

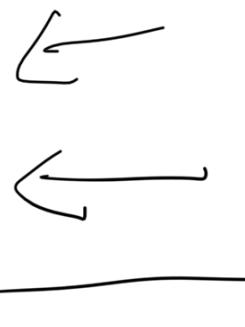
# Model :

chefController

```

Employee Manager
- employeeHashMap : HashMap<String, Employee>
- servers : Server[]
- chefs : Chef[]
- inventory : Inventory2
- orderManager : OrderManager
- tableManager : TableManager
- int i, j = 0
+ EmployeeManager(OrderManager, TableManager, inventory)
+ getEmployeeHashMap() : HashMap<String, Employee>
+ addEmployee(name: String, employeePosition: String)
+ addEmployee(name: String, manager: Manager) : void
~ removeEmployee(name: String) : void
- writeEmployeeFile() : void

```

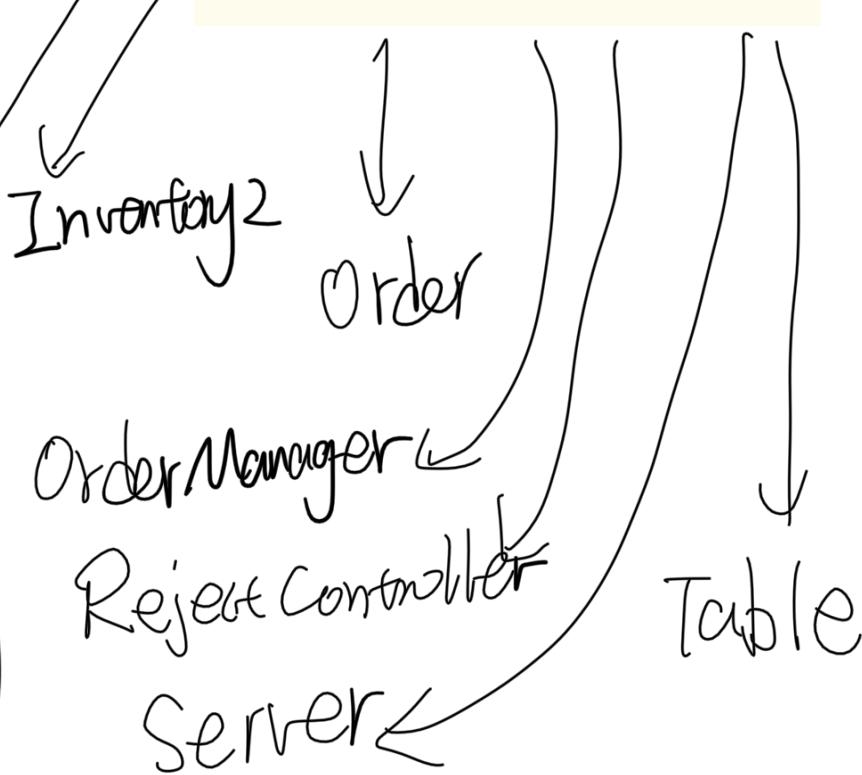


## Phase2Log

```

- Logger = Logger.getLogger(Phase2Log.class.getName())
- instance : Phase2Log
+ getInstance() : Phase2Log
- phase2Log()
+ putLog(Level level, message: String): void

```



```

Manager
+ setName : ArrayList<String>
+ employeeManager : EmployeeManager
- payment : HashMap<String, ArrayList<String>>
+ setRequest(name: String, quantity: Integer) : void
+ hire(name: String, position: String) : void
+ fire(name: String) : void
+ getEmployeeManager() : EmployeeManager
+ getIncomes() : void
+ getPayment(date: String) : ArrayList<String>
+ getInventory() : Inventory2
+ validateDate(date: String) : boolean
+ getTotal(date: String) : double

```

<Interface> Observable

```

register(observer: Observer) : void
notifyObservers() : void.

```

<Interface> Observer

```

update() : void

```

**Employee**

- observers : ArrayList<Observer>
- ~ name : String
- ~ inventory : Inventory

```

+ Employee(name: String, inventory: Inventory2)
+ getInventory() : Inventory2
+ receive(receiveName: String, receiveQuantity: int) : void
~ findOrder(orderName: String, tableNum: int, orderList: ArrayList<Order>) : Order
+ register(observer: Observer) : void
+ notifyObserver() : void

```

**Inventory 2**

- items : ArrayList<Ingredient>
- requestIngredients : HashMap<String, Integer>

+ Inventory2()

```

+ getQuantity(ingredientName: String) : int
+ getItems() : ArrayList<Ingredient>
+ getRequestedIngredients() : HashMap<String, Integer>
~ contains(ingredientName: String) : boolean
- writefile() : void
~ addItems(ingredient: String, quantity: int) : void
~ useItems(ingredient: String, quantity: int) : void
~ selfCheck() : void
~ writeRequest() : void
+ readRequest() : void

```

**Ingredient**

- ingredientName : String
- quantity : int

```

+ Ingredient(name: String, amount: int)
+ getIngredientName() : String
+ getQuantity() : int
+ setQuantity() : void

```

**Server**

- tableManager : TableManager
- orderManager : OrderManager

```

+ Server(name: String, inventory: Inventory2, tableManager: TableManager, orderManager: OrderManager)
- isValidOrder(orderName: String) : boolean
+ createOrder(orderName: String, tableNum: int) : Order
+ cancelOrder(order: Order) : void
+ deliverOrder(orderName: String, tableNum: int) : void
+ freeOrder(orderName: String, tableNum: int) : void
+ getOrderManager() : OrderManager
+ getTableManager() : TableManager
+ refrigerate(orderName: String, tableNum: int) : void
+ reheat(orderName: String, tableNum: int) : void
+ writeDetails(orderName: String, tableNum: int) : void
~ updateInMenu(details: String) : void
- updateInventory(order: Order) : void

```

**Chef**

- orderManager : OrderManager
- toPrepare : ArrayList<Order>

```

+ Chef(name: String, inventory: Inventory2, orderManager: OrderManager)
+ getOrderManager() : OrderManager
+ seeOrder(orderName: String, tableNum: int) : void
+ getToPrepare() : ArrayList<Order>
+ fillOrder(orderName: String, tableNum: int) : void
+ cancelOrder(order: Order) : void

```

## OrderManager

```
- orders: ArrayList<Order>
- orders_wok: ArrayList<Order>
- orders_deliver: ArrayList<Order>
- refrigerator: ArrayList<Order>

+ OrderManager()
+ getOrder (orderName: String): Order
+ getOrders_wok(): ArrayList<Order>
+ getOrders_deliver(): ~
+ getRefrigerator(): ~
+ cancelOrder (order: Order): void
+ addOrder_refrigerator (order: Order): void
+ addOrder_cook (order: Order): void
+ addOrder_deliver (order: Order): void
```

## Order

```
- name: String
- tableNum: int
- ingredients: HashMap<String, Integer>
- price: double
- freeOrder: boolean
- LIMIT = 5: int

+ Order (dishName: String, tableNumber: int)
+ isFreeOrder(): boolean
+ getTableNum(): int
+ getIngredients (order: Order): HashMap<String, Integer>
+ getPrice (order: Order): double
+ getName (): String
+ getPrice (): double
+ getIngredients (): HashMap<String, Integer>
+ addIngredient (ingredientName: String): void
+ subtractIngredient (ingredientName: String): void
+ equals (orderName: String, tableNum: int): boolean
+ toString (): String
```

## TableManager

```
- tables

+ TableManager()
+ addTable (table: Table): void
+ getTable (tableNumber: int): Table
```

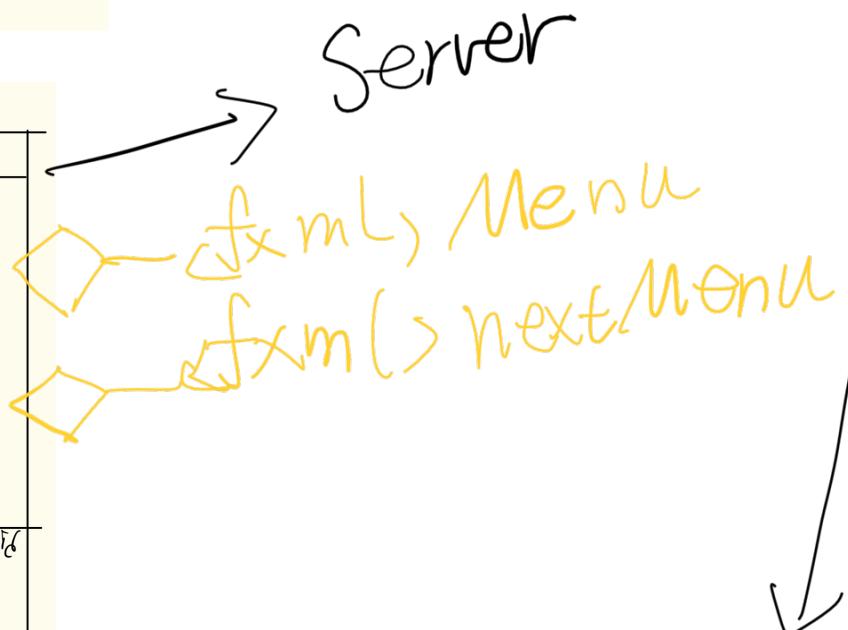
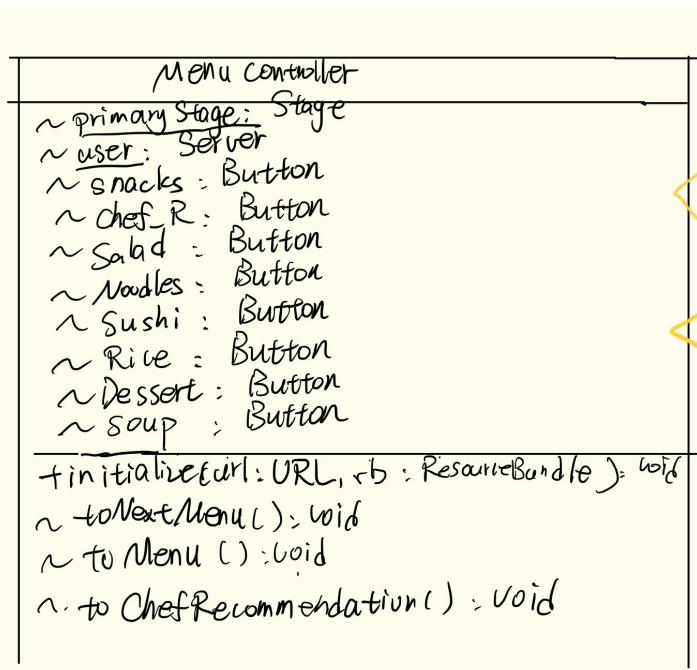
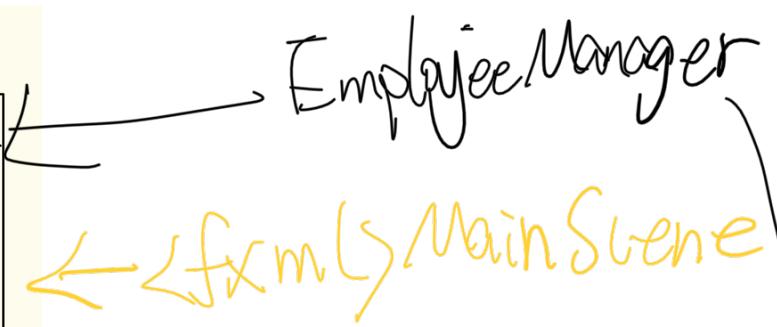
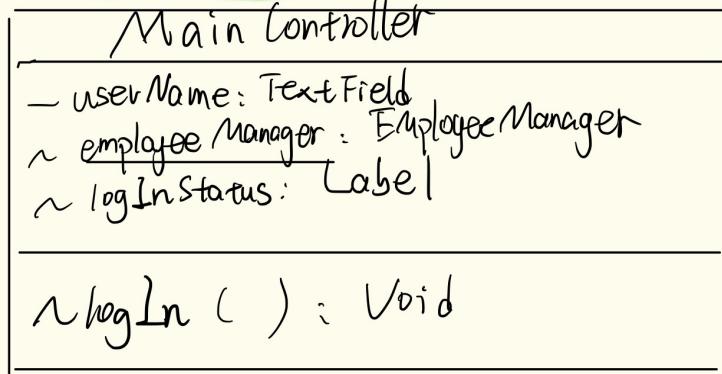


## Table

```
- orders: ArrayList<Order>
- tableNumber: int
- numPeople: int

+ Table (tableNum: int)
+ getTableNumber (): int
+ getNumPeople (): int
+ setNumPeople (): void
+ overEight (): boolean
+ getOrders (): ArrayList<Order>
+ addOrder (): void
+ calculateBill (): double
+ clean (): void
```

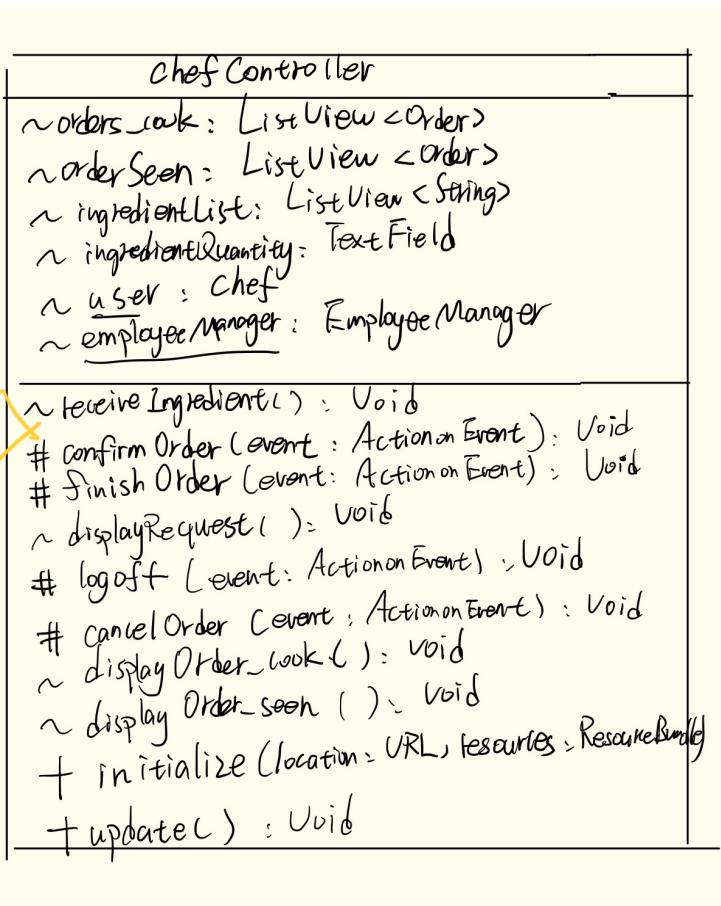
# GUI



Chef

FXML Chef Scene

FXML Order View



### Order Controller

```

~upperSelection : CheckBox
~lowerSelection : CheckBox
~Add : Button
~Sub : Button
~Clear : Button
~Submit : Button
~Cart : Button
~Back : Button
~Next : Button
~upperList : ListView
~lowerList : ListView
~upperImage : ImageView
~lowerImage : ImageView
~Title : Text
~upperPrice : Text
~lowerPrice : Text
~Notation : Text
~cat : ImageView
- ingredient : HashMap<String, ArrayList<String>>
~click : int
~scene : int
~orderPicture : HashMap
~name : String

~uniqueChoice() : void
- displayIngredients() : void
- getList() : ListView
- getUnList() : ListView
- toString(ingredients : ObservableList<String>, type : String) : String
~BoxSelect() : void
~PressSubmit() : void
- getSelected() : CheckBox
- getIngredient(Order : String) : HashMap<String, Integer>
~toMenu():
- findIndex(type : String, name : String) : int
- findType(type : String) : int
- changeImage() : void
~toCart() : void
~addPicture() : void
- helper(map : HashMap<String, ArrayList<String>>, list : String[])
- getPrice(Order : String) : double
+ initialize(location : URL, resource : ResourceBundle)

```

### Reject Controller

```

~user : Server
~order : Order
~otherReason : TextField
~reasons : ChoiceBox<String>

```

```

~rejectOrder() : void

```

```

+ initialize(location : URL,
             resources : ResourceBundle)
             : void

```

Server

ifxml  
RejectScene

**Server Controller**

- ~ user : Server
- ~ employeeManager : EmployeeManager
- currentTable : Table
- ~ ingredientList : ListView<String>
- ~ numPeople : TextField
- ~ orderOnTable : ListView<String>
- ~ ingredientQuantity : TextField
- ~ currentSelectedTable : Label
- ~ billsLeft : Label
- ~ tipsAdded : Label
- ~ numOfPeople : Label
- ~ stillOrderToDeliver : Label
- ~ orderPlaced : ListView<Order>
- ~ orderToDeliver : ListView<Order>

---

- ~ addOrder() : void
- ~ reject() : void
- + initialize(location: URL, resources: ResourceBundle) : void

---

- ~ receiveIngredient() : void
- displayRequest() : void
- + specifyTable(event: Event) : void
- # logOff() : void
- displayOrderOnTable() : void
- displayOrderPlaced() : void
- displayOrderToDeliver() : void
- displayBill() : void
- ~ cancelOrder() : void
- ~ payOrders() : void
- ~ deliverOrder() : void
- ~ setNumPeople() : void
- + update() : void

**AlertWindow**

- stage: Stage

---

- ~ AlertWindow(window: Stage)
- ~ display(title: String, scene: Scene) : void

**Add Order**

**ServerScene**

**Employee Manager**

**Table**

**Cart Controller**

---

- CartView : ListView<String>
- ~ tankYou: ImageView
- ~ orderManager: OrderManager
- ~ order: HashMap<String, ArrayList>
- ~ currentTable: Table
- ~ user: Server

---

- ~ toMenu(): void
- ~ showCart(): void
- ~ removeOrder(): void
- ~ createOrder(): void

---

- toString(Order: HashMap) : String

+ initialize(location: URL, resources: ResourceBundle) : void

FXML  
Cart

Server

Order Manager

FXML  
Manager Scene

**Manager Controller**

---

- ~ user : Manager
- ~ employeeManager: EmployeeManager
- ~ Ingredient : ListView<String>
- ~ Request : ListView<String>
- ~ Records : ListView<String>
- ~ EmployeeNames : ListView<String>
- ~ requestList : ListView<String>
- ~ dates : TextField
- ~ newEmployeePosition : ChoiceBox<String>
- ~ totalIncome : Label

---

~ receiveIngredient(): void

~ fire(): void

~ hire(): void

~ displayIngredients(): void

~ displayRequests(): void

~ displayEmployees(): void

~ SetRequest(Event.ActionEvent): void

# logOff(): void

# check(): void

+ initialize(location: URL, resources: ResourceBundle) : void

+ update(): void