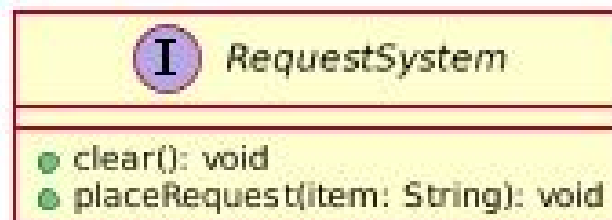
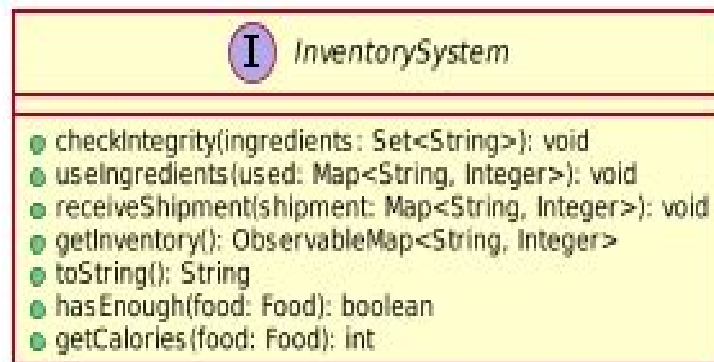
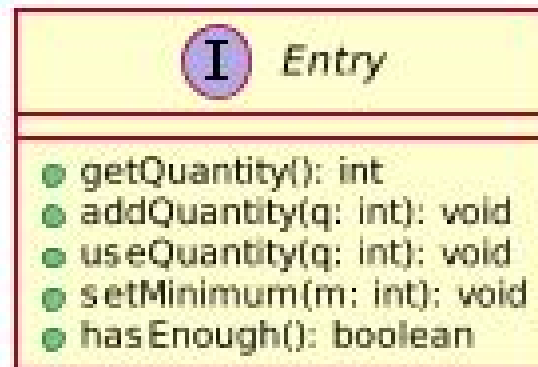
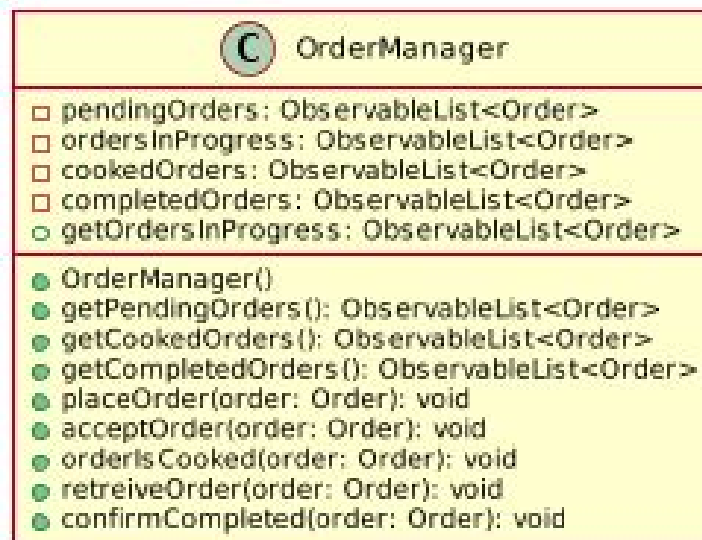
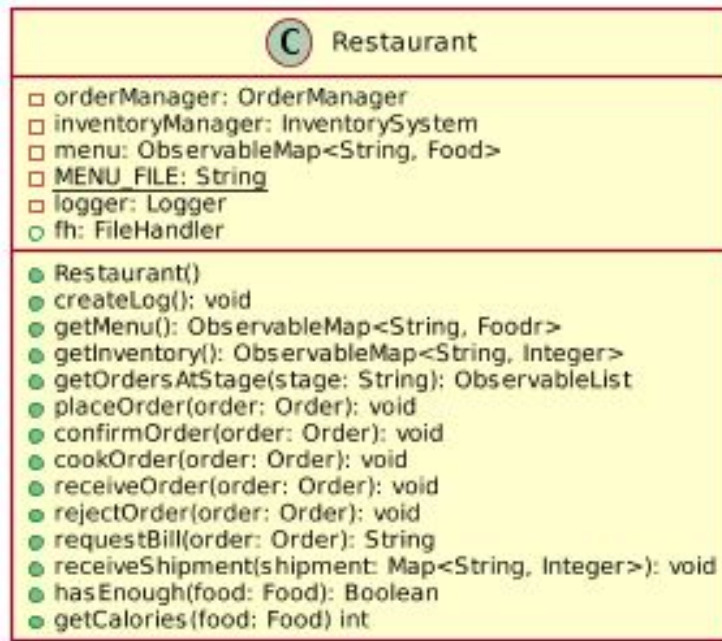


# RestaurantModel- Interfaces



# RestaurantModel- Managers



C InventoryManager
<ul style="list-style-type: none"> <li>□ inventory: Map&lt;String, SimpleEntry&gt;</li> <li>□ requests: RequestManager</li> <li>□ <u>INVENTORY_FILE</u>: String</li> <li>□ <u>MINIMUM_FILE</u>: String</li> </ul>
<ul style="list-style-type: none"> <li>● InventoryManager()</li> <li>● checkIntegrity(ingredients: Set&lt;String&gt;): void</li> <li>● useIngredients(used: Map&lt;String, Integer&gt;): void</li> <li>● receiveShipment(shipment: Map&lt;String, Integer&gt;): void</li> <li>● getInventory(): ObservableMap&lt;String, Integer&gt;</li> <li>● toString(): String</li> <li>● hasEnough(food: Food): boolean</li> <li>● getCalories(food: Food): int</li> </ul>

C ComplexInventory
<ul style="list-style-type: none"> <li>□ inventory: Map&lt;String, CalorieEntry&gt;</li> <li>□ requests: RequestManager</li> <li>□ <u>INVENTORY_FILE</u>: String</li> <li>□ <u>MINIMUM_FILE</u>: String</li> <li>□ <u>CALORIE_TABLE</u>: String</li> </ul>
<ul style="list-style-type: none"> <li>● ComplexInventory()</li> <li>● checkIntegrity(ingredients: Set&lt;String&gt;): void</li> <li>● useIngredients(used: Map&lt;String, Integer&gt;): void</li> <li>● receiveShipment(shipment: Map&lt;String, Integer&gt;): void</li> <li>● getInventory(): ObservableMap&lt;String, Integer&gt;</li> <li>● hasEnough(food: Food): boolean</li> <li>● getCalories(food: Food): int</li> <li>● toString(): String</li> </ul>

C RequestManager
<ul style="list-style-type: none"> <li>□ requested: Set&lt;String&gt;</li> <li>□ <u>REORDER_FILE</u>: String</li> </ul>
<ul style="list-style-type: none"> <li>● RequestManager()</li> <li>● clear(): void</li> <li>● placeRequest(item: String): void</li> </ul>

## RestaurantModel- Objects



### SimpleEntry

- quantity: int
- minimum: int

- SimpleEntry(q: int)
- SimpleEntry(q: int, m: int)
- getQuantity(): int
- addQuantity(q: int): void
- useQuantity(q: int): void
- setMinimum(m: int): void
- hasEnough(): boolean



### CalorieEntry

- quantity: int
- minimum: int
- calorieCount: int

- CalorieEntry(q: int)
- CalorieEntry(q: int, m: int)
- getCalorieCount(): int
- setCalorieCount(c: int): void
- getQuantity(): int
- addQuantity(q: int): void
- useQuantity(q: int): void
- setMinimum(m: int): void
- hasEnough(): boolean

## C Order

- ❑ foods: ObservableList<Food>
- ❑ tableNumber: int
- ❑ classOrderNumber: int
- ❑ nickname: String
- ❑ instructions: String
- ❑ orderNumber: int
- setNickname(nickname: String): void
- getInstructions(): String
- setInstructions(instructions: String): void
- getOrderNumber(): int
- Order()
- getAllIngredients(): Map<String, Integer>
- addFood(food: Food): void
- removeFood(food: Food): void
- getPrices(): Map<String, float>
- getTotalPrice(): double
- toString(): String

## C Food

- ❑ ingredients: Map<String, Integer>
- ❑ price: float
- ❑ name: String
- ❑ instructions: String
- ❑ changedIngredients: Map<String, Integer>
- Food(name: String, price: float, ingredients: Map<String, Integer>)
- Food(name: String, price: float)
- Food()
- Food(another: Food)
- getChangedIngredients(): Map<String, Integer>
- getName(): String
- getInstructions(): String
- setInstructions(instructions: String): void
- addIngredient(ingredientName: String, ingredientQuantity: int): void
- removeIngredient(ingredientName: String, ingredientQuantity: int): void
- setPrice(price: float): void
- getPrice(): float
- getIngredients(): Map<String, Integer>
- toString(): String

# Restaurant GUI-Controllers

## I WorkerController

- `init(restaurant: RestaurantModel): void`

## C TakeOrderController

- `previousScene: Scene`
- `menuList: ListView`
- `orderList: ListView`
- `ingredientList: ListView`
- `tableNumberInput: TextField`
- `ingredientBox: TextField`
- `orderNickname: TextField`
- `orderInstructions: TextField`
- `customName: TextField`
- `customPrice: TextField`
- `customInstructions: TextFields`
- `orderLabel: Label`
- `order: Order`
- `restaurant: RestaurantModel`
- `ingredientsFiltered: FilteredList<String>`

- `init(restaurant: RestaurantModel): void`
- `addIngredient(): void`
- `removeIngredient(): void`
- `submitOrder(event: ActionEvent): void`
- `cancelOrder(event: ActionEvent): void`
- `removeFood(): void`
- `addCustomFood(): void`

## C BillController

- `parentGrid: GridPane`
- `finalPrice: Label`
- `fullBill: Label`
- `tipPrice: Label`
- `subTotal: Label`
- `tipInput: TextField`
- `prevButton: Button`
- `tipAmount: double`
- `totalPrice: double`

- `init(orders: ObservableList<Order>, restaurant: RestaurantModel, button: Button): void`
- `setLabels(taxAmount: double, tipAmount: double, totalPrice: double): void`
- `closeBill(): void`



## CookController

- receiveShipment: Button
  - prevLabel: Label
  - postLabel: Label
  - infoLabel: Label
  - prevOrderButton: Button
  - postOrderButton: Button
  - prevOrderList: ListView
  - postOrderList: ListView
  - localToCook: ObservableList<Order>
  - restaurant: RestaurantModel
- 
- init(restaurant: RestaurantModel): void
  - initialize(): void
  - receiveShipment(event: ActionEvent): void
  - cancelOrder(): void



## ManagerController

- confirmList: ListView
  - cookList: ListView
  - deliverList: ListView
  - billList: ListView
  - inventoryList: ListView
  - restaurant: restaurantModel
- 
- init(restaurant: RestaurantModel): void
  - receiveShipment(event: ActionEvent): void
  - addServer(): void
  - addCook(): void



## ReceiveShipmentController

- previousScene: scene
  - ingredientsList: ListView<String>
  - shipmentsList: ListView<String>
  - quantityBox: TextField
  - searchBar: TextField
  - ingredientsFiltered: FilteredList<String>
  - shipment: ObservableMap<String, Integer>
  - restaurant: RestaurantModel
- 
- init(restaurant: RestaurantModel): void
  - removeIngredient(): void
  - confirmShipment(event: ActionEvent): void



## ServerController

- receiveShipment: Button
  - prevLabel: Label
  - postLabel: Label
  - prevOrderButton: Button
  - postOrderButton: Button
  - takeOrderButton: Button
  - prevOrderList: ListView
  - postOrderList: ListView
  - restaurant: RestaurantModel
- 
- init(restaurant: RestaurantModel): void
  - initialize(): void
  - receiveShipment(event: ActionEvent): void
  - cancelOrder(): void



# Restaurant GUI-Initializer



