PART 2 :

Refik Orkun Arslan

Administarator :

-Add branch O(1)

```
public boolean addBranch(int branchCode)
{
    return getCompany().getBranch().add(new branch(getCompany(),branchCode));
}
```

getCompany() $=\theta(1)$

getBranch() $=\theta(1)$

add(new branch(getCompany(),branchCode))=linked list add so $\theta(1)$

-Remove branch O(1)

```
public boolean removeBranch(int branchCode)
{
    return getCompany().getBranch().remove(new branch(getCompany(),branchCode));
}
```

getCompany() $=\theta(1)$

getBranch() $=\theta(1)$

add(new branch(getCompany(),branchCode))=*linked* list remove so $\theta(1)$

-Add branchEmployee O(1)

```
public boolean addBranchEmployee(int branchCode)
{
    int index=0;
    return getCompany().getBranch().get(branchCode).getBranchEmployee().add(new
branchEmployee(getCompany().getBranch().get(index)));
}
```

getCompany() $=\theta(1)$

getBranch() $=\theta(1)$

get(branchCode) $=\theta(1)$

getBranchEmployee()$=\theta(1)$

add(new branchEmployee(getCompany().getBranch().get(index)));=*linked* list add so $\theta(1)$

-Remove branch Employee O(n)

```
public boolean removeBranchEmployee(int branchCode, String name, String surname, int
id)
{
    int index=0;
    if((index = getCompany().getBranch().indexOf(new branch(getCompany(),branchCode)))
== -1)
        return false;
    return getCompany().getBranch().get(index).getBranchEmployee().remove(new
branchEmployee(getCompany().getBranch().get(index)));
}
```

indexOf(new branch(getCompany(),branchCode))=θ(n)

getCompany() =θ(1)

getBranch() =θ(1)

getBranchEmployee()=θ(1)

remove(new branchEmployee(getCompany().getBranch().get(index)));=*Linked* list remove so θ(1)

BranchEmployee :

-add User O(1)

```
public boolean addUser(String name, String surname, String email,int password)
{
    return getBranch().getCompany().getCustomer().add(new
person(name,surname,email,password));
}
```

getBranch() =θ(1)

getCompany() =θ(1)

getCustomer() =θ(1)

add(new person(name,surname,email,password))=*ArrayList* addlast so θ(1)

-remove User O(n)

```
public boolean removeUser(String name, String surname,String email,int password)
{
    return getBranch().getCompany().getCustomer().remove(new
person(name,surname,email,password));
}
```

getBranch() =θ(1)

getCompany() =θ(1)

getCustomer() =θ(1)

remove(new person(name,surname,email,password))=*ArrayList* remove so θ(n)

-Add product O(1)

```
public boolean enterShipmentInformation(officeChairModel a,officeChairColor c)
{
    return getBranch().getOfficeChair().add(new officeChair(a,c));
}
```

getBranch() =θ(1)

getOfficeChair()=θ(1)

add(new officeChair(a,c));=θ(1) linkedlist and arraylist addlast

***-Remove product* O(1)**

```
public void removeShipmentInformation(officeChairModel a,officeChairColor c)
{
    int j;

    for( j=0 ; j < getBranch().getOfficeChair().size() ; ++j)
    {

        if(getBranch().getOfficeChair().get(j).getOfficeChairModel()== a &&
getBranch().getOfficeChair().get(j).getOfficeChairColor()== c )
        {
            getBranch().getOfficeChair().get(j).clear();
        }
    }
}
```

getBranch() =θ(1)

getOfficeChair()=θ(1)

getOfficeChairModel()=θ(1)

getOfficeChairColor()=θ(1)

clear()=θ(1)

Company:

***-User Login* O(1)**

```
public person login(String email,int password)
{
    for(int i=0;i<customer.size();i++)
    {
        if(customer.get(i).getEmail()==email &&
customer.get(i).getPassword()==password)
        {
            return customer.get(i);
        }

    }
    System.out.println("Not member account");

    return null;

}
```

customer.get(i)=ArrayList get θ(1)

getEmail() =θ(1)

getPassword() =θ(1)

Person(user):

-Search Product O(*branch.getSize() * getOfficeChair().size()*)

```
public officeChair searchProduct(KWLinkedList<branch> branch,officeChairModel model,
officeChairColor color)
{

     for(int i=0 ; i < branch.getSize() ; ++i)
    {

        for(int j=0 ; j < branch.get(i).getOfficeChair().size() ; ++j)
        {
            if(branch.get(i).getOfficeChair().get(j).getOfficeChairModel()== model &&
branch.get(i).getOfficeChair().get(j).getOfficeChairColor()== color )
            {
                System.out.println("Found it");
                System.out.println("Branch :" + branch.get(i).getBranchCode());
                return  branch.get(i).getOfficeChair().get(j);
            }
            else
            {
                System.out.println(" NOT Found ");



            }

        }
    }

    return null;
}
```

`branch.get(i)`=Linked list get method θ(1)

`getOfficeChair()`=θ(1)

-Buy Product O(*branch.getSize() * getOfficeChair().size()*)

```
public void buy(KWLinkedList<branch> ranch,String Adres,int phone,officeChairModel
a,officeChairColor c)
 {
    officeChair b=searchProduct(ranch,a,c);
```

=SearchProduct method θ(*branch.getSize() * getOfficeChair().size()*)

```
     b.setOfficeChairModel(null);
     b.setOfficeChairColor(null);
     System.out.println("Buy it");

 }
```

-Querying O(branch.getSize() * getOfficeChair().size()*get index)

```
public officeChair querying(KWLinkedList<branch> branch,officeChairModel model,
officeChairColor
        color)
{
    for(int i=0 ; i < branch.size() ; ++i) ==> O(branch.size())
    {
        for(int j=0 ; j < branch.at(i).getOfficeChair().size() ; ++j)
        { ==> O(getOfficeChair().size())
            if(branch.at(i).getOfficeChair().at(j).getOfficeChairModel()== model &&
                    branch.at(i).getOfficeChair().at(j).getOfficeChairColor()== color )
        ==>O(1)
            {
                System.out.println("Found it"); ==> O(1)
                System.out.println("Branch :" + branch.at(i).getBranchCode()); ==> O(1)
                return branch.at(i).getOfficeChair().at(j); ==> O(1)
            }
            else
            {
                System.out.println(" NOT Found "); ==> O(1)
            }

}
```