```
import java.util.*;

public class Main {

    static Scanner in = new Scanner(System.in);
    static general gen = new general();
    public static void main(String[] args) throws Exception {

        int repeat =1;
        while(repeat==1){

            gen.market_status();
            gen._login_();
            gen.pass_day();

            System.out.println("login/leave shop? (1/0) : ");
            repeat = Integer.parseInt(in.nextLine());
        }
    }
}

import java.util.*;

public class HM_expDate {
    private Integer item_count=13;
    private HashMap<Integer,Integer> mp=new HashMap<Integer,Integer>();

    HM_expDate(){
        //ID , days until it expires

        mp.put(1, 2);//Spinach
        mp.put(2, 5 );//Cauli-flower
        mp.put(3, 7);//Mushrooms
        mp.put(4, 3);//Beet
        mp.put(5, 1);//Brocolli
        mp.put(6, 7);//Cabbage

        mp.put(7, 5);//Apple
        mp.put(8, 2);//Orange
        mp.put(9, 7);//Bananas
        mp.put(10, 11);//Cherries

        mp.put(11, 6);//Chicken
        mp.put(12, 5);//Fish
        mp.put(13, 3);//Crabs
    }

    public Integer get_expDate(Integer item_key){
        return mp.get(item_key);
    }

    public Boolean pass_day(){
        Boolean flag=false;
        for(Integer ct=1; ct<=item_count; ct++){
            if(mp.get(ct) != 0)
                mp.put(ct, mp.get(ct) - 1);
            if(mp.get(ct)==0){
                flag=true;
            }
        }
        return flag;
    }//reduce expiry by a day

    public void set_exp_date(){
        //FIXEDD

        mp.put(1, 2);//Spinach
        mp.put(2, 5 );//Cauli-flower
        mp.put(3, 7);//Mushrooms
        mp.put(4, 3);//Beet
        mp.put(5, 1);//Brocolli
        mp.put(6, 7);//Cabbage

        mp.put(7, 5);//Apple
        mp.put(8, 2);//Orange
        mp.put(9, 7);//Bananas
        mp.put(10, 11);//Cherries

        mp.put(11, 6);//Chicken
        mp.put(12, 5);//Fish
        mp.put(13, 3);//Crabs
    }
}

import java.util.*;

public class HM_count {

    private HashMap<Integer,Integer> mp=new HashMap<Integer,Integer>();

    HM_count(){
        // Inventory before user purchase/restock

        //VEGETABLES;
        mp.put(1, 4);//Spinach
        mp.put(2, 5 );//Cauli-flower
        mp.put(3, 20);//Mushrooms
        mp.put(4, 5);//Beet
        mp.put(5, 0);//Brocolli
        mp.put(6, 3);//Cabbage

        //FRUITS
        mp.put(7, 10);//Apple
        mp.put(8, 7);//Orange
        mp.put(9, 20);//Bananas
        mp.put(10, 20);//Cherries

        //MEAT
        mp.put(11, 1);//Chicken
        mp.put(12, 5);//Fish
        mp.put(13, 10);//Crabs
    }

    public boolean buy(Integer item_key, Integer item_bought_count){
        if(mp.get(item_key)>=item_bought_count){
            mp.put(item_key, mp.get(item_key) - item_bought_count);
            return true;
        }
        return false;
    }

    public Integer get_count(Integer item_key){
        return mp.get(item_key);
    }

    public void _set_(Integer item_key, Integer count){
        mp.put(item_key, count);
    }//for re-stocking

    public void set_0(Integer item_key){
        mp.put(item_key, 0);
    }//for when food get expired

}


import java.util.*;


public class general {
    static Scanner in = new Scanner(System.in);

    static accounts acch = new accounts();      //change to account_handling for Database support
    static HM_count qty = new HM_count();
    static HM_expDate exp = new HM_expDate();
    static ARR_names itmn = new ARR_names();
    static int item_count=13;

    general(){
        System.out.println("\n\n**** WELCOME TO BIG-BASKET ****");
    }
    void market_status(){
        System.out.println("Market Status : ");
        System.out.printf( "\n| %2s.| %-15s| %5s   | %-10s |\n ","ID", "Item Name", "QTY", "Expires in" );
        System.out.println("------------------------------------------");

        for(int ct=0; ct<item_count; ct++){
            System.out.printf( "| %2d.| %-15s| %5d   |%5d days  |\n",
                            ct + 1,
                            itmn.get_item_name(ct),
                            qty.get_count(ct+1),
                            exp.get_expDate(ct+1)
                        );
        }
    }

    void pass_day(){
        Boolean _status_ = exp.pass_day();
        System.out.println("A DAY HAS PASSED");
        if(_status_)
            System.out.println("*** please ask admin to restock\n");
        System.out.println("------------------------------------------");


        for(Integer ct=0; ct<item_count; ct++){
            if(exp.get_expDate(ct+1)==0)
                qty.set_0(ct+1);
        }
    }

    void _login_(){
        int fl=1;
        while(fl==1){
            System.out.print("\nSign-up/ Login (1/0) : ");
            int new_acc = Integer.parseInt(in.nextLine());
            if(new_acc == 1){
                acch.new_user_login();
                customer_menu();
                fl=0;
            }
            else{
                int opt = acch.old_user_login();

                if(opt==0){ fl=1; }
                if(opt==1){ fl=0; customer_menu(); }
```

```
                if(opt==2){ fl=0; admin_menu(); }
            }
        }
    }

    void admin_menu(){
        System.out.println("***Admin Mode");
        System.out.println("Re-stock per category :");
        Integer quan;
        quan = Integer.parseInt(in.nextLine());

        System.out.println("Restocking...");
        for(Integer ct=0; ct<item_count; ct++){
            qty._set_(ct+1, quan);
        }
        exp.set_exp_date();
    }

    void customer_menu(){
        Integer count;
        System.out.print("\nNo. of items you want to BUY : ");
        count = Integer.parseInt(in.nextLine());
        for(int ct=0; ct<count; ct++){
            Integer item_id, quan;
            System.out.print("\nItem_id  : ");
            item_id = Integer.parseInt(in.nextLine());
            System.out.print("Quantity : ");
            quan = Integer.parseInt(in.nextLine());

            if(qty.buy(item_id, quan) != true){
                System.out.println("** not in stock **");
                ct--;
            }
        }
        System.out.println("-------------------------------------------");
        System.out.println("\nThanks for supporting us!");
    }


}


public class ARR_names {
    private Integer item_count = 13;
    private String items[] = new String[item_count];

    ARR_names(){
        items = new String[]{
                                "Spinach",
                                "Cauli-flower",
                                "Mushrooms",
                                "Beet",
                                "Brocolli",
                                "Cabbage",
                                "Apple",
                                "Orange",
                                "Bananas",
                                "Cherries",
                                "Chicken",
                                "Fish",
                                "Crabs"
                            };//order must be maintained according to HM_count.java
    }

    public String get_item_name(Integer i){
        return items[i];
    }

}

import java.util.*;

public class accounts {


    Scanner in = new Scanner(System.in);
    Set<String> hash_Set = new HashSet<String>();
    private HashMap<String,String> mp=new HashMap<String,String>();

    boolean username_already_exists(String user_name){

        if (! hash_Set.contains(user_name)){
            hash_Set.add(user_name);
            return false;
            }
        else{
            System.out.println(" **username exists");
            return true;
        }

    }

    void new_user_login(){
        String pass1, pass2,user_name;
        do{
            System.out.print("\nUsername : ");
            user_name = in.nextLine();
        }while(username_already_exists(user_name));
        username_already_exists(user_name);

        do{
            System.out.print("Password          : ");
            pass1 = in.nextLine();
            System.out.print("Re-enter Password : ");
            pass2 = in.nextLine();
        }while(!pass1.equals(pass2));

        hash_Set.add(user_name);
        mp.put(user_name, pass1);

        System.out.println("*** account created ***");

    }

    int old_user_login(){
        String pass1, user_name;
        String retrieved="";
        do{
            System.out.print("\nUsername : ");
            user_name = in.nextLine();
        }while(!username_already_exists(user_name));

        retrieved = mp.get(user_name);

        String backup = retrieved;
        do{
            retrieved=backup;
            pass1="";
            System.out.print("\nPassword : ");
            pass1 = in.nextLine();

            if( !pass1.equals(retrieved)){
                System.out.println("wrong pass");
                System.out.println("Try again? (1/0) : ");
                int opt = Integer.parseInt(in.nextLine());

                if(opt==0){
                    return 0;
                }
            }
        }while(!pass1.equals(retrieved));

        System.out.println("** Successful login ***");
        if(user_name.equals("admin")){
            return 2;
        }
        return 1;
    }

}

import java.sql.*;
import java.util.*;


public class account_handling {

    Scanner in = new Scanner(System.in);

    boolean username_already_exists(String user_name){
        String query = "select user_name from users where user_name='"+user_name+ "'";

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = 
DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query);
            if (!rs.next()){
                con.close();
                return false;
                }
            else{
                System.out.println(" **username exists");
                con.close();
                return true;
            }
        }catch(Exception e){System.out.println(e);}

        return true;
    }
    void new_user_login(){
        String pass1, pass2,user_name;
        do{
            System.out.print("\nUsername : ");
            user_name = in.nextLine();
        }while(username_already_exists(user_name));
        username_already_exists(user_name);

        do{
        System.out.print("Password          : ");
        pass1 = in.nextLine();
        System.out.print("Re-enter Password : ");
        pass2 = in.nextLine();
        }while(!pass1.equals(pass2));
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = 
DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
            Statement stmt = con.createStatement();
            String updt = "insert into users values('"+user_name+ "','"+ pass1 +"')";
            stmt.executeUpdate(updt);
            con.close();
            System.out.println("*** account created ***");
        }catch(Exception e){System.out.println(e);}
    }

    int old_user_login(){

        String pass1, user_name;
        String retrieved="";
        do{
            System.out.print("\nUsername : ");
            user_name = in.nextLine();
        }while(!username_already_exists(user_name));

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = 
DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
            Statement stmt = con.createStatement();
            String query = "select password from users where user_name='"+user_name+ "'";
            ResultSet rs = stmt.executeQuery(query);

            rs.next();
            retrieved = rs.getString(1);
            //System.out.println("test : ");
            //System.out.println(retrieved);
            con.close();


        }catch(Exception e){System.out.println(e);}
        String backup = retrieved;
        do{
            retrieved=backup;
            pass1="";
            System.out.print("\nPassword : ");
            pass1 = in.nextLine();
            //in.nextLine();
            if( !pass1.equals(retrieved)){
                System.out.println("wrong pass");
                System.out.println("Try again? (1/0) : ");
                int opt = Integer.parseInt(in.nextLine());
                if(opt==0){
                    return 0;
                }
                //System.out.println("pass1 = "+pass1);
                //System.out.println("r = "+retrieved);
            }
        }while(!pass1.equals(retrieved));

        System.out.println("** Successful login ***");
        if(user_name.equals("admin")){
            //System.out.println("admin 1");
            return 2;
        }
        return 1;
    }

}
```

OUTPUT

```
**** WELCOME TO BIG-BASKET ****
Market Status :

| ID.| Item Name      |   QTY   | Expires in |
 ---------------------------------------------
|  1.| Spinach        |     4   |    2 days  |
|  2.| Cauli-flower   |     5   |    5 days  |
|  3.| Mushrooms      |    20   |    7 days  |
|  4.| Beet           |     5   |    3 days  |
|  5.| Brocolli       |     0   |    1 days  |
|  6.| Cabbage        |     3   |    7 days  |
|  7.| Apple          |    10   |    5 days  |
|  8.| Orange         |     7   |    2 days  |
|  9.| Bananas        |    20   |    7 days  |
| 10.| Cherries       |    20   |   11 days  |
| 11.| Chicken        |     1   |    6 days  |
| 12.| Fish           |     5   |    5 days  |
| 13.| Crabs          |    10   |    3 days  |

Sign-up/ Login (1/0) : 1
```

```
Username : p
 **username exists
Password         : 123
Re-enter Password : 123
*** account created ***

No. of items you want to BUY : 2

Item_id  : 1
Quantity : 4

Item_id  : 2
Quantity : 5
---------------------------------------------

Thanks for supporting us!
A DAY HAS PASSED
*** please ask admin to restock

---------------------------------------------
login/leave shop? (1/0) :
1
Market Status :

| ID.| Item Name      |   QTY   | Expires in |
 ---------------------------------------------
|  1.| Spinach        |     0   |    1 days  |
|  2.| Cauli-flower   |     0   |    4 days  |
|  3.| Mushrooms      |    20   |    6 days  |
|  4.| Beet           |     5   |    2 days  |
|  5.| Brocolli       |     0   |    0 days  |
|  6.| Cabbage        |     3   |    6 days  |
|  7.| Apple          |    10   |    4 days  |
|  8.| Orange         |     7   |    1 days  |
|  9.| Bananas        |    20   |    6 days  |
| 10.| Cherries       |    20   |   10 days  |
| 11.| Chicken        |     1   |    5 days  |
| 12.| Fish           |     5   |    4 days  |
| 13.| Crabs          |    10   |    2 days  |

Sign-up/ Login (1/0) : 0

Username : q

Username : q
 **username exists

Password : qq
wrong pass
```

```
Try again? (1/0) :
1

Password : qq
wrong pass
Try again? (1/0) :
0

Sign-up/ Login (1/0) : 0

Username : p
 **username exists

Password : 123
** Successful login ***

No. of items you want to BUY : 3

Item_id  : 1
Quantity : 1
** not in stock **

Item_id  : 3
Quantity : 1

Item_id  : 4
Quantity : 1

Item_id  : 5
Quantity : 1
** not in stock **

Item_id  : 6
Quantity : 1
--------------------------------------------

Thanks for supporting us!
A DAY HAS PASSED
*** please ask admin to restock

--------------------------------------------
login/leave shop? (1/0) :
0
```

Assignment no. 1

Title: Use of collections & generics

Problem Statement:
Design a system withe the help of advanced data structure in java & enhance the system using collections & generics.

Objective: To implement java generics/collections.

Outcome: We will understand basics of java

S/W & H/W Req: JAVA IDE, Intel I5 processor, 8GB RAM

Theory:-
- Collections in java is a framework that helps in storing & manipulating a group of objects.
- All operations like searching, sorting can be performed using collections.
- Types:
  1) Ordered lists:
     Programmer inserts items in an order to get back later
     Interfaces:-
     1) List
     2) Queue.

Vector: Similar to array but size
is dynamic

Lists: Ordered collection of objects
duplication allowed.

Set : Insertion order not maintained
no duplication

linked list: Uses doubly linked list
to store elements.

Tree Set : Homogeneous collection of
elements & underline as
balanced tree in ascending order.

Queue : List in which order is maintained
(FIFO)

Map : Organises objects as key value
pair.

Arraylist methods:
ArrayList <String> al = new ArrayList <string>
1) void add (int ~~order~~ index, object element)
2) void clear
3) boolean addAll()
4) object clone()

Hashset class:
1) void clear()
2) boolean contains(object D)
3) boolean add (object D)
4) boolean isEmpty()
5) HashSet <String> hs = new hashSet <String>();

:) Generics in Java

Generics are a facility of generic programming that allows a type or method to operate on objects of various types while providing compile time safety.

Syntax: class or interface (Type)
eg Arraylist <String>

Generic class:

```
class   A <T> {
          T obj ;
          void  add (T obj) {
             this.obj = obj;
          }
          T get () {
             return  obj ;
          }
};
```

T -> any data type (template)

★ Class structure:

```
public class account_handling {

    boolean usernam_already-exists (String
    username) {}
    void new-user_login () {}
    void old-user_login () {}

}

public class ARR_names {

    ARR_names() {}
    public String get_item_name() {}

}

public class general {
    general () {}
    void market_status () {}
    void pass_day () {}
    void login () {}
    void admin_menu () {}
    void customer_menu () {}
}
```

```
public class HM-count {

    HM-count () {}
    public boolean buy(Integer item-key,
    Integer Item-bought-count) {}
    public Integer get-count (Integer item_key) {}
    public void _set_(Integer item-key,
    Integer count) {}
    public void set-zero (Integer item_key){}
}

public class HM_expDate {

    HM_expDate() {}
    public Integer get-exp-date(Integer item-key)
    public boolean pass-day();
    public void set-exp-date();
}

public class Main {

  public static void main( String []args)
  throws Exception {}

}
```

Conclusion:- Understood & implemented Java generics & collections & incorporated them in our system.

classmate
Date
Page

* Test Cases:

1) Signup / Login (1/0) : 1

Usernome: P
password : 123
Re-enter : 123

* Successfully made acc.
* Logged in

No. of items to buy : 1
Item code : 1
quantity : 2

* thanks for supporting us.

— Status : Succesful.

2) Signup / Login (1/0) : 0
Username : 1
Password : 1
* Wrong credentials
Username : 1
Password : 2
* Success
No. of items to buy : 1
Item code : 1
quantity : 3
* thanks for supporting us

— Status : Succesful$