInstructions: List<string> << get>> {override} +ToString(): string {override}

halmorTriple

+PropertyChanged: PropertyEventHandler<<get>>> {override} +Bun: bool <<get, set>>

+Ketchup: bool <<get, set>> +Mustard: bool <<get, set>> +Pickle: bool <<get, set>> +Cheese: bool << get, set>> +Tomato: bool <<get, set>> +Lettuce: bool <<get, set>> +Mayo: bool <<get, set>> +Bacon: bool <<get, set>> +Egg: bool <<get, set>>

+ToString(): string {override}

+Price: double <<get>> {override} +Calories: uint <<get>> {override} +Description: string <<get>> {override} +SpecialInstructions: List<string> <<get>> {override}

+PropertyChanged: PropertyEventHandler <<get>>> {override}

+Sirloin: bool <<get, set>> +Onion: bool <<get, set>>

+Price: double <<get>> {override} +Calories: uint <<get>> {override} +Description: string <<get>>> {override} +SpecialInstructions: List<string> << get>> {override}

ardenOrcOmelette | +PropertyChanged: PropertyEventHandler <<get>> {override}

+ToString(): string {override}

+Broccoli: bool <<get, set>> +Mushrooms: bool <<get, set>> +Tomato: bool <<get, set>>

+Cheddar: bool <<get, set>> +Price: double <<get>> {override} +Calories: uint <<get>> {override}

+Description: string <<get>> {override} +SpecialInstructions: List<string> << get>> {override} +ToString(): string {override}

nokehouseSkeleton

+PropertyChanged: PropertyEventHandler <<get>> {override} +SausageLink: bool <<get, set>>

+Egg: bool <<get, set>> +HashBrowns: bool <<get, set>> +Pancake: bool <<get, set>>

+ToString(): string {override}

+Price: double <<get>> {override} +Calories: uint <<get>> {override} +Description: string <<get>> {override} +SpecialInstructions: List<string> << get>> {override}

nugsTBone

+PropertyChanged: PropertyEventHandler <<get>>> {override}

+Price: double <<get>> {override} +Calories: uint <<get>> {override} +Description: string <<get>> {override} +SpecialInstructions: List<string> << get>> {override} +ToString(): string {override}

+PropertyChanged: PropertyEventHandler <<get>>

T-----

+Size: Size << get, set>> +Price: double <<get>>

> +Calories: uint <<get>> +Description: string<<get>>

+SpecialInstructions: List<string> <<get>> +ItemName: string<<get>>

etino Apple Juice

+PropertyChanged: PropertyEventHandler <<get>>> {override}

+lce: bool <<get, set>> +Size: Size << get, set>> {override} +Price: double <<get>> {override}

+Calories: uint <<get>> {override} +Description: string <<get>> {override}

+SpecialInstructions: List<string> << get>> {override} +ToString(): string {override}

CandlehearthCoffee

+PropertyChanged: PropertyEventHandler <<get>>> {override}

+lce: bool <<get, set>> +Decaf: bool <<get, set>>

+RoomForCream: bool <<get, set>> +Size: Size << get, set>> {override} +Price: double <<get>> {override}

+Calories: uint <<get>> {override} +Description: string <<get>> {override}

+SpecialInstructions: List<string> << get>> {override} +ToString(): string {override}

/larkarthMilk

+PropertyChanged: PropertyEventHandler <<get>>> {override}

+lce: bool <<get, set>> +Size: Size << get, set>> {override} +Price: double <<get>> {override}

+Calories: uint <<get>> {override} +Description: string <<get>> {override}

+SpecialInstructions: List<string> << get>> {override}

+ToString(): string {override}

SailorSoda

+PropertyChanged: PropertyEventHandler <<get>>> {override} +Ice: bool <<get, set>> +Size: Size <<get, set>>

+Flavor: SodaFlavor << get, set>> +Price: double <<get>> {override}

+Calories: uint <<get>>> {override} +Description: string <<get>>> {override} +SpecialInstructions: List<string> << get>> {override}

NarriorWater

+PropertyChanged: PropertyEventHandler <<get>> {override}

+lce: bool <<get, set>> +Size: Size <<get, set>>

+ToString(): string {override}

+Lemon: bool <<get, set>>

+Price: double <<get>> {override} +Calories: uint <<get>> {override}

+Description: string<<get>> +SpecialInstructions: List<string> << get>> {override}

+ToString(): string {override}

+PropertyChanged: PropertyEventHandler <<get>>

+Size: Size <<get, set>> +Price: double <<get>>

+Calories: uint <<get>>

+Description: string<<get>> +SpecialInstructions: List<string> <<get>> +ItemName: string<<get>>

agonbornWaffleFries

+PropertyChanged: PropertyEventHandler <<get>>> {override}

+Size: Size << get, set>> {override} +Price: double <<get>> {override}

+Calories: uint <<get>> {override} +Description: string <<get>> {override}

+ToString(): string {override}

riedMiraak

+SpecialInstructions:List<string> <<get>> {override}

+PropertyChanged: PropertyEventHandler <<get>> {override}

+Size: Size << get, set >> {override} +Price: double <<get>> {override}

+Calories: uint <<get>> {override} +Description: string <<get>> {override}

+SpecialInstructions: List<string> << get>> {override} +ToString(): string {override}

//adOtarGrits

+PropertyChanged: PropertyEventHandler <<get>> {override}

+Size: Size << get, set>> {override} +Price: double <<get>> {override}

+Calories: uint <<get>> {override}

+Description: string <<get>> {override}

+SpecialInstructions: List<string> << get>> {override} +ToString(): string {override}

okunSalad

+PropertyChanged: PropertyEventHandler <<get>>> {override}

+Size: Size << get, set>> {override} +Price: double <<get>> {override}

+Calories: uint <<get>> {override}

+Description: string <<get>> {override} +SpecialInstructions: List<string> << get>> {override}

+ToString(): string {override}

kwindBuffet.Data.Enum

numeration>>

Small

Medium Large

Enumeration>>

daFlavor

Blackberry Cherry Grapefruit

Watermelon

Enumeration>>

aymentTypes

Lemon Peach

Cash Card