

System Requirements:

Functional Requirements:

- There is 4 Users in my Cargo Company Design.
- First user is Administrators
 - Admins can adding and removing Branches to the system.
 - Admins can adding and removing Branch Employees to the system.
 - Admins can adding and removing Transportation Personnel to the system.
- Second user is Branch Employees
 - Branch Employees can enter and remove information about package
 - Branch Employees can add or remove customers to the system .
- Third user is Transportation Personnel
 - Transportation personnel when package is delivered update the package status
- Fourth user is Customers
 - All Customers have a tracking number .
 - With this number if they have a package they can see its current statu in system.
- Also There are Branch and ShipmentSent class too.
- Administrator, BranchEmployee, Transportation Personel and Customers inherited from UserInterface.
- UserInterface has LoginSystem and toString methods. This methods override for subclasses.
- In system there is a error checking system. If a user try to call his methods without login system(ex admin.LoginSystem()) an exception will be thrown and program finish with an error message.

Non-Functional Requirements:

- This Program runs at JDK 13
- IntelliJ IDE is used as IDE

Rapor:

In my Design I created a User class that is superclass for Administrators, BranchEmployees, Transportation personnels and Customers.I define User class as Abstract class because I don't want to have a User class object. And I define a System class .My Administrators class object Admin1 can adding and removing branches, Branch Employees and Transportation Personnels to the system. Only Administators can do these jobs because only admin has methods for adding and removing branches, Branch emps and transportation personnel to the system. Like Adminstrator class I have Branch employee, Transportation personnel and Customer class too Branch Employee can enter and remove Information about package and add or remove customers in system with its methods. Transportation Personnel can update package statu with its methods.

Customer class has a data field called Tracking number and a method for check there is a package that match with this tracking number and see its status.

Also I have Branch and ShipmentSent class too.

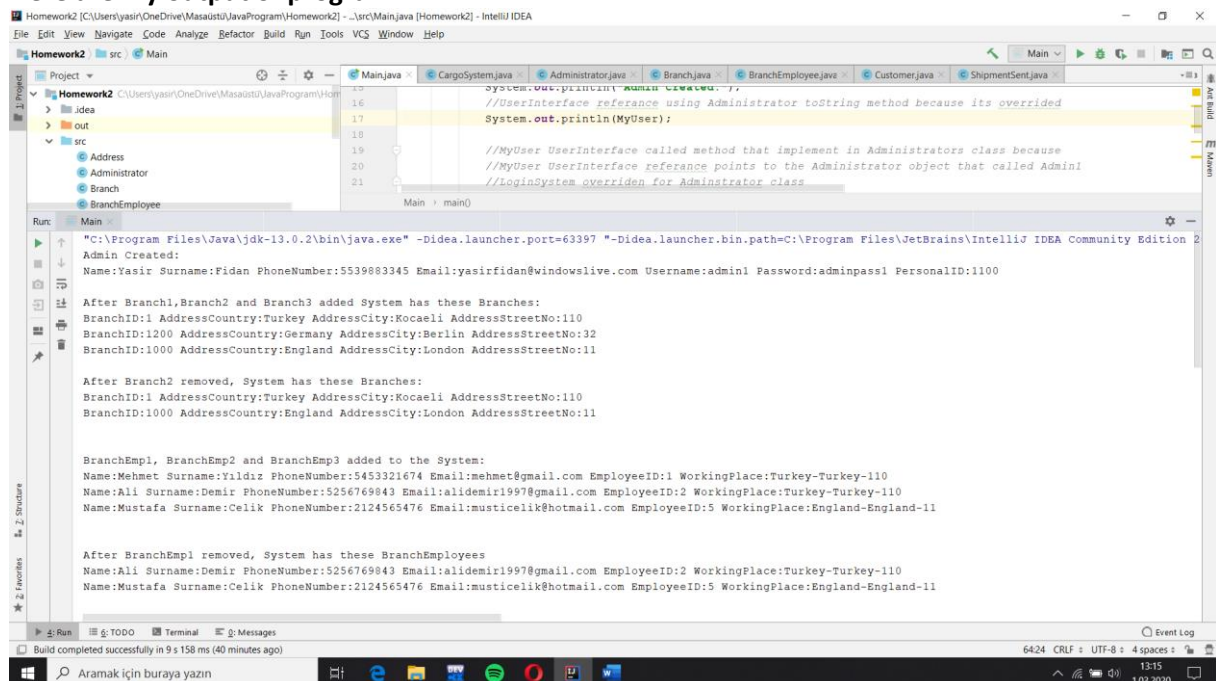
I keep this Administrators, Branch employee, Transportation personnel and Customers in System class with array.

ShipmentSent same as a package.

After Homework 2 I create a UserInterface and inherit User abstract class and Administrators, BranchEmployee, Transportation Personel and Customers inherited from User class. I declare UserInterface reference and points users. With this reference I call overridden method LoginSystem. With this I use polymorphism. UserInterface has LoginSystem and toString abstract methods. I override this methods in Administrator, Branch Employee, Transportation personel and Customer classes.

I test all methods in main and print the results.

Here are my output of program:



```
Homework2 [C:\Users\yasir\OneDrive\Masaüstü\JavaProgram\Homework2] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Homework2 src Main
Project: Homework2
src
  Address
  Administrator
  Branch
  BranchEmployee
  CargoSystem
  Customer
  ShipmentSent
  UserInterface
  User

Main.java
15 //System.out.println("Admin Created.");
16 //UserInterface reference using Administrator toString method because its overridden
17 System.out.println(MyUser);
18
19 //MyUser UserInterface called method that implement in Administrators class because
20 //MyUser UserInterface reference points to the Administrator object that called Admin
21 //LoginSystem overridden for Administrator class

Run: Main
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" -Didea.launcher.port=63397 "-Didea.launcher.bin.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2
Admin Created:
Name:Yasir Surname:Fidan PhoneNumber:5539883345 Email:yasirfidan@windowslive.com Username:admin1 Password:adminpass1 PersonalID:1100

After Branch1,Branch2 and Branch3 added System has these Branches:
BranchID:1 AddressCountry:Turkey AddressCity:Kocaeli AddressStreetNo:110
BranchID:1200 AddressCountry:Germany AddressCity:Berlin AddressStreetNo:32
BranchID:1000 AddressCountry:England AddressCity:London AddressStreetNo:11

After Branch2 removed, System has these Branches:
BranchID:1 AddressCountry:Turkey AddressCity:Kocaeli AddressStreetNo:110
BranchID:1000 AddressCountry:England AddressCity:London AddressStreetNo:11

BranchEmp1, BranchEmp2 and BranchEmp3 added to the System:
Name:Mehmet Surname:Yildiz PhoneNumber:5453321674 Email:mehmet@gmail.com EmployeeID:1 WorkingPlace:Turkey-Turkey-110
Name:Ali Surname:Demir PhoneNumber:5256769843 Email:alidemir1997@gmail.com EmployeeID:2 WorkingPlace:Turkey-Turkey-110
Name:Mustafa Surname:Celik PhoneNumber:2124565476 Email:musticelik@hotmail.com EmployeeID:5 WorkingPlace:England-England-11

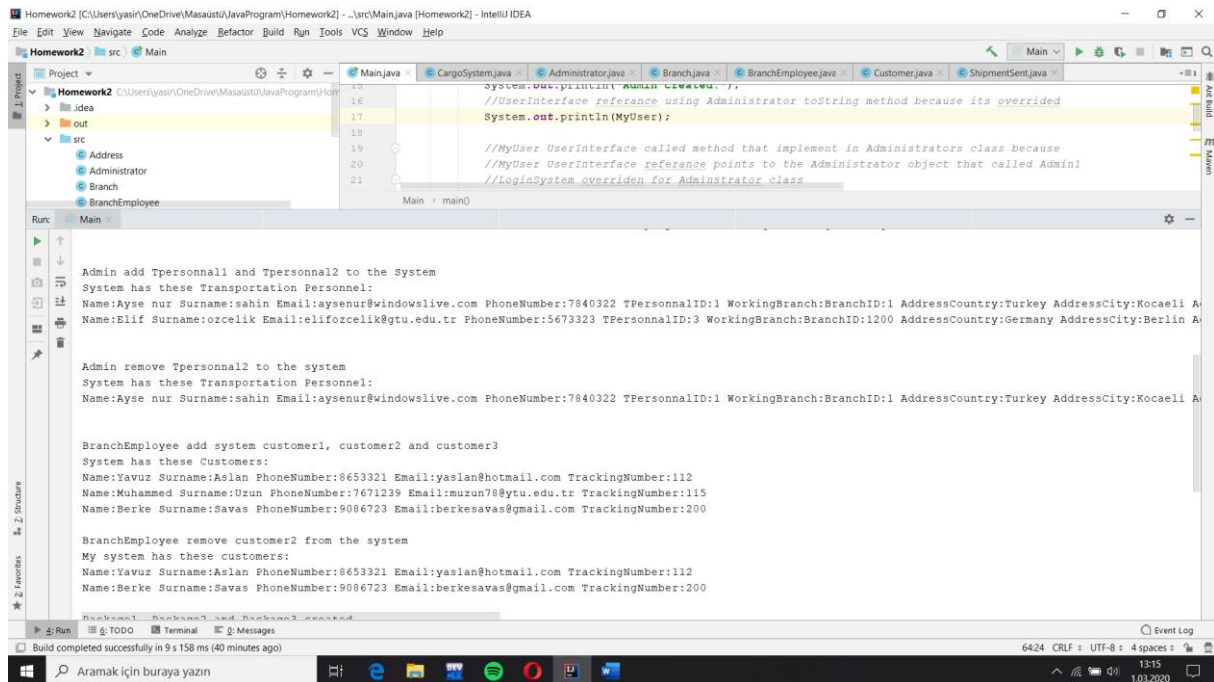
After BranchEmp1 removed, System has these BranchEmployees
Name:Ali Surname:Demir PhoneNumber:5256769843 Email:alidemir1997@gmail.com EmployeeID:2 WorkingPlace:Turkey-Turkey-110
Name:Mustafa Surname:Celik PhoneNumber:2124565476 Email:musticelik@hotmail.com EmployeeID:5 WorkingPlace:England-England-11

Build completed successfully in 9 s 158 ms (40 minutes ago)
6424 CRLF UTF-8 4 spaces
13:15
1.03.2020
```

Firstly, I created an administration object called Admin1 and prints its data fields to the screen. Then I created 3 Branches objects called Branch1,Branch2,Branch3. Then with Admin1 I added Branch1 to the system and prints all Branches that System has. After that I added Branch2 and Branch3 objects too via to admin and prints all Branches that my system has. Then admin remove Branch2 with Admin.removeBranchToSystem method so there is no Branch2 in my system anymore. Then i print branches that system has to Show branch2 removed.

After that, I created BranchEmployee object called BranchEmp1, BranchEmp2, BranchEmp3. Then again with admin1 object I add this employees to the system and print the all BranchEmployees that system has. Then with admin1 I removed BranchEmp2 to the system and prints all BranchEmployee objects that system has to Show BranchEmp2 removed successfully.

Again, I created Transportation Personnel object called TrasportationPersonnel1 and TransportationPersonnel2, then with admin1 I added TransportationPersonnel1 to the system and prints all TransportationPersonnel object that system has.



After adding TransportationPersonnel1 object to the system, I added TransportationPersonnel2 object to the system too and prints all TransportationPersonnel objects that system has. So it prints TransportationPersonnel1 and TransportationPersonnel2. After that I removed TransportationPersonnel1 with admin then print all TransportationPersonnel system has so its only prints Transportation2 because there was only TransportationPersonnel2 in the system. So I test all methods that admin can do in system.

Secondly, I created 3 Customer for my system. And using my BranchEmployee I added this customers to the system and prints all customers that system has to the screen. Then BranchEmployee3 removed Customer2 in the system and again all print all customers to the system has. so it prints only customer1 and customer3 so my branch employees can add and remove customers to the system successfully.

After adding/removing customer I created 3 packages and prints its informations to the screen than via to BranchEmployees I changed its Informations and prints their Informations after the change. So as you can see BranchEmployees changed packages informations successfully

```

16 //System.out.println("Admin User Login.");
17 //UserInterface reference using Administrator toString method because its overridden
18 System.out.println(MyUser);
19
20 //MyUser UserInterface called method that implement in Administrators class because
21 //MyUser UserInterface reference points to the Administrator object that called Admin
22 //LoginSystem overridden for Administrator class

```

Run: Main

```

Package1, Package2 and Package3 created
First Packages Info
Package1 Info:unknown
Package2 Info:unknown
Package3 Info:Yolda

After BranchEmployees change the Informations of Packages:
Package1 Info:PackageOnTheRoad
Package2 Info:PackageOnTheRoad
Package3 Info:unknown

After Tpersonall deliver the package
Package2 Info:PackageDelivired

Customer1 try to check package1 with his Tracking Number:
This Package Sender Information:
Sender Name:Yavuz Sender Surname:Aslan
This Package Receiver Information:
Receiver Name:Berke Receiver Surname:Savas
Package Current Status:PackageOnTheRoad

Process finished with exit code 0

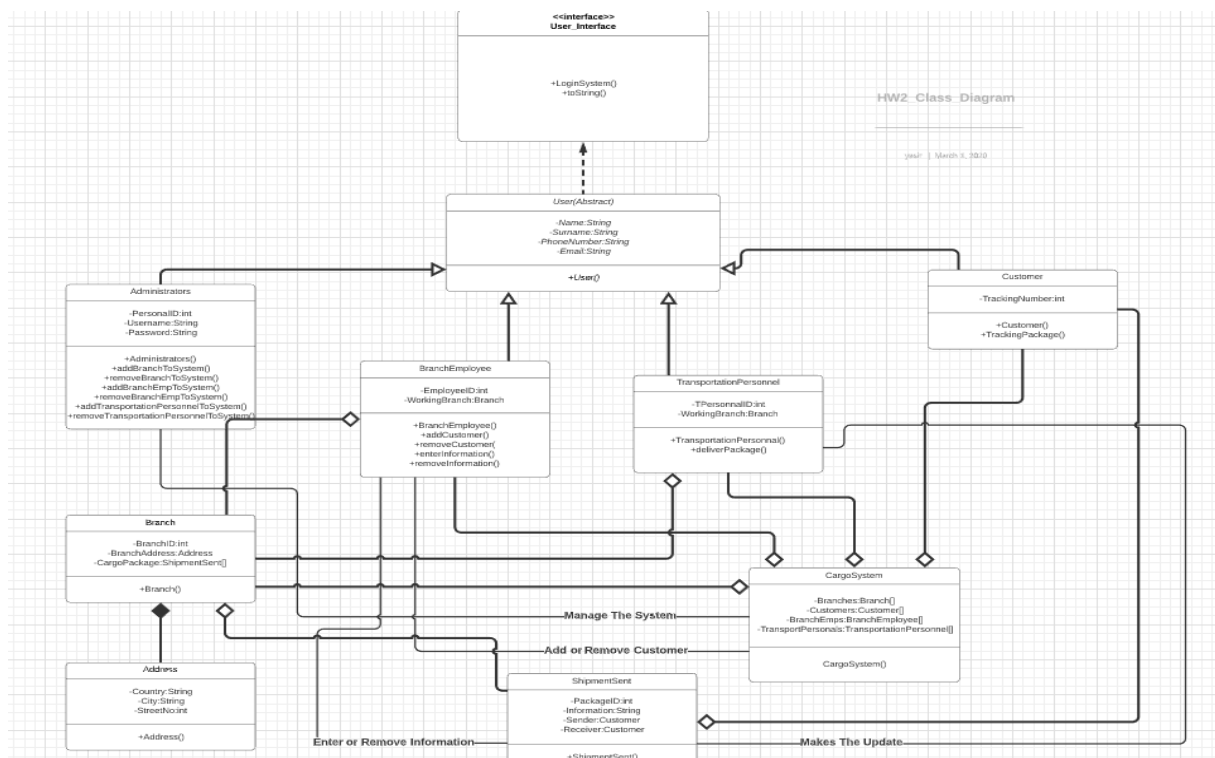
```

After that, When the package arrived to the customer my TransportationPersonnel change package Information to the "PackageDelivired".

And Customers can check their Package Sender informations, Recive Informations and package status With using their TrackingNumber. I write a method that take this Tracking number and check all Packages if its find prints these informations.

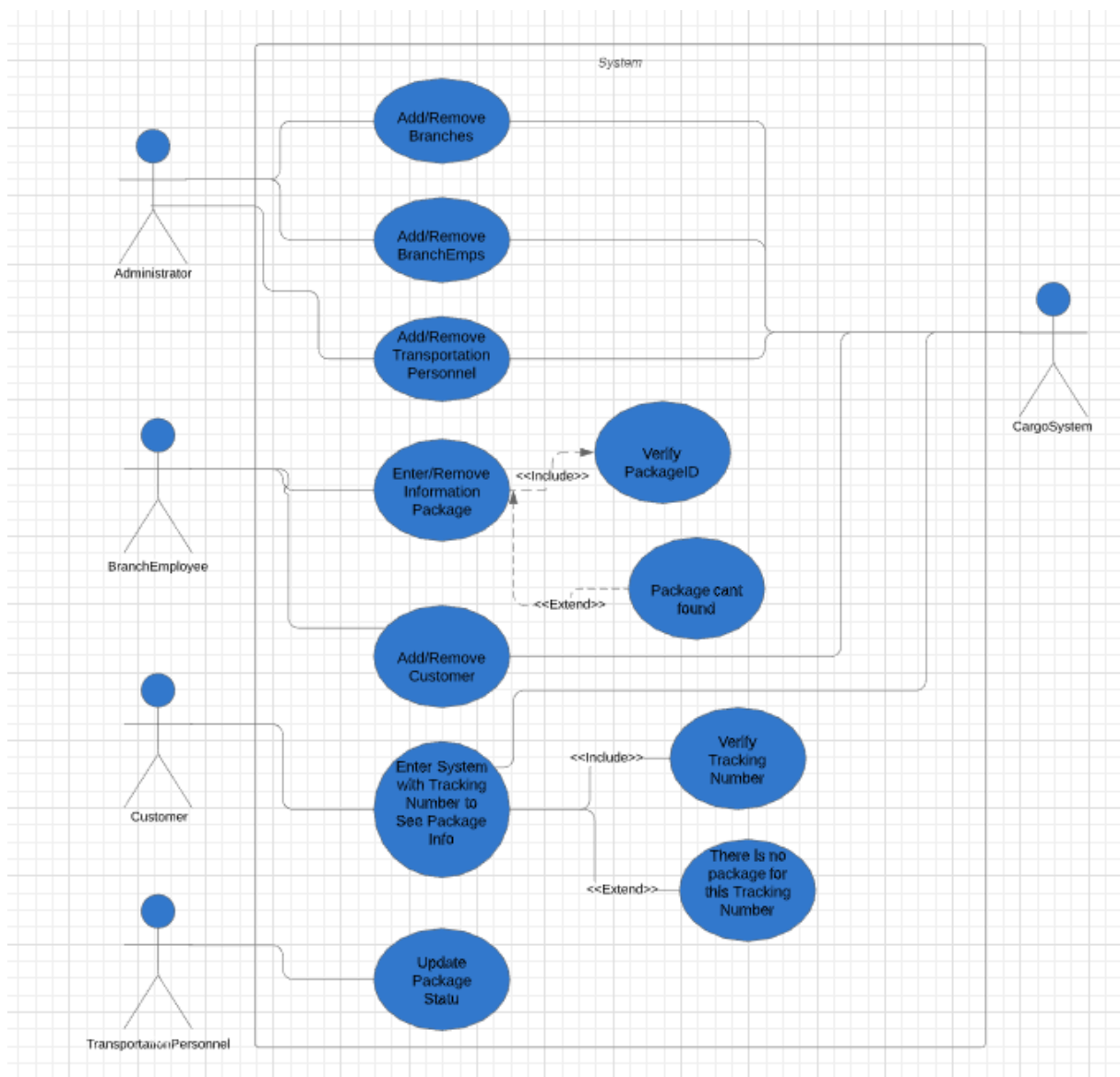
So in main I check all methods that users can.

Lets check my design with Class and Usecase diagrams:



I used LucidChart Diagram tools for drawing my class and use case diagrams.

In my design there are 10 class. I derived Administrator, BranchEmployee, TransportationPersonnel ,Customer and User class to the User Interface. These are the users of the system. For my system I define CargoSystem class. My CargoSystem class hold Branches, Customers , BranchEmployees and TransportationPersonnals with using an array. So there is "has a" relation between CargoSystem with Branches, BranchEmps, TransportationPersonnals and customers. Because of this I used aggregation arrows. There is also aggregation between ShipmentSent and Customers because in ShipmentSent class there are Sender and Receiver with Customer type. And I have a branch class and address class. All Branches have a BranchID , ShipmentsSend array and Address. There is a composition between Branch and Address because without Branch there will be no Address. And I hold my shipmentsSents(Packages) in Branche class.



In my use case diagram there are 4 primary user for system and a CargoSystem for respond to methods in system. An administrator can add/remove Branches, BranchEmployees and Transportation Personnals to the system. System actor has a line between these methods because its respond this methods.

Like administrators, Branch Employees can Enter/Remove information about package and add/remove Customers to the system. System has a line between add/remove customers because these customers hold in system so there must be a respond in system actor.

But there is no relationship between Enter/remove information about packages and system, because in my design packages keep in Branch class. But Enter/remove information has include and extend relation in my use case diagram. Verify PackageID <<include>> because system do this all time but Package cant found << extend >> because doesn't its work for all time enter/remove information called.

Customer actor has Access to Enter system with its tracking number to see package information. Also there is a <<include>> and <<extend>>relation. There is a <<include>> relation with Verify Tracking number because my system always try to verify tracking number when a customer try to enter system with its tracking number to see a package statu. But there is a <<extend>> relation with There is no package for this tracking number because doesnt nessessary customer get this error always.

Lastly, there is TransportationPersonnel user that can update package statu but there is no respond with the system because packages keep in Branches not in the CargoSystem class.

Error Handling:

```
13 UserInterface MyUser = Admin1;
14 System.out.println("Admin Created:");
15 //UserInterface reference using Administrator toString method because its overridden
16 System.out.println(MyUser);
17
18 //MyUser UserInterface called method that implement in Administrators class because
19 //MyUser UserInterface reference points to the Administrator object that called Admin1
20 //LoginSystem overridden for Administrator class
21 //MyUser.LoginSystem();
22 Admin1.addBranchToSystem(MyCargoSystem, Branch1);
23 Admin1.addBranchToSystem(MyCargoSystem, Branch2);
24 Admin1.addBranchToSystem(MyCargoSystem, Branch3);
25 System.out.println("\nAfter Branch1,Branch2 and Branch3 added System has these Branches:");
26
27 for (int i = 0; i < MyCargoSystem.getBranches().size(); i++) {
28 }
29 Main > main()
```

Run: Main
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" -Didea.launcher.port=64138 "-Didea.launcher.bin.path=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2
Admin Created:
Name:Yasir Surname:Fidan PhoneNumber:5539883345 Email:yasirfidan@windowslive.com Username:admin1 Password:adminpass1 PersonalID:1100
After Branch1,Branch2 and Branch3 added System has these Branches:
ERROR:For calling methods user must login to the System.
Exception Caught!

As you can see MyUser is UserInterface reference that point administrate object. But when I make a comment line of line 22 there will be an exception will thrown because of line 30 31 32. Because Admin try to add somethings in system without Login to the system.

Muhammed Yasir Fidan
161044056