**Assignment-6-Final\_Design Document**

1. **UML Diagram**

BevShopInterface

- processOrder(order: Order)

- totalMonthlySale(): boolean

- generateMonthlyReport(): String

-getMaxOrderForAlcohol(): int

-getNumOfAlcoholDrink(): int

-getMinAgeForAlcohol(): int

-isMaxFruit(numOFFruits): int

-getCurrentOrder():

OrderInterface

addNewBeverage(beverage: Beverage)

totalAmount(): double

generateOrderReport(): String

Order

orderNumber: int orderTime: String

orderDay: DAY

customer: Customer

beverages: List<Beverage>

addNewBeverage(beverage: Beverage) compareTo(Order other): int

toString(): String

getCustomer(): Customer

Customer

name: String

age: int

toString(): String

Customer(String, int)

Customer(Customer)

BevShop

- orders: List<Order>

- alcoholCount: int

processOrder(order: Order)

generateMonthlyReport(): String

toString(): String

Smoothie

numFruits: int

proteinAdded: boolean

calcPrice(): double

toString(): String

equals(Object obj) : boolean

Alcohol

isWeekend: Boolean

calcPrice(): double

toString(): String

Beverage

- name: String

- type: TYPE

- size: SIZE

- BASE\_PRICE: double

- SIZE\_PRICE: double

calcPrice(): double

toString(): String

equals(Object obj): boolean

Cofee

extraShot: Boolean

extraSyrup: Boolean

calcPrice(): double

toString(): String

equals(Object obj): boolean

enumeration

**TYPE**

COFFEE

SMOOTHIE

ALCOHOL

enumeration

**SIZE**

SMALL

MEDIUM

LARGE

enumeration

**DAY**

MONDAY

TUESDAY

THURSDAY

FRIDAY

SATURDAY

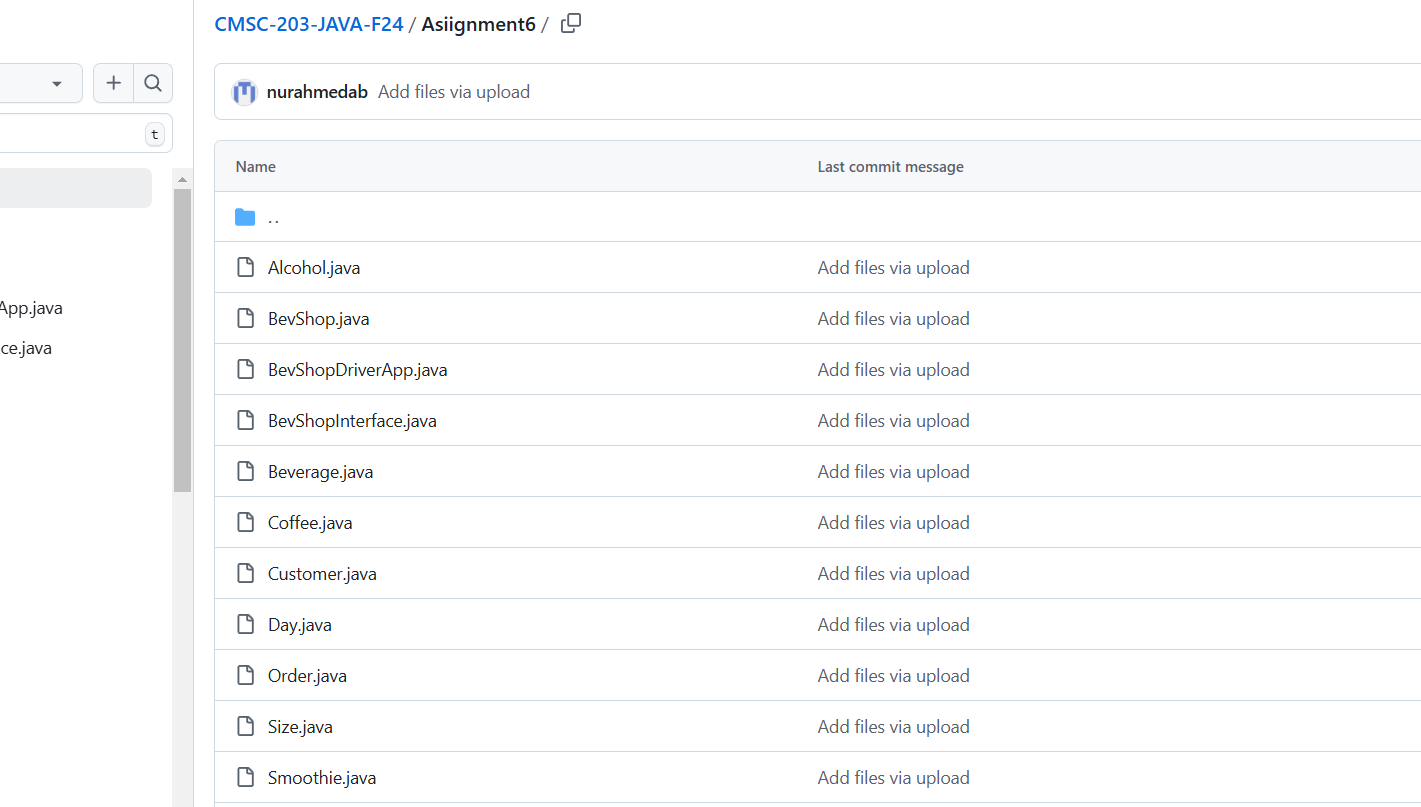
SUNDAY

1. **Lesson Learned**  
   Working on this project allowed me to learn a lot about object-oriented programming (OOP) concepts, particularly how to design and implement interfaces, handle data encapsulation, and manage collections of objects. I learned how to structure an application that interacts with user input and performs business logic like checking for valid times, managing orders, and enforcing rules such as age restrictions and the maximum number of beverages in an order. I also learned how to implement the selection sort algorithm to sort orders, as well as how to calculate and maintain order totals and shop sales.

One of the challenges I faced was properly managing state across the various methods in the BevShop class. For example, ensuring that the alcohol order count (alcoholOrders) accurately reflects the number of alcohol drinks in the current order and managing changes to customer information.

1. **ScreenshotsA screenshot of a computer

   Description automatically generated**

****

some junit test screenshotsA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated