TABLE OF CONTENTS

İçindekiler

LIST OF FIGURES

1. INTRODUCTION

The gas station management system is software that simplifies and facilitates the operation of fuel stations.

1.1 Purpose

Our problem is the complexity of gas stations, the heavy workload at gas stations and the lack of privacy of personal data.

In order to solve these problems, it is necessary to develop an easy-to-use system with the customer at the center, and to provide a data view for the administrator in the system to control and augment the data. The program we designed aims to provide these solutions.

The aim of this project is to simplify the operation at gas stations. Also, the purpose of this program is to design a useful system for administrators or customers.

1.2 Scope

Within the scope of the project, a system plan was created by observing and analyzing the deficiencies and malfunctions in the gas stations. Then, an algorithm was designed to solve the problems and a class diagram was designed to determine how the application would work. Finally, the algorithm was converted into code.

2. DESIGN

2.1 Approach

The program has features that can meet all the needs of everyone at a fuel station, for example, the manager, the employee, the customer. The use of the program is very understandable and simple. In addition, the program offers quick access to the information to be displayed.

2.2 System

The main screen of the program is as in the figure.


```
C:\Users\Ramazan\Desktop\project_asil (1)\project_asil\x64\Debug\project_asil.exe
Welcome To Your GasStation
1. Administrator
2. Worker
3. Customer
4. Exit
Your choice:
```
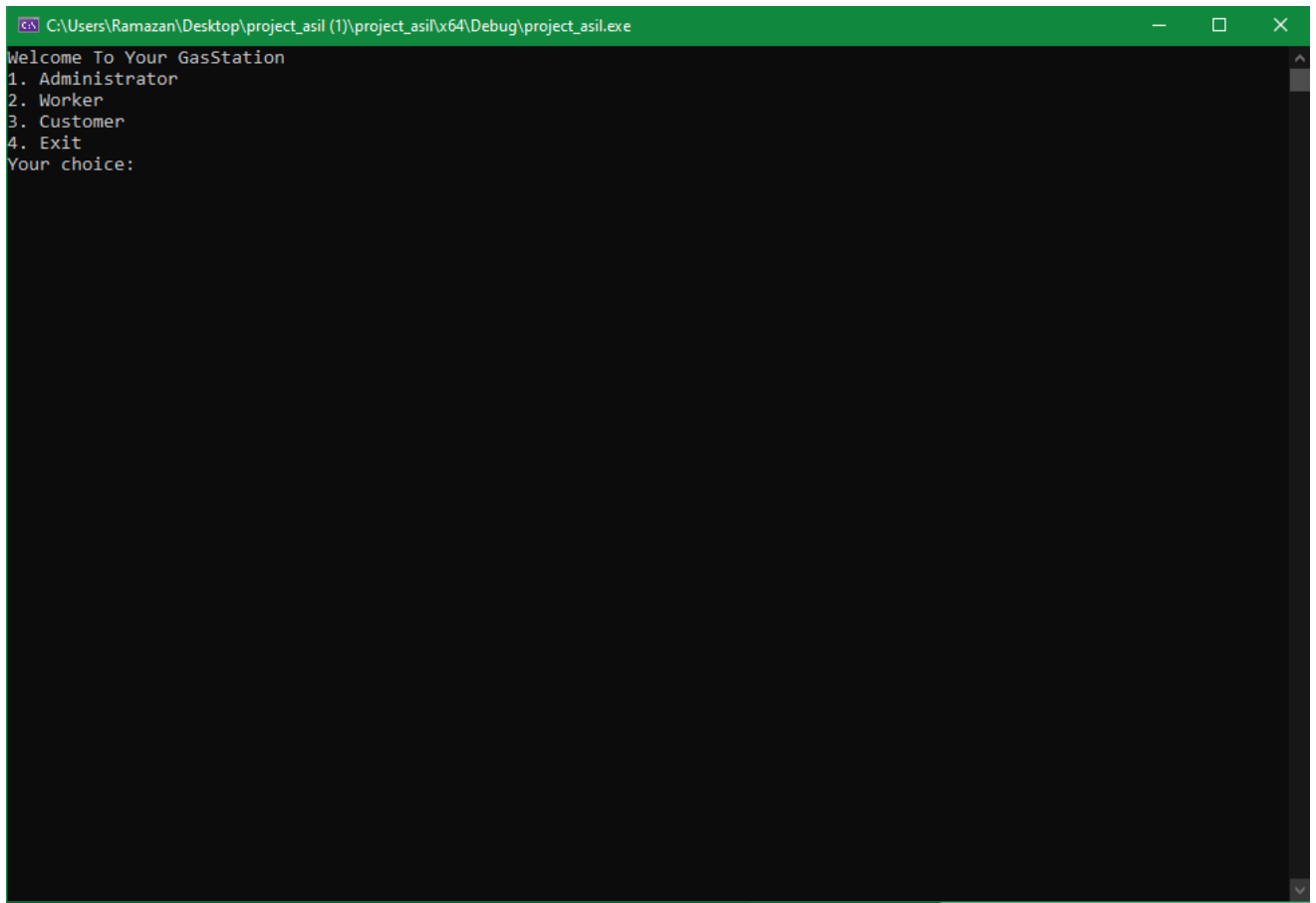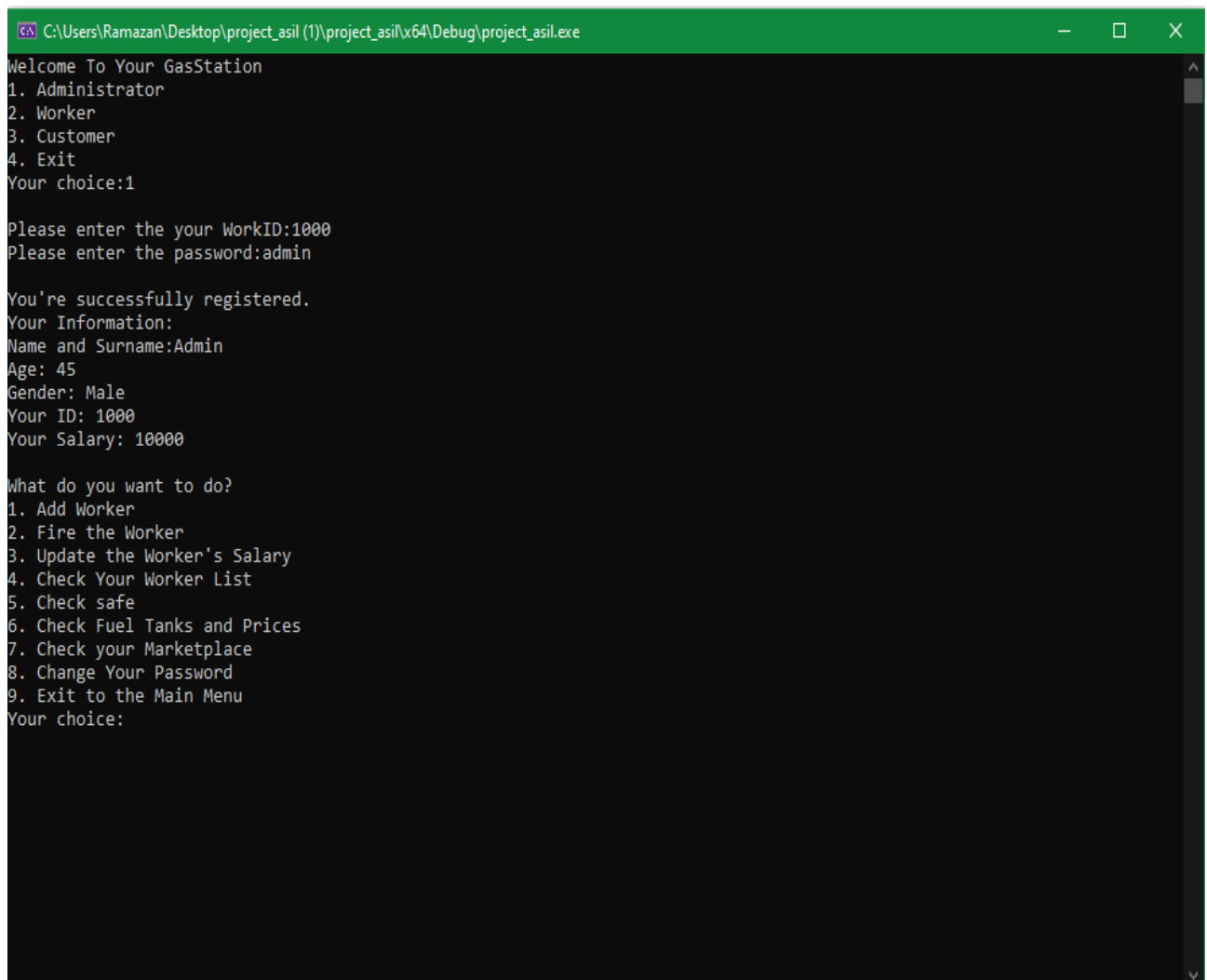
Figure 1. Main Page

We run the program by entering the process number of our job description in the selection section.

For example, we can continue our program by choosing admin first. We enter the transaction code into the system and press enter. When we enter the admin for the first time, we verify the admin username and password for security, and if there is verification, we reach the admin menu.


```
C:\Users\Ramazan\Desktop\project_asil (1)\project_asil\x64\Debug\project_asil.exe                    —    □    X
Welcome To Your GasStation
1. Administrator
2. Worker
3. Customer
4. Exit
Your choice:1

Please enter the your WorkID:1000
Please enter the password:admin

You're successfully registered.
Your Information:
Name and Surname:Admin
Age: 45
Gender: Male
Your ID: 1000
Your Salary: 10000

What do you want to do?
1. Add Worker
2. Fire the Worker
3. Update the Worker's Salary
4. Check Your Worker List
5. Check safe
6. Check Fuel Tanks and Prices
7. Check your Marketplace
8. Change Your Password
9. Exit to the Main Menu
Your choice:
```
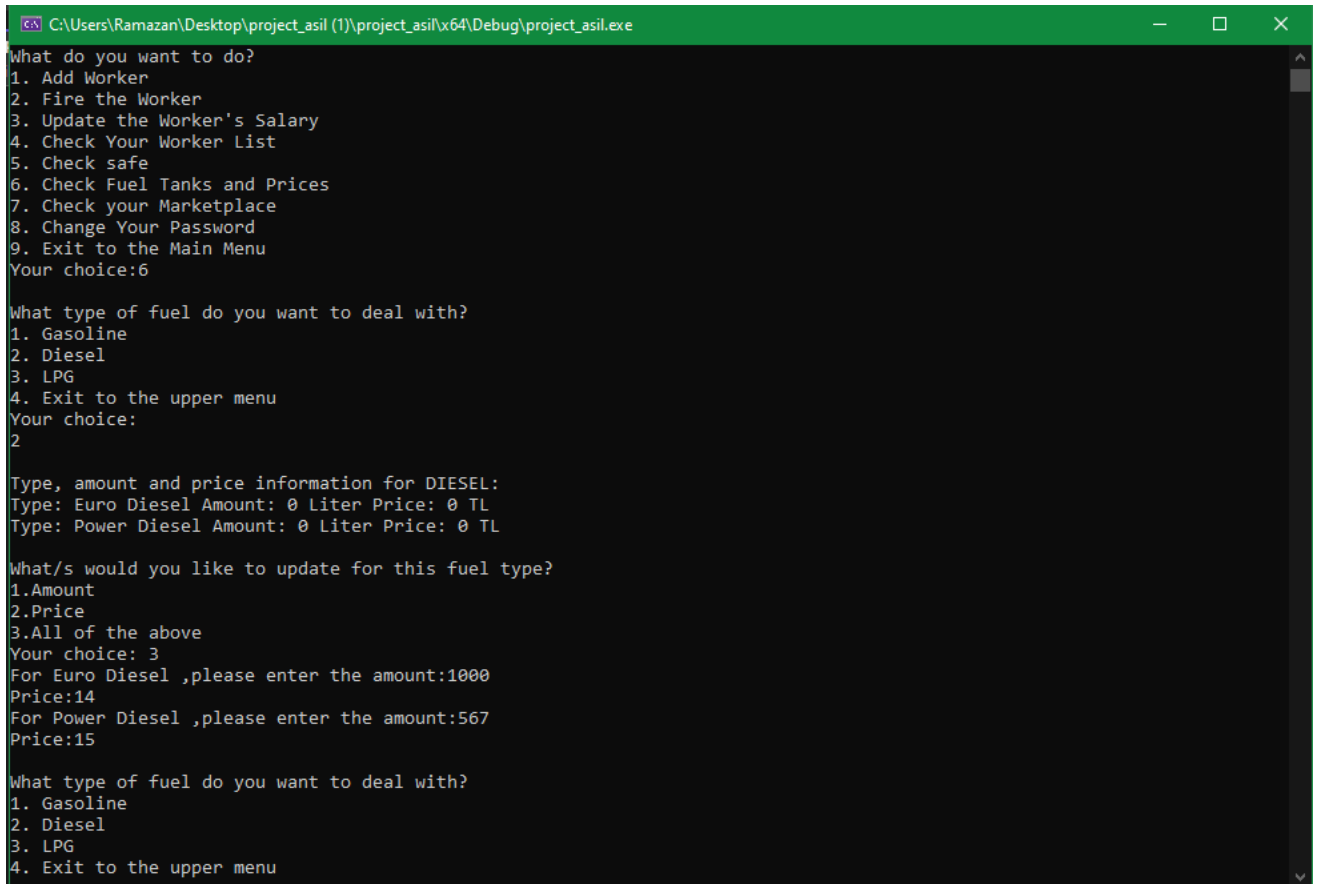
Figure 2. administrator menu

In the admin menu, there are options to add worker, fire rhe worker, update the salary of the workers, check the worker list, check the safe, check the fuel amount and fuel prices in the tank, check the market at the station and update the password for the admin. Checking the fuel amount and fuel prices in the tank and checking the market in the station has sub-menus within itself.

First, we can try "Check Fuel Tanks and Prices" by pressing 6.

```
C:\Users\Ramazan\Desktop\project_asil (1)\project_asil\x64\Debug\project_asil.exe                    —    □    ×
What do you want to do?
1. Add Worker
2. Fire the Worker
3. Update the Worker's Salary
4. Check Your Worker List
5. Check safe
6. Check Fuel Tanks and Prices
7. Check your Marketplace
8. Change Your Password
9. Exit to the Main Menu
Your choice:6

What type of fuel do you want to deal with?
1. Gasoline
2. Diesel
3. LPG
4. Exit to the upper menu
Your choice:
2

Type, amount and price information for DIESEL:
Type: Euro Diesel Amount: 0 Liter Price: 0 TL
Type: Power Diesel Amount: 0 Liter Price: 0 TL

What/s would you like to update for this fuel type?
1.Amount
2.Price
3.All of the above
Your choice: 3
For Euro Diesel ,please enter the amount:1000
Price:14
For Power Diesel ,please enter the amount:567
Price:15

What type of fuel do you want to deal with?
1. Gasoline
2. Diesel
3. LPG
4. Exit to the upper menu
```
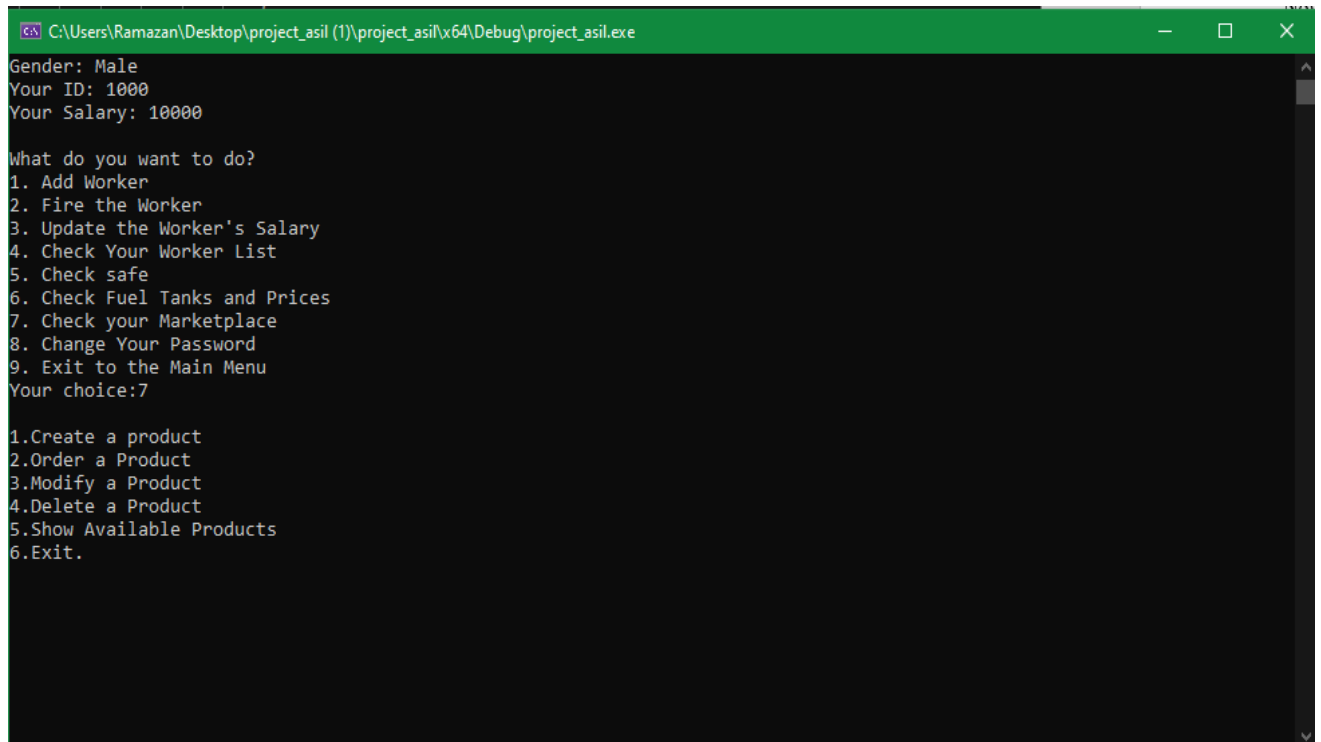
Figure 3. administrator menu(check fuel tanks and prices)

When we choose to check fuel tanks and prices, first we see the fuel types, then the admin chooses which type of fuel he wants to update. After the selection, we first see the current fuel amount and price in the tank. If there are different types of fuel, we can see each of them. In the new menu that comes up, the admin chooses to update the quantity, price or both and enters the new data.

We can try Check your Marketplace by typing 7 from the admin menu. In this section, we encounter the own menu of the market section..
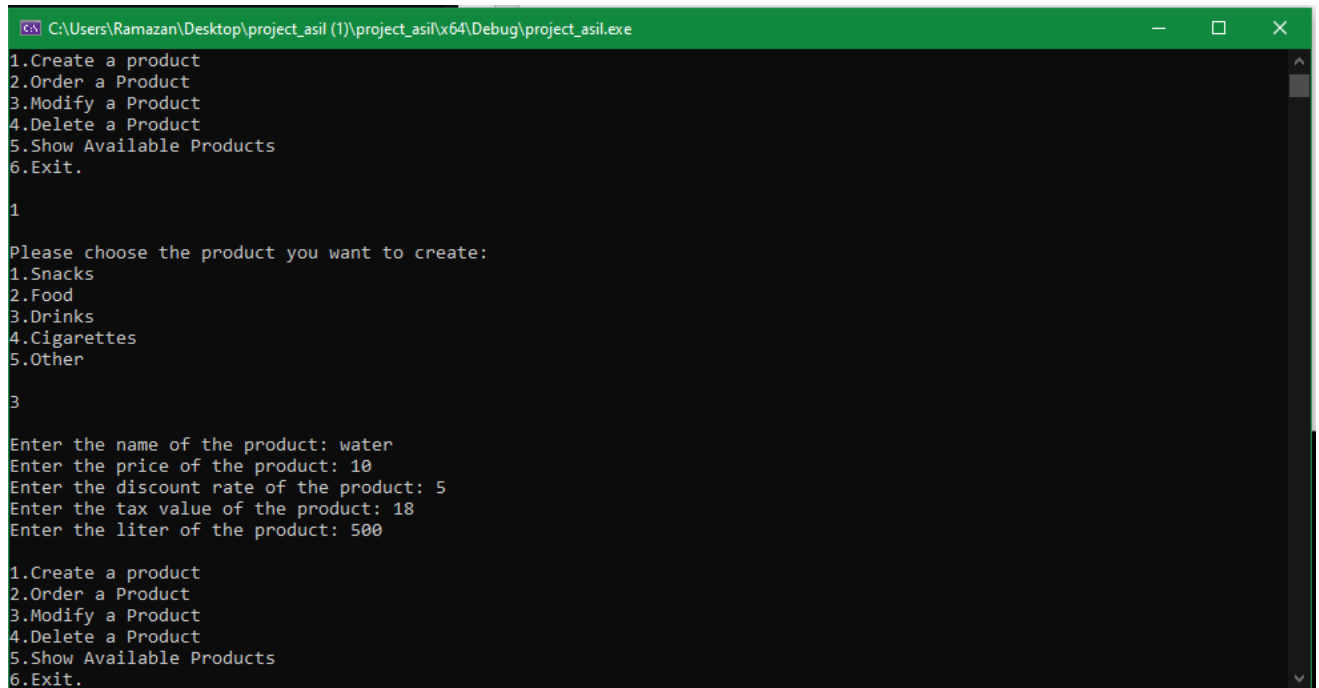
```
C:\Users\Ramazan\Desktop\project_asil (1)\project_asil\x64\Debug\project_asil.exe                    —    □    ×

Gender: Male
Your ID: 1000
Your Salary: 10000

What do you want to do?
1. Add Worker
2. Fire the Worker
3. Update the Worker's Salary
4. Check Your Worker List
5. Check safe
6. Check Fuel Tanks and Prices
7. Check your Marketplace
8. Change Your Password
9. Exit to the Main Menu
Your choice:7

1.Create a product
2.Order a Product
3.Modify a Product
4.Delete a Product
5.Show Available Products
6.Exit.
```

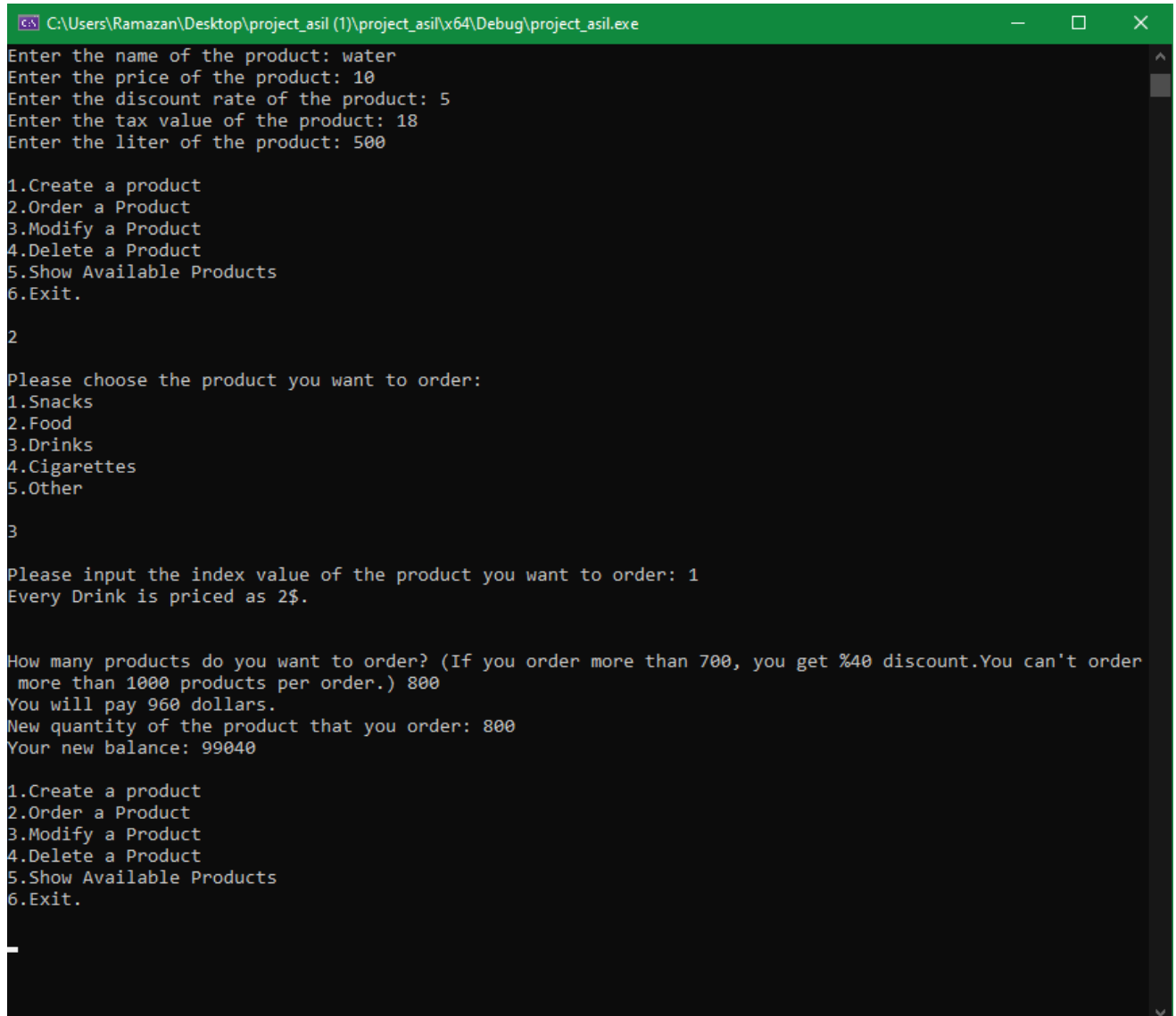Figure 4. administrator menu(Check your marketplace)

Figure 5. administrator menu(check your marketplace-create product)

When we enter 1 in the market menu section, we first see the create a product section. The system adds the product to the stock after receiving the product information from the admin.

In the "order a product" section, which is the second option of the market section, we can increase the stock number of a product that has been added to the stock before.

```
C:\Users\Ramazan\Desktop\project_asil (1)\project_asil\x64\Debug\project_asil.exe                    —    □    ×

Enter the name of the product: water
Enter the price of the product: 10
Enter the discount rate of the product: 5
Enter the tax value of the product: 18
Enter the liter of the product: 500

1.Create a product
2.Order a Product
3.Modify a Product
4.Delete a Product
5.Show Available Products
6.Exit.

2

Please choose the product you want to order:
1.Snacks
2.Food
3.Drinks
4.Cigarettes
5.Other

3

Please input the index value of the product you want to order: 1
Every Drink is priced as 2$.


How many products do you want to order? (If you order more than 700, you get %40 discount.You can't order
 more than 1000 products per order.) 800
You will pay 960 dollars.
New quantity of the product that you order: 800
Your new balance: 99040

1.Create a product
2.Order a Product
3.Modify a Product
4.Delete a Product
5.Show Available Products
6.Exit.

_
```
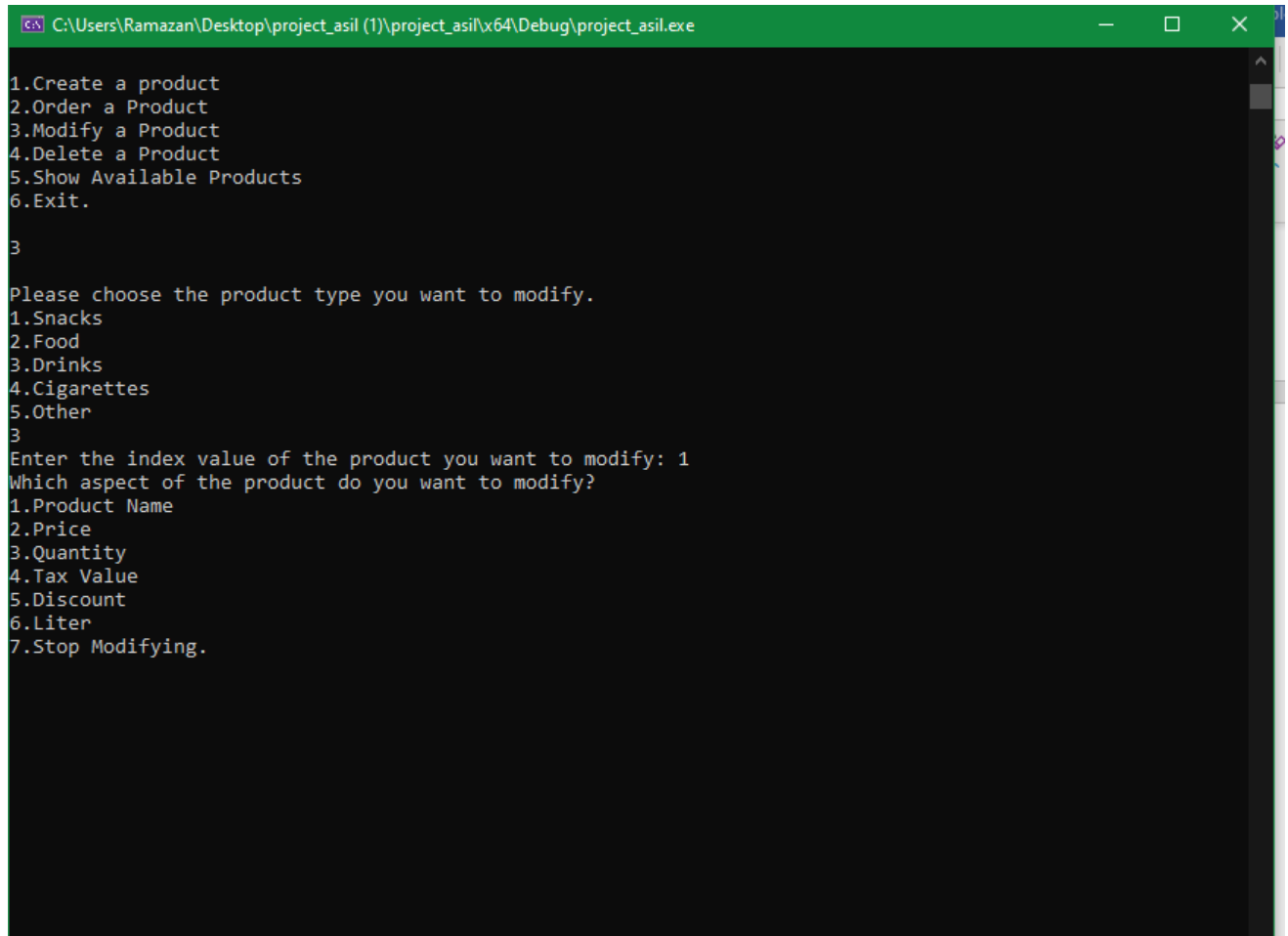
Figure 6. administrator menu(check your marketplace-order a product)

In the "modify a product" section, which is the 3rd option of the market section, we can update the information of a product that has been added to stock before.

Figure 7. administrator menu(check your marketplace-modify product)

Figure 8. administrator menu(check your marketplace-delete a product and show available products)

In the 4th option of the market section, "delete a product" is to delete a product that is in stock, in the "show available products" section of the 5th option, the products in stock are checked.

When we log into the main menu of our program as a worker, we cannot access any information if we are not first hired by the admin.

Figure 9. worker menu

Figure 10. worker menu

When we log into the system as a hired worker, we can see our personal information in the system.

Figure 11. customer menu

Finally, we can log in to the system as a customer. When we make our selection, we see a submenu of the customer.

Figure 12. customer menu(market)

When the market option is entered in the customer's submenu, we see the "display available products" option where the products that the customer can buy are listed, the "buy product" option to buy the product, and the "display total price" option, where we can see the total expenditure in the shopping.

Figure 13. customer menu(refueling)

When you go to the "refueling" option from the customer menu, we first ask the customer for a license plate information to understand whether he is a registered customer, in this way, we enable customer's to spend the points customer has earned from customer's previous purchases. If fuel is in stock, customer continues his/her shopping and earns points according to the amount of shopping customer makes.

Figure 14. customer menu(refueling)

If the amount of fuel in the tank is insufficient, the system shows the maximum amount of fuel that can be taken in the tank and informs the customer.
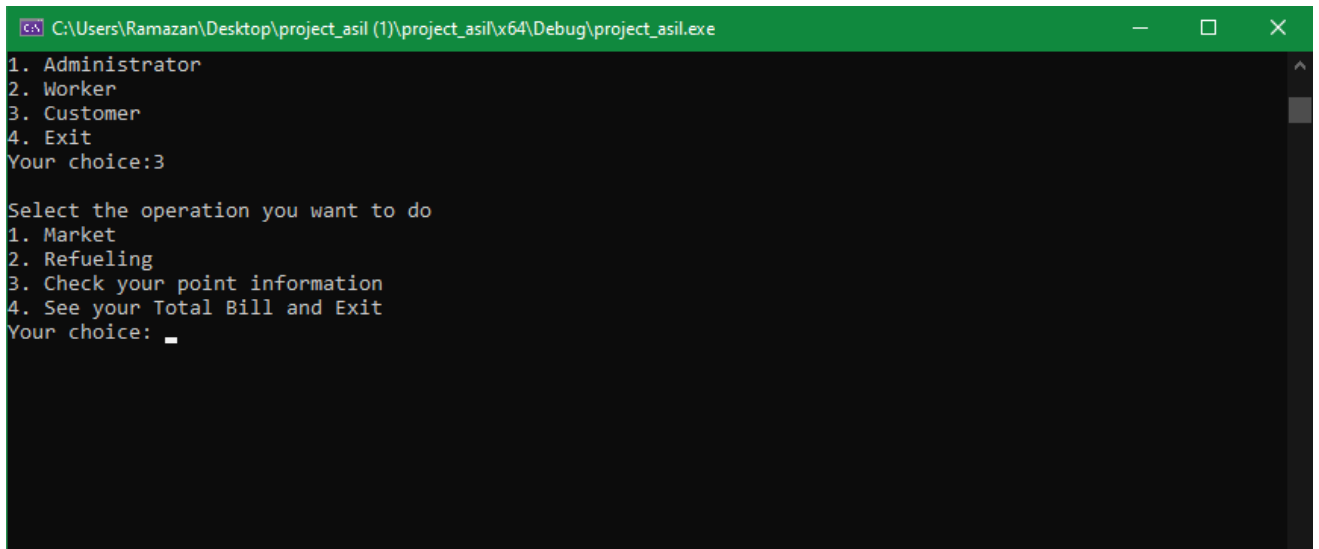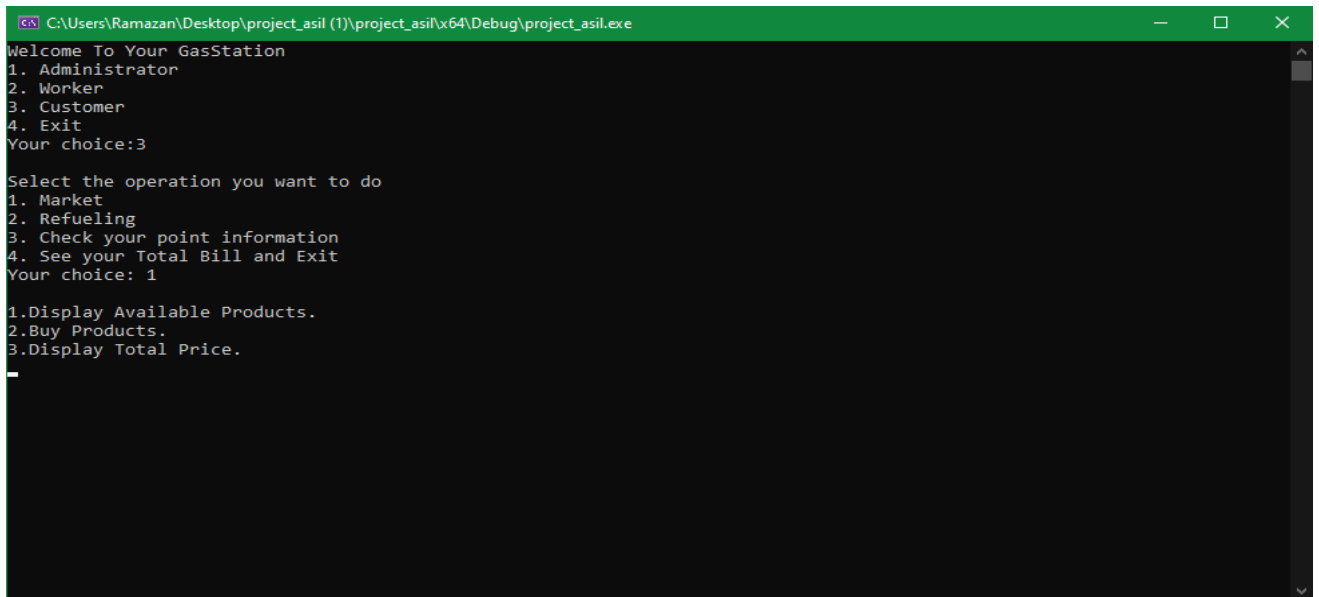
```
Seç C:\Users\Ramazan\Desktop\project_asil (1)\project_asil\x64\Debug\project_asil.exe

1. Administrator
2. Worker
3. Customer
4. Exit
Your choice:3

Select the operation you want to do
1. Market
2. Refueling
3. Check your point information
4. See your Total Bill and Exit
Your choice: 1

1.Display Available Products.
2.Buy Products.
3.Display Total Price.
2
What do you want to buy?
1.Snacks
2.Foods
3.Drinks
4.Cigarettes
5.Others
3
Please enter the index value of the drink that you want to buy.
1
How many waters do you want to buy?10

1.Display Available Products.
2.Buy Products.
3.Display Total Price.
3
Total price you need to pay: 95$
Thanks for shopping from us.
Select the operation you want to do
1. Market
2. Refueling
3. Check your point information
4. See your Total Bill and Exit
Your choice: 2
Please enter the your plate information (Ex: 06YY68):51hy717
Select your vehicle's fuel type
1. Gasoline
2. Diesel
3. LPG
4. Exit
Your choice: 3
How much fuel do you want to buy? :500
As a result of your request you made a purchase of 500 TL from LPG fuel type.
And you earned 25 points in this way
Select the operation you want to do
1. Market
2. Refueling
3. Check your point information
4. See your Total Bill and Exit
Your choice: 4
Would you like to use the 25 points you have accumulated before (1point==1TL) (y/n)?
y
Your 25 points have been successfully converted to 25 TL.
Your plate: 51hy717, Your total spend: 570 TL,25 point earned and 25 point spend
You are redirected to the main menu.
```

Figure 15. customer menu(total bill)

Finally, in the "check your point informaiton" section of the customer menu, customers will be able to see the points they have earned. In the "see yout total bill and exit" option, which is the 4th option, customers will be able to see their total expenditures in the market and refueling sections.

2.3 System Design

While designing the system, our priority was customer class in terms of the operability of our system. Thus, thanks to the admin part, we followed the stability of the objects we created.

The "admin" class performs all kinds of additions and subtractions in our system (for example: adding fuel, products to the market, adding or removing money from the cash register).

The gas station class has made our job easier by parenting other (gasoline, diesel, lpg) classes.

In our customer class, we did not allow access beyond our control by defining the information we received from the customer specially.

Our Market class includes functions such as product definition, product ordering, and product listing by parenting other classes (snacks, food, drinks etc.) just like the GasStation class.

3. CONCLUSION

In this project, we did a study to eliminate the problems we observed in gas stations. Our project was designed to meet the needs of every fuel office that we may encounter in daily life. We tried to exemplify all the topics (functions, class structures, constructors, destructors, etc.) we saw in the content of the Ceng241 course this semester, in accordance with our project. .We created our project with C++ object-oriented software language using Dev-C++ and Visual Studio. Finally, our project is suitable for further development and customization upon request.

Person

-nameAndSurname: string
-age: int

+Person(): default const
+Person(string,string,age): overload const
+getNameAndSurname(): return type
+getAge(): return type

Employee

-salary: double
-workID : int

+Employee(): default const
+Employee(double,int):overload const
+updateSalary(double): virtual void
+getSalary(): double
+getworkID(): int
+displayInformation() : virtual void

FuelSupplier

+FuelSupplier():const.
+FuelSupplier(string, int, bool , int ,double):overload const.

Administrator

-password: string
-safe: double

+Administrator(): const
+updateSafe(double) : void
+setPassword(string): void
+getPassword(): return type
+getSafe(): return type
+depositAndWithdraw() : void

Customer

-license_plate: string
-total: float
-point: float

-Customer(): const.
-Customer(string,float,float):overload const.
+updateTotal(float):void
+updatePoint(float):void
+getPoint(): float

GasStation

-tank : float
-price : float

+GasStation(): default const
+GasStation(float,float): overload const.
+updateTank(float) : void
+setPrice(float) : void
+getTank() : return type
+getPrice() : return type
+displayInfo() : virtual void

Gasoline

-type: string

+Gasoline():const.
+Gasoline(float,float): overloaded const.
+displayInfo() : virtual void
+setType(string) : void
+getType() : return type

Diesel

-type : string

+Diesel() : const.
+Diesel(float, float, striing) : overload const.
+displayInfo() : virtual void
+getType() : string

LPG

+LPG();const.
+LPG(float, float):overloaded const.

Marketplace

#product_name : string
#price : float
#quantity : int
#discount : int
#tax_value : float

+Marketplace(): default const
+Marketplace(string,float,int,float,int):overloaded const.
+~Marketplace():default dest
+setName(string):void
+getName():return type
+setPrice(float):void
+getPrice():return type
+setQuantity(int):void
+getQuantity():return type
+setDiscount():void
+getDiscount():return type
+setTax():void
+getTax():return type

Snacks

+Snacks():default const
+Snacks(string,float,int,float,int):overloaded const
+~Snacks():default dest.
+displaySnack():void
+orderSnack():float

Food

#prepare_time1: int
#prepare_time2: int

+Food():default const.
+Food(string,float,int,int,float,int,int):overloaded const.
+~Food():default dest.
+set_Preptime1(int):void
+get_Preptime1():return type
+set_Preptime2(int):void
+get_Preptime2():return type
+displayFood():void
+orderSnack():float

Drinks

#liter:int

+Drinks():default const.
+Drinks(string,float,int,int,float,int):overloaded const.
+~Drinks():default dest
+setDrink_liter(int):void
+getDrink_liter():return type
+displayDrink():void
+orderDrink():float

Cigarettes

+Cigarettes():default const.
+Cigarettes(string,float,int,int,float):overloaded const.
+~Cigarettes():default dest
+displayCigarette():void
+orderCigarette():float

Other

+Other():default const.
+Other(string,float,int,int,float):overloaded const.
+~Other():default dest
+displayOther():void
+orderOther():void