

GTU Department of Computer Engineering
CSE 222/505 - Spring 2021
Homework 3 Report

AHMET OKUR
1801042655

1. SYSTEM REQUIREMENTS

Functional Requirements:

ADMIN:

- 1) Admin can add a new branch
- 2) Admin can remove branch
- 3) Admin can add employee to branches
- 4) Admin can remove employee to branches

COSTUMER:

- 1) Costumer can view the products and features in the store
- 2) Costumer can search for the desired product in the store
- 3) Costumer can shop online
- 4) Costumer can login system
- 5) Costumer can Register

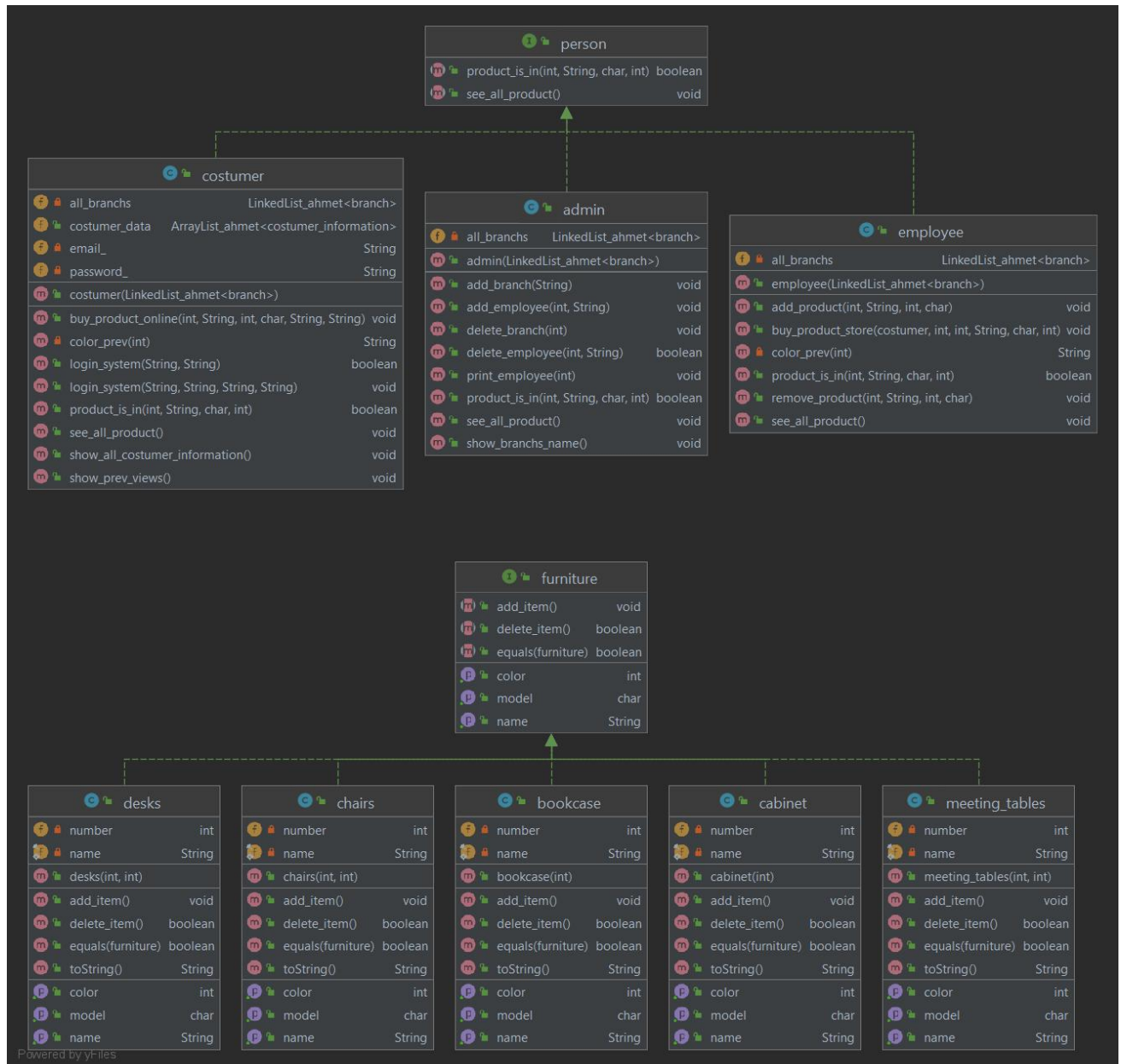
EMPLOYEE:

- 1) Employee can add a product
- 2) Employee can remove product
- 3) Employee can show products in stock
- 4) Employee can add members
- 5) Employee can view previous order information

Non-Functional Requirements:

1. Back-end software: Java 11
2. Hardware Interfaces: Mac, Linux and Windows operating systems
3. The system has an internal bug reporting system that keeps track of what types errors happened and when.
4. The system holds all the data to keep track of everything available in itself.
5. The system follows an email address policy.
6. The system follows a password policy.

2. CLASS UML DIAGRAMS



LinkedList_ahmet<E>	
head	Node<E>
tail	Node<E>
LinkedList_ahmet()	
Iterator(int)	ListIterator<E>
add(E, int)	void
addFirst(E)	void
addLast(E)	void
get(int)	E
indexOf(E)	int
remove(int)	void
removeFirst()	void
removeLast()	void
set(int, E)	E
toString()	String
first	E
last	E
size	int

HybridList_ahmet<E>	
head	Node<E>
MAX_VALUE	int
HybridList_ahmet()	
add(int, ArrayList_ahmet<E>, String)	void
addFirst(ArrayList_ahmet<E>, String)	void
addLast(ArrayList_ahmet<E>, String)	void
add_new_item(furniture)	boolean
get(int)	ArrayList_ahmet<E>
is_in(String, char, int)	boolean
remove(int)	void
removeFirst()	void
removeLast()	void
remove_item(String, int, char)	boolean
set(int, ArrayList_ahmet<E>, ArrayList_ahmet<E>)	ArrayList_ahmet<E>
toString()	String
size	int

ArrayList_ahmet<E>	
data	E[]
capacity	int
MAX_NUMBER	int
ArrayList_ahmet()	
add(int, E)	void
contains(E)	int
get(int)	E
reallocate()	void
remove(int)	E
set(int, E)	E
toString()	String
size	int

costumer_information	
name	String
surname	String
email	String
password	String
phone_number	String
address	String
prev_view	ArrayList_ahmet<String>
id	int
costumer_information()	
print_all_information()	void

branch<E>	
employee_name	ArrayList_ahmet<String>
data	HybridList_ahmet<furniture>
branch_name	String
branch(String)	
toString()	String

MyListIterator<E>	
nextItem	Node<E>
lastItemReturned	Node<E>
index	int
MyListIterator(int)	
add(E)	void
hasNext()	boolean
hasPrevious()	boolean
next()	E
nextIndex()	int
previous()	E
previousIndex()	int
remove()	void
set(E)	void

Node<E>	
data	E
next	Node<E>
prev	Node<E>
Node(E)	
Node(E, Node<E>)	

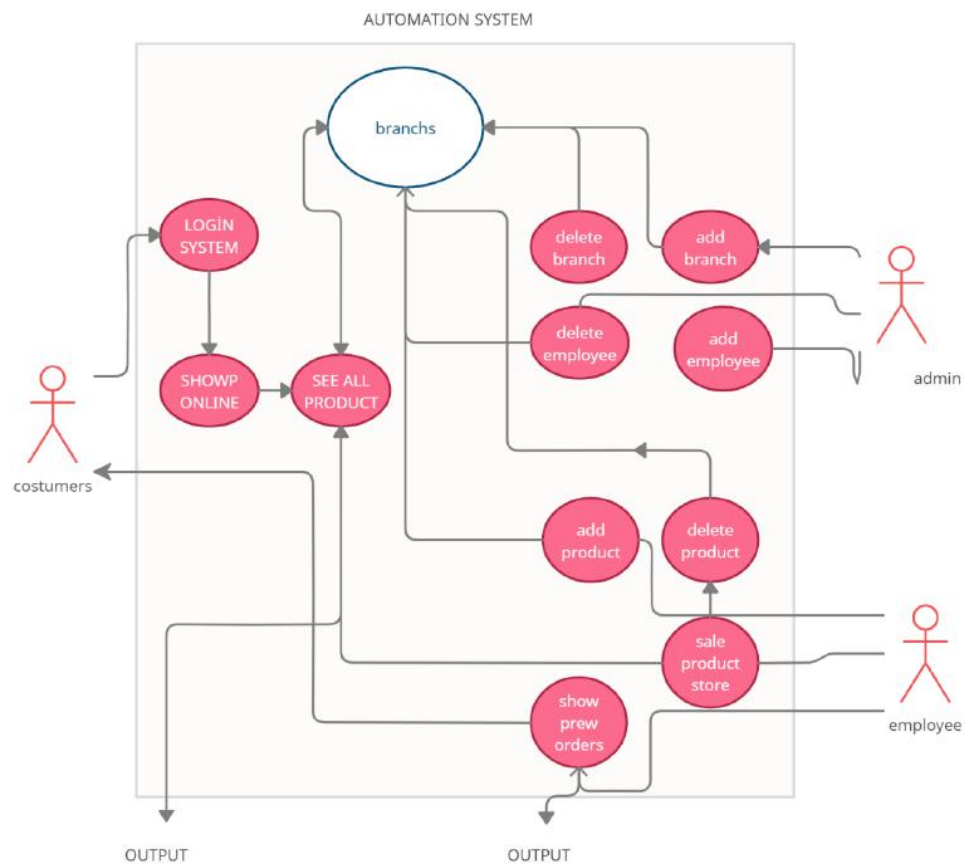
Node<E>	
data	ArrayList_ahmet<E>
next	Node<E>
which_furniture	String
Node(ArrayList_ahmet<E>, Node<E>)	
Node(ArrayList_ahmet<E>, String)	

ahmets_exceptions	
ahmets_exceptions(String)	

automation_system	
branches	LinkedList_ahmet<branch>
costumer_skills	costumer
employee_skills	employee
admin_skills	admin
automation_system()	

test	
test()	
driver_func()	void
main(String[])	void

3. USE CASE DIAGRAM



4. PROBLEM SOLUTION APPROACH

I choose java language for problem solving. I think the problem with this assignment is that the classes in the system can be aware of each other and keep themselves up-to-date. I created a class under the name of automation system admin, costumer, employee and branches. For the required arrays, I sometimes used classes as structures, for example costumer information consists of information in the class and these class and these classes are kept in an array. I inherited the

employee and the costumer to the person; admin independent of these two .I have defined admin,employee,costumer and branches as inner class in the automation class. In this way,they could be aware of each other,the processes in these 3 classes can update each other and the changes in the branches can stay up-to-date.In this way,the system becomes interactive.

5. TEST CASES RUNNING AND RESULTS

ADMIN

In this section the branch is deleted and added

```
/****** TEST ADMIN AREA *****/
System.out.println("BEFORE ADD/DELETE BRANCH");
test_outamation.admin_skills.show_branchs_name();

System.out.println("AFTER ADD KASTAMONU");
test_outamation.admin_skills.add_branch( branch_name: "KASTAMONU");
test_outamation.admin_skills.show_branchs_name();

System.out.println("AFTER DELETE İZMİT");
test_outamation.admin_skills.delete_branch( index: 4);
test_outamation.admin_skills.show_branchs_name();
```

branch delete and add output

```
BEFORE ADD/DELETE BRANCH
    ***BRANCHS NAME**
1)GÜMÜŞHANE
2)ANKARA
3)İSTANBUL
4)İZMİT

AFTER ADD KASTAMONU
    ***BRANCHS NAME**
1)KASTAMONU
2)GÜMÜŞHANE
3)ANKARA
4)İSTANBUL
5)İZMİT

AFTER DELETE İZMİT
    ***BRANCHS NAME**
1)KASTAMONU
2)GÜMÜŞHANE
3)ANKARA
4)İSTANBUL
```

In this section, the employee is added and deleted to the desired branch.

```
System.out.println("***BEFORE ADD/DELETE EMPLOYEE***");
test_outamation.admin_skills.print_employee( branch_id: 0);

System.out.println("\n***AFTER ADD ARGIT***");
test_outamation.admin_skills.add_employee( branch_id: 0, employee_name: "ARGIT");
test_outamation.admin_skills.print_employee( branch_id: 0);

System.out.println("\n***AFTER DELETE HATICE***");
test_outamation.admin_skills.delete_employee( branch_id: 0, employee_name: "HATICE");
test_outamation.admin_skills.print_employee( branch_id: 0);
```

In this section, the employee is added and deleted to the desired branch. OUTPUT

```
***BEFORE ADD/DELETE EMPLOYEE***
HATICE SENA AHMET

***AFTER ADD ARGIT***
ARGIT HATICE SENA AHMET

***AFTER DELETE HATICE***
ARGIT SENA AHMET
```

EMPLOYEE

Employee remove product method and product is in method test

```
System.out.println("\n*****\t\tTEST EMPLOYEE AREA\t\t*****");
System.out.println("\n***BRANCH 3 is in product BLUE DESK REMOVE 2 TIMES");
test_outamation.employee_skills.remove_product( branch_id: 2, furniture: "DESK", color_id: 1, model_id: 'A');
test_outamation.employee_skills.remove_product( branch_id: 2, furniture: "DESK", color_id: 1, model_id: 'A');

System.out.println("\n***AFTER REMOVE BRANCH 3 is in product BLUE DESK CONTROL(EMPLOYEE)***");
System.out.println(test_outamation.employee_skills.product_is_in( branch_id: 2, furniture: "DESK", model_id: 'A', color_id: 1));
```

Employee remove product method and product is in method output (before product number=2)

```
***BRANCH 3 is in product BLUE DESK REMOVE 2 TIMES  
***AFTER REMOVE BRANCH 3 is in product BLUE DESK CONTROL(EMPLOYEE)***  
false
```

Employee add product method and product is in method test

```
System.out.println("\n***BRANCH 3 is in product BLUE DESK ADD 2 TIMES");  
test_outamation.employee_skills.add_product( branch_id: 2, WhichFurniture: "DESK", color_id: 1, model_id: 'A');  
test_outamation.employee_skills.add_product( branch_id: 2, WhichFurniture: "DESK", color_id: 1, model_id: 'A');  
  
System.out.println("***AFTER ADD BRANCH 3 is in product BLUE DESK CONTROL(EMPLOYEE)***");  
System.out.println(test_outamation.employee_skills.product_is_in( branch_id: 2, furniture: "DESK", model_id: 'A', color_id: 1));
```

Employee add product method and product is in method output (before product number=0)

```
***BRANCH 3 is in product BLUE DESK ADD 2 TIMES  
***AFTER ADD BRANCH 3 is in product BLUE DESK CONTROL(EMPLOYEE)***  
true
```

Employee sell product store MEHMET OKUR

```
System.out.println("***BUY PRODUCT STORE TEST FOR MEHMET OKUR DESK B RED***\n");  
try {  
    test_outamation.employee_skills.buy_product_store(test_outamation.costumer_skills, id: 1, branch_id: 1, which_furniture: "DESK", model_id: 'B', color_id: 2);  
}  
catch (Exception e){  
    System.out.println(e);  
}  
  
test_outamation.costumer_skills.show_prev_views();
```


Employee sell product store MEHMET OKUR OUTPUT and SEE prev view

```
***BUY PRODUCT STORE TEST FOR MEHMET OKUR DESK B RED***
```

```
MEHMET OKUR
```

```
FURNITURE= DESK      MODEL= B      COLOR= Red
```

```
FURNITURE= DESK      MODEL= A      COLOR= Blue
```

COSTUMER

First login system method two people and just email and password login.if the password or e-mail incorrect,then gives error

```
System.out.println("\n\n***FIRST LOGIN SYSTEM 2 PEOPLE***");
test_outamation.costumer_skills.login_system( name: "AHMET", surname: "OKUR", email: "a.okur2018@gtu.edu.tr", password: "asdasd");
test_outamation.costumer_skills.login_system( name: "MEHMET", surname: "OKUR", email: "a.mehmet@gtu.edu.tr", password: "sasasa");

System.out.println("***LOGIN SYSTEM FOR MEHMET OKUR***");
test_outamation.costumer_skills.login_system( email: "a.mehmet@gtu.edu.tr", password: "sasasa");
System.out.println("***LOGIN SYSTEM INCORRECT EMAIL***");
test_outamation.costumer_skills.login_system( email: "a.mehmeast@gtu.edu.tr", password: "sasasa");
```

First login system method output

```
***FIRST LOGIN SYSTEM 2 PEOPLE***
NAME=AHMET
SURNAME=OKUR
EMAIL=a.okur2018@gtu.edu.tr
PASSWORD=asdasd
REGISTERED

NAME=MEHMET
SURNAME=OKUR
EMAIL=a.mehmet@gtu.edu.tr
PASSWORD=sasasa
REGISTERED

***LOGIN SYSTEM FOR MEHMET OKUR***
LOG IN
***LOGIN SYSTEM INCORRECT EMAIL***
PLEASE REGISTER
```

Costumer product is in using method

```
System.out.println("***BRANCH 3 is in product BLUE DESK CONTROL***");
System.out.println(test_outamation.costumer_skills.product_is_in( branch_id: 2, furniture: "DESK", model_id: 'A', color_id: 1));
```

Costumer product_is_in using method output

```
***BRANCH 3 is in product BLUE DESK CONTROL***
true
```

Costumer online shopping “DESK”,BLUE,A

```
test_outamation.costumer_skills.login_system(email: "a.mehmet@gtu.edu.tr", password: "888888");
System.out.println("***BUY PRODUCT ONLINE DESK A MODEL COLOR=BLUE ***");
test_outamation.costumer_skills.buy_product_online(branch_number: 1, which_furniture: "DESK", color: 1, model_id: 'A', address: "kocaeli/izmit", phone_number: "213123");
```

Output

```
*****DESK*****
Color = Blue   Model = A= 1
Color = Red    Model = A= 2
Color = black  Model = A= 2
```

Costumer can see all branch and product

```
System.out.println("***COSTUMER SEE ALL BRANCH AND FURNITURE***");
test_outamation.costumer_skills.see_all_product();
```

```
*****DESK*****
Color = Blue   Model = A= 1
Color = Red    Model = A= 2
Color = black  Model = A= 2
Color = yellow Model = A= 2
Color = Blue   Model = B= 2
Color = Red    Model = B= 2
Color = black  Model = B= 2
Color = yellow Model = B= 2
Color = Blue   Model = C= 2
Color = Red    Model = C= 2
Color = black  Model = C= 2
Color = yellow Model = C= 2
Color = Blue   Model = D= 2
Color = Red    Model = D= 2
Color = black  Model = D= 2
Color = yellow Model = D= 2
Color = Blue   Model = E= 2
Color = Red    Model = E= 2
Color = black  Model = E= 2
Color = yellow Model = E= 2

*****MEETINGTABLE*****
Color = Blue   Model = A= 2
Color = Red    Model = A= 2
Color = black  Model = A= 2
```

customer shopping history

```
System.out.println("*****Show PREVIOUS*****");  
test_outamation.costumer_skills.show_prev_views();
```

output

```
*****Show PREVIOUS*****  
MEHMET OKUR  
  
FURNITURE= DESK      MODEL= A      COLOR= Blue
```