PA5 UML

|  |
| --- |
| Class Name: Menu Driver |
| Fields/Properties:  -Menu/Driver Class/Flow of Control (MAIN)  userInput: int |
| Methods:  displayMenu(): int  menuSelection(int userInput): void  if (menuSelection == 1)  Listing.Add(listings array[])  If(menuSelection == 2)  Listing.ViewAll(listings[])  Renter.rentListing(listings[])  If(menuSelection == 3)  Reports.view(listings[]) |

|  |
| --- |
| Class Name: Listing |
| Fields/Properties:  Listing  - Listing ID: int  - Address: string  - Listing end date: string  - Listing price: long  - Owner email: string |
| Methods:  +Listing() //No arg  +Listing(with arguments)  +getListingID(): int  + getAddress(): string  + getEndDate(): string  + getPrice(): long  + getEmail(): string  + setListingID(int): void  + setAddress(string): void  + setEndDate(string): void  + setPrice(long): void  + setEmail(string): void  +populateListingArray(): void  +sortListings(array) void  +compareTo(): void  +toString(listings[]): void |

|  |
| --- |
| Class Name: Transaction |
| Fields/Properties:  -Renter Name  -Renter Email  -Date of transaction |
| Methods:  +getRenterName(): void  +getRenterEmail(): void  +getTransactionDate(): void  +setRenterName(string): void  +setRenterEmail(string): void  +setTransactionDate(string): void  +compareTo(listings[], listingCount): void  displayTransaction(listings[], listingCount, renter[]): void |

|  |
| --- |
| Class Name: Reporting |
| Fields/Properties:  -Reporting |
| Methods:  +sortByCustomer(listings[], listingCount): void  +sortByDate(listings[], listingCount): void  +sortByRevenue(listings[], listingCount): void |

Menu:

1. Add listing
   1. Obtain new listing info (List ID, address, price, etc.)
   2. Save listing to its transaction file.
2. Lease condo
   1. View condos available for lease
   2. Rent out condo
3. Run reports
   1. Individual customer rental report
      1. Take email and find previous rentals for that customer
   2. Historical customer rentals
      1. Sort rentals by customer
      2. Sort rentals by date
   3. Historical revenue report
      1. List rental revenue by month and by yearint
4. Exit

using System;  
using System.IO;  
  
namespace TaylorBurchPA5  
{  
    class MainClass  
    {  
        public static void Main(string[] args)  
        {  
  
            //EXTRAS:  
            //Autogenerate Listing ID  
  
  
            int menuSelection = 0;   
  
            menuSelection = displayMenu(); //Display main menu  
            while (menuSelection != 4)  
            {  
                if (menuSelection == 1)  
                {  
                    Listing.addListing(listingCount: 0); //Add a new listing with all information to the program and save to file.  
                }  
                else if (menuSelection == 2)  
                {  
                    Listing.viewListings(); //View all condos available for rent  
                }  
                else if (menuSelection == 3)  
                {  
                    Listing.individualCustomerReport();  
                    Listing.historicalCustomerReports();  
                    Listing.historicalRevenueReports();  
  
                }  
                else  
                {  
                    Console.WriteLine("Sorry, that input is invalid. Please try again.");  
                }  
  
                menuSelection = displayMenu();  
            }  
        }  
  
        public static int displayMenu()  
        {  
            Console.WriteLine("Welcome to Rent My Place!");  
            Console.WriteLine("Please select a menu option by inputing the corresponding number.");  
            Console.WriteLine("Input 1 to add a listing.");  
            Console.WriteLine("Input 2 to lease a condo.");  
            Console.WriteLine("Input 3 to run reports.");  
            Console.WriteLine("Input 4 to exit.");  
  
            int menuSelection = int.Parse(Console.ReadLine());  
            return menuSelection;  
        }  
    }  
}

namespace TaylorBurchPA5  
{  
    class Listing  
    {  
        private int listingID = 0;  
        private string address = "";  
        private string endDate = "";  
        private long price = 0;  
        private string email = "";  
        int listingCount = 0; //Count total number of listings  
  
        public void setListingID(int ID)  
        {  
            listingID = ID;  
        }  
  
        public void setAddress(string A)  
        {  
            address = A;  
        }  
  
        public void setEndDate(string date)  
        {  
            endDate = date;  
        }  
  
        public void setPrice(long P)  
        {  
            price = P;  
        }  
  
        public void setEmail(string E)  
        {  
            email = E;  
        }  
  
        public int getListingID()  
        {  
            return listingID;  
        }  
  
        public string getAddress()  
        {  
            return address;  
        }  
  
        public string getEndDate()  
        {  
            return endDate;  
        }  
  
        public long getPrice()  
        {  
            return price;  
        }  
  
        public string getEmail()  
        {  
            return email;  
        }  
  
        public static void addListing(int listingCount)  
        {  
            string userInput = "";  
            listingCount++;  
            Listing newListing = new Listing;  
            newListing.setListingID(listingCount);  
            Console.WriteLine("Please enter the prompted information to add a new listing!");  
  
            Console.WriteLine("Enter the address of the listing: ");  
            userInput = Console.ReadLine();  
            newListing.setAddress(userInput);  
  
            Console.WriteLine("Enter the end date of the listing: ");  
            userInput = Console.ReadLine();  
            newListing.setEndDate(userInput);  
  
            Console.WriteLine("Enter the price you would like to list at: ");  
            long input = long.Parse(Console.ReadLine());  
            newListing.setPrice(input);  
  
            Console.WriteLine("Please enter seller email: ");  
            userInput = Console.ReadLine();  
            newListing.setEmail(userInput);  
  
            //\*\*\*\*\*\*\*\*\*\*\*SAVE TO FILE  
        }  
  
        public static viewListings()  
        {  
              
        }  
  
    }  
  
}