Home Screen

[1] - New Order
[0] - Exit application

Order Screen

[1] - Add Sandwich
[2] - Add Drink
[3] - Add Chips
[4] - Checkout
[0] - Cancel Order - delete order

Order Class

- sandwiches: List<Sandwich>
- drinks: List<Drink>
- chips: List<Chip>

+ addSandwich(sandwich: Sandwich)
+ addDrink(drink: Drink)
+ addChip(chip: Chip)
+ calculateTotalPrice(): double
+ generateReceipt(): void
+ displayOrderDetails(): void

+clearOrder(): void

Add Sandwich Screen

1) select bread
2) select sandwich size
3) Choose toppings:
   Meat
   Cheese
   Other toppings:
   Select sauces:

4)Toasted?

Add Drink Screen

1) select size
2) select flavor

Add Chips Screen

1) select type/flavor

Checkout Screen

1) Display order details and price

[1] - Checkout, create receipt file
[0] - Cancel Order - delete order

Sandwich Class

- size: enum Size
- bread: enum BreadType
- toppings: List<Topping>
- sauces: List<Sauce>
- toasted: boolean
- price

+Sandwich(size: Size, bread: Bread)
+ setSize(size: Size)
+ setBread(bread: Bread)
+ addTopping(topping: Topping)
+ addSauce(sauce: Sauce)
+ setToasted(toasted: boolean)
+ calculatePrice(): double
+ toString(): String

Drink Class

- size: enum Size
- flavor: enum DrinkType

+ Drink(size: Size, flavor: String)
+ setSize(size: Size)
+ setFlavor(flavor: String)
+ calculatePrice(): double
+ toString(): String

Chip Class

- type: enum ChipsType

+ Chip(type: String)
+ setType(type: String)
+ calculatePrice(): double
+ toString(): String

Order File Manager

Write new order.csv file using
order details

Receipt Folder

Hold receipt.csv files

Topping Class

- name: String
- category: Category
- price: double

+ Topping(name: String, category:
Category)
+ getName(): String
+ getCategory(): Category
+ getPrice(): price

Sauce Class

- name: String

+ Sauce(name: String)
+ getName(): String