GIT Department of Computer Engineering

CSE 222/505 - Spring 2021

Homework 1 Report

FATİH DOĞAÇ

1901042654

SYSTEM REQUIREMENTS

The user must have java v1.8.0 to run the program.And at least 5 MB space to run the program properly.

In Company class;

public boolean check(String model , String color , String name);

Checks if the wanted product (which is stated as model color name) exists in the company.If so returns true.

public void list_all() ;

Lists all products in every branches.

public void search(String model , String color , String name) ;

Searches all of the branches for the wanted product and sends a feedback. **public**

Admin check_admin_info(String email , String password) ;

Checks if the inputs match with an existing admin.If so returns the corresponding Admin. If not returns null.

public Customer check_customer_info(String email , String password);

Checks if the inputs match with an existing customer.If so returns the corresponding Customer. If not returns null.

public void check_stocks(Branch b1) ;

Asks from an admin to check the stocks of branch b1 .

public void add_customer(User u);

Adds customer u to the system.

public void show_customers() ;

Shows the current customers.

public void add_branch(User u1 , Branch b1) ;

Asks from admin u1 to add a branch b1 to the company. **public**

void del_branch(User u1 , Branch b1) throws Exception;

Asks from admin u1 to delete a branch b1 from the company. Throws exception if the branch doesn't exist.

public void show_branches() ;

Shows the current branches.

public void del_employee(User u1, User u2 , Branch b1) throws Exception ;

Asks from admin u1 to delete an employee from branch b1 Throws exception if the user u2 is not an employee of the branch

public void show_employees(Branch b) ;

Shows the current employees of branch 'b'.

public void make_sale(User u1 , Branch b1 , Product x) ;

Sales the product x in the branch b1 to the customer u1.

In Branch Class;

public void make_sale(Product x , User u) ;

Asks from a random employee to sell the product x to the customer u.

public void add_product(Product x);

Orders an employee to create a product x. **public void del_product(Product x) ;**

Orders and employee to create a product x.

public void requested_amount(Product x) ;

If the stocks are not enough, orders to an employee to inform the manager about stocks.

public void sale(User u , String model , String color , String name);

A customer can buy a product by himself/herself with online shopping.

public boolean check_product(String model , String color , String name);

Checks the products if the wanted one exists.

public void list_products();

List all of the products in this branch.

public void check_all_stocks() ;

Checks all the stocks of the products.

public void Add_Employee(User u2);

Hires u2 to the branch as an employee.

public void Del_Employee(User u2);

Fires u2 to the branch as an employee. In

Admin Class;

public void query_branch(Branch b1);

Orders to an Admin to check the stocks of a branch.

In Customer Class;

public void addHistory(Product a) ;

Adds a product to the customer's purchase histroy.

public void addHistory(String model , String color , String name) ;

Adds a product to the customer's purchase histroy.

public void see_history() ;

Shows the customers purchase history.

In Branch Employee class;

public void inform_the_manager(Product x) ;

Inform the manager about stocks of a product.

public void delete_product(Product x) ;

Deletes a product.

public void add_prod(Product x) ;

Creates a product.

public void sale(Product x , User u);

Sales a product x to the user u. **public**

void check_history(User u);

Checks the purchase history of an user. **In**

Desk class;

public void increaseStock() ;

Increases the corresponding desk's stock by one. **public**

void decreaseStock() ;

Decreases the corresponding desk's stock by one.

In Cabinet Class;

public void increaseStock() ;

Increases the corresponding desk's stock by one. **public**

void decreaseStock() ;

Decreases the corresponding desk's stock by one.

In Bookcase Class;

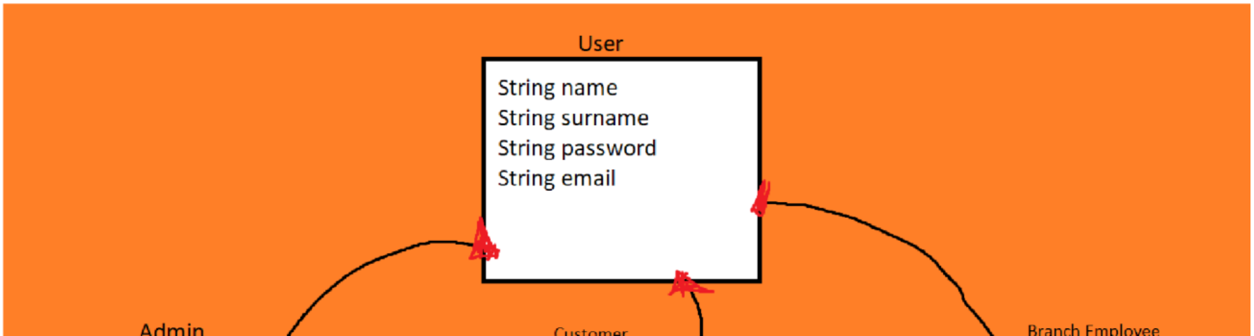
public void increaseStock() ;

Increases the corresponding desk's stock by one. **public**

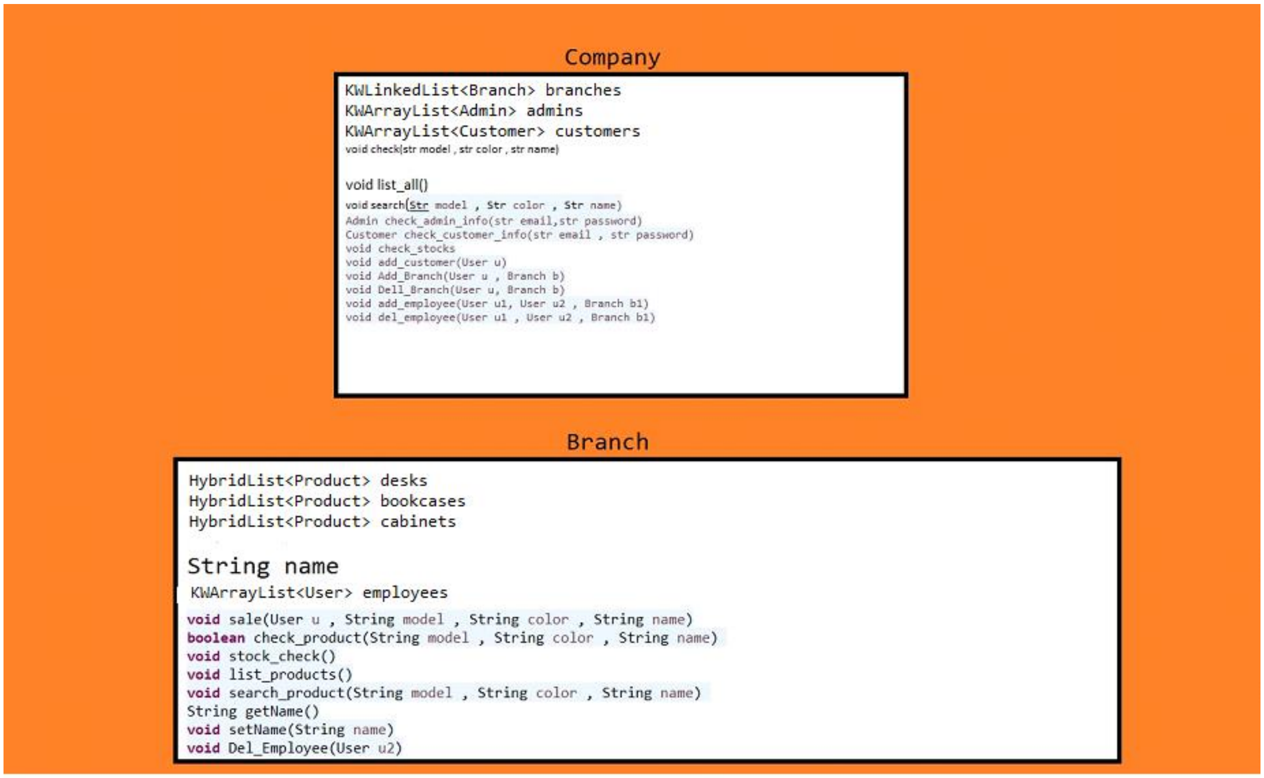
void decreaseStock() ;

Decreases the corresponding desk's stock by one.

User and Product Classes are abstract classes.



USE CASE AND CLASS DIAGRAMS

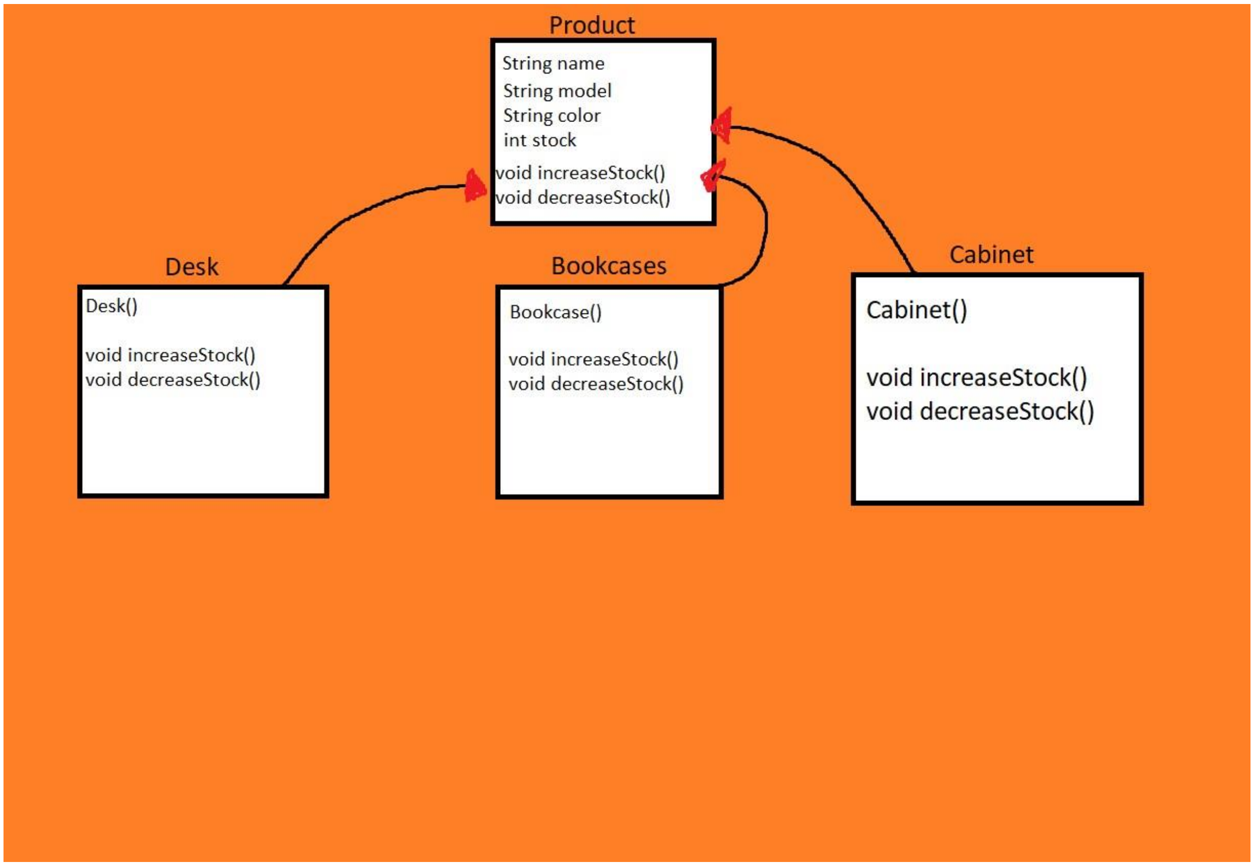


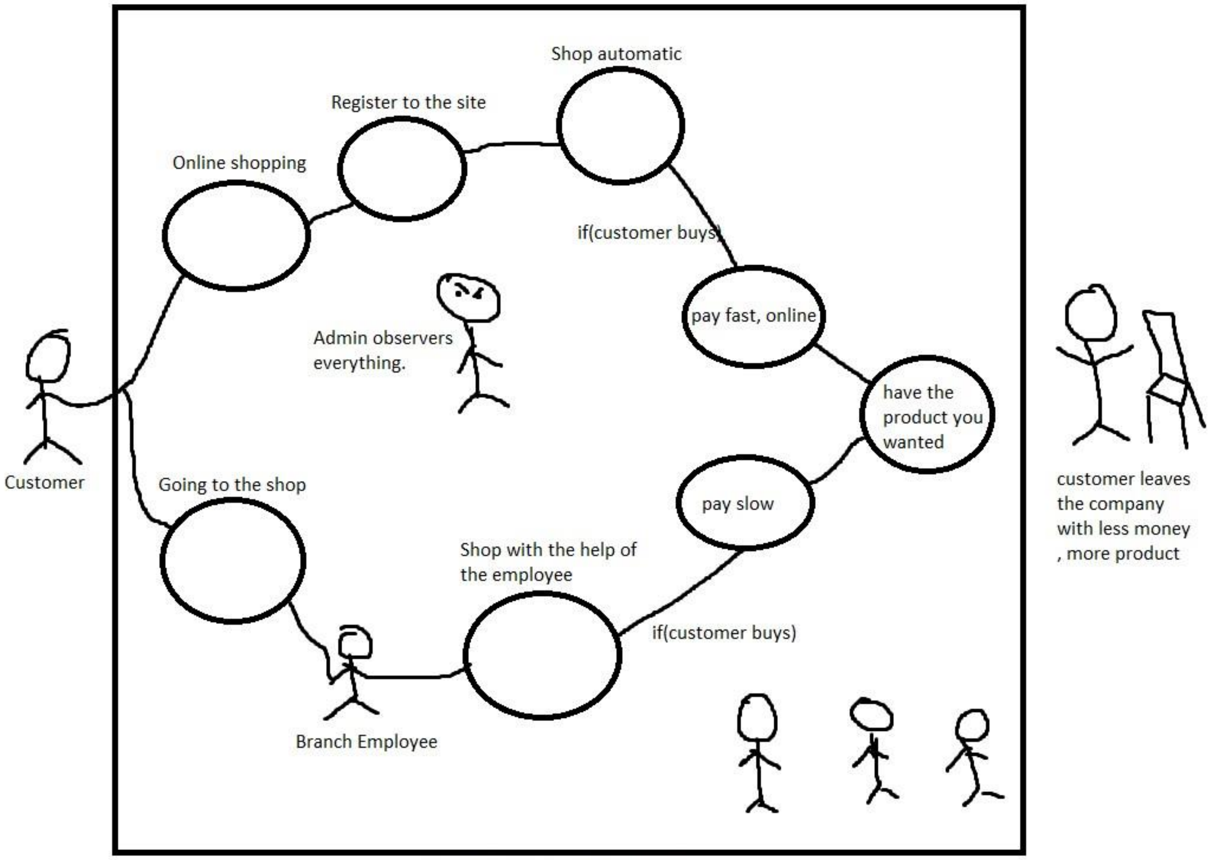
Company

```
KWLinkedList<Branch> branches
KWArrayList<Admin> admins
KWArrayList<Customer> customers
void check(str model , str color , str name)

void list_all()
void search(Str model , Str color , Str name)
Admin check_admin_info(str email,str password)
Customer check_customer_info(str email , str password)
void check_stocks
void add_customer(User u)
void Add_Branch(User u , Branch b)
void Dell_Branch(User u, Branch b)
void add_employee(User u1, User u2 , Branch b1)
void del_employee(User u1 , User u2 , Branch b1)
```

Branch

```
HybridList<Product> desks
HybridList<Product> bookcases
HybridList<Product> cabinets

String name
KWArrayList<User> employees
void sale(User u , String model , String color , String name)
boolean check_product(String model , String color , String name)
void stock_check()
void list_products()
void search_product(String model , String color , String name)
String getName()
void setName(String name)
void Del_Employee(User u2)
```


Product
String name
String model
String color
int stock
void increaseStock()
void decreaseStock()
Desk
Desk()
void increaseStock()
void decreaseStock()
Bookcases
Bookcase()
void increaseStock()
void decreaseStock()
Cabinet
Cabinet()
void increaseStock()
void decreaseStock()



COMPANY (FATIH FURNITURES)

PROBLEM SOLUTION APPROACH

I had a lot of problems with java. Because it doesn't have dynamic arrays and pointers. But I knew what should I do. So I searched the internet , some resources and pdfs to learn java's approach to pointers and alternatives of dynamic arrays. After all , I figured it out how to handle java with my knowledge in programming.

TEST CASES

All the test cases are in the driver.java. You just need to execute it.