# 03. Cocktail Bar

A group of different colored drinks on a table

Description automatically generated

A person holding a phone and a sign

Description automatically generated

*You are tasked with creating a program that simulates a menu for a cocktail bar. Imagine a warm summer evening where guests are eager to enjoy refreshing cocktails. The menu should be able to store and manage various types of cocktails. Guests will be able to scan a QR code and view this virtual menu on their devices. You should create the necessary classes and methods to implement this functionality.*

**Preparation**

Download the skeleton provided in Judge. **Do not** change the **Program** class or its **namespace**.

Pay attention to **name the project CocktailBar**, all the **classes**, their **fields**, and **methods** the **same** way they are presented **in the following document.** It is also important to **keep the project structure as described.**

**Problem Description**

Your task is to create a **repository that stores cocktails** by creating the classes described below.

### Cocktail

You are given a class **Cocktail** with the following properties:

* **Name – string**
* **Price - decimal**
* **Volume – double**
* **Ingredients – List<string>**

The class **constructor** should receive **name, price, volume** and **ingredients**.

Override the **ToString()** method, to return a **string**, **separated on two lines**, in the following format:  
**"{Name}, Price: {Price:F2} BGN, Volume: {Volume} ml"  
"Ingredients: {ingredient1, ingredient2, … ,ingredientn}"**

**** We have implemented the **Ingredients** property for you!

### Menu

**Next**, you are given a class **Menu** that has **Cocktails** (a List that stores Cocktails). The **Menu** class should have the following **properties**:

* **Cocktails – List<Cocktail>**
* **BarCapacity - int** - **returns** the **maximum number of cocktails**, the bar is able to support, without compromising on quality.

The class **constructor** should receive **BarCapacity**, also it should initialize the **Cocktails** with a new instance of the collection.

Implement the following features:

* **Method AddCocktail(Cocktail cocktail)**
  + **Adds** an **entity** to the **collection** of Coctails, **if** the **BarCapacity** **allows it**.
  + Should **NOT accept** cocktail **names duplication**. If such exists, just **exit the operation**.
* **Method RemoveCocktail(string name)**
  + **Removes** a **cocktail by** a **given name,** if such **exists**, and **returns boolean**(**true** if it is removed / **false if not found**)
* **Method GetMostDiverse()**
  + **Returns** the **Cocktail** with the **largest variety of ingredients (count).**
* **Method Details(string cocktailName)**
  + R**eturns** detailed information about the **Cocktail** with the **given name, using the overridden ToString() method ot the specific cocktail**.
  + There will also be a cocktail with the given name added, before calling **Details** method**.**
* **Method GetAll()** – **returns the names of all cocktails** in the menu**, order alphabetically. Each name on a new line.** A **string** in the following **format**:
  + **"All Cocktails:  
    {Cocktail1.Name}  
    {Cocktail2.Name}  
    (…)**
  + **{Cocktailn.Name}"**

**Constraints**

* You will **always have drinks** **added before** receiving methods, **manipulating** the drinks in the **Menu**.
* There will always be **exactly one cocktail**, that will have **the greatest count of ingredients**.

**Examples**

This is an example of how the **Menu** class is **intended to be used**.

|  |
| --- |
| **Sample code usage** |
| //Initialize the repository (CocktailBar)Menu cocktailMenu = newMenu(5);  //Initialize entity (Cocktail)  Cocktail mojito = new Cocktail("Mojito", 8.50m, 200, "White Rum, Soda Water, Fresh Mint, Lime, Brown Sugar");  Cocktail fakeMojito = new Cocktail("Mojito", 18.50m, 180, "Red Rum, Soda Water, Fresh Mint, Lime, Brown Sugar");  Cocktail pinaColada = new Cocktail("Pina Colada", 7.00m, 150, " White Rum, Coconut Cream, Pineapple Juice");  Cocktail sexOnTheBeach = new Cocktail("Sex On The Beach", 11.00m, 200, "Vodka, Peach Schnapps, Orange Juice, Cranberry Juice, Glase Cherry");  Cocktail margarita = new Cocktail("Margarita", 10.50m, 150, " Tequila, Triple Sec, Lime Juice, Salt");  Cocktail dryMartini = new Cocktail("Dry Martini", 7.50m, 120, "Gin, Vermouth");  Cocktail longIsland = new Cocktail("Long Island", 13.00m, 300, " Vodka, Tequilla, White Rum, Cointreau, Gin, Lemon Juice, Cola");  //Adding coctails to the repositorycocktailMenu.AddCocktail(mojito);  //The first cocktails is added, 4 more positions available  //Name duplication is NOT allowed  cocktailMenu.AddCocktail(fakeMojito);  //The cocktail should NOT be added, 4 more positions available  cocktailMenu.AddCocktail(pinaColada);  cocktailMenu.AddCocktail(sexOnTheBeach);  cocktailMenu.AddCocktail(margarita);  cocktailMenu.AddCocktail(dryMartini);  //The capacity is full after adding dryMartini  //Try to add cocktail over the allowed capacity  cocktailMenu.AddCocktail(longIsland);  //The last cocktail should not be added, because there are no space for it//Removing cocktails from the repository  //Try to remove not existing cocktail, should return FalsecocktailMenu.RemoveCocktail("Long Island");  //Try to remove existing cocktail, should return True  cocktailMenu.RemoveCocktail("Pina Colada");  //Now there is one position availabe, for adding a cocktail, that should not be added previously  cocktailMenu.AddCocktail(longIsland);  //Finding the cocktail with the greatest variety of ingredientsConsole.WriteLine(cocktailMenu.GetMostDiverse());  //Long Island, Price: 13.00 BGN, Volume: 300 ml  //Ingredients: Vodka, Tequilla, White Rum, Cointreau, Gin, Lemon Juice, Cola  //Getting cocktail details  Console.WriteLine(cocktailMenu.Details("Mojito"));//Mojito, Price: 8.50 BGN, Volume: 200 ml  //Ingredients: White Rum, Soda Water, Fresh Mint, Lime, Brown Sugar  //Get a list of all cocktailsConsole.WriteLine(cocktailMenu.GetAll());  // All Cocktails:  //Dry Martini  //Long Island  //Margarita  //Mojito  //Sex On The Beach |

**Submission**

Zip all the files in the project folder except **bin** and **obj** folders.

A screenshot of a computer menu

Description automatically generated