InventoryController

- ingredientTableView: TableView<Ingredient>
- refID: TableColumn<Ingredient, String>
- name: TableColumn<Ingredient, String>
- number: TableColumn<Ingredient, String>
- email: TableColumn<Ingredient, String>
- searchValue: TextField
- numberAddValue: TextField
- refIDValue: Label
- numberValue: Label
- nameValue: Label
- errorInfo: Label
- needListView: ListView<String>
- stringValue TextArea

- toMain(event: ActionEvent): void
- initialize(): void
- search(event: ActionEvent): void
- addStock(event: ActionEvent): voidi
- addStockByString(event: ActionEvent): void

CookController

- cookQueueTableView: TableView<Order>
- deliverQueueTableView: TableView<Order>
- cTime: TableColumn<Order, String>
- cStatus: TableColumn<Order, String>
- cTime: TableColumn<Order, String>
- dTable: TableColumn<Order, String>
- errorInfo: Text
- cookOrderDetail: Text
- deliverOrderDetail: Text

- initialize(): void
- setAcknowledge(event: ActionEvent): void
- setComplete(event: ActionEvent): void
- setDeliver(event: ActionEvent): void
- setCancel(event: ActionEvent): void

ManagingController

- ingredientLogTV: TableView<IngredientLog>
- orderLogTV: TableView<BillLog>
- ingreTimeTC: TableColumn<IngredientLog, String>
- ingreNameTC: TableColumn<IngredientLog, String>
- ingreNumberTC: TableColumn<IngredientLog, String>
- orderTimeTC: TableColumn<Bill, String>
- orderIDTC: TableColumn<Bill, String>
- tableIDTC: TableColumn<Bill, String>
- orderDetail: Text
- timePicker1: DatePicker
- timePicker2: DatePicker

- toMain(event: ActionEvent): void
- initialize(): void
- toStats(event: ActionEvent): void
- toModify(event: ActionEvent): void
- search(event: ActionEvent): void

NewRecipeIngreController

- recipeNameTF: TextField
- recipeRefIDTF: TextField
- recipePriceTF: TextField
- ingreNameTF: TextField
- ingreRefIDTF: TextField
- shortTF: TextField
- addTF: TextField
- ingrePriceTF: TextField
- emailTF: TextField
- recipeIngreTF: TextArea
-ingreErrorInfo: Label
- ingreSuccessInfo: Label
- recipeErrorInfo: Label
- recipeSuccessfulInfo: Label
- recipeIngreErrorInfo: Label
- cold: RadioButton
- hot: RadioButton
- sides: RadioButton
- topping: RadioButton
- typeChoice: ToggleGroup

- initialize(): void
- addNewIngre(event: ActionEvent): void
- toModify(event: ActionEvent): void

MenuController

- refID: TableColumn<Ingredient, String>
- name: TableColumn<Ingredient, String>
- number: TableColumn<Ingredient, String>
- scrollpane: ScrollPane
- newOrder: Order
- menuFlow: FlowPane
- size: int
- totalPrice: double

- initFlowPane(): void
- topMenuButton(event: ActionEvent): void
- toMain(event: ActionEvent): void
- initialize(): void
- backButton(event: ActionEvent): void
- confirmButton(event: ActionEvent): void
- addDishButton(event: ActionEvent): void
- removeDishButton(event: ActionEvent): void
- addDish(addId: int): String
- checkDish(dish: Dish): boolean
- removeDish(addId: int): void
- addToppingPane(recipe: Recipe, id: int): ScrollPane
- addSizeBox(recipe:Recipe): ChoiceBox<String>
- addGridPane(recipe: Recipe, id: int): GridPane
- setGridLayout(grid GridPane): void

StatsController

- timePicker1: DatePicker
- timePicker2: DatePicker
- errorInfo: Label
- moneyC: LineChart<String, Number>
- ingredientLV: ListView<String>
- dishLV: ListView<String>

- initialize(): void
- search(event: ActionEvent): void
- toManaging(event: ActionEvent): void
- search(event: ActionEvent): void
- getChart(date1: LocalDate, date2: LocalDate): List<XYChart.Series<String, Number>>

TableController

- orderedItems: ListView<Dish>
- currentTableID: Label
- peopleNumberLabel: Label
- brokenButton: Button
- errorInfo: Text

handlePayByDish(event: ActionEvent): void
handleDishCancelled(event: ActionEvent): void
- toMain(event: ActionEvent): void
- getTable(tableName(event: ActionEvent): Table
- initialize(): void
handleSeatedButton(event: ActionEvent): void
- setPeopleNumOfTable(int peopleNum): void
handlePayTogether(event: ActionEvent): void
handlePaySeparately(event: ActionEvent): void
handleViewMenu(event: ActionEvent): void
handleReheat(event: ActionEvent): void
handleRemake(event: ActionEvent): void
handleTableSelected(event: ActionEvent): void
updateTableInfo(currentTable: Table): void
handleBroken(event: ActionEvent): void

RecipeIngreModifierController

- orderLogTV: TableView<BillLog>
- ingreTimeTC: TableColumn<IngredientLog, String>
- ingreNameTC: TableColumn<IngredientLog, String>
- moneyC: LineChart<String, Number>
- ingredientLV: ListView<String>
- dishLV: ListView<String>
- ingreNameTF: TextField
- ingreRefIDTF: TextField
- shortTF: TextField
- addTF: TextField
- ingrePriceTF: TextField
- emailTF: TextField

- toMain(event: ActionEvent): void
- initialize(): void
- toAddNew(event: ActionEvent): void
- toManaging(event: ActionEvent): void
- setRecipe(event: ActionEvent): void
- setIngredient(event: ActionEvent): void
- deleteIngredient(event: ActionEvent): void
- deleteRecipe(event: ActionEvent): void
- convertIngredientsText(ingreText String): Map<Ingredient, Double>

WelcomeController

- toTable(event: ActionEvent): void
- toInventory(event: ActionEvent): void
- toManaging(event: ActionEvent): void
- toCook(event: ActionEvent): void