

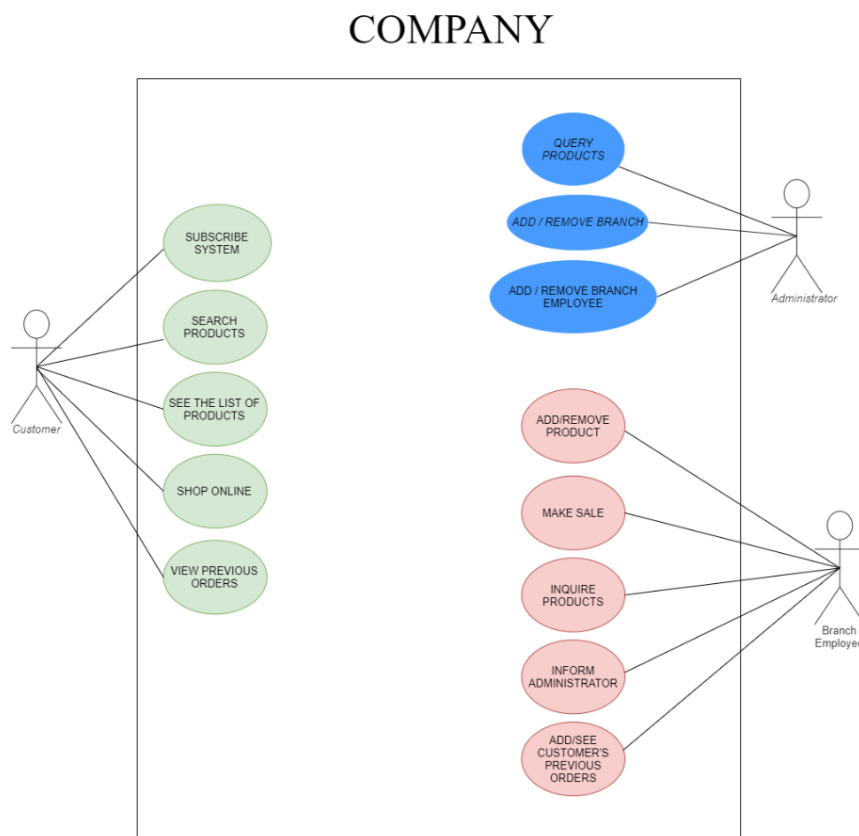
**GIT Department of Computer Engineering
CSE 222/505 - Spring 2021
Homework 3 Report**

**Student Name
Samet NALBANT
Student Number
1801042614**

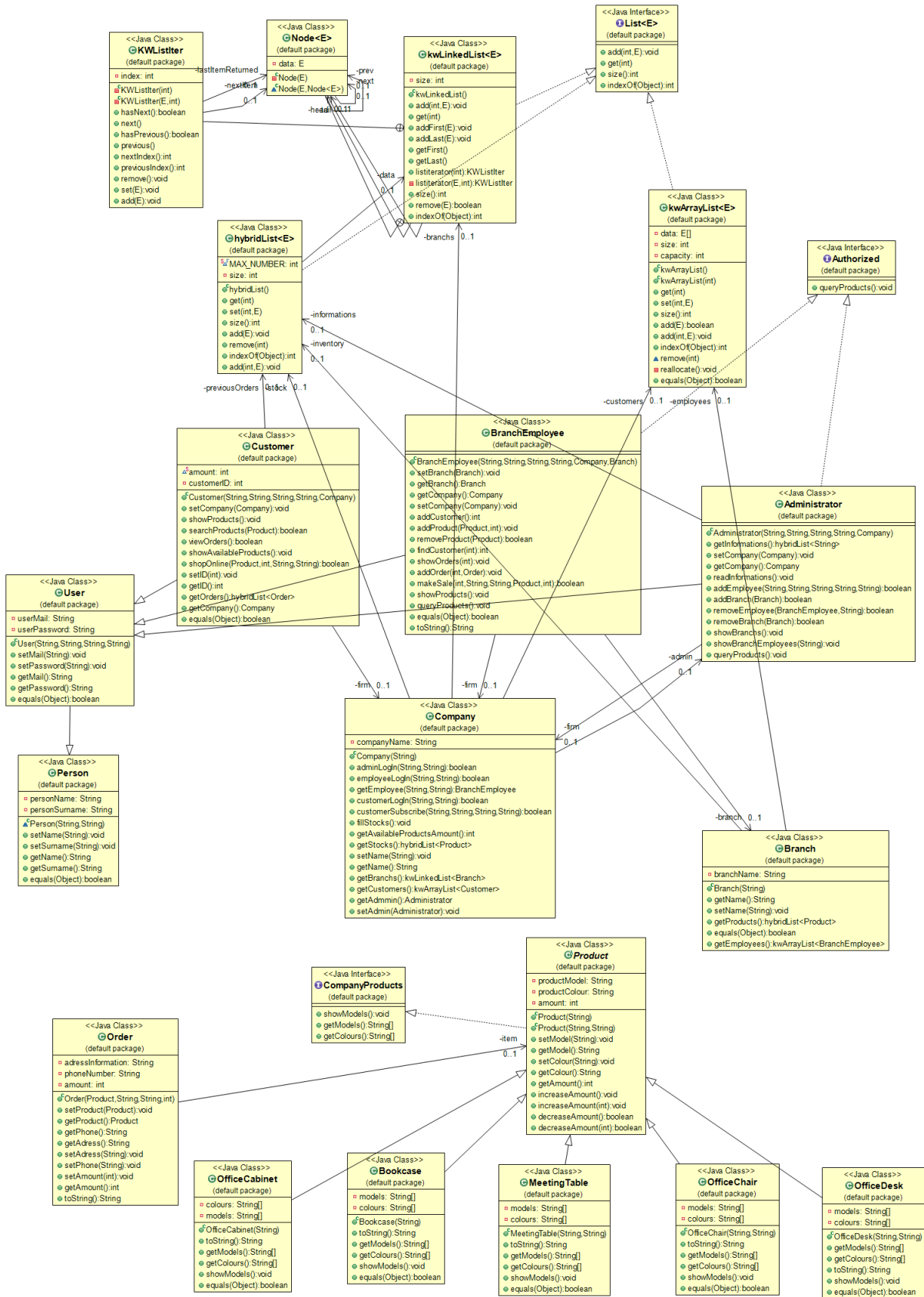
1. SYSTEM REQUIREMENTS

- 1) Firstly company must be created.
- 2) Administrator must be set to use administrator functions. Branches, branch employees can create or remove using the administrator functions.
- 3) Branch Employees can add products, remove products and make sales to customers. Customers need to be subscribed company to buy something. If customer shopping first time branch employee can do subscription.
- 4) Customers search products, shop online, show products, and can view their previous orders. To be able to do all of these customers should subscribe to company. After the subscription they will informed about their customer ID.

USE CASE DIAGRAM



CLASS DIAGRAM



2. PROBLEM SOLUTION APPROACH

I list the necessary functions for automation system. I create a necessary classes according to homework file such as arraylist, linkedlist, hibrit list. I create product class and furniture classes (bookcase, office desk etc.) as a subclass of product class. I create product class as an abstract class because furniture classes almost have same functions and same implementation and also they have some differences. So I can use polymorphism easily. I implement an interface to product class, this way I can create functions which has different implementation. After the complete product part I create person class to keep people's specific information such as name, surname etc. Then I create user class as a subclass of person to hold necessary information to use entering automation system. After I implement employee, administrator, customer subclasses of user with functions which defined in homework and other helper functions. After that I create company and branch classes appropriate way. Finally I need to order class to hold customer's previous orders.

3. TEST CASES

To be able to test system I create company and administrator of the company.

```
Company gtu = new Company("GTU BUTTERFLY A.S");  
gtu.setAdmin(new Administrator("Anonymous", "Anonymous", "admin@gtu", "gtu123", gtu));
```

After that I use administrator functions for the automation system.

```
admin.addBranch(new Branch("Gebze"));  
admin.addBranch(new Branch("Pendik"));  
admin.addEmployee("Ahmet", "Aksoy", "ahmet@gtu", "123", "Gebze");  
admin.addEmployee("Fatma", "Tekin", "fatma@gtu", "123", "Gebze");  
admin.addEmployee("Mehmet", "Kayhan", "mehmet@gtu", "123", "Pendik");
```

Automation system needs to some products. So using branch employee functions I add some products.

```
employee = x.getEmployee("ahmet@gtu", "123");  
employee2 = x.getEmployee("mehmet@gtu", "123");
```

```
employee.addProduct(new Bookcase("Middle Size"), 3);  
employee.addProduct(new Bookcase("Rectangle"), 7);  
employee.addProduct(new Bookcase("Square"), 2);  
employee.addProduct(new Bookcase("Wooden"), 0);
```

I didn't add the all model of each product because using the automation system all models all colors can be add according to user's requests.

```
x.customerSubscribe("Samet", "Nalbant", "sam@gtu", "123");  
x.customerSubscribe("Feyza", "Kaya", "feyza@gtu", "123");
```

I create two customer. To shop some products using their ID.

```
Your customer id is: 0  
Your customer id is: 2
```

```
customer1 = x.getCustomers().at(employee.findCustomer(0));  
customer1.shopOnline(new OfficeDesk("Wooden", "Gray"), 2, "Gebze Home", "123456");  
customer1.shopOnline(new OfficeCabinet("Reverse"), 3, "Gebze Office", "123456");
```

Finally I try to remove not existing branch to see the throw exception.

```
admin.removeBranch(new Branch("yjtrhe"));
```

```
Branch couldn't Found
```

4. RUNNING AND RESULTS

Entering system as an administrator.

```
Welcome to GTU BUTTERFLY A.S  
1)Enter as an Administrator.  
2)Enter as a Branch Employee.  
3)Enter as Customer.  
4)Subscribe to system as a Customer.  
0)EXIT  
  
1  
Enter Mail Address:  
admin@gtu  
Enter Password:  
gtu123  
Succesfully Loged In
```

Trying to add new branch.

```
Enter the name of the new Branch  
Tuzla  
Operation Succesful!
```

```
----BRANCHS----  
1-)Gebze  
----BRANCHS----  
2-)Pendik  
----BRANCHS----  
3-)Tuzla  
-----
```

-Trying to add already exist branch.

```
Enter the name of the new Branch  
Tuzla  
This Branch is already exist!
```

-Trying to remove branch.

```
----BRANCHS----  
1-)Gebze  
----BRANCHS----  
2-)Pendik  
----BRANCHS----  
3-)Tuzla  
-----  
  
Enter the number of the Branch:  
3  
Operation Succesful!
```

After remove:

```
----BRANCHS----  
1-)Gebze  
----BRANCHS----  
2-)Pendik  
-----
```

-Adding employee

```
----BRANCHS----
1-)Gebze
-----

Enter the number of the Branch to Add new Employee.
1
Enter the name of the Employee.
Ali
Enter the surname of the Employee.
Kaya
Enter the mail of the Employee.
ali@gtu
Enter the password of the Employee.
gtu123
Operation Succesful!
```

-Removing Employee

```
----BRANCHS----
1-)Gebze
-----

Enter the number of the Branch to Remove Employee:
1
Gebze
Employees of Gebze :
1) Name: Ahmet - Surname: Aksoy - Mail: ahmet@gtu - Password: 123
2) Name: Fatma - Surname: Tekin - Mail: fatma@gtu - Password: 123
3) Name: Ali - Surname: Kaya - Mail: ali@gtu - Password: gtu123
```

-Query Products

```
----Consumed Product List----

----- Gebze -----
Wooden Bookcase
Narrow Black MeetingTable
Middle size OfficeCabinet
Plastic Red OfficeDesk
```

-Entering as a branch employee

```
Welcome to GTU BUTTERFLY A.S
1)Enter as an Administrator.
2)Enter as a Branch Employee.
3)Enter as Customer.
4)Subscribe to system as a Customer.
0)EXIT
```

```
2
Enter Mail Address:
ahmet@gtu
Enter Password:
123
Succesfully Loged In
```

```
1)Add product.
2)Remove product
3)Make Sale
4)Show Previous Order
5)Inquire Products
0)EXIT
```

-Add product

```
1
1)Bookcase
2)Meeting Table
3)Office Cabinet
4)Office Chair
5)Office Desk
Enter the number of Item Which You Want to Add:
```

```
Models:
1) Long Size
2) Short size
3) Middle size
4) Narrow
5) Wide
6) Circle
7) Square
8) Rectangle
9) Wooden
10) Plastic
11) Grid
12) Reverse

Enter the number of model:
3
Enter the amount:
2
```


-Removing product.

```
19)Wide OfficeCabinet x1
20)Wooden Gray OfficeDesk x12
21)Wooden Black OfficeDesk x12
22)Wooden Red OfficeDesk x1
23)Plastic Red OfficeDesk x0
24)Plastic Gray OfficeDesk x10
25)Rectangle Red OfficeDesk x6
26)Square White OfficeDesk x5
27)Square Red OfficeDesk x2
28)Narrow White OfficeDesk x3
29)Middle size Bookcase x2

Enter the number of the product:
```

```
19)Wide OfficeCabinet x1
20)Wooden Gray OfficeDesk x12
21)Wooden Black OfficeDesk x12
22)Wooden Red OfficeDesk x1
23)Plastic Red OfficeDesk x0
24)Plastic Gray OfficeDesk x10
25)Rectangle Red OfficeDesk x6
26)Square White OfficeDesk x5
27)Square Red OfficeDesk x2
28)Narrow White OfficeDesk x3

1)Add product.
```

-Make sale

```
Enter the number of the product:
2
Enter the amount of product 7
Enter the CustomerID:
If you are not subscribed enter -2
0
Enter the phone number
123456
Enter the customer address
Gebze University
```

```
1)Middle Size Bookcase x3
2)Rectangle Bookcase x7
```

-Show previous orders

```
4
Enter the customer id:
0
Address: Gebze Home - Phone Number: 123456 - Product: Wooden Gray OfficeDesk- x 2
Address: Gebze Office - Phone Number: 123456 - Product: Reverse OfficeCabinet- x 3
Address: Gebze University - Phone Number: 123456 - Product: Rectangle Bookcase- x 7
1)Add product.
```

-Inquire Products

```
5
1)Middle Size Bookcase x3
2)Rectangle Bookcase x0
3)Square Bookcase x2
4)Wooden Bookcase x0
5)Wide Red OfficeChair x5
6)Wide Black OfficeChair x2
7)Wooden White OfficeChair x1
8)Wooden Orange OfficeChair x2
9)Plastic White OfficeChair x5
10)Long Size Black MeetingTable x24
11)Narrow Black MeetingTable x0
12)Wide Black MeetingTable x1
```

...

-Entering as a customer

```
3
Enter Mail Address:
sam@gtu
Enter Password:
123
Succesfully Loged In
```

-See list of products show which stores has how many products.

```
----- Branch Gebze-----
Middle Size Bookcase      x3
Rectangle Bookcase        x0
Square Bookcase           x2
Wooden Bookcase           x0
Wide Red OfficeChair      x5
Wide Black OfficeChair    x2
Wooden White OfficeChair  x1
Wooden Orange OfficeChair x2
Plastic White OfficeChair x5
Long Size Black MeetingTable x24
Narrow Black MeetingTable x0
Wide Black MeetingTable   x1
Square Red MeetingTable   x3
Suare Gray MeetingTable   x7
Narrow OfficeCabinet      x3
Short size OfficeCabinet  x12
Middle size OfficeCabinet x0
Rectangle OfficeCabinet   x7
Wide OfficeCabinet        x1
Wooden Gray OfficeDesk    x12
Wooden Black OfficeDesk   x12
Wooden Red OfficeDesk     x1
```

-Search products takes model and colour as an number again.And shows which branch has it.

```
Enter the model number:
7
----- Branch Gebze-----
Square Bookcase          x2
```

-Shop online shows the all products in branches.

```
1)Middle Size Bookcase
2)Rectangle Bookcase
3)Square Bookcase
4)Wooden Bookcase
5)Wide Red OfficeChair
6)Wide Black OfficeChair
7)Wooden White OfficeChair
8)Wooden Orange OfficeChair
9)Plastic White OfficeChair
10)Long Size Black MeetingTable
11)Narrow Black MeetingTable
12)Wide Black MeetingTable
13)Square Red MeetingTable
14)Suare Gray MeetingTable
15)Narrow OfficeCabinet
16)Short size OfficeCabinet
17)Middle size OfficeCabinet
18)Rectangle OfficeCabinet
19)Wide OfficeCabinet
20)Wooden Gray OfficeDesk
21)Wooden Black OfficeDesk
22)Wooden Red OfficeDesk
23)Plastic Red OfficeDesk
24)Plastic Gray OfficeDesk
25)Rectangle Red OfficeDesk
26)Short Size Bookcase
```

```
27)Middle Size Bookcase
Enter the number of the product to buy:
21
Enter the amount of product:
5
Enter the adress:
Pendik Home
Enter the phone:
123456
Order Completed!
```

-View previous orders.

```
4
Adress: Gebze Home - Phone Number: 123456 - Product: Wooden Gray OfficeDesk- x 2
Adress: Gebze Office - Phone Number: 123456 - Product: Reverse OfficeCabinet- x 3
Adress: Gebze University - Phone Number: 123456 - Product: Rectangle Bookcase- x 7
Adress: Pendik Home - Phone Number: 123456 - Product: Wooden Black OfficeDesk- x 5
1)See the list of Products
```

-Subscribe system as a customer

```
4
Enter Name:
Necati
Enter Surname:
Sasmaz
Enter Mail Adress:
necati@gtu
Enter Password:
123
Your customer id is: 6
You can log in now!
```

-----Function Anlayze-----

Add Product

```

public void addProduct(Product newProduct, int amount){
    int index = getBranch().getProducts().indexOf(newProduct);  $O(m \cdot n)$ 
    if(index == -1){
        getBranch().getProducts().add(newProduct);  $O(n)$ 
        index = getBranch().getProducts().indexOf(newProduct);  $O(m \cdot n)$ 
    }
    getBranch().getProducts().get(index).increaseAmount(amount);  $O(n)$ 
    firm.fillStocks();  $O(x \cdot y)$ 
}

int indexOf(E target){
    for(int i=0; i<size; i++){
        if(data.get(i / MAX_NUMBER).get(i % MAX_NUMBER).equals(target)){
            return i;
        }
    }
    return -1;
}

public void add(E anEntry){
    if(size % MAX_NUMBER == 0){
        data.addLast(new KwArrayList<E>());
    }
    data.get(size / MAX_NUMBER).add(anEntry);
    size++;
}

public void fillStocks(){
    for(int i=0; i<getBranches().getSize(); i++){
        for(int j=0; j<getBranches().at(i).getProducts().getSize(); j++){
            stock.addElement(getBranches().at(i).getProducts().at(j));
        }
    }
}

```

$O(m \cdot n + x \cdot y)$
 $O(m)$ array list get method
 $O(n)$ linked list get method
 $O(n)$
 $O(n)$
 $O(n)$
 $T(m, n) = O(m \cdot n)$

Remove Product

```

public boolean removeProduct(Product oldProduct){
    try {
        getBranch().getProducts().remove(getBranch().getProducts().indexOf(oldProduct));
    } catch (Exception e) {
        System.out.println("Product couldn't Found\n");
        return false;
    }
    firm.fillStocks();
    return true;
}

public void fillStocks(){
    for(int i=0; i<getBranches().getSize(); i++){
        for(int j=0; j<getBranches().at(i).getProducts().getSize(); j++){
            stock.addElement(getBranches().at(i).getProducts().at(j));
        }
    }
}

int indexOf(E target){
    for(int i=0; i<size; i++){
        if(data.get(i / MAX_NUMBER).get(i % MAX_NUMBER).equals(target)){
            return i;
        }
    }
    return -1;
}

public E remove(int index){
    if(index < 0 || index >= size){
        throw new ArrayIndexOutOfBoundsException(index);
    }
    for(int i=index; i<size-1; i++){
        if(i % MAX_NUMBER == MAX_NUMBER - 1){
            data.get(i / MAX_NUMBER).set(i % MAX_NUMBER, data.get((i+1) / MAX_NUMBER).get((i+1) % MAX_NUMBER));
        }
        else{
            data.get(i / MAX_NUMBER).set(i % MAX_NUMBER, data.get((i+1) / MAX_NUMBER).get((i+1) % MAX_NUMBER));
        }
    }
    size--;
    return null;
}

public E set(int index, E anEntry){
    if(index < 0 || index >= size){
        throw new ArrayIndexOutOfBoundsException(index);
    }
    E temp = data.get(index / MAX_NUMBER).get(index % MAX_NUMBER);
    data.get(index / MAX_NUMBER).set(index % MAX_NUMBER, anEntry);
    return temp;
}

```

$O(m \cdot n)$
 $O(t^3 + m \cdot n)$
 $O(t^3)$
 $O(n)$
 $T(m, n) = O(m \cdot n)$
 $O(m)$ array list get method
 $O(n)$ linked list get method
 $O(m \cdot n)$
 $O(n^3)$
 $O(n^3)$
 $O(n)$
 $O(n)$
 $O(n)$
 $O(n)$
 $O(n)$

ADD BRANCH

```
public boolean addBranch(Branch newBranch){
    getCompany().getBranches().addLast(newBranch);
    return true;
}
```

kw linked list add last

$O(1)$ } $O(1)$

REMOVE BRANCH

```
public boolean removeBranch(Branch oldBranch){
    return getCompany().getBranches().remove(oldBranch);
}
```

$O(n)$

```
public boolean remove(E obj){
    KWListIter temp;
    try {
        temp = listiterator(obj, 0);
    } catch (Exception e) {
        return false;
    }
    temp.remove();
    return true;
}
```

$O(n)$ } $O(n)$

```
private KWListIter(E obj, int flag){
    lastItemReturned = null;
    nextItem = head;
    boolean flag2 = false;
    for(index = 0; index < size; index++){
        if(nextItem.data.equals(obj)){
            flag2 = true;
            break;
        }
        nextItem = nextItem.next;
    }
    if(flag2 == false){
        throw new UnsupportedOperationException();
    }
}
```

$O(n)$

SEARCH PRODUCT

```
public boolean searchProducts(Product product){
    int index;
    boolean flag = false;
    for(int i = 0; i < getCompany().getBranches().size(); i++){
        index = getCompany().getBranches().get(i).getProducts().indexOf(product);
        if(index != -1){
            flag = true;
            System.out.println("----- Branch " + getCompany().getBranches().get(i).getName() + "-----");
            System.out.println(getCompany().getBranches().get(i).getProducts().get(index).toString());
        }
    }
    return flag;
}
```

$O(mn)$

$O(n)$ } $O(n)$

$O(n^2)$

$O(n^2 + mn)$

```
public E get(int index){
    return data.get(index/MAX_NUMBER).get(index%MAX_NUMBER);
}
```

$O(n)$

$O(1)$

ADD EMPLOYEE

```
public boolean addEmployee(String name,String surname,String mail,String password,String branchName){
    int index = getCompany().getBranches().indexOf(new Branch(branchName));  $\rightarrow O(n)$  linked list
    int index2;
    if(index == -1){
        return false;  $O(1)$ 
    }
    else{
        index2 = getCompany().getBranches().get(index).getEmployees().indexOf(new BranchEmployee(name, surname, mail, p
        if(index2 != -1){
            System.out.println("Employee is already registered!");
            return false;
        }
        getCompany().getBranches().get(index).getEmployees().add(new BranchEmployee(name, surname, mail, password, get
        return true;
    }
}

int indexOf(Object target){
    for(int i=0;i<size();i++){
        if(target.equals(data[i])){
            return i;
        }
    }
    return -1;
}
```

REMOVE EMPLOYEE

```
public boolean removeEmployee(BranchEmployee employee,String branchName){
    int index = getCompany().getBranches().indexOf(new Branch(branchName));  $O(n)$ 
    if(index == -1){
        return false;  $O(1)$ 
    }
    else{
        try {
            getCompany().getBranches().get(index).getEmployees().remove(getCompany().getBranches().get(index).getEmployees().indexOf(empl
        } catch (Exception e) {
            System.out.println("Employee couldn't Found\n");
            return false;
        }
        return true;
    }
}
```

Search Product