GEBZE TECHNICAL UNIVERSITY

COMPUTER ENGINEERING

CSE222-2021

HOMEWORK 01 REPORT

AHMET FURKAN KURBAN

1801042674

1 INTRODUCTION

1.1 Problem Definition

The company has 4 different branches and sells both in-store and online. The store sells office chairs (7 models and for each, 5 colors), office desks (5 models and for each, 4 colors), meeting tables (10 models and for each, 4 colors), bookcases (12 models) and office cabinets (12 models). The automation system has users such as administrators, branch employees and customers. Administrators manage the system by adding and removing branches and branch employees. They can also query whether there are any products that need to be supplied. A special customer number is defined for each customer in the system. When a person subscribes to the system, he/she uses his/her name, surname and e-mail address, and defines a password. When the registration is completed, the customer number is automatically defined in the system and the user is informed about it. When the customer logs into the system, they can search for products, see the list of products, see which store a product is in, shop online by entering address and phone information, and view their previous orders. Branch employees can inquire about the products in stock, inform the manager that the product should be purchased when any product is less than the requested amount, add / remove products, make sales, access the information of the previous orders of a customer by using the customer number and add new order to this section. Branch employees should update the customers' previous orders section during their sales from the store. If the customer is shopping for the first time, then they should create a new subscription.

System Requirements

First of all the company must be created.While creating the company name of company and name ,surname,id of admin is determined with constructur.

```
Administrator(Company company,String name, String surname, int id) {
		super(name, surname, id);
		this.company=company;
}
Company company=new Company("Kurban Holding","Adem","Catal",10);
```

Administrator can add and delete branch ,branch employee They can also query whether there are any products that need to be supplied.Adding removing branch ,branch employee can be done with branchcode.

```
public void addBranch(int branchCode)
public void removeBranch(int branchCode)
```

Adding removing branch employee can be done with branch code and personel information of branch employee.

```
public boolean addBranchEmployee(int branchCode,String name,String surname,int id)
public boolean removeBranchEmployee(int branchCode,String name,String surname,int id)
```

The bound function for supplying the product with administrator.

```
public void add1Product(Branch branch,productClass product,int number,int number1)
```

Branch employee can add / remove customer ,add / remove products from branchs .They can be make sales to the customer.Also they can be access the up to date previous orders of customer.They can be add new order and update order.Lastly Branch employees can inquire about the products in stock, inform the manager that the product should be purchased when any product is less than the requested amount.

Adding / removing the customer are needed personal features and special number of customers.

```
public void addCustomer(String name,String surname,int id,String email,String password,String SpecialNumber)
```

```
public void removeCustomer(String name,String surname,int id,String email,String password,String SpecialNumber)
```

Adding / removing products are needed product and which model and which color informations.

```
public void addProducts(productClass product,int number,int number1)
```

```
public void removeProducts(productClass product,int number,int number1)
```

To reach previous orders of customer the personal information and special number are needed.

```
public void accesPreviousOrders(String name,String surname,int id,String email,String password,String SpecialNumber)
```

To add / remove order to / from customer the personel information special number and product information are needed.

```
public void addNewOrder(String name,String surname,int id,String email,String password,String SpecialNumber,productClass product,int number,int number1)
```

```
public void updateOrders(String name,String surname,int id,String email,String password,String SpecialNumber,productClass product,int number,int number1)
```

To inquire about products, company (To reach the admin) ,product information and indexes of branch and customer in the array are needed.

```
public boolean findProduct(Company company,productClass product,int number1,int number2,int number3,int number4)
```

To makesales personal information special number product information are needed.

```
public void makeSales(String name,String surname,int id,String email,String password,String SpecialNumber,productClass product,int number,int number1)
```

Customer can search products ,see list products ,find product in which branch,see the his/her previous orders and but product with online or normally.

To search products ,branch and product information are necessary.

```
public int searchProducts(Branch branch,productClass product,int number,int number1)
```

Customer can see the list of products via branch whichever is registered.

```
public void listofProducts()
```

Customer can find the products via company->branch whichever is registered and product informations.

```
public int findProductWhichBranch(Company company,productClass product,int number,int number1)
```

Customer can see the previous orders via branch whichever is registered this orders is not deleted after shopping.

```
public void previousOrders()
```

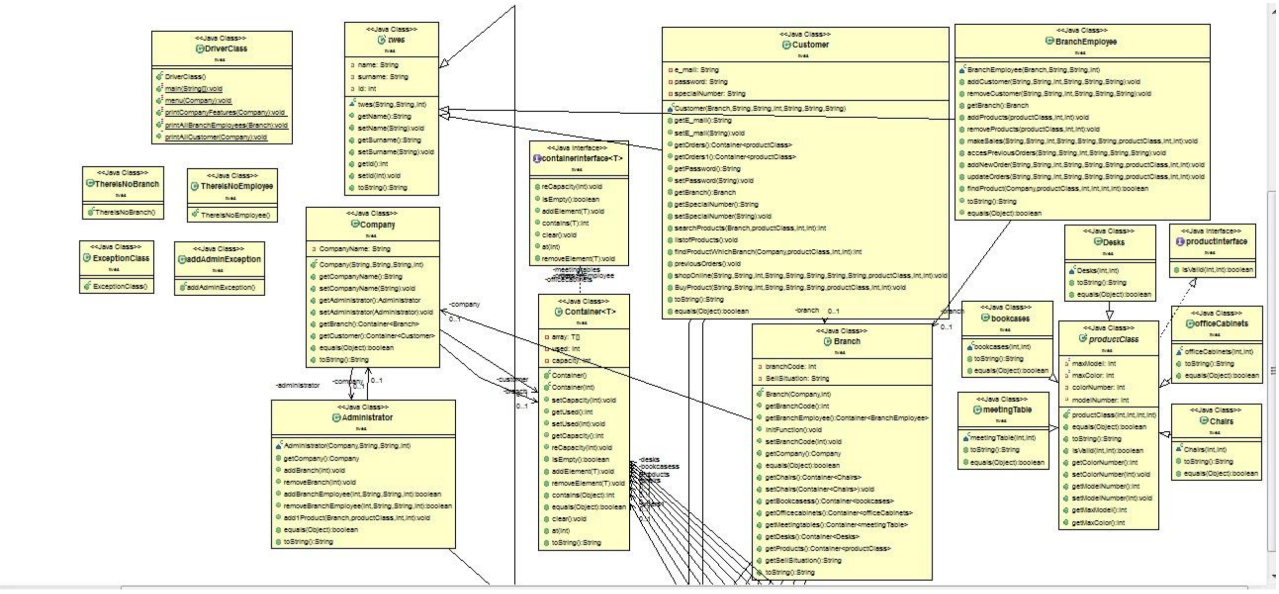
To buy something customer informations(because of adding order) and product informations are necessary.

```
public void BuyProduct(String name,String surname,int id,String email,String password,String SpecialNumber,productClass product,int number,int number1
```

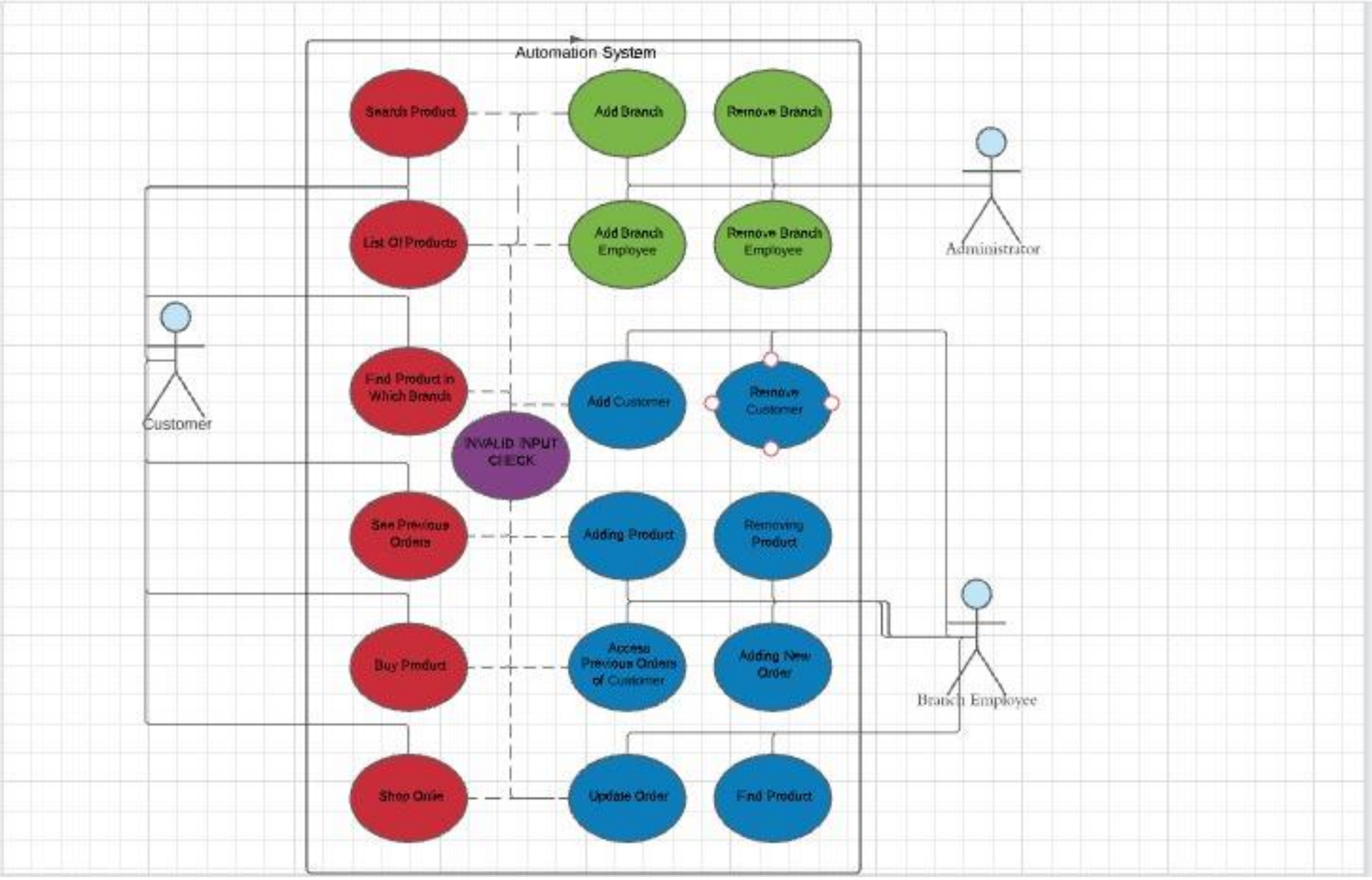
2 METHOD:

2.1 Class Diagrams

UML DİAGRAM



2.2 USE CASE DIAGRAM



2.3 Problem Solution Approach

First of all,I have implemented and Company Class which has all the information about the automation system.This class carries the data portion about the company.Then i created branch class that has branch employee and products there is company object in branch to reach company easily using the getCompany().I created a container class to keep and modify the datas easily.Then ı set up a hierarchy and find solution correctly determining the class relationships.Some class i have branch and company object this situation made it easy to reach all data easily (getCompany(),getBranch())

3 RESULT

3.1 Test Cases

Create a company with administrator-

```
Company company=new Company("Kurban Holding","Adem","Catal",10);
```

```
Kurban Holding
Administrator : Adem  Catal
```

Administrator:

-Adding Branch:

Branch is added to the this company by Administrator

This branch has already exist at company (Already Exist Branch added one more)

Branch is deleted from this company by Administrator

There is no branch like that at company (Already Exist Branch removed one more)

Branches:

Kurban Holding --> 1. Branch

Kurban Holding --> 2. Branch

-Adding Branch Employee

Adding branch employee is successful.

Company Name: Kurban Holding

Branch Code: 2

Name: Mehmet

Surname: kurban

ID: 1234

-Removing Branch Employee

```
Removing Branch Employee

Removing branch employee is successful.

Company Name: Kurban Holding

Branch Code: 2

Name: Ali

Surname: kurban

ID: 12345
```

```
-------------------------------
```

Branch Employee:

```
-------------------------------
```

-Adding a Customer

```
Adding customer operation is successful.

Customer : alibaba@gmail.com21323512321

Name: Ali

Surname: Baba

ID: 123457
```

-Removing a Customer

```
Removing customer operation is successful.

Customer : alibaba@gmail.com21323512321

Name: Ali

Surname: Baba

ID: 123457
```

-Adding Products

```
Employee is adding the product

DesksproductClass [maxModel=5, maxColor=4, colorNumber=5,
modelNumber=3]

DesksproductClass [maxModel=5, maxColor=4, colorNumber=5,
modelNumber=4]
```

-Removing Products

```
Employee is deleting the product

DesksproductClass [maxModel=5, maxColor=4, colorNumber=5,
modelNumber=3]

DesksproductClass [maxModel=5, maxColor=4, colorNumber=3,
modelNumber=3]
```

Here firstly i add then i delete desk 3(model),3(color) then i added desk 3,3 again i want to show you what happens after adding and removing.

-Access Previous Orders of Customer

```
Employee is looking previous orders of customer

DesksproductClass [maxModel=5, maxColor=4, colorNumber=1,
modelNumber=1]

DesksproductClass [maxModel=5, maxColor=4, colorNumber=1,
modelNumber=2]

DesksproductClass [maxModel=5, maxColor=4, colorNumber=1,
modelNumber=3]
```

-Adding New Order

```
Orders succesfully Added by customer

OfficeChairproductClass [maxModel=7, maxColor=5, colorNumber=1,
modelNumber=1]

OfficeChairproductClass [maxModel=7, maxColor=5, colorNumber=2,
modelNumber=2]
```

```
OfficeChairproductClass [maxModel=7, maxColor=5, colorNumber=3, modelNumber=3]
```

-Update Order After makesale-

```
After Make Sales (makesales function have updatedorders function)

DesksproductClass [maxModel=5, maxColor=4, colorNumber=1, modelNumber=3]

DesksproductClass [maxModel=5, maxColor=4, colorNumber=1, modelNumber=2]



Orders succesfully Updated after make sales

DesksproductClass [maxModel=5, maxColor=4, colorNumber=1, modelNumber=2]
```

-FindProduct

```
employee.findProduct(company,desks,1,3,1,1);
```

Element is not founded at that branch so we said admin then admin have added product the branch.(if element is not in branch)

Customer:

-Search Product

```
customer.searchProducts(branch,desk,3,3);
```

```
Customer is searching the product...

Products is founded index :10
```

-list Of Products

```
customer.listofProducts();
```

There is whole products but i added 4 of them to show (you can see the terminal whole products)

```
meetingTableproductClass [maxModel=10, maxColor=4, colorNumber=10, modelNumber=3]

meetingTableproductClass [maxModel=10, maxColor=4, colorNumber=10, modelNumber=4]

DesksproductClass [maxModel=5, maxColor=4, colorNumber=1, modelNumber=1]

DesksproductClass [maxModel=5, maxColor=4, colorNumber=1, modelNumber=2]
```

-find product in which branch

```
customer.findProductWhichBranch(company, desk,3, 3);
```

```
Customer is searching the product is which Branch...

Products is founded at that branch index :1
```

-see previous orders

```
customer.previousOrders();
```

```
Customer is looking the his/her previous orders...

OfficeChairproductClass [maxModel=7, maxColor=5, colorNumber=1, modelNumber=1]

OfficeChairproductClass [maxModel=7, maxColor=5, colorNumber=1, modelNumber=2]

OfficeChairproductClass [maxModel=7, maxColor=5, colorNumber=1, modelNumber=3]
```

-Buy Product

```
customer.BuyProduct("Elon","Musk",1234567,"ElonMusk@gmail.com","231212312","1",desk,4,4);

employee.addNewOrder(name,surname, id, e_mail, password, specialNumber, product, number, number1);

employee.accesPreviousOrders(name, surname, id, email, password, SpecialNumber);

employee.makeSales(name,surname, id, e_mail, password, specialNumber, product, number, number1);

removeProducts(product,number,number1);

updateOrders(name,surname,id,email,password,SpecialNumber,product,number,number1);
```

```
Customer is buying product...
```

Here if customer want to buy product first function is working then employee add new order to customer orders and with previous orders we can see the customers orders then employee make sale to customer , bought product is removing from that branch lastly with update orders function employee remove the orders that customer needed from customer orders but previous orders is still has customer so we can see if we want anytime.