

GIT Department of Computer Engineering

CSE 222/505 - Spring 2020

Homework 1 Report

ŞEYDA ÖZER

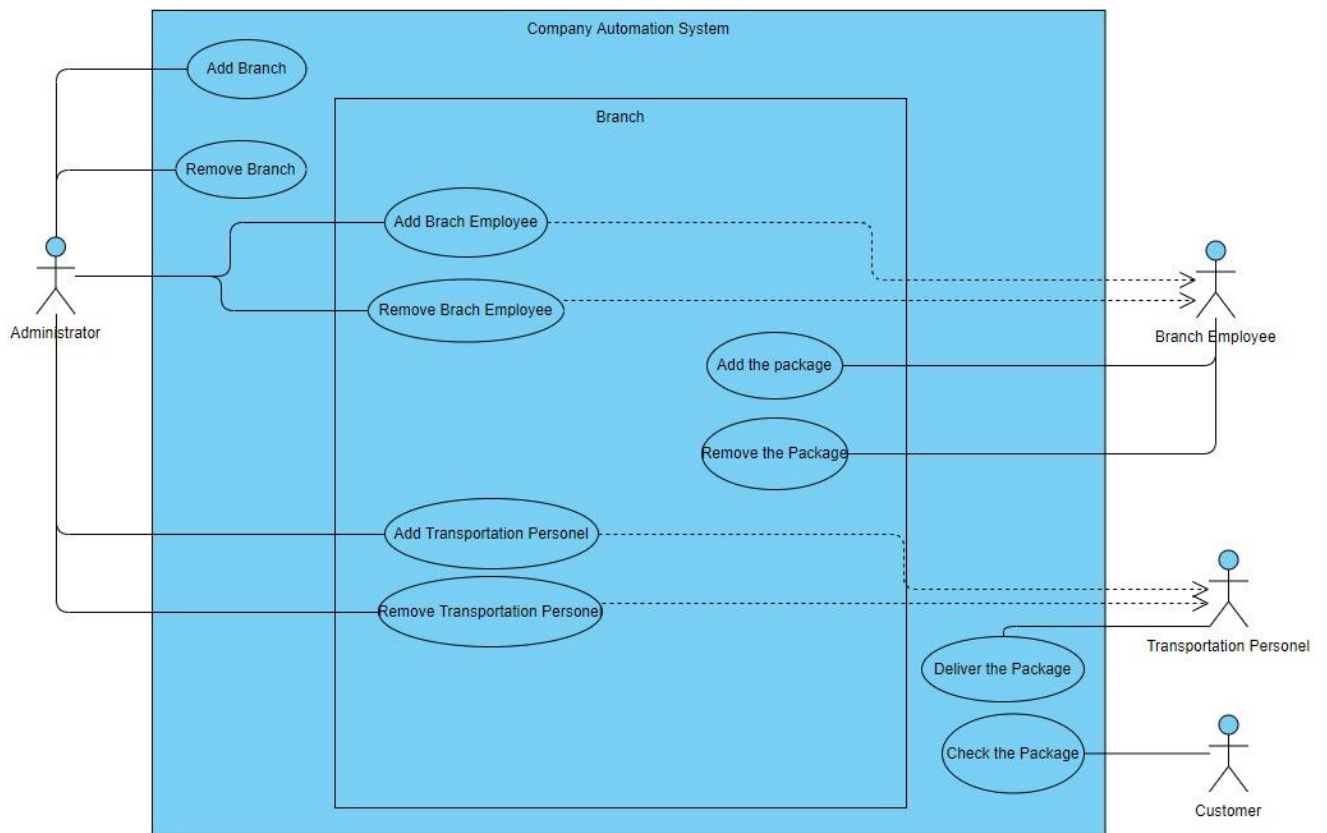
171044023

SYSTEM REQUIREMENTS

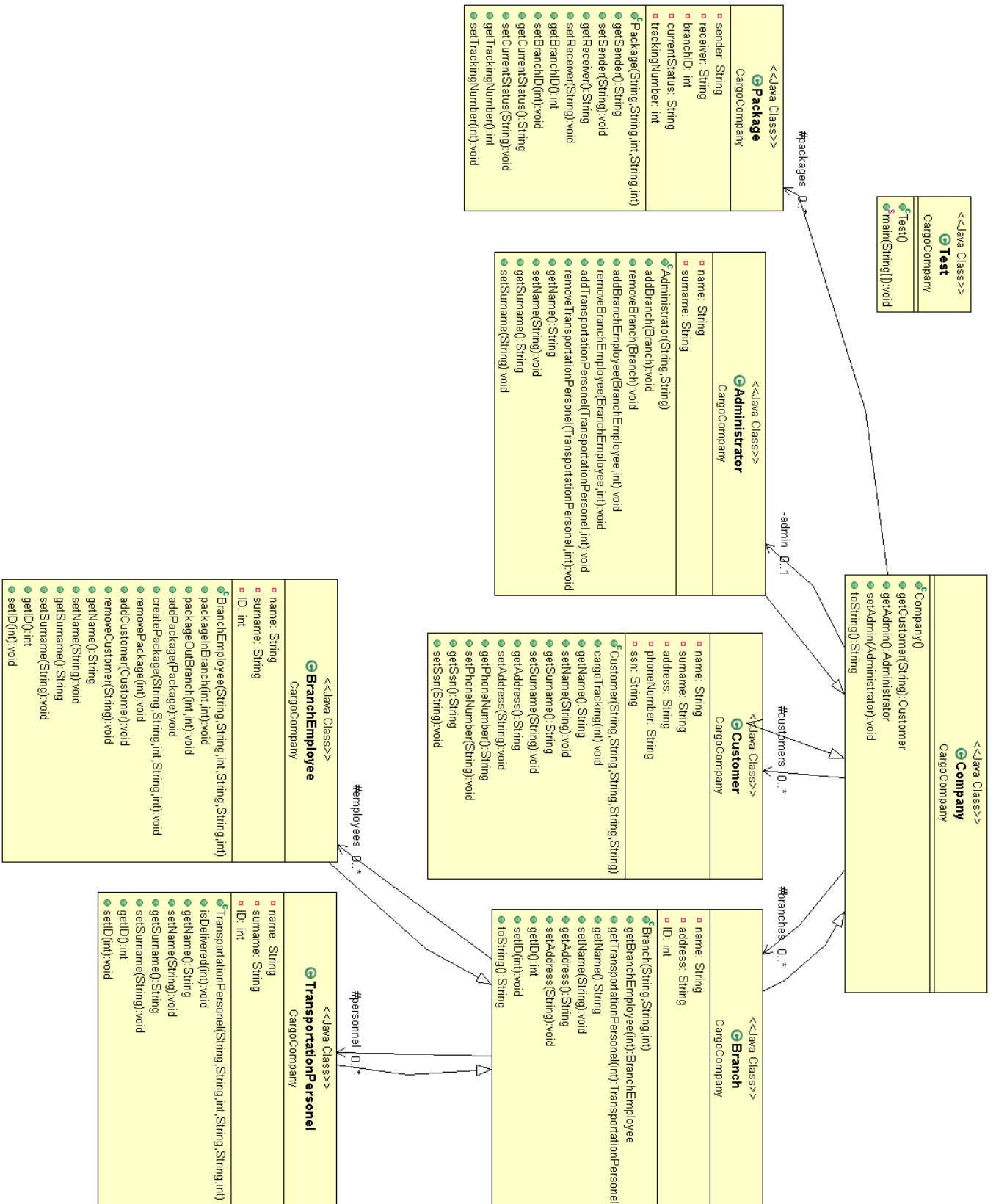
In this assignment, we design and implement the automation system for a cargo company. Users in the automation system are administrators, branch employees, transportation personnel and customers. These administrators can add and remove branches, branch employees, transportation personnel. When customers entered the system with tracking number, only they see the receiver, sender and current status of the package. This package can be added to the system by branch employee and can be removed. When a package is delivered, the transportation personnel enters the system and updates its current status.

There is a need for a class for the company's data. This class has information about packages, customers, branches. These branches in the company have branch employees and transportation personnel. Administrators manage these branches and their employees. So, there is a class needed. This administrator class should include administrator's individual data, and some methods that add and remove the branches, branch employees and transportation personnel. Since branches need to keep branch employees' and transportation personnel's data, a class called branch is needed. A branch employee can add and remove packages and customers so, a class is needed for branch employee. In this class, branch employee adds package to company and enters the system its information. There is also a need for a class for transportation personnel's system update task and his/her personal information. Customers should have a social security number because it can be confusing. There is a need for a class for customer's social security and their other information and this class has a method that takes a tracking number and gives information of the package. It seems that the information of the package should be kept in a class that is called package. In this case, this automation system has a lot of classes and it has connections between these classes.

USE CASE DIAGRAMS



CLASS DIAGRAMS



PROBLEM SOLUTION APPROACH

Firstly, I created a Company class. Then I decide to put the branch array, customer array, package array and a administrator into Company class. I needed the Branch class, Customer class, Package class and Administrator class for these data. I created the first Package class among these classes. I set the fields for Package class as a sender, a receiver, a ID of the branch where package is located, current status and tracking number. Then I started creating Branch class. I have determined the fields of Branch class as the name, address, ID, branch employee array and transportation personnel array. So I had to create two classes again. These classes are BranchEmployee and TransportationPersonel. in BranchEmployee class and TranportationPersonel class, I kept the name, surname and ID of a branch employee. Later I created Administrator class. Fields of the Administrator class are the name and the surname. I created methods for adding and removing a branch to the array of Branch in Company class. Then I created methods for adding and removing a branch employee to the array of BranchEmployee in a Branch(one of branches in Company class). I did also same operation for transportation personel. I added methods like addingCustomer and removingCustomer to the BranchEmployee class for customer related tasks. I added also same methods for packages tasks. I also added methods for update current status of the package about this package arriving at the branch or leaving the branch. I created a method for TransportationPersonel class and this method updates the current status of the package, when the package is delivered. I used keyword static for the arrays of Company class because I want all Company class objects to use the same array. Then I created Customer class. I decided to fields of Customer class should be the name, surname, phone number, address and social security number. I created a method for Customer class. This method shows the information about the package with tracking number to customer. Finally, I added getters and setters to my classes.

TEST CASES

Test Case 1: adding the same branch in the company again.

Test Case 2: removing a branch that is not in the company.

Test Case 3: adding the same branch employee in the same branch again.

Test Case 4: removing a branch employee that is not in the branch.

Test Case 5: adding the same transportation personel in the same branch again.

Test Case 6: removing a transportation personel that is not in the branch.

Test Case 7: adding the same package in the company again.

Test Case 8: removing a package that is not in the company.

Test Case 9: adding the same customer in the company again.

Test Case 10: removing a customer that is not in the company.

Test Case 11: trying to look at the information of the package with an invalid tracking number.

Test Case 12: trying to enter the system with a wrong branch employee.

Test Case 13: trying to enter the system with a wrong transportation persone.

Test Case 14: trying to enter the system with a wrong customer.

RUNNING AND RESULTS

Test Case 1:

I added Gebze, Darica and Cayirova as a branch and I try to add Gebze again.

```
Branch b1 = new Branch("Gebze", "Kocaeli", 1);
Branch b2 = new Branch("Darica", "Kocaeli", 2);
Branch b3 = new Branch("Cayirova", "Kocaeli", 3);

company.getAdmin().addBranch(b1);
company.getAdmin().addBranch(b2);
company.getAdmin().addBranch(b3);
System.out.println(company);
company.getAdmin().addBranch(b1);
```

Gebze branch is added.

Darica branch is added.

Cayirova branch is added.

```
-----  
This company's branches(name):  
Gebze Darica Cayirova  
-----
```

Gebze branch is not added. Because it is already added.

Test Case 2:

I try to remove a branch two times.

```
company.getAdmin().removeBranch(b2);  
System.out.println(company);  
company.getAdmin().removeBranch(b2);
```

Darica branch is removed.

```
-----  
This company's branches(name):  
Gebze Cayirova  
-----
```

Darica branch is not removed because this branch is not found!

Test Case 3:

I added Ayse and Guler as a branch employee and I try to add Ayse again.

```
BranchEmployee e1 = new BranchEmployee("Gebze", "Kocaeli", b1.getID(), "Ayse", "Kaya", 1);  
BranchEmployee e2 = new BranchEmployee("Gebze", "Kocaeli", b1.getID(), "Guler", "Kaya", 2);
```

```
company.getAdmin().addBranchEmployee(e1, b1.getID());  
company.getAdmin().addBranchEmployee(e2, b1.getID());  
System.out.println(b1);  
company.getAdmin().addBranchEmployee(e1, b1.getID());
```

```
-----  
Cayirova branch's branch employees(name-surname):  
Fatma Kaya  
-----
```

Ayse Kaya is not added to branch employees. Because it is already added.

Test Case 4:

I try to remove a branch employee two times.

```
company.getAdmin().removeBranchEmployee(e2, b1.getID());
System.out.println(b1);
company.getAdmin().removeBranchEmployee(e2, b1.getID());
```

Guler Kaya is removed from branch employees.

```
-----
Gebze branch's branch employees(name-surname):
Ayse Kaya
-----
```

Guler Kaya is not removed from branch employees because this is not found.

Test Case 5:

I added Gamze and Hulya as a transportation personel and I try to add Gamze again.

```
TransportationPersonel p1 = new TransportationPersonel("Gebze", "Kocaeli", b1.getID(), "Gamze", "Celik", 1);
TransportationPersonel p2 = new TransportationPersonel("Gebze", "Kocaeli", b1.getID(), "Hulya", "Celik", 2);
```

```
company.getAdmin().addTransportationPersonel(p1, b1.getID());
company.getAdmin().addTransportationPersonel(p2, b1.getID());
System.out.println(b1);
company.getAdmin().addTransportationPersonel(p1, b1.getID());
```

Gamze Celik is added to transportation personnel in Gebze branch.

Hulya Celik is added to transportation personnel in Gebze branch.

```
-----
Gebze branch's transportation personnel(name-surname):
Gamze Celik, Hulya Celik
-----
```

Gamze Celik is not added to transportation personnel. Because it is already added.

Test Case 6:

I try to remove a transportation personel two times.

```
company.getAdmin().removeTransportationPersonel(p2, b1.getID());
System.out.println(b1);
company.getAdmin().removeTransportationPersonel(p2, b1.getID());
```

Hulya Celik is removed from transportation personnel.

```
-----
Gebze branch's transportation personnel(name-surname):
Gamze Celik
-----
```


Hulya Celik is not removed from transportation personnel because this is not found.

Test Case 7:

I added the packages with 1001, 1002 and 1003 tracking numbers and I try to add 1001 again.

```
Package shipment1 = new Package("Ali", "Ayse", b1.getID(), b1.getName(), 1000);
Package shipment2 = new Package("Fatma", "Hasan", b1.getID(), b1.getName(), 1001);
Package shipment3 = new Package("Mehmet", "Cemal", b1.getID(), b1.getName(), 1002);
b1.getBranchEmployee(e1.getID()).addPackage(shipment1);
b1.getBranchEmployee(e1.getID()).addPackage(shipment2);
b1.getBranchEmployee(e1.getID()).addPackage(shipment3);
System.out.println(company);
b1.getBranchEmployee(e1.getID()).addPackage(shipment2);
```

package with 1000 tracking number is added.

package with 1001 tracking number is added.

package with 1002 tracking number is added.

```
-----
This company's packages(tracking number):
1000 1001 1002
-----
```

package with 1001 tracking number is not added because it is already added.

Test Case 8:

I try to remove a package two times.

```
b1.getBranchEmployee(e1.getID()).removePackage(shipment2.getTrackingNumber());
System.out.println(company);
b1.getBranchEmployee(e1.getID()).removePackage(shipment2.getTrackingNumber());
```

package with 1001 tracking number is removed.

```
-----
This company's packages(tracking number):
1000 1002
-----
```

package with 1001 tracking number is not removed because it is not found.

Test Case 9:

I added Mustafa, Alihan and Yasemin as a customers and I try to add Yasemin again.

```
Customer c1 = new Customer("Mustafa", "Demir", "Atalar", "05551112323", "1111111111");
Customer c2 = new Customer("Alihan", "Kale", "Maltepe", "05552222323", "2222222222");
Customer c3 = new Customer("Yasemin", "Gul", "Kadikoy", "05553332323", "3333333333");
b1.getBranchEmployee(e1.getID()).addCustomer(c1);
b1.getBranchEmployee(e1.getID()).addCustomer(c2);
b1.getBranchEmployee(e1.getID()).addCustomer(c3);
System.out.println(company);
b1.getBranchEmployee(e1.getID()).addCustomer(c2);
```

Customer named Mustafa Demir is added.

Customer named Alihan Kale is added.

Customer named Yasemin Gul is added.

```
-----
This company's customers(name-surname):
Mustafa Demir, Alihan Kale, Yasemin Gul
-----
```

Customer named Alihan Kale is not added because it is already added.

Test Case 10:

I try to remove a customer two times.

```
b1.getBranchEmployee(e1.getID()).removePackage(shipment2.getTrackingNumber());
System.out.println(company);
b1.getBranchEmployee(e1.getID()).removePackage(shipment2.getTrackingNumber());
```

Customer named Yasemin Gul with 3333333333 SSN is removed.

```
-----
This company's customers(name-surname):
Mustafa Demir, Alihan Kale
-----
```

Customer with 3333333333 SSN is not removed because it is not found.

Test Case 11:

Entering the valid and invalid tracking number

```
company.getCustomer(c1.getSsn()).cargoTracking(shipment1.getTrackingNumber());
company.getCustomer(c1.getSsn()).cargoTracking(shipment2.getTrackingNumber());
```

```
-----
Informations about the package with 1000 tracking number:
Sender: Ali
Receiver: Ayse
Current Status: in branch Gebze - 1
-----
```

```
-----
This package with 1001 tracking number is not found!
-----
```

Test Case 12:

Entering the system with a wrong branch employee

Branch employee e2 is removed before this operation.

```
b1.getBranchEmployee(e2.getID()).packageOutBranch(shipment1.getTrackingNumber(), b1.getID());
```

[java.lang.NullPointerException](#): This branch has not the employee.

Test Case 13:

Entering the system with a wrong transportation personnel

Transportation personnel p2 is removed before this operation.

```
b3.getTransportationPersonel(p2.getID()).isDelivered(shipment1.getTrackingNumber());
```

[java.lang.NullPointerException](#): This branch has not the personnel.

Test Case 14:

Entering the system with a wrong customer

Customer c3 is removed before this operation.

```
company.getCustomer(c3.getSsn()).cargoTracking(shipment1.getTrackingNumber());
```

[java.lang.NullPointerException](#): This company has not the customer.