GTU Department of Computer Engineering
CSE 222/505 - Spring 2021
Homework #1 Report

Yunus Emre Öztürk
1901042619

1. SYSTEM REQUIREMENTS

All classes and containers in this project implements serializable so creating a local copy with the data inside of them is not a problem.

To start using the functionalities client must create a company object with using company constructor. Constructor needs company name as input.

```
public Company(String name)
```

After the company creation, company object can be used for calling menus with a scanner object that will be determined by client:

```
public static void adminMenu(Company comp, Scanner in)
public static void employeeMenu(Company comp, Scanner in)
public static void customerMenu(Company comp, Scanner in)
```

These menus has specialized functionalities for Admin, Employee and Customers.

Starting with Admins:

Admins can add branch to the company with giving a unique special name which has been never used for creating branches:

```
public boolean addBranch(String name)
```

Admins can Access already created branches by giving the correct name of it:

```
public Branch getBranch(String name);
```

Admins can remove branches again with giving the unique name of the branch.

```
public boolean removeBranch(String name);
```

As can see here, names of the branches acts like the ID of the branches.

Admins can add or remove branch employees to/from branches with giving the branch information:

To add an employee admin should use only employee name and surname then program will generate a unique ID and information data for employee.

```
public Employee addBranchEmployee(Branch branch, String name, String surname);
```

To remove an employee, admin should use unique employee ID

```
public boolean removeBranchEmployee(Branch branch, String ID);
```

Admins can communicate with other parts of the company with Information Boxes, these boxes includes messages to the manager or admins. For example a branch employee can inform the manager about restock etc.

Because of each branch has unique box inside of it, client must provide branch to the function and system will give the messages inside of it to the user.

```
public String lookInfoBoxes(Branch branch);
```

Admins can delete messages using the associated number with message .

```
public boolean removeInfo(Branch branch, int ind);
```

Branch Employees:

Each employee has unique ID to login the system.

Branch employees can create new customer subscription to sell product in branches. Employee should provide name, surname, mail and password information about the customer.

```
Customer createNewCustomer(String name, String surname, String email, String password);
```

Branch employees can Access the order list of the customer by providing the customer ID to the system.

```
public Order getOrder(String ID);
```

After getting the Order list, they can sell items to the customers after providing the item information.

```
public boolean addOrder(Order order, Item item)
```

They can remove the items from the Order list by providing the index number associated with the item in the Order list, they can only remove not shipped items.

```
public boolean removeOrder(Order order, int index);
```

Branch employees has full Access to their branch depots, they can add item by providing the item information to the system.

```
public boolean addItem(Item item);
```

And they can remove items from the depots again by providing the item information.

```
public boolean removeItem(Item item);
```

Client can directly Access branch depot information by using **branch employee Access.**

```
public Depot getBranchDepot();
```

And branch employees can leave restock messages to the managers by providing the item information.

```
public void restockInform(Item item);
```

Customer:

Customers should login the system by using their passwords and mail adresses, each mail address can be associated only one customer account. Customers can create accounts with branches or they can create in online.

Customer should provide name, surname, mail and password information, then system will give the Online Depot information.

```
Customer(String name, String surname, String email, String password, OnlineDepot onlineDep)
```

Again each customer has unique ID, which can accessed by :

```
public String getCustomerID();
```

Each customer can look the product list and branch information of the products by:

```
public String getProductList();
```

Each customer can search specific product by providing the name of it

```
public String searchProduct(String name);
```

They can add item to their order list with providing the item information:

```
public boolean addToOrder(Item item);
```

They can get their order information:

```
public Order getOrder();
```

And they can remove items from their order list by providing the associated number of the item in the order list. (Only not shipped orders):

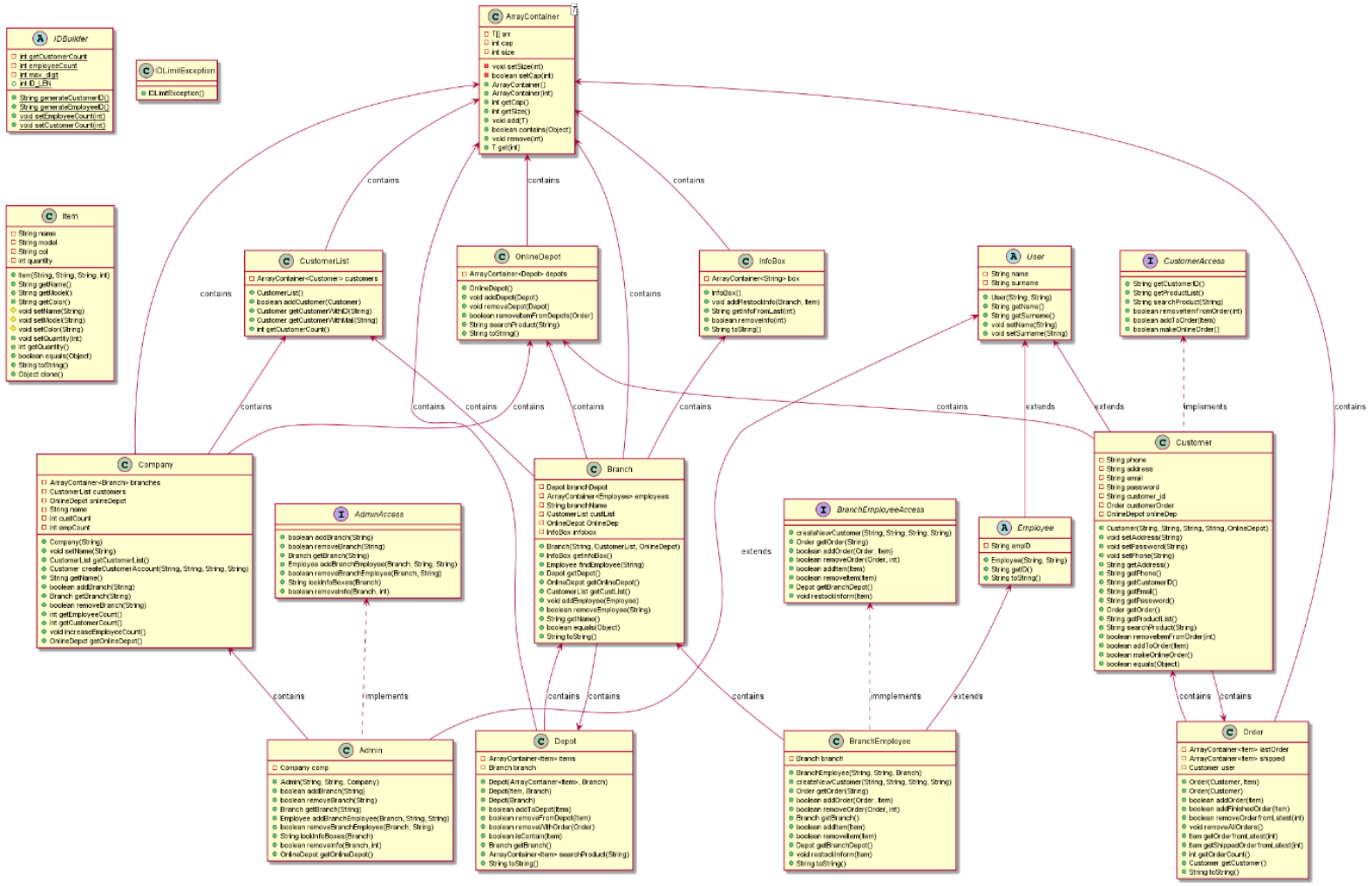
```
public boolean removeItemFromOrder(int ind);
```

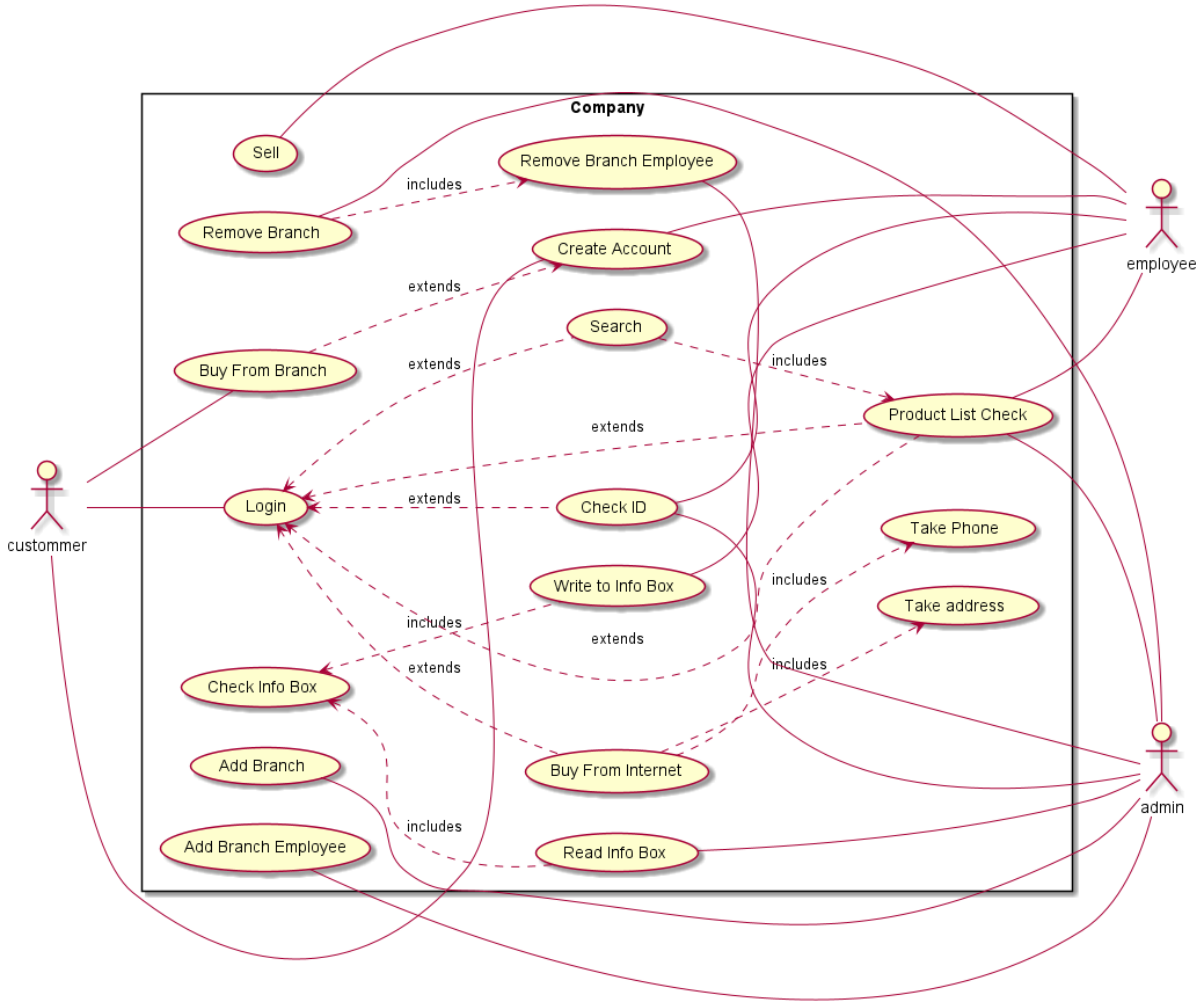
And they can finish their order:

```
public boolean makeOnlineOrder();
```

Notes: Most of the objects which can accessed by simply get methods can be instantly print because of their overriden toString() methods, for example client can write a code and print the Order object directly.

USE CASE AND CLASS DIAGRAMS





2.

3. PROBLEM SOLUTION APPROACH

First of all I write down what will I do before start the coding, and draw same simple class diagrams. First I decided to create a class (ArrayContainer) to modify the arrays easily, this was the first problem that I solved. Then I used this container and create more high level data holders for specific jobs, for example I created CustomerList which has ArrayContainer in it, with this CustomerList I can access the customers by their ID's if I wouldn't done this probably I would write the same code over and over again for all needed functions. CustomerList, Depot, InfoBox, Order, OnlineDepot are my higher level data holders, at the end all of them is created by the arrays.

Aside from data holding problems, other parts of the project can be done without that much problem because we don't have restrictions about them.

4. TEST CASES

Admin Part:

Creates new branch:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
1
Please write unique name of Branch: branch1
branch1:

Employees:
```

Tries to create new branch with same name:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
1
Please write unique name of Branch: branch1
Please write a unique name for Branch!
```

Removes a branch:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
2
Please write unique name of Branch: branch1
Branch removed successfully.
```

Tries to remove same branch again:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
2
Please write unique name of Branch: branch1
Please write a unique name for Branch!
```

Tries to add an employee to a removed branch:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
3
Please write unique name of Branch: branch1
Please write the name of the employee: Gordon
Please write the surname of the employee: Freeman
Branch doesn't exist!
```

I created the branch again and added the employee:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
3
Please write unique name of Branch: branch1
Please write the name of the employee: Gordon
Please write the surname of the employee: Freeman
Employee Information: Name: Gordon;Surname: Freeman;ID: E00001;Branch: branch1;
```

Removes employee from branch:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
4
Please write unique name of Branch: branch1
Please wrrite the ID of employee: E00001
Employee successfully removed from the branch.
```

Tries to remove same employee again:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
4
Please write unique name of Branch: branch1
Please wrrite the ID of employee: E00001
Please provide correct name of the branch and ID of the employee!
```

I put some item to the branch depot and tried to Show branch depot and employee list.

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
5
Please write unique name of Branch: branch1
branch1:
   Name: office_desk;Model: model1;Color: white;Quantity: 150;
   Name: office_chair;Model: model1;Color: brown;Quantity: 100;

Employees:   Name: Gordon;Surname: Freeman;ID: E00002;Branch: branch1;
```

I created another branch and add some item to it, to test 6th choice:

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
6
branch1:
   Name: office_desk;Model: model1;Color: white;
   Name: office_chair;Model: model1;Color: brown;
;branch2:
   Name: office_chair;Model: model2;Color: black;
   Name: office_cabinet;Model: model1;Color: white;
```

I sent a message from branch1 employee to the manager for restock:

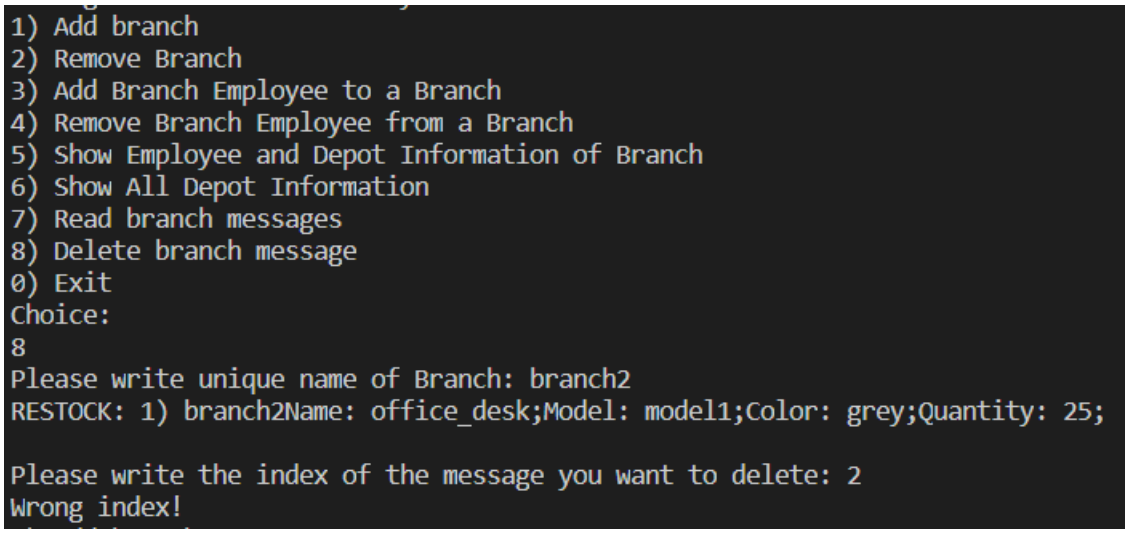
```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
7
Please write unique name of Branch: branch1
RESTOCK: 1) branch1Name: office_cabinet;Model: model1;Color: white;Quantity: 50;
```

Deletes the message already read.

```
1) Add branch
2) Remove Branch
3) Add Branch Employee to a Branch
4) Remove Branch Employee from a Branch
5) Show Employee and Depot Information of Branch
6) Show All Depot Information
7) Read branch messages
8) Delete branch message
0) Exit
Choice:
8
Please write unique name of Branch: branch1
RESTOCK: 1) branch1Name: office_cabinet;Model: model1;Color: white;Quantity: 50;

Please write the index of the message you want to delete: 1
Message deleted successfully
```

Tries to delete the message with wrong index:



Employee Part:

Login with Employee ID:

```
1) Admin login
2) Employee login
3) Customer login
0) Exit
2
Please write the name of your Branch: branch2
Please write your ID: E00003
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice:
```

Login with wrong ID and branch, both of them tested:

```
1) Admin login
2) Employee login
3) Customer login
0) Exit
2
Please write the name of your Branch: branch2
Please write your ID: AAAAAAA
System couldn't find employee! Please try again!
Please write the name of your Branch: AAAAA
Please write your ID: E00003
Branch name is not correct!
```

Creates new customer:

```
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 1
Write customer name: Frodo
Write customer surname: Baggins
Write customer mail: fbaggin@mail.com
Write customer password: 1234
Subscription Created, customer ID: C00001
```

Checks depot:

```
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 3
branch2:
   Name: office_chair;Model: model2;Color: black;Quantity: 250;
   Name: office_cabinet;Model: model1;Color: white;Quantity: 40;
```

Sells item and stock situation after sell:

```
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 4
Please write the ID of the customer: C00001
Please write the name of the Item: office_chair
Please write the model of the Item: model2
Please write the color of the Item: black
Please write the quantity of the Item: 55
Order completed and decreased from stock.
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 3
branch2:
   Name: office_chair;Model: model2;Color: black;Quantity: 195;
   Name: office_cabinet;Model: model1;Color: white;Quantity: 40;
```

Checks customer order list, bought office_chair and it is shipped to customer directly:

```
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 2
Please write the ID of customer: C00001
Last order(s):
Shipped order(s):
1) Name: office_chair;Model: model2;Color: black;Quantity: 55;
```

Try to sell something to non-registered customer ID:

```
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 4
Please write the ID of the customer: C00002
Please write the name of the Item: office_chair
Please write the model of the Item: model2
Please write the color of the Item: black
Please write the quantity of the Item: 10
Customer ID is not correct!
```

Removes last order:

```
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 5
Please write the ID of the customer: C00001
Last order(s):
1) Name: office_chair;Model: model2;Color: black;Quantity: 5;
Shipped order(s):
1) Name: office_chair;Model: model2;Color: black;Quantity: 55;

Please write the index of the item: 1
Item removed from order successfully.
```

Tries to remove order of non-registered customer:

```
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 5
Please write the ID of the customer: AAAAAAAA
Customer ID is not correct!
```

Adds item to the depot, this screenshot shows last situation in depot:

```
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 6
Please write the name of the Item: office_lamp
Please write the model of the Item: model
Please write the color of the Item: white
Please write the quantity of the Item: 25
Item added to depot.
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 3
branch2:
   Name: office_chair;Model: model2;Color: black;Quantity: 195;
   Name: office_cabinet;Model: model1;Color: white;Quantity: 40;
   Name: office_lamp;Model: model;Color: white;Quantity: 25;
```

Removes item from depot, shows last situation of depot:

```
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 7
Please write the name of the Item: office_lamp
Please write the model of the Item: model
Please write the color of the Item: white
Please write the quantity of the Item: 20
Item removed from the depot.
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 3
branch2:
   Name: office_chair;Model: model2;Color: black;Quantity: 195;
   Name: office_cabinet;Model: model1;Color: white;Quantity: 40;
   Name: office_lamp;Model: model;Color: white;Quantity: 5;
```

Tries to remove 15 items while depot has 5:

```
branch2:
   Name: office_chair;Model: model2;Color: black;Quantity: 195;
   Name: office_cabinet;Model: model1;Color: white;Quantity: 40;
   Name: office_lamp;Model: model;Color: white;Quantity: 5;

1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 7
Please write the name of the Item: office_lamp
Please write the model of the Item: model
Please write the color of the Item: white
Please write the quantity of the Item: 15
There is no such item in depot or enough quantity!
```

Informs manager about restock:

```
1) Create New Customer Subscription
2) Check Customer Order
3) Check Depot
4) Sell Item
5) Remove Customer Order
6) Add Item to Depot
7) Remove Item from Depot
8) Inform Manager
0) Exit
Choice: 8
Please write the name of the Item: office_lamp
Please write the model of the Item: model
Please write the color of the Item: white
Please write the needed restock quantity of the Item: 25
Manager informed!
```

Customer Part:

Creates new account:

```
1) Login
2) Create New Account
0) Exit
2
Write your name: Peter
Write your surname: Parker
Write your mail: parker@mail.com
Write your password: 1234
Subscription Created, customer ID: C00002
1) See list of the products
2) Search product
```

Log-in:

```
Please write your email: parker@mail.com
Please write your password: 1234
1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
```

Tries to log-in with unknows email:

```
Please write your email: ppppp@mail.com
Please write your password: 22222
Wrong password or mail!
1) Login
2) Create New Account
0) Exit
```

Checks the list of the products

```
1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
1
branch1:
   Name: office_desk;Model: model1;Color: white;
   Name: office_chair;Model: model1;Color: brown;
;branch2:
   Name: office_chair;Model: model2;Color: black;
   Name: office_cabinet;Model: model1;Color: white;
   Name: office_lamp;Model: model;Color: white;
```

Search for products:

```
1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
2
Please write the name of the item:
office_chair
branch1:
   Name: office_chair;Model: model1;Color: brown;Quantity: 100;
branch2:
   Name: office_chair;Model: model2;Color: black;Quantity: 195;
```

Adds item to the order list:

```
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
4
Please write the name of the Item: office_chair
Please write the model of the Item: model2
Please write the color of the Item: black
How many you want: 50
Item added to your order list!
1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
3
Last order(s):
1) Name: office_chair;Model: model2;Color: black;Quantity: 50;
Shipped order(s):
```

Checks the order list:

```
Last order(s):
1) Name: office_chair;Model: model2;Color: black;Quantity: 50;
Shipped order(s):

Please choose the item number you want to remove:1
Item removed successfully.
1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
```

Stock wont decrease if user didn’t finish the order:

```
2
Please write the name of the item:
office_chair
branch1:
   Name: office_chair;Model: model1;Color: brown;Quantity: 100;
branch2:
   Name: office_chair;Model: model2;Color: black;Quantity: 195;

1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
3
Last order(s):
1) Name: office_chair;Model: model2;Color: black;Quantity: 55;
Shipped order(s):
```

Now it is decreased:

```
6) Make the order
7) Check your ID
0) Exit
Choice:
6
Please write your address: addressssssssss
Please write your phone: 01111111111
Order completed successfully!
1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
2
Please write the name of the item:
office_chair
branch1:
   Name: office_chair;Model: model1;Color: brown;Quantity: 100;
branch2:
   Name: office_chair;Model: model2;Color: black;Quantity: 140;
```

Checks customer ID:

```
1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
7
C00002
```

Tries to buy not listed item, it will be added to the order list but user cant buy it:

```
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
4
Please write the name of the Item: random
Please write the model of the Item: random
Please write the color of the Item: random
How many you want: 1
Item added to your order list!
1) See list of the products
2) Search product
3) Check Order List
4) Add item to your order
5) Remove item from your order list
6) Make the order
7) Check your ID
0) Exit
Choice:
6
Please write your address: randomrandomrandom
Please write your phone: 01111111111
Couldn't complete the order!
```