

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

**Yeşim Yalçın  
200104004094**

# 1-) System Requirements

## A-) Functional Requirements

### -Log In/Registration Requirements

- The company's automation system will have users who will use the system. Users can be classified as "Admin", "Employee" or "Customer". The system will offer the users different things to do according to their status.
- The system saves all the new users who registers into the system.
- Users can register by putting their name, surname, email and password information.
- Once registered, the user is saved as a customer.
- Branch Employees can be added to the system by Admins, they cannot register by themselves.
- Admins can be added to the system only by editing the AdminsInfo.txt file.
- Every user can be registered with their own unique email address. The system will not accept two users with the same email address.
- Every customer has their own unique customer number, this number will be created by the system when they register for the first time.

### -Menu Requirements

#### -Admin Menu

- \* ADDBranch: Creates a new branch for the company in the system.
- \* REMOVEBranch: Removes a branch of the company in the system.
- \* ADDEmployee: Adds a new employee for the company in the system.
- \* REMOVEEmployee: Removes an employee of the company in the system.
- \* CHECK: Checks what products should be supplied.
- \* QUIT: Logs off.

#### -Employee Menu

- \* STOCK: Shows the stock of a branch/online store or all.
- \* ARRANGESales: Arrange the things related to sales.
- \* ADDProduct: Adds a product to a branch's stock or to the online store's stock.
- \* REMOVEProduct: Removes a product from a branch's stock or from the online store's stock.
- \* SUPPLYProducts: Supplies the products that are needed to be supplied.
- \* MAKESale: Makes a sale through branch.
- \* PASTOrders: Shows the past orders of a customer.
- \* QUIT: Logs off.

#### -Customer Menu

- \* SEARCH: Searches for a product in the stores.
- \* LIST: Shows all the products this company produces and sells.
- \* SHOP: Starts shopping.
- \* PASTOrders: Views past orders.
- \* QUIT: Logs off.

### -Store Requirements

- When the company's automation system is run for the first time, it creates 4 branches and 1 online store. However more branches can be added or branches can be removed later on by the admins. These changes will be saved.

- If a branch is removed, all the other branch numbers will be changed accordingly. Example: If the Branch3 is deleted, old Branch4 will be the new Branch3. Previous orders from that branch will be changed as if they were done & will be done from the online store not to affect the customers. Also the Supply List will be reseted because of the changes in the branch numbers. Employees can create a new Supply List afterwards.

-There can be only one online store, if more will be tried to created, it will not have any affect.

-The online store cannot be removed.

-All branches and the online stock have their own stock.

-Branch employees can edit the stock of both branches and the online stock.

### **-Product Requirements**

-Products are created by determining their type, model and the colour.

-Types can be Office\_Chair, Office\_Desk, Meeting\_Table, Bookcase or Office\_Cabinet.

-All types have different available colours and models.

-When the type is determined, the available colours and the models for that product will be set.

-Office\_Chair have 7 models and can be Black, White, Red, Blue, Green

-Office\_Desk have 5 models and can be Black, White, Red, Blue

-Meeting\_Table have 10 models and can be Black, White, Red, Blue

-Bookcase have 12 models and can be Black

-Office\_Cabinet have 12 models and can be Black

-If a non existing product is tried to be created, the system will ask for new information input.

### **-Shopping Requirements**

-If a customer wants to buy from a branch and the product exists in their stock, the purchase is done instantly. The product will be added as a past order.

-If a customer wants to buy from a branch and the product does not exist in their stock, the purchase will be added as a new order. When the product is supplied to the stock of the branch, the purchase can be completed.

-If a customer wants to buy from the online store, the purchase will be added as a new order and when a branch employee arrange the sales, the purchase will be done.

### **- Arranging Sales Requirements**

-Branch employees can arrange the sales of the customer. They edit a new order of a customer as a past order to complete the purchase. They have to decrease the product amount from the store's stock. If the product does not exist in the stock, they cannot change the order as a past order. They firstly need to add the product to the Supply List. Supply the necessary product to the store, then arrange that sale again.

### **B-) Non-Functional Requirements**

-You can use the makefile then use java AutomationSystem command to run the program.

-The system comes with the files AdminsInfo.txt, BranchEmployeesInfo.txt, CustomersInfo.txt and BranchCount.txt. These files are important for the system to start properly. If these files are deleted, there will not be any admins, employees or customers that are registered to begin with. Also there will not be any branches. Other files will be created by the system when they are needed.

-To be able to log in as an admin, you can check the AdminsInfo.txt file and log in with one of the existing admin information.

- To be able to log in as a branch employee, you can check the BranchEmployeesInfo.txt file and log in with one of the existing employee information.
- To be able to log in as a customer, you can check the CustomersInfo.txt file and log in with one of the existing customer information or you can register as a new customer.
- The input for the model information for the products is expected to be an integer. If a string is entered, the system will be exited.
- The input for all the string information such as name, surname, email information etc is expected not to have spaces inside. If a string with spaces inside is entered, the information might be saved wrongly and need to be edited later on.
- The status of the users is to determine if they were registered from before or not. If a wrong status is entered, InvalidStatus will be thrown and the system will be exited.  
Example: if a user is created with “old” status but the email address is not registered, it is in an invalid status situation.
- If the branch employees try to remove more products than there is, InvalidAmount will be thrown and the system will be exited.

## **2-) Class Diagram-Use Case Diagram**

- The class diagram is added as ClassDiagram.png to the homework file.
- The use case diagram is added as UseCaseDiagram.png to the homework file.

## **3-)Problem Solution Approach**

- While starting designing the software, I decided to create the small essential pieces first. These pieces are the Users, Products and the Stores. I have implemented all of their qualities in their own classes.
- To be able to create Admin, BranchEmployee and Customer classes, I created an abstract User class. The reason why it is an abstract class is, Admin, BranchEmployee and Customer classes have many common fields and methods.
- I decided to create one Product class rather than creating different classes for all different type of products because that would be too much code repetition and too many classes to deal with.
- I decided to make a Store interface and implemented it on Branch and OnlineStore classes. The reason why I decided to create an interface was wanting to be able to use both Branches and OnlineStores in the same array with the same functions.
- Another thing I paid attention was saving everything to files. The system saves the data to the arrays but I wanted the changes to stay even when I closed the program. Because of that I created a file for everything that might be changed.
- After creating every detail about these classes I created the CompanySystem class that gathers all those small pieces all together. This class connects every method of the other classes that need to work together to perform proper tasks like adding a product to a customer’s past orders.
- I wanted to make the general flow of the system to be like this:
  - \*User logs into the system or gets registered.
  - \*Their status is determined: Admin/Employee/Customer.
  - \*They can choose the actions they want to do through the menu of the system.
  - \*All the changes done to the informations are saved to the files.
  - \*When the system is opened again, the changes will still be there. Thanks to this, all the users can interact with eachother in a way with the changes they make in the system.

#### 4-) Test Cases and Running Results

-These tests are done in the specified order. The orders are specified with the numbers.

-Note: While doing all the tests, the files were also examined. The changes in the files were done successfully.

##### A-)Logging In/Registration Tests

Test No	Scenario	Expected Result	Real Result	Pass/Fail
1	Entering wrong password as admin then fixing it.	Asking for a new password. If entered the right password, logs in.	The same	Pass
2	Entering wrong password as employee then fixing it.	Asking for a new password. If entered the right password, logs in.	The same	Pass
3	Entering wrong password as customer then fixing it.	Asking for a new password. If entered the right password, logs in.	The same	Pass
4	Entering as a new user and registering as a new customer.	Asking for information and then registering the customer.	The same	Pass

##### B-)Customer Menu Tests

Test No	Scenario	Expected Result	Real Result	Pass/Fail
5	Putting a wrong command	Asking for a new command	The same	Pass
6	Searching for a product with wrong type&colour&model information as customer and then fixing it	Asking for new proper information. If entered properly, proceeds with searching for the product.	The same	Pass
7	Using LIST command	Lists all the products available	The same	Pass
8	Using PASTOrders command while	Showing No Past Orders	The same	Pass

	having no past orders			
9	Shopping from Online Store with wrong product information then fixing it	Asking for new proper information. If entered properly, proceeds with buying the product.	The same	Pass
10	Shopping from a branch with wrong product information then fixing it	Asking for new proper information. If entered properly, proceeds with buying the product.	The same	Pass
11	Trying to shop from a store that does not exist then fixing it.	Asking for new proper information. If entered properly, proceeds with shopping.	The same	Pass
12	Trying to shop from a store that does not have the specified product in its stock then changing the store.	Asking for a new store to choose then proceeds with shopping.	The same	Pass
13	Trying to shop from a store that does not have the specified product in its stock	Adding the order as a new order and waiting it to be supplied.	The same	Pass
14	Using PASTOrders command while having past orders.	Printing out the past orders.	The same	Pass
15	Checking if the stocks are updated after making purchases with STOCK command	Stocks should have been updated and the updated version should be printed out.	The same	Pass

#### C-)Admin Menu Tests

Test No	Scenario	Expected Result	Real Result	Pass/Fail
---------	----------	-----------------	-------------	-----------

16	Using CHECK command when there is no products registered in the SupplyList	Printing out There is no Products that need to be supplied	The same	Pass
17	Trying to add an already existing employee then fixing it	Asking for another email that does not exist. If entered properly, adds the employee.	The same	Pass
18	Trying to remove an employee who does not exist	Throwing InvalidStatus	The same	Pass
19	Using REMOVEEmployee command successfully	Removing the employee	The same	Pass
20	Using ADDBranch command successfully	Adding a new branch	The same	Pass
33	Using CHECK function when there are products that need to be supplied	Printing out all the products that need to be supplied	The same	Pass
34	Using REMOVEBranch function with a wrong branch number then fixing it	Asking for a proper branch number. If entered properly, removes it.	The same	Pass

#### D-)Employee Menu Test

Test No	Scenario	Expected Result	Real Result	Pass/Fail
21	Using STOCK command for all stocks	Showing the stocks of all stores in total	The same	Pass
22	Using STOCK command with a wrong command then for ONE	Asking for a proper input then showing the given store's stock only	The same	Pass
23	Using PASTOrders command with a wrong customer number	Printing out that there is no customers with the given number	The same	Pass

<b>24</b>	Using PASTOrders command successfully	Showing the customer's past orders with the given number	The same	Pass
<b>25</b>	Using SUPPLYProducts order when there is nothing to be supplied.	Printing out there is nothing that needs to be supplied	The same	Pass
<b>26</b>	Using ADDProduct with wrong product informations to a branch then fixing it	Asking for proper information. If entered, edits the stock of the branch accordingly.	The same	Pass
<b>27</b>	Using ADDProduct with wrong product informations to Online Store then fixing it	Asking for proper information. If entered, edits the stock of the online store accordingly.	The same	Pass
<b>28</b>	Using REMOVEProduct command successfully.	Editing the store's stock accordingly.	The same	Pass
<b>29</b>	Trying to remove a product by some amount that would make the stock of it negative	Throwing InvalidAmount	The same	Pass
<b>30</b>	Using ARRANGESales command successfully	Arranging all the sales. Editing new orders, editing the Supply List etc	The same	Pass
<b>31</b>	Using MAKESale command for an existing customer when the product is not in the stock	Adding the order to the customer's new order's section and adding the product to the Supply List.	The same	Pass
<b>32</b>	Using MAKESale command for a new customer firstly from a	Asking for a new store number, then making the sale.	The same	Pass



	branch that does not exist, then from a branch which has the product in the stock.			
35	Using SUPPLYProducts command when there are product that need to be supplied.	Supplying all the products, editing the stocks.	The same	Pass
36	Using ARRANGESales after supplying the products.	Arranging all the sales. Editing new orders	The same	Pass

#### E-) Extra Tests

Test No	Scenario	Expected Result	Real Result	Pass/Fail
37	Trying to put a string input for model instead of integer.	Might cause improper data	The same	Pass
38	Trying to put spaces in string inputs	Throws exception, exists the program	The same	Pass
39	Using QUIT command	Exits the program	The same	Pass

#### F-) The Results

1-) Entering wrong password as admin then fixing it.

```

Enter Email:
y.yalcin072@gmail.com
Enter your password:
122232
You have entered a wrong password. Try again:
12345y

*****Admin Menu*****

Commands List:
ADDBranch: Creates a new branch for the company in the system.
REMOVEBranch: Removes a branch of the company in the system.
ADDEmployee: Adds a new employee for the company in the system.
REMOVEEmployee: Removes an employee of the company in the system.
CHECK: Checks what products should be supplied.
QUIT: Logs off.

*****
Enter command:

```

## 2-) Entering wrong password as employee then fixing it.

```
Enter Email:
keira02@gmail.com
Enter your password:
123453
You have entered a wrong password. Try again:
1234

*****Employee Menu*****

Commands List:
STOCK: Shows the stock of a branch/online store or all.
ARRANGESales: Arrange the things related to sales.
ADDProduct: Adds a product to a branch's stock or to the online store's stock.
REMOVEProduct: Removes a product from a branch's stock or from the online store's stock.
SUPPLYProducts: Supplies the products that are needed to be supplied.
MAKESale: Makes a sale through branch.
PASTOrders: Shows the past orders of a customer.
QUIT: Logs off.

*****
Enter command:
```

## 3-) Entering wrong password as customer then fixing it

```
Enter Email:
fayron@gmail.com
Enter your password:
23231
You have entered a wrong password. Try again:
23323

*****Customer Menu*****

Commands List:
SEARCH: Searches for a product in the stores.
LIST: Shows all the products this company produces and sells.
SHOP: Starts shopping.
PASTOrders: Views past orders.
QUIT: Logs off.

*****
Enter command:
```

## 4-) Entering as a new user and registering as a new customer

```
Enter Email:
emily@gmail.com
You can register the system by entering a password, your name and surname.
Enter your password:
4321
Enter your name:
Emily
Enter your surname:
Cara
The registration is completed. Your customer number is: 4
Enter your password:
4321

*****Customer Menu*****

Commands List:
SEARCH: Searches for a product in the stores.
LIST: Shows all the products this company produces and sells.
SHOP: Starts shopping.
PASTOrders: Views past orders.
QUIT: Logs off.

*****
Enter command:
```

### 5-) Putting a wrong command

```
Commands List:
SEARCH: Searches for a product in the stores.
LIST: Shows all the products this company produces and sells.
SHOP: Starts shopping.
PASTOrders: Views past orders.
QUIT: Logs off.

*****
Enter command:
Search
You have entered an invalid command. Please enter again.
```

### 6-) Searching for a product with wrong type&colour&model information as customer and then fixing it

```
Enter command:
Search
You have entered an invalid command. Please enter again.
SEARCH
Enter the type of the product you want to search:
Book
Enter the model of the product you want to search:
13
Enter the colour of the product you want to search:
Blue
InvalidType: This store does not produce Book.Please enter the type again.
Bookcase
InvalidModel: There is no Model 13 for Bookcase.Please enter the model number again.
2
InvalidColour: Bookcase cannot be Blue.Please enter the colour again.
Black
Bookcase 2 Black can be found in the Online Store.
There are 5
Bookcase 2 Black can be found in Branch 1.
There are 3
Bookcase 2 Black can be found in Branch 2.
There are 3
Bookcase 2 Black can be found in Branch 3.
There are 3
Bookcase 2 Black can be found in Branch 4.
There are 3
Enter command:
```

### 7-) Using LIST command (Not showing all the result because it is long)

```
LIST
All Products:
Office_Chair 1 Black
Office_Chair 1 White
Office_Chair 1 Red
Office_Chair 1 Blue
Office_Chair 1 Green
Office_Chair 2 Black
Office_Chair 2 White
Office_Chair 2 Red
Office_Chair 2 Blue
Office_Chair 2 Green
Office_Chair 3 Black
Office_Chair 3 White
Office_Chair 3 Red
Office_Chair 3 Blue
Office_Chair 3 Green
Office_Chair 4 Black
Office_Chair 4 White
Office_Chair 4 Red
Office_Chair 4 Blue
Office_Chair 4 Green
```

### 8-) Using PASTOrders command while having no past orders

```
Enter command:
PASTOrders
Your Past Orders:
No Past Orders

Enter command:
```

### 9-) Shopping from Online Store with wrong product information then fixing it

```
Enter command:
SHOP
Enter the type of the product you want to buy:
Book
Enter the model of the product you want to buy:
19
Enter the colour of the product you want to buy:
RED
Enter the store number you want to buy from. For Online Store enter 0.
0
InvalidType: This store does not produce Book.Please enter the type again.
Meeting_Table
InvalidModel: There is no Model 19 for Meeting_Table.Please enter the model number again.
2
InvalidColour: Meeting_Table cannot be RED.Please enter the colour again.
Red
You have successfully done your purchase from the online store. Your purchase will be delivered as soon as possible.

Enter command:
```

### 10-) Shopping from a branch with wrong product information then fixing it

```
Enter command:
SHOP
Enter the type of the product you want to buy:
Pen
Enter the model of the product you want to buy:
12
Enter the colour of the product you want to buy:
Green
Enter the store number you want to buy from. For Online Store enter 0.
2
InvalidType: This store does not produce Pen.Please enter the type again.
Office_Desk
InvalidModel: There is no Model 12 for Office_Desk.Please enter the model number again.
4
InvalidColour: Office_Desk cannot be Green.Please enter the colour again.
White
You have successfully done your purchase from Branch 2.

Enter command:
```

### 11-) Trying to shop from a store that does not exist then fixing it.

```
Enter command:
SHOP
Enter the type of the product you want to buy:
Meeting_Table
Enter the model of the product you want to buy:
9
Enter the colour of the product you want to buy:
Black
Enter the store number you want to buy from. For Online Store enter 0.
6
There is no store with this number, please enter again.
-1
There is no store with this number, please enter again.
3
You have successfully done your purchase from Branch 3.

Enter command:
```

12-) Trying to shop from a store that does not have the specified product in its stock then changing the store.

```
Enter command:
SHOP
Enter the type of the product you want to buy:
Office_Cabinet
Enter the model of the product you want to buy:
12
Enter the colour of the product you want to buy:
Black
Enter the store number you want to buy from. For Online Store enter 0.
1

Currently this product is not in the stock of the store you want to buy from.
You can make your order and wait for the stock to be updated or you can purchase it from other stores.
This product can be found in these stores' stocks:
Office_Cabinet 12 Black can be found in the Online Store.
There are 5
Office_Cabinet 12 Black can be found in Branch 2.
There are 3
Office_Cabinet 12 Black can be found in Branch 3.
There are 3
Office_Cabinet 12 Black can be found in Branch 4.
There are 3

Please enter the store number you want to buy from again. You can also enter the same number to proceed with your order.
2
You have successfully done your purchase from Branch 2.

Enter command:
```

13-) Trying to shop from a store that does not have the specified product in its stock

```
Enter command:
SHOP
Enter the type of the product you want to buy:
Office_Cabinet
Enter the model of the product you want to buy:
12
Enter the colour of the product you want to buy:
Black
Enter the store number you want to buy from. For Online Store enter 0.
1

Currently this product is not in the stock of the store you want to buy from.
You can make your order and wait for the stock to be updated or you can purchase it from other stores.
This product can be found in these stores' stocks:
Office_Cabinet 12 Black can be found in the Online Store.
There are 5
Office_Cabinet 12 Black can be found in Branch 2.
There are 2
Office_Cabinet 12 Black can be found in Branch 3.
There are 3
Office_Cabinet 12 Black can be found in Branch 4.
There are 3

Please enter the store number you want to buy from again. You can also enter the same number to proceed with your order.
1
You have successfully done your order from Branch 1. Your order will be delivered as soon as it gets into stock.

Enter command:
█
```

14-) Using PASTOrders command while having past orders.

```

Enter command:
PASTOrders
Your Past Orders:
Meeting_Table 9 Black
Office_Desk 4 White
Office_Cabinet 12 Black
Enter command:

```

### 15-) Checking if the stocks are updated after making purchases with STOCK command

```

SEARCH
Enter the type of the product you want to search:
Office_Cabinet
Enter the model of the product you want to search:
12
Enter the colour of the product you want to search:
Black
Office_Cabinet 12 Black can be found in the Online Store.
There are 5
Office_Cabinet 12 Black can be found in Branch 2.
There are 2
Office_Cabinet 12 Black can be found in Branch 3.
There are 3
Office_Cabinet 12 Black can be found in Branch 4.
There are 3
Enter command:

```

### 16-) Using CHECK command when there is no products registered in the SupplyList

```

Enter command:
CHECK
There are no products that need to be supplied.
Enter command:

```

### 17-) Trying to add an already existing employee then fixing it

```

Enter command:
ADDEmployee
Enter the name of the employee you want to add:
Menma
Enter the surname of the employee you want to add:
Taylor
Enter the email of the employee you want to add:
menma03@gmail.com
Enter the password of the employee you want to add:
12345
This email already exists. Enter another email.
menma@gmail.com
New employee is successfully added.
Enter command:

```

### 18-) Trying to remove an employee who does not exist

```

Enter command:
REMOVEEmployee
Enter the name of the employee you want to remove:
kara
Enter the surname of the employee you want to remove:
catlin
Enter the email of the employee you want to remove:
kara@gmail.com
Enter the password of the employee you want to remove:
323424
InvalidStatus: There is not an already existing branch employee with this email address.cse312@ubuntu:~/Documen
ts$

```

### 19-) Using REMOVEEmployee command successfully

```
Enter command:
REMOVEEmployee
Enter the name of the employee you want to remove:
Menma
Enter the surname of the employee you want to remove:
Taylor
Enter the email of the employee you want to remove:
menma@gmail.com
Enter the password of the employee you want to remove:
12345
The employee is successfully removed.

Enter command:
```

## 20-) Using ADDBranch command successfully

```
Enter command:
ADDBranch
New branch is successfully added.

Enter command:
```

## 21-) Using STOCK command for all stocks (Not showing all because it is too long)

```
Enter command:
STOCK
Do you want to see all stock or one stock of a store? Enter ALL or ONE.
ALL
Office_Chair 1 Black      20
Office_Chair 1 White      20
Office_Chair 1 Red        20
Office_Chair 1 Blue       20
Office_Chair 1 Green      20
Office_Chair 2 Black      20
Office_Chair 2 White      20
Office_Chair 2 Red        20
Office_Chair 2 Blue       20
Office_Chair 2 Green      20
Office_Chair 3 Black      20
Office_Chair 3 White      20
Office_Chair 3 Red        20
Office_Chair 3 Blue       20
Office_Chair 3 Green      20
Office_Chair 4 Black      20
Office_Chair 4 White      20
```

## 22-) Using STOCK command with a wrong command then for ONE

```
STOCK
Do you want to see all stock or one stock of a store? Enter ALL or ONE.
all
You have entered a wrong command. Please enter again.
ONE
If you want to see Online Store stock, enter 0. If you want to see a branch's stock, enter their branch number.
1
Office_Chair 1 Black      3
Office_Chair 1 White      3
Office_Chair 1 Red        3
Office_Chair 1 Blue       3
Office_Chair 1 Green      3
Office_Chair 2 Black      3
Office_Chair 2 White      3
Office_Chair 2 Red        3
Office_Chair 2 Blue       3
Office_Chair 2 Green      3
Office_Chair 3 Black      3
Office_Chair 3 White      3
Office_Chair 3 Red        3
Office_Chair 3 Blue       3
Office_Chair 3 Green      3
Office_Chair 4 Black      3
Office_Chair 4 White      3
Office_Chair 4 Red        3
Office_Chair 4 Blue       3
Office_Chair 4 Green      3
Office_Chair 5 Black      3
Office_Chair 5 White      3
Office_Chair 5 Red        3
```

#### 24-) Using PASTOrders command with a wrong customer number

```
PASTOrders
Enter the customer no of the customer you want to see their past orders.
2
The Customer's Past Orders:
Office_Chair 3 Blue
Meeting_Table 6 Black
Enter command:
```

#### 25-) Using SUPPLYProducts order when there is nothing to be supplied.

```
Enter command:
SUPPLYProducts
There are no products that are needed to be supplied.
Enter command:
```

#### 26-) Using ADDProduct with wrong product informations to a branch then fixing it

```
Enter command:
ADDProduct
Enter the type of the product you want to add:
Notebook
Enter the model of the product you want to add:
5
Enter the colour of the product you want to add:
Green
Enter how many you want to add:
3
Enter the store number you want to add. For Online Store enter 0.
4
InvalidType: This store does not produce Notebook.Please enter the type again.
Office_Desk
InvalidColour: Office_Desk cannot be Green.Please enter the colour again.
Blue
The product has been successfully added.
Enter command:
```

#### 27-) Using ADDProduct with wrong product informations to Online Store then fixing it

```
Enter command:
ADDProduct
Enter the type of the product you want to add:
Green
Enter the model of the product you want to add:
19
Enter the colour of the product you want to add:
Black
Enter how many you want to add:
5
Enter the store number you want to add. For Online Store enter 0.
0
InvalidType: This store does not produce Green.Please enter the type again.
Office_Chair
InvalidModel: There is no Model 19 for Office_Chair.Please enter the model number again.
2
The product has been successfully added.
Enter command:
```

#### 28-) Using REMOVEProduct command successfully.

```
Enter command:
REMOVEProduct
Enter the type of the product you want to remove:
Office_Chair
Enter the model of the product you want to remove:
3
Enter the colour of the product you want to remove:
Red
Enter how many you want to remove:
1
Enter the store number you want to add. For Online Store enter 0.
5
The product has been successfully removed.
Enter command:
```

#### 29-) Trying to remove a prpduct by some amount that would make the stock of it negative



```

Enter command:
REMOVEProduct
Enter the type of the product you want to remove:
Bookcase
Enter the model of the product you want to remove:
12
Enter the colour of the product you want to remove:
Black
Enter how many you want to remove:
6
Enter the store number you want to add. For Online Store enter 0.
5
InvalidAmount: There cannot be negative amount of products.cse312@ubuntu:~/Documents$

```

### 30-) Using ARRANGESales command successfully

```

Enter command:
ARRANGESales
Product has been added to the Supply List to inform the manager: Office_Cabinet 12 Black
All sales have been successfully arranged.
Enter command:

```

### 31-) Using MAKESale command for an existing customer when the product is not in the stock

```

Enter command:
MAKESales
You have entered an invalid command. Please enter again.
MAKESale
Enter the email of the customer who is shopping
emily@gmail.com
Enter the type of the product customer wants to buy:
Office_Cabinet
Enter the model of the product customer wants to buy:
12
Enter the colour of the product customer wants to buy:
Black
Enter the branch number customer wants to buy from.
1
Product has been added to the Supply List to inform the manager: Office_Cabinet 12 Black
You have successfully done your order from Branch 1. Your order will be delivered as soon as it gets into stock.
Enter command:

```

### 32-) Using MAKESale command for a new customer firstly from a branch that does not exist, then from a branch which has the product in the stock.

```

Enter command:
MAKESale
Enter the email of the customer who is shopping
geraldt@gmail.com
Enter the name of the customer who is shopping
Grealt
Enter the surname of the customer who is shopping
Rivia
Enter the password of the customer who is shopping
3422
The new customer is successfully added.
The customer number is: 5
Enter the type of the product customer wants to buy:
Bookcase
Enter the model of the product customer wants to buy:
4
Enter the colour of the product customer wants to buy:
Black
Enter the branch number customer wants to buy from.
0
There is no branch with this number, please enter again.
5
You have successfully done your purchase from Branch 5.
Enter command:

```

### 33-) Using CHECK function when there are products that need to be supplied

```
Enter command:
CHECK
These products should be supplied:
Office_Cabinet 12 Black to Store 1
Office_Cabinet 12 Black to Store 1
Enter command:
```

### 34-) Using REMOVEBranch function with a wrong branch number then fixing it

```
Enter command:
REMOVEBranch
Enter the branch number you want to remove.
6
InvalidIndex: There is no branch with this number
Please enter again.
4
The branch is successfully removed.
Enter command:
```

### 35-) Using SUPPLYProducts command when there are product that need to be supplied.

```
Enter command:
SUPPLYProducts
All products are successfully supplied.
Enter command:
```

### 36-) Using ARRANGESales after supplying the products.

```
Enter command:
ARRANGESales
All sales have been successfully arranged.
Enter command:
```

### 37-) Trying to put a string input for model instead of integer.

```
Enter the model of the product you want to add:
ModelA
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:913)
    at java.util.Scanner.next(Scanner.java:1534)
    at java.util.Scanner.nextInt(Scanner.java:2164)
    at java.util.Scanner.nextInt(Scanner.java:2123)
    at AutomationSystem.main(AutomationSystem.java:184)
cse312@ubuntu:~/Documents$
```

### 38-) Trying to put spaces in string inputs

```
Enter command:
ADDEmployee
Enter the name of the employee you want to add:
keira@gmail.com
Enter the surname of the employee you want to add:
Clark Simpson
Enter the email of the employee you want to add:
23123
Enter the password of the employee you want to add:
23123
New employee is successfully added.
Enter command:
```

### 39-) Using QUIT command

```
Enter command:
QUIT
cse312@ubuntu:~/Documents$
```