ÇANKAYA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARMENT

CENG 200

SUMMER INTERNSHIP

REPORT

Yazgı KAVUKÇU

ID: 201811041

Performed at

Ministry of Transport and Infrastructure of the Republic of Turkey

17.06.2021 – 14.07.2021

Contents

Abstract

I performed my CENG200 internship at the Ministry of Transport and Infrastructure of the Republic of Turkey. My department was Strategy Development. I worked at İZLEM which is a web application program that has six modules. Specifically, I worked cash advance module. Coded in C# because of that I made many console applications in C# latest version and I used visual studio. I watched records of some meetings about starting the İZLEM project and analyzed the requirements. After that, I met the users and checked the system issues, and reported them.

Introduction

The main focus of the Strategy development department is money management of all Ministry Transport and Infrastructure of the Republic of Turkey. I was working on a web application project which is using for monthly or yearly money management.

I was involved in the project cash advanced module. In my report on the part of work done, I mentioned more details.

Company Information

Ministry Transport and Infrastructure of the Republic of Turkey is based on transportation. Company responsible Not only the highways but also airways and maritime lines too. They generally made roads but some times they made mega projects like airports or harbours.

Davut Cihan who works in Strategy development department is one of my supervisors, and Tutku Özarslan is my other supervisor, who graduated at Hacettepe University in the department of Computer Engineering. He is conductor of İZLEM project. His e-mail is davut.cihan@uab.gov.tr There were sixty interns in Ministry Transport and Infrastructure of the Republic of Turkey. Strategy development department had six interns. I worked with Metehan who is studying at K.T.Ü. in the department of Computer Engineering. I and Metehan were only the engineering students in the department. So when people received errors they asked us to if we can fix it.

Work Done

The purpose of the İZLEM project is prosecute the money flow. Institutions make a cash request every month in a period, then record it into the system, and the authorized person examines the request, if the request is okay then approves it. Next month authorized person compares if the spending money and the demanded money match. So the government budget can follows-up. Because İZLEM is a web application, I need to research C#, msSQL, javaScript, HTML, and CSS. Also, I read the book; Clean code.

The first week I watched some records of commissioning the İZLEM project, my supervisor and the firm which made the İZLEM application, was discussing the details, and I analyzed them. My supervisor and I discussed meetings, then he wanted me to do a diagram for revision transactions.

Figure of revision diagram

In this diagram, there are days of months. For the revision can it be possible if a week has a start date and end date they are included in a month. At least one day is needed to include the week for we can say that start week or end week. If we can say that it is start or end then we should check are there three or more days include in this week, if it is then revision can possible other way revision is not possible.

At the end of the first week, my supervisor and I visited the admin users and talked about İZLEM's absence or issues. Lastly, asked them satisfaction level, it was quite well.

The second week my other supervisor Tutku who works in the data processing department wanted me to create some console applications for me to consolidate C#. While I was doing these console applications, she ensured that my code has pure and understandable. At first, I made some control mistakes or control lacks like take input as strings and turn it into an integer, then checks if it is an integer. These kinds of mistakes or lacks made the program crash, so I gave more attention to do not miss out on any controls. In my codes, the long part was the control or the menu parts. I saw that the most important thing is the program works correctly, and the second important thing is writing a sustainable code because after I wrote the code other people may improve or use it for a long time and they made the updates. So after read the clean code I debugged my mistakes in my older console applications.

In the third week, I proceed to console applications but this time attach particular importance to the algorithmic methods. For instance, I made a console application which is a memory card game. In this application, they wanted to be a basic card game. In this game, there will be sixteen numbers 1 to 16, and every card has a character A to H, which makes eight characters so, the characters are duplicate and randomly matches a number. The player can see the numbers 1 to 16 and, every turn can choose two of them. Chosen numbers tuns and their characters show up if the characters are the same, then they will be open till the end. When the player opens all the cards then the number of turns shows, and the game ends. I did that game properly to the "clean code" book.

Last week I authenticated as an admin in the İZLEM's for student version and for controlling the database need to research entity framework and msSQL queries. Entity framework is a code-first approach, so it is more professional than changing the data manually. I made basic queries like intersecting two tables with my own data. This was not challenging because I already gave the course for the database and wanted queries that were not complicated.

Conclusion

Before the İZLEM, cash advanced request has demanded with a petition a lot of paperwork for monthly. And tracking money flow was very hard. Now everything is systematically tracking approving and demanding.

But I think İZLEM still needs improvement. For example, once while demanding cash advanced one zero added extra mistakenly, so the money is about 90.000 Turkish liras returned to the public purse. In my opinion, one SQL query can add to compare before the demanded amount of money and the new demanded amount if their difference is more or less than half of the average or duplicate the average a warning can add for the user to be sure about the demanded amount. I have talked about that idea and my supervisor confirmed that can make it in İZLEM2 which is more comprehensive than İZLEM.

Appendix[0]

List of Source Codes

Calculator

```
using System;

namespace hesapMachine

{  class Program

   { public double num_1 { get; set; }

      public double num_2 { get; set; }

      //Default Constructor

      public Program(double num_1, double num_2)

      {  this.num_1 = num_1;

         this.num_2 = num_2; }

      //sumation method
```

```
public string sum() => Convert.ToString(num_1 + num_2);
//extraction method
public string extract() => Convert.ToString(num_1 - num_2);
//multiply method
public string mult() => Convert.ToString(num_1 * num_2);
//dived method
public string div() => Convert.ToString(num_1 / num_2);
//mod method
public string mod() => Convert.ToString(num_1 % num_2);
//printing menu method
public static void printMenu()
{System.Console.WriteLine("Enter 1 for +\nEnter 2 for -\nEnter 3 for *\nEnter 4 for /\nEnter 5 for %\nEnter 0 for exit");}
//control method -> checks the value is double
public static double take_double()
{double num;
   //checks inpot is a deacimal or not
   while(!double.TryParse(Console.ReadLine(), out num))
   {System.Console.Write("Wrong input!! Enter a double number : ");}
   return num;
}
static void Main(string[] args)
{  Program p;
   double num_1, num_2;
   int num1, num2;
   string my_operator ;
   //endles loop
   while(true)
   {printMenu();
      //takes menu option input
      my_operator = Console.ReadLine();
      switch(my_operator)
      {case "1":
            System.Console.Write("Enter first double number : ");
```

```
    num_1 = take_double();
    System.Console.Write("Enter second double number : ");
    num_2 = take_double();
    p = new Program(num_1, num_2);
    System.Console.WriteLine(num_1 + " + " + num_2 + " = " + p.sum());
break;

case "2":
    System.Console.Write("Enter first double number : ");
    num_1 = take_double();
    System.Console.Write("Enter second double number : ");
    num_2 = take_double();
    p = new Program(num_1, num_2);
    System.Console.WriteLine(num_1 + " - " + num_2 + " = " + p.extract());
break;
case "3":
    System.Console.Write("Enter first double number : ");
    num_1 = take_double();
    System.Console.Write("Enter second double number : ");
    num_2 = take_double();
    p = new Program(num_1, num_2);
    System.Console.WriteLine(num_1 + " * " + num_2 + " = " + p.mult());
break;
case "4":
    System.Console.Write("Enter first double number : ");
    num_1 = take_double();
    System.Console.Write("Enter second double number : ");
    //endless loop for checks the second number not equal to zero
    while(true)
    {num_2 = take_double();
      if(num_2 == 0)
        System.Console.WriteLine("Second number can not equal to 0");
      else    { break; }}
```

```
   p = new Program(num_1, num_2);
   System.Console.WriteLine(num_1 + " / " + num_2 + " = " + p.div());
break;
case "5":
   while(true)
   {  System.Console.Write("Enter first integer number : ");
      //checks the input is an integer
      while(!int.TryParse(Console.ReadLine(), out num1))
      {System.Console.Write("Wrong input!! Enter an integer : ");}
      //checks the input less than zero
      if(num1 < 0)
         System.Console.WriteLine("Numbers can not less than 0!");
      else    { break; }}
   while(true)
   {System.Console.Write("Enter second integer number : ");
      //checks the input is an integer
      while(!int.TryParse(Console.ReadLine(), out num2))
      {System.Console.Write("Wrong input!! Enter an integer : ");}
      //checks the input less than or equal to zero
      if(num2 <= 0)
         System.Console.WriteLine("Numbers can not less than or equal to 0!");
      else    { break; }};
   p = new Program(num1, num2);
   System.Console.WriteLine(num1 + " % " + num2 + " = " + p.mod());
break;
case "0":
   //if input equal to zero than exit
   Environment.Exit(0);
break;
default:
   //checks input value different from wanted value
   System.Console.WriteLine("Invalid menu option!");
break; }}}}}
```

I wrote this calculator for learning and improving C#. This was my first console application during internship.

Memory Card Game

```
using System;
namespace MemoryCardGame
{class OutOFBounds : Exception
   {public OutOFBounds(string Message) : base(Message)
      { }}
   class AlreadyOpen : Exception
   {public AlreadyOpen(string Message) : base(Message)
      { }}
   class AlreadyChoosen : Exception
   {public AlreadyChoosen(string Message) : base(Message)
      { }}
   class Program
   {
      public char[] arrCards = new char[16];
      public bool[] arrIs_Open = new bool[16];
      public char[] arrChar  = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H' };
      public int firstCard;
      public static Random rand;

      public Program()
      {firstCard = 0;
         rand = new Random();
         randomFill();}
      //randomly fills cards array
      public void randomFill()
```

```
{arrCards[0] = arrChar[rand.Next(8)];
   for (int i = 1; i < 16; i++)
   {again:
      int tmp = rand.Next(8);
      int count = 0;
      for (int j = 0; j < i; j++)
      {if (arrCards[j].Equals(arrChar[tmp]))
            count++;
         if (count == 2)
            goto again; }
      arrCards[i] = arrChar[tmp];  }
}
//prints current cards
public void findCards(int firstCard, int secondCard)
{for (int i = 0; i < 9; i++)
   { if(i % 4 == 0 && i != 0)
         Console.WriteLine("| ");
      if (arrIs_Open[i])
         Console.Write("| " + arrCards[i] + " ");
      else if (firstCard == i + 1 || secondCard == i + 1)
         Console.Write("| " + arrCards[i] + " ");
      else
      {Console.Write("| " + (i + 1) + " ");}}
   for (int i = 9; i < 16; i++)
   {if (i % 4 == 0)
         Console.WriteLine("|");
      if (arrIs_Open[i])
         Console.Write("| " + arrCards[i] + " ");
      else if (firstCard == i + 1 || secondCard == i + 1)
         Console.Write("| " + arrCards[i] + " ");
      else
      {Console.Write("| " + (i + 1)); }
   }
```

```
    Console.WriteLine("|");
    if (arrCards[firstCard - 1].Equals(arrCards[secondCard - 1]))
    {arrIs_Open[firstCard - 1] = true;
        arrIs_Open[secondCard - 1] = true; }
}
//prints cards situation
public void printCards()
{for(int i = 0; i < 9; i++)
    {if(i % 4 == 0 && i != 0)
            Console.WriteLine("| ");
        if(arrIs_Open[i])
            Console.Write("| " + arrCards[i] + " ");
        else
        {Console.Write("| " + (i + 1) + " ");}
    }
    for (int i = 9; i < 16; i++)
    {if (i % 4 == 0 )
            Console.WriteLine("|");
        if (arrIs_Open[i])
            Console.Write("| " + arrCards[i] + " ");
        else
        {Console.Write("|" + (i + 1) + " ");}
    }
    Console.WriteLine("|");
}
//checks input is a integer or not
public static int take_int()
{int num;
    while (!int.TryParse(Console.ReadLine(), out num))
    {System.Console.Write("Wrong input!! Enter an integer number : ");}
    return num;
}
public int checkInput()
```

```
{int tmp;
    bool success1 = false;
    bool success2 = false;
    bool success3 = false;

    while (true)
    {jump:
        tmp = take_int();
        try
        {if (tmp <= 0 || tmp > 16)
                throw new OutOFBounds("Input must between 1 to 16 !");
            success1 = true;
        }
        catch (OutOFBounds outOfBounds)
        {Console.WriteLine(outOfBounds.Message);
            Console.Write("Enter a proper integer : ");
            goto jump;
        }
        try {
            if (arrIs_Open[tmp - 1])
                throw new AlreadyOpen("This card is already open!");
            success2 = true;
        }
        catch (AlreadyOpen alreadyOpen)
        {Console.WriteLine(alreadyOpen.Message); }
        try {
            if (firstCard != 0 && tmp == firstCard)
                throw new AlreadyChoosen("This card has choosen at first!");
            success3 = true;
        }
        catch (AlreadyChoosen alreadyChoosen)
        {Console.WriteLine(alreadyChoosen.Message); }
        if (success1 && success2 && success3)
```

```
            return tmp;
        else
        {  success1 = false;
            success2 = false;
            success3 = false;
            Console.Write("Enter a proper input : ");}
    }}
public bool is_allCardsOpen(int index)
{if(index == 15)
        return arrIs_Open[15];
    return arrIs_Open[index] && is_allCardsOpen(index + 1);
}
static void Main(string[] args)
{Program p = new Program();
    int firstCard;
    int secondCard;
    int count = 0;
    DateTime start = DateTime.Now;
    while (!p.is_allCardsOpen(0))
    {p.printCards();
        Console.Write("Enter first card : ");
        firstCard = p.checkInput();
        p.firstCard = firstCard;
        Console.Write("Enter second cars : ");
        secondCard = p.checkInput();
        Console.WriteLine();
        p.findCards(firstCard, secondCard);
        p.firstCard = 0;
        Console.WriteLine();
        count++;
    }
    Console.WriteLine("Making move : " + count);
    Console.WriteLine("Time : " + (DateTime.Now - start));
```

}}}

This console application was my last one, I ended my internship whit that. I added that my first and last console application in the appendix part but for the others I link my github in references part.

References

[1] Ministry of Transport and Infrastructure of the Republic of Turkey's website is available:

https://sgb.uab.gov.tr/

[2] My github link:s

https://github.com/KodMachine/internshipSourceCodes

[3] Sadık Turan web development Html, Css, Sass, Flexbox, Bootstrap, Javascript, Angular, JQuery, Asp.Net Mvc&Core Mvc Udemy Course.

[4] Clean Code A Handbook of Agile Software Craftsmanship The Object Mentors: Robert C. Martin Michael C. Feathers Timothy R. Ottinger Jeffrey J. Langr Brett L. Schuchert James W. Grenning Kevin Dean Wampler Object Mentor Inc.