**Team:**

Trey Pero

Mark Bambino

Sean Robb

Harvey Weyandt

Benjamin Wargo

**Class Diagram:**

Coffee Class – Manipulating coffee data for the menu/ordering

* Constructor – string name, double/Decimal price
* Properties: name and price

Topping Class - Manipulating topping data for the menu/ordering

* Constructor – string name, double/Decimal price
* Properties: name and price

Maybe a FileManager namespace???

CoffeeFileManager Class – To manage the coffee file, i.e. coffee.txt

* AddCoffee(string name, double/Decimal price) – Adds given coffee name and price to file
* DeleteCoffee(string name) – Deletes given coffee from file
* GetCoffeeList() - Returns a list of all current coffee objects

ToppingsFileManager Class – To manage the toppings file, i.e. toppings.txt

* AddTopping(string name, double/Decimal price) - Adds given topping name and price to file
* DeleteTopping(string name) - Deletes given topping from file
* GetToppingsList() - Returns a list of all current toppings objects

OrderHistoryFileManager Class – To manage the order history file, i.e. orderHistory.txt

* AddOrder(Order order) - Adds the order data to the orderHistoryFile
* GetOrderHistory(Date firstDate, Date lastDate) - Returns a list of order objects to the user for displaying

MenuUI Class – An observer to display menu options to the user and manage receiving/building orders and passing them off to the OrderHistoryFileManager when the user checks out

ManagerUI Class – A UI to let the manager interact with the various file managers

* Private constructor
* Static getManagerUI (string userName, string password – returns new instance of class if userName and password are valid, returns null if they're not

Order Class - Used to store order history data for report generation

* Properties: Date dateOfOrder, List<Coffee> coffees, List<Topping> toppings, string customerName, double/Decimal totalPrice

**Patterns Used:** Observer, Singleton, and Decorator?

**File Storage Design:**

JSON file parsing

**Report Generation Design:**

**Map Objects:**

Toppings file maps to Toppings object: Name, Price

Coffee file maps to Coffee object: Name, Price

**User Interface and Menu Options:**

Menu – A list of coffee choices and when you click on them, their available toppings appear in a list to choose from. An "out of the way" button will be on the menu panel to allow a manager to log in and make updates to the menu options. Total can be dynamically updated next to a "checkout" button, as the user makes choices, and when the user clicks checkout, a receipt could be displayed?

Manager Panel -

**Language of choice:** C#. Majority rules.