GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 1 Report

Buğra Eren Yılmaz
1801042669

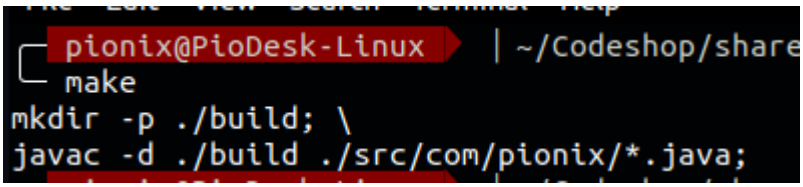
1. System Requirements

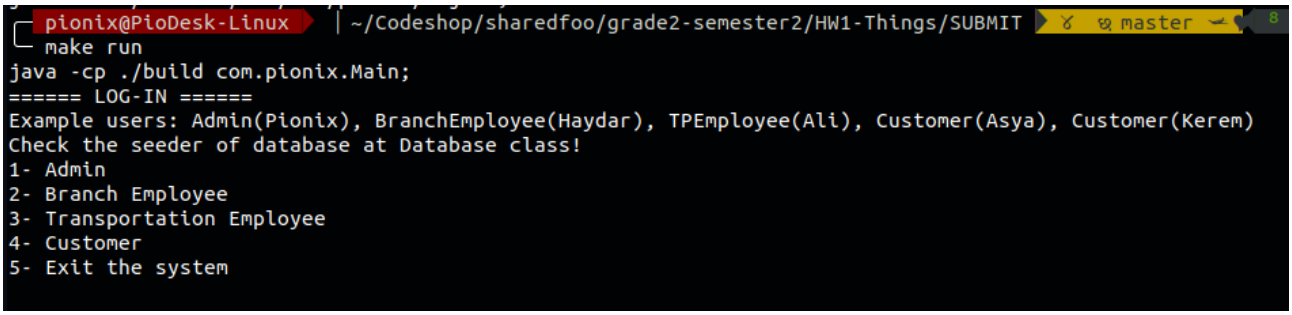
From the problem definition we see that we need a Log-in system for user authentication to permit user to manipulate parts of the “Database”.

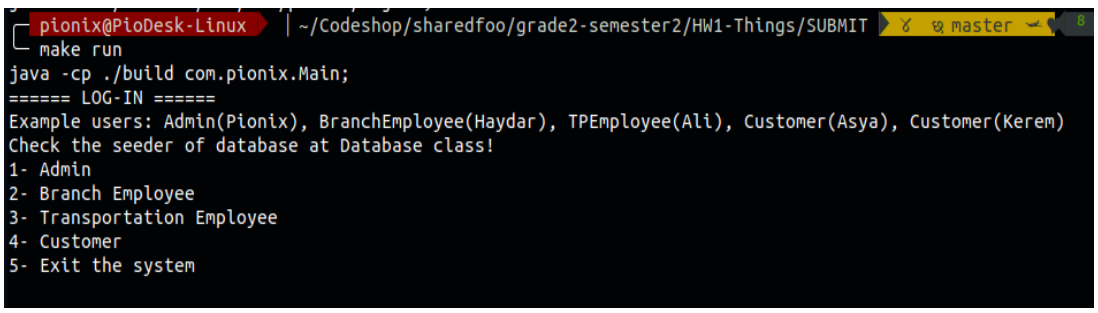
Here I refer the in-memory fake database system as Database. It utilizes the database object sets with the DbArray and DbController system which I designed.

The system is similar to MVC Architecture. I tried to implement it primarily using Dependency Injection techniques.

To compile the system use the provided Makefile.
- make all
- make javadoc
- make run



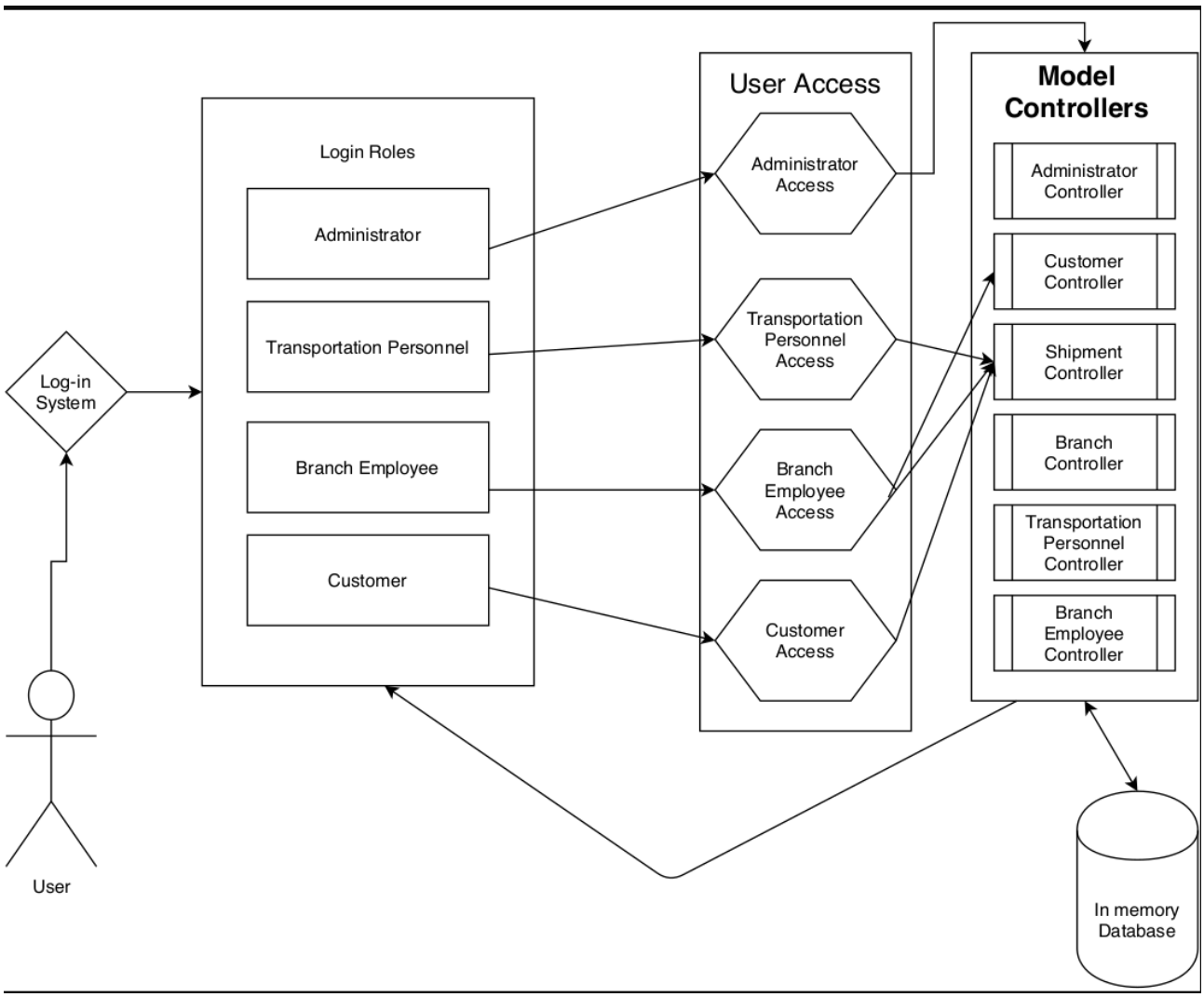




2. Use Case Diagrams

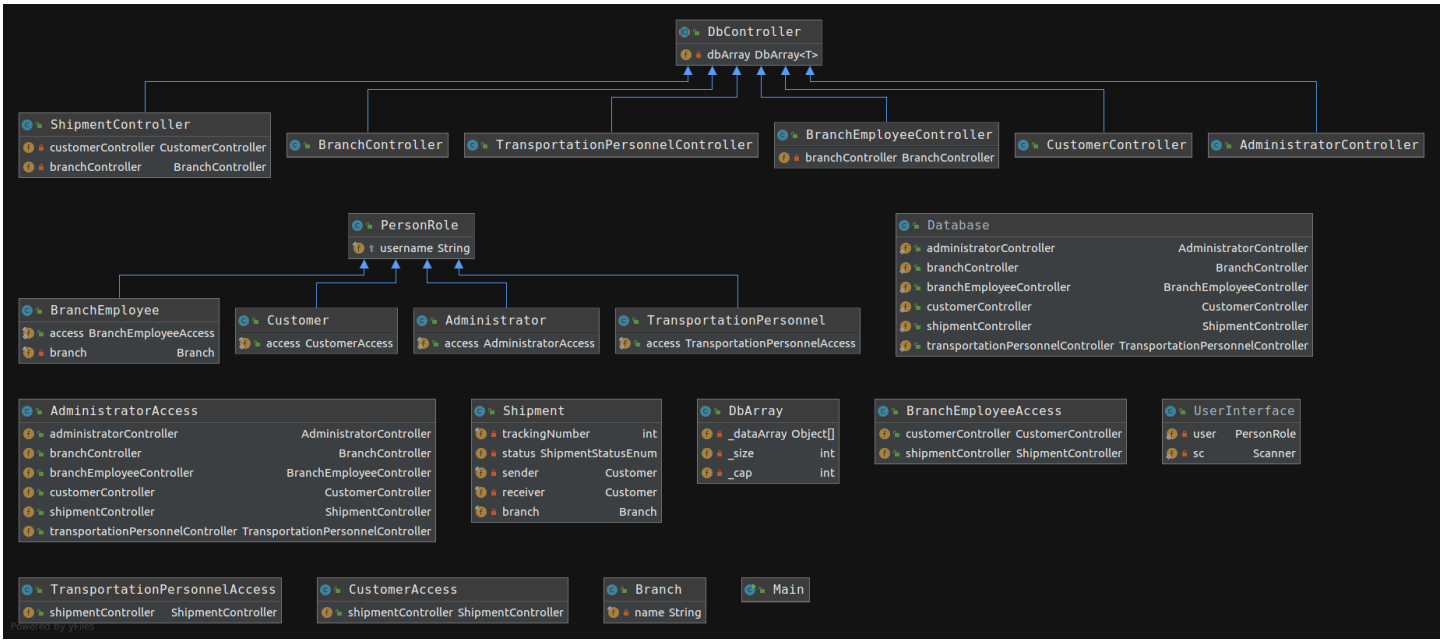
The user can login as 4 roles. Every role has different access levels to Model Controllers. Model Controllers control and manipulates the Models in Database.

For example a Branch Employee has User Access of a BranchEmployeeAccess. Which this access level has the access to ShipmentController and CustomerController. This way a BranchEmployee can manipulate shipments and customers in database.

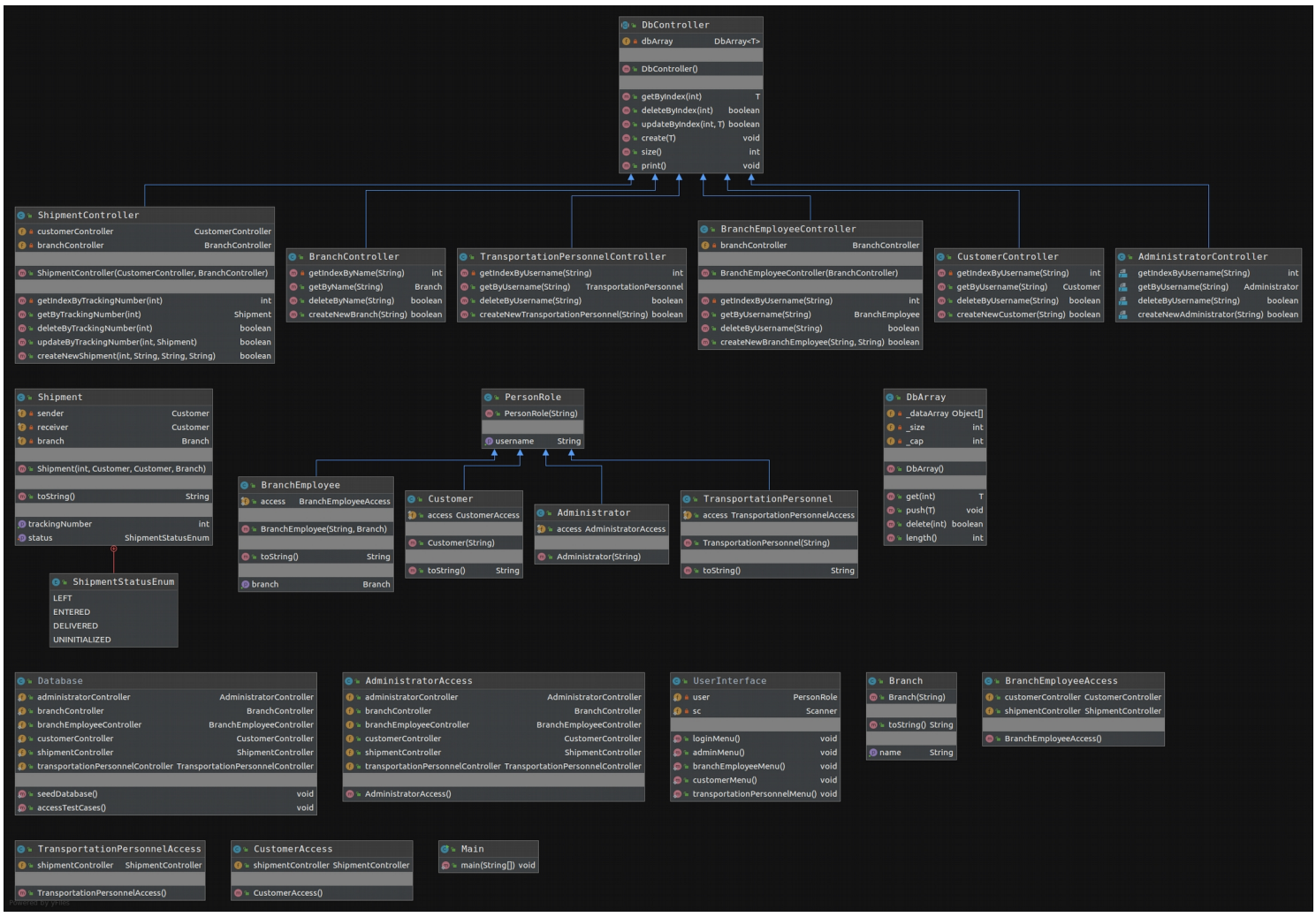


3. Class Diagrams

Minimal Class UML



Full Class UML



4. Problem Solution Approach

Problem was requiring me to somehow authorize users and restrict their access to the Database according to their roles. So I tried to make a system like MVC database architecture.

I started with building access units to database for every model. First I write every available model. Then made a Database Set class **DbArray** which holds the data as an array, can delete, add, get the data.

After that I made a Database controller class **DbController** which utilizes the given **DbArray** to manipulate the data inside it more.

So these classes are all abstract and we need the actual controllers so I started writing down every controller for every available model. Generally every model needs a specific way of get, set, delete and update.

For example **Shipment** model needs to be created with a tracking number, sender, receiver and an operating branch. So when creating this model I need to check if the sender is actually available in **Customers** database or the branch is an actual branch. These type of data checks are made in the controllers.

After writing controllers, I write access classes for roles. For instance Branch Employee person role needs to access Shipment and Customer controllers so I give that access to these access classes, BranchEmployeeAccess.

To Mimic a real life database I made the Database class which has the seedData function and Controllers as **Singleton classes.**

After that it was all about dependency injecting these access classes into Role classes. For example injectin BranchEmployeeAccess class into BranchEmployee class.

5. Test Cases

Test case for user access authorizations.

6. Running and Results

Admin menu example

```
└─ make run
java -cp ./build com.pionix.Main;
====== LOG-IN ======
Example users: Admin(Pionix), BranchEr
Check the seeder of database at Datab
1- Admin
2- Branch Employee
3- Transportation Employee
4- Customer
5- Exit the system
1
Username: Pionix
====== ADMIN ======
1- Add Branch
2- Remove Branch
3- List Branches
4- Add Branch Employee
5- Remove Branch Employee
6- List Branch Employees
7- Add Transportation Employee
8- Remove Transportation Personnel
9- List Transportation Personnel
10- List Shipments
11- List Customers
0- Logout
3
Branch name: Ankara
Branch name: Istanbul
Branch name: Izmir
Branch name: Adana
Operation was successful!
```

Adding branch employee to existing branch

```
====== ADMIN ======
1- Add Branch
2- Remove Branch
3- List Branches
4- Add Branch Employee
5- Remove Branch Employee
6- List Branch Employees
7- Add Transportation Employee
8- Remove Transportation Personnel
9- List Transportation Personnel
10- List Shipments
11- List Customers
0- Logout
4
Enter username of the new employee: Batu
Enter name of the branch to station employee: Giresun
Operation was successful!
```

Trying to add a duplicate branch

```
====== ADMIN ======
1- Add Branch
2- Remove Branch
3- List Branches
4- Add Branch Employee
5- Remove Branch Employee
6- List Branch Employees
7- Add Transportation Employee
8- Remove Transportation Personnel
9- List Transportation Personnel
10- List Shipments
11- List Customers
0- Logout
1
Enter name of the new branch: Ankara
Branch name duplicate when creating new branch!
Operation was NOT successful!
```

Adding new branch

```
====== ADMIN ======
1- Add Branch
2- Remove Branch
3- List Branches
4- Add Branch Employee
5- Remove Branch Employee
6- List Branch Employees
7- Add Transportation Employee
8- Remove Transportation Personnel
9- List Transportation Personnel
10- List Shipments
11- List Customers
0- Logout
1
Enter name of the new branch: Giresun
Operation was successful!

====== ADMIN ======
1- Add Branch
2- Remove Branch
3- List Branches
4- Add Branch Employee
5- Remove Branch Employee
6- List Branch Employees
7- Add Transportation Employee
8- Remove Transportation Personnel
9- List Transportation Personnel
10- List Shipments
11- List Customers
0- Logout
3
Branch name: Ankara
Branch name: Istanbul
Branch name: Izmir
Branch name: Adana
Branch name: Giresun
Operation was successful!
```

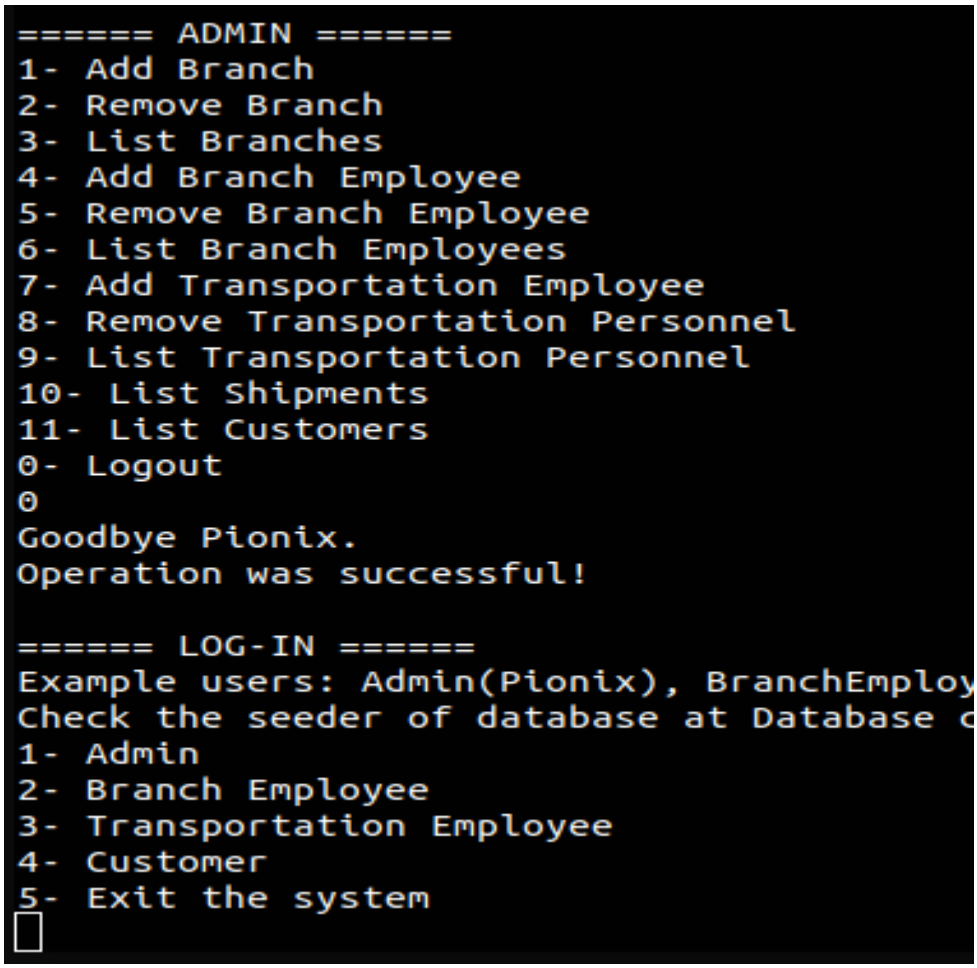
Trying to add a employee to non existing branch

```
====== ADMIN ======
1- Add Branch
2- Remove Branch
3- List Branches
4- Add Branch Employee
5- Remove Branch Employee
6- List Branch Employees
7- Add Transportation Employee
8- Remove Transportation Personnel
9- List Transportation Personnel
10- List Shipments
11- List Customers
0- Logout
4
Enter username of the new employee: JohnDoe
Enter name of the branch to station employee: NonExistin
Branch not found when creating new Branch Employee!
Operation was NOT successful!
```

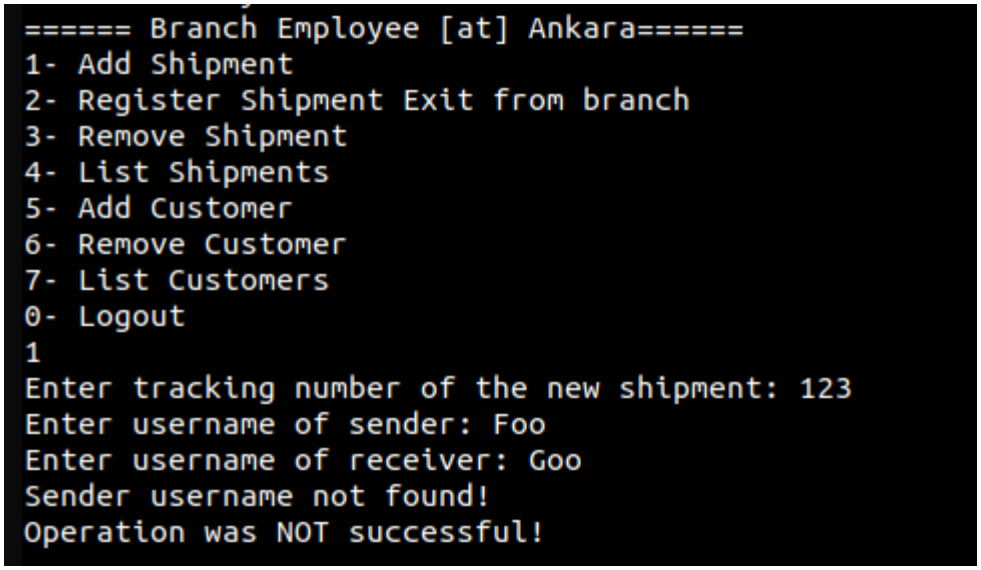
Listing shipments

```
====== ADMIN ======
1- Add Branch
2- Remove Branch
3- List Branches
4- Add Branch Employee
5- Remove Branch Employee
6- List Branch Employees
7- Add Transportation Employee
8- Remove Transportation Personnel
9- List Transportation Personnel
10- List Shipments
11- List Customers
0- Logout
10
100001: [Hakan] --> [Fatma]
        shipment entered the Ankara branch.
100002: [Asya] --> [Kerem]
        shipment entered the Izmir branch.
Operation was successful!
```

Logging out



Branch Employee Menu and trying to add shipment with non existing customers



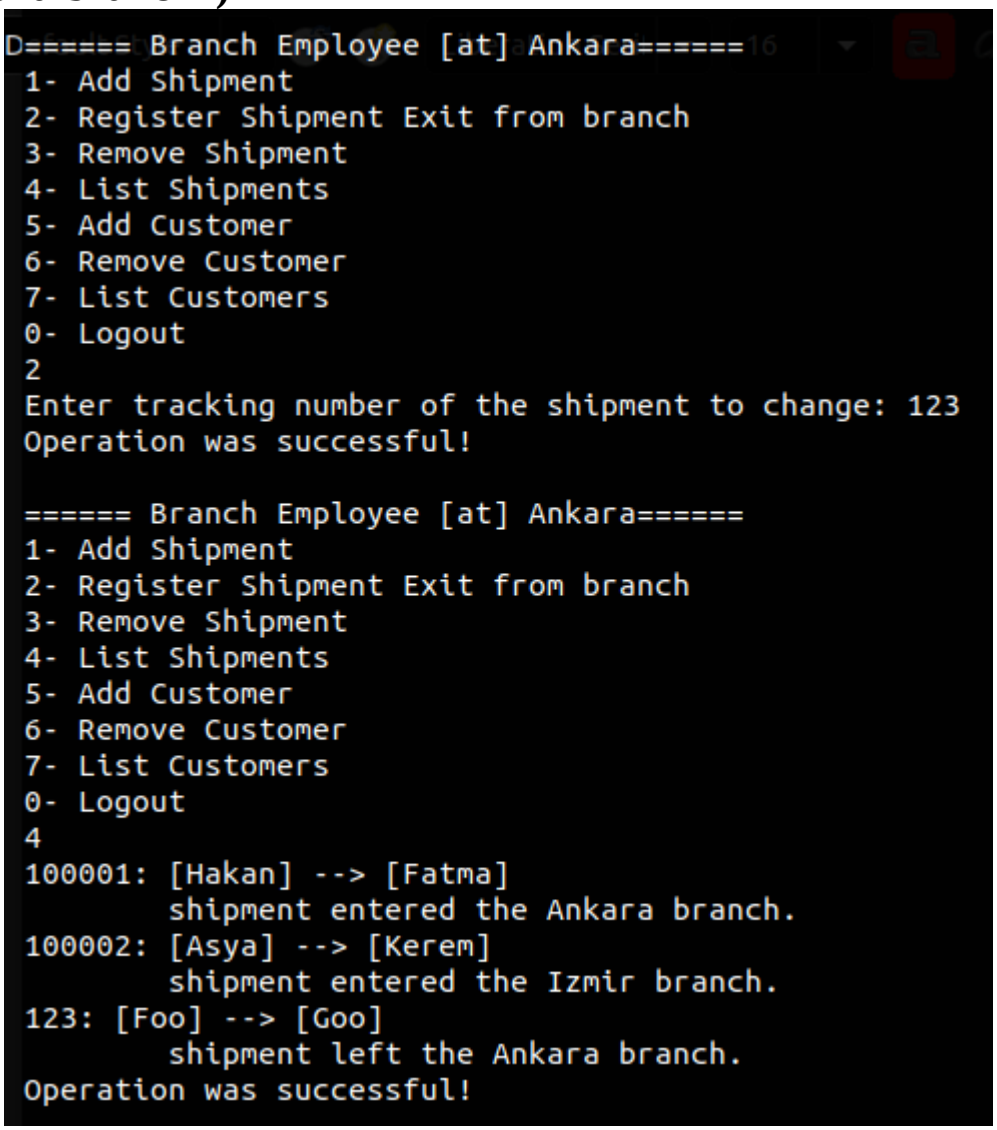
Adding the non existing customers

```
====== Branch Employee [at] Ankara======
1- Add Shipment
2- Register Shipment Exit from branch
3- Remove Shipment
4- List Shipments
5- Add Customer
6- Remove Customer
7- List Customers
0- Logout
5
Enter username of the new customer: Goo
Operation was successful!

====== Branch Employee [at] Ankara======
1- Add Shipment
2- Register Shipment Exit from branch
3- Remove Shipment
4- List Shipments
5- Add Customer
6- Remove Customer
7- List Customers
0- Logout
7
Customer name: Kerem
Customer name: Eren
Customer name: Asya
Customer name: Fatma
Customer name: Hakan
Customer name: Foo
Customer name: Goo
Operation was successful!
```

Creating a shipment (its status is “entered the ankara branch”)

```
====== Branch Employee [at] Ankara======
1- Add Shipment
2- Register Shipment Exit from branch
3- Remove Shipment
4- List Shipments
5- Add Customer
6- Remove Customer
7- List Customers
0- Logout
1
Enter tracking number of the new shipment: 123
Enter username of sender: Foo
Enter username of receiver: Goo
Operation was successful!

====== Branch Employee [at] Ankara======
1- Add Shipment
2- Register Shipment Exit from branch
3- Remove Shipment
4- List Shipments
5- Add Customer
6- Remove Customer
7- List Customers
0- Logout
4
100001: [Hakan] --> [Fatma]
        shipment entered the Ankara branch.
100002: [Asya] --> [Kerem]
        shipment entered the Izmir branch.
123: [Foo] --> [Goo]
        shipment entered the Ankara branch.
Operation was successful!
```

Sending shipment out to transportation personnel (now status is “left the ankara branch”)



Logout and login as transportation personnel and deliver cargo(now status is “delivered”)

```
6- Remove Customer
7- List Customers
0- Logout
0
Goodbye Haydar.
Operation was successful!

====== LOG-IN ======
Example users: Admin(Pionix), BranchEmployee(Haydar), TR
Check the seeder of database at Database class!
1- Admin
2- Branch Employee
3- Transportation Employee
4- Customer
5- Exit the system
3
Username: Ali
====== Transportation Personnel ======
1- Deliver Shipment
0- Logout
1
Enter tracking number of the shipment: 123
123: [Foo] --> [Goo]
        shipment delivered to the customer from  Ankara
Operation was successful!

====== Transportation Personnel ======
1- Deliver Shipment
0- Logout
```

Logout and login as customer and check the cargo status

```
0- Logout
0
Goodbye Ali.
Operation was successful!

====== LOG-IN ======
Example users: Admin(Pionix), BranchEmployee(Haydar), TPEmployee
Check the seeder of database at Database class!
1- Admin
2- Branch Employee
3- Transportation Employee
4- Customer
5- Exit the system
4
Username: Foo
====== Customer ======
1- Check Shipment
0- Logout
1
Enter tracking number of the shipment: 123
123: [Foo] --> [Goo]
        shipment delivered to the customer from  Ankara branch.
Operation was successful!

====== Customer ======
1- Check Shipment
0- Logout
```

Logout and exit system

```
====== Customer ======
1- Check Shipment
0- Logout
0
Goodbye Foo.
Operation was successful!

====== LOG-IN ======
Example users: Admin(Pionix),
Check the seeder of database a
1- Admin
2- Branch Employee
3- Transportation Employee
4- Customer
5- Exit the system
5
Bye bye..
```