

**University of Sadat City**



**Faculty of Computers and Artificial Intelligence (FCAI)**

**Course: Computer Programming-2**



# Blood Bank Management System

By

**Mohammed Shaaban Ali**

**Hisham Mahmoud Khairallah**

**Mostafa Safwat Mohammed**

**Sahar Gamal Tawfik**

**Noora Abdelrahman Zakaria**

**Under Supervision**

**Dr / Sara Shehab**

**Eng / Mohamed Hammad**

# **Blood Bank Management System**

## **Abstract :**

Blood Bank Management System is a browser-based system that is designed to store, process, retrieve and analyses information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassles free and corruption free and make the process of obtaining blood bank management effective. our client is not interested in blood stocking instead we are stocking blood donor's information. The donors who are interested in donating blood has to register in the array list. There is no storage of blood so no complications in the array list. The software is fully integrated with customer relationship management as well as content management system solution. It is developed in a manner that is easily manageable, time saving and relieving one from manual works .

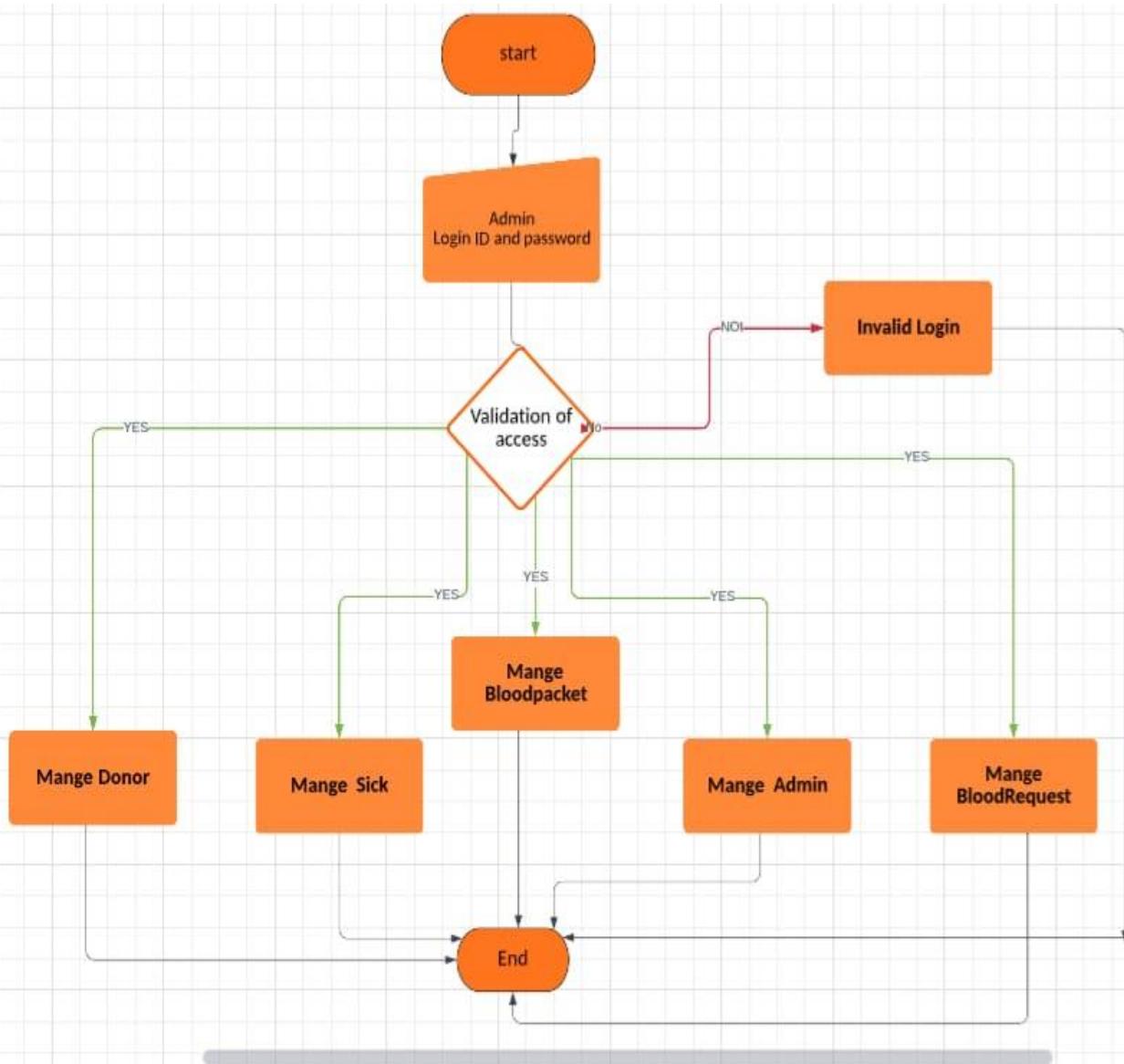
## **Keywords :**

Admin, Sick , Donor , Blood,Blood Request

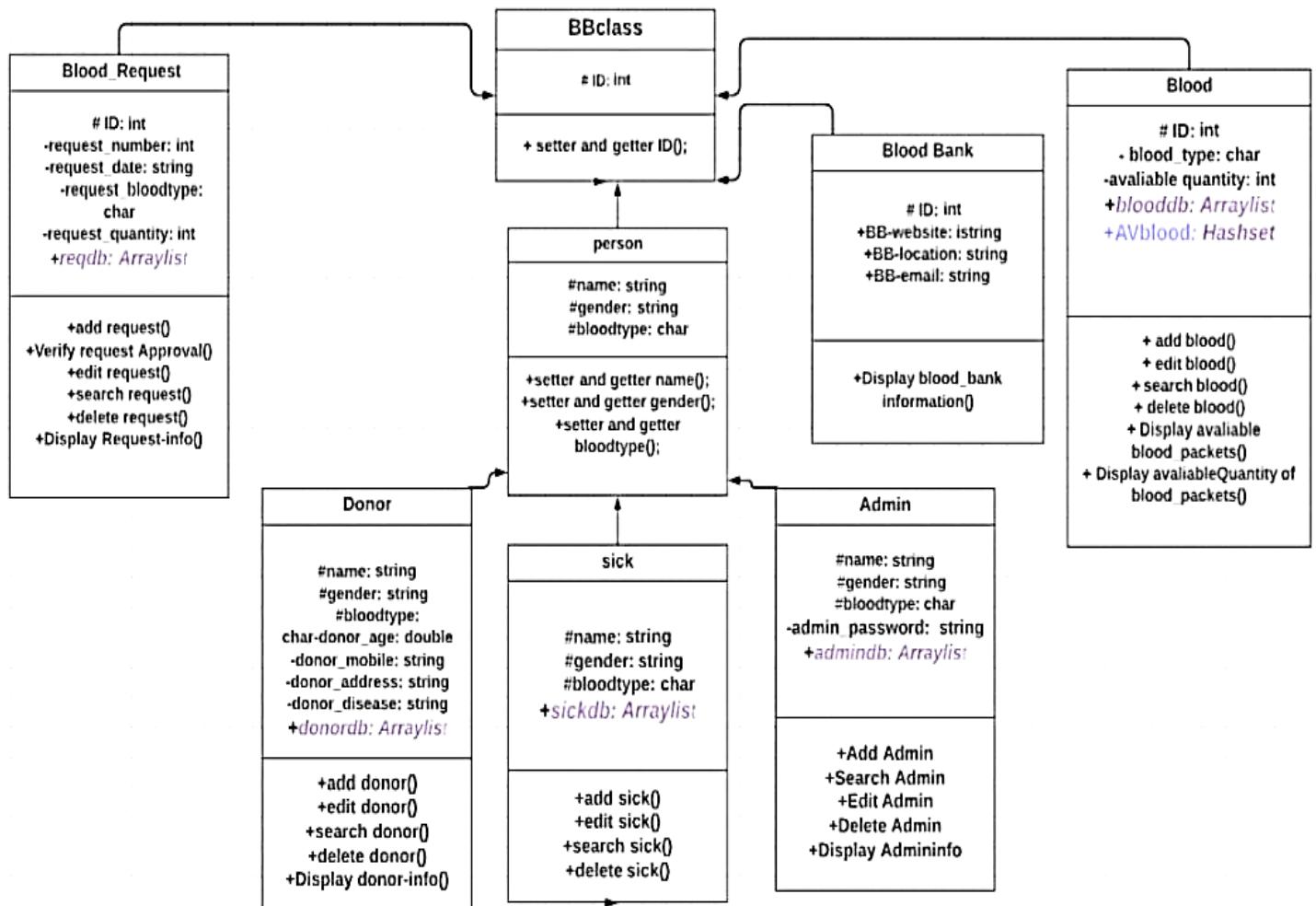
## **INTRODUCTION**

The "Blood Bank Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Blood Bank Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. Every organization, whether big or small, has challenges to overcome and managing the information of Donor, Blood Bank, Blood, Patient, Blood Group. Every Blood Bank Management System has different Blood Bank needs; therefore, we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executives who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources .

## Flowchart for our Project (BBM System) :-



## Uml Class Diagram :-



❖ This First class (BBclass) in the project(BBSYSTEM)

at this class we define an [ id ] that will be inherited in others classes



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package BBS;
6  public class BBclass {
7      protected int id;
8
9      public BBclass() {
10
11
12      public int getId() {
13          return id;
14
15      public void setId(int id) {
16          this.id = id;
17
18      public void display(BBclass e) {
19          System.out.println(e.id);
20
21      }
22
23  }
24
25 }
```

❖ This class(person) that inherited from (BBclass)

Then we define an string (name) , string (blood type) and string (gender) .All this will inherit in others classes , So we have a multilevel inheritance that will be Shown later.



```
4  /*
5   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
6   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
7   */
8  package BBS;
9
10 public abstract class person extends BBclass {
11     protected String name;
12     String bloodtype;
13
14     protected String gender;
15     public person() {
16
17     public String getGender() {
18         return gender;
19
20     public void setGender(String gender) {
21         this.gender = gender;
22
23     public String getName() {
24         return name;
25
26
27     public void setName(String name) {
28         this.name = name;
29
30     public String getBloodtype() {
31         return bloodtype;
32
33
34     public void setBloodtype(String bloodtype) {
35         this.bloodtype = bloodtype;
36
37
38 }
```

We've been said that we have a multilevel inheritance as shown

{BBclass} → {person} → {Admin}

Here in this class(Admin) that inherit from previous class , then we define a string (password) and make an arraylist called (admindb )that we will use in this class , then invoke the constructors for this class



The screenshot shows a Java code editor with the following code:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package BBS;

import static BBS.Donor.donordb;
import static BBS.sick.sickdb;
import java.util.ArrayList;
import java.util.Scanner;

public class Admin extends person {

    String admin_password;
    public static ArrayList<Admin> admindb = new ArrayList();

    public Admin() {
    }

    public Admin(int admin_id, String admin_username, String admin_password) {
        this.id = admin_id;
        this.name = admin_username;
        this.admin_password = admin_password;
    }

    public String getAdmin_password() {
        return admin_password;
    }

    public void setAdmin_password(String admin_password) {
        this.admin_password = admin_password;
    }

    public void addadmin(Scanner s) {
        System.out.println("plz enter number of admin");
        // Start Handel Exception
        String numberString;
        int number;
    }
}
```

After that we will start to build the methods for this class that will be invoked later in the main Class.

we built a method called (addadmin) , in this method first ask user to enter the number of admins that he want to add in array list , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value , then we have (for loop) to enter the number of admin that the user need to add them in the array list , then we take an object from this class(admin )and start with entering the admin id , and check if this id had been entered to the array list , if no this id of admin will added to the array list.

```
33
34 public void addadmin(Scanner s) {
35     System.out.println("plz enter number of admin");
36     // Start Handel Exception
37     String numberString;
38     int number;
39     numberString = s.next();
40     try {
41         number = Integer.parseInt(s.numberString);
42     }catch (Exception e){
43         System.out.println("plz enter the integer value");
44         number = s.nextInt();
45     }
46     // End Handel Exception
47     if (number == 0) {
48         System.out.println("error,enter number>0");
49     } else {
50         for (int i = 0; i < number; i++) {
51             Admin a = new Admin();
52             System.out.println("enter the ID of Admin " + (i + 1) + ":");
53             // Start Handel Exception
54             String idString;
55             int id;
56             idString = s.next();
57             try {
58                 id = Integer.parseInt(s.idString);
59             }catch (Exception e){
60                 System.out.println("plz enter the integer value");
61                 id = s.nextInt();
62             }
63             // End Handel Exception
64             if (Admin.admindb.isEmpty()) {
65                 a.setId(id);
66             } else {
67                 for (Admin e : admindb) {
68                     if (e.id == id) {
69                         System.out.println("Error,this ID is exist");
70                         return;
71                     }
72                 }
73             }
74         }
75     }
76 }
```

After then ask user to entering the username and password of this admin , then add this admin in the array list and so on to the other admins .

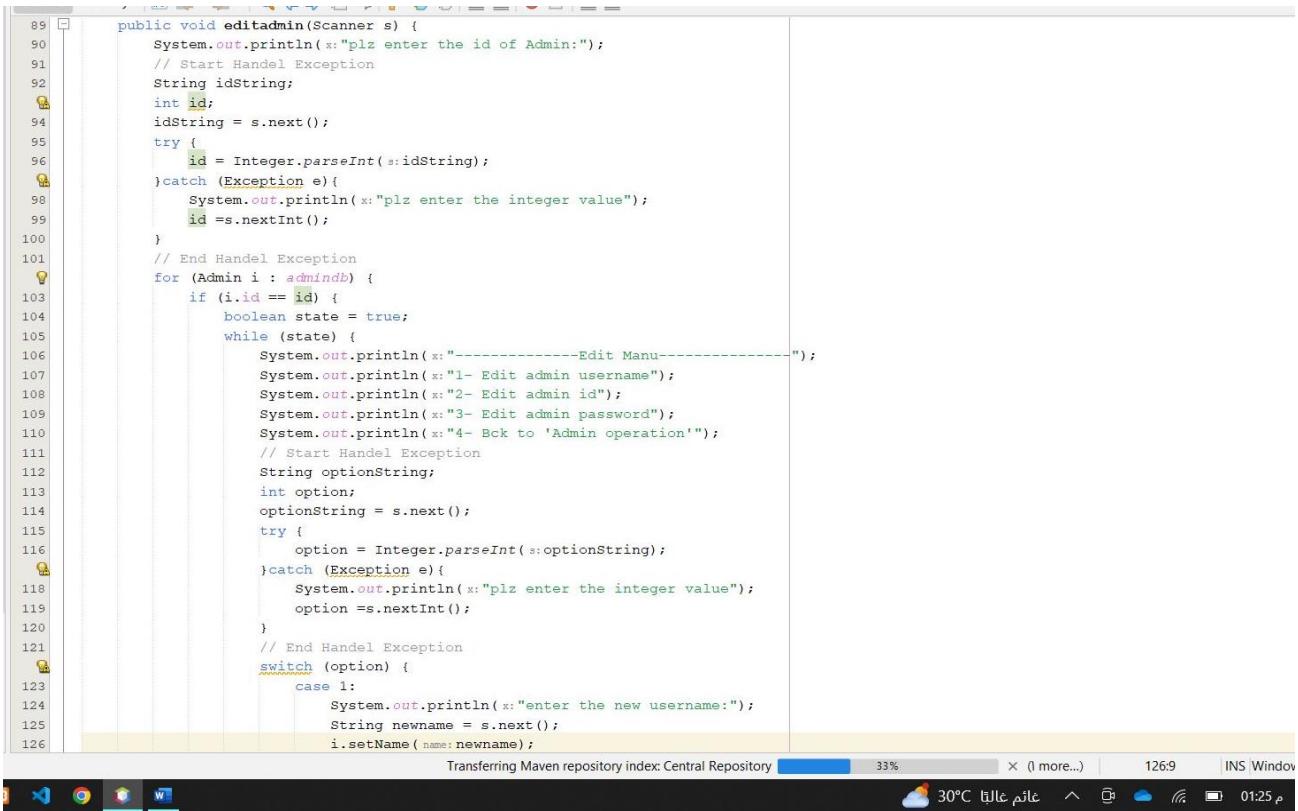


```
72     }
73 }
74
75     System.out.println("enter the username of Admin " + (i + 1) + ":");
76     String username = s.next();
77     System.out.println("enter the password of Admin " + (i + 1) + ":");
78     String password = s.next();
79     a.setId(id);
80     a.setAdmin_password(admin_password);
81     a.setName(name);
82     Admin.admindb.add(a);
83 }
84
85 }
86
87 }
```

Transferring Maven repository index: Central Repository 33% X (1 more...) 82:38 INS Windows (CRLF)

Cloud 30°C July 10 12:38 01:24 p

Then we build a method called(editadmin) Then ask user to enter an id of the admin then check if this admin id is already added to Array list . in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value, if true an edit menu will appear to the user and some options and make a switch to an entered option in this switch case1 to till user to enter the new name



```
89 public void editadmin(Scanner s) {
90     System.out.println("plz enter the id of Admin:");
91     // Start Handel Exception
92     String idString;
93     int id;
94     idString = s.next();
95     try {
96         id = Integer.parseInt(idString);
97     } catch (Exception e) {
98         System.out.println("plz enter the integer value");
99         id = s.nextInt();
100    }
101    // End Handel Exception
102    for (Admin i : admindb) {
103        if (i.id == id) {
104            boolean state = true;
105            while (state) {
106                System.out.println("-----Edit Manu-----");
107                System.out.println("1- Edit admin username");
108                System.out.println("2- Edit admin id");
109                System.out.println("3- Edit admin password");
110                System.out.println("4- Bck to 'Admin operation'");
111                // Start Handel Exception
112                String optionString;
113                int option;
114                optionString = s.next();
115                try {
116                    option = Integer.parseInt(optionString);
117                } catch (Exception e) {
118                    System.out.println("plz enter the integer value");
119                    option = s.nextInt();
120                }
121                // End Handel Exception
122                switch (option) {
123                    case 1:
124                        System.out.println("enter the new username:");
125                        String newname = s.next();
126                        i.setName(newname);
```

Transferring Maven repository index: Central Repository 33% X (1 more...) 126:9 INS Windows

Cloud 30°C July 10 12:38 01:25 p

After add the new name and then print the information after editing .

Case 2 to edit admin id and a message till user to enter the new id and then print the information after editing.

Case 3 to edit admin password and a message till user to enter the new password and then print the information after editing .

When the id that has been entered is false there is a message to tell user that admin is unavailable

```
126 i.setName( name: newname );
127 System.out.println( x:"****information about Admin after Edit****");
128 this.display(i);
129 break;
130 case 2:
131 System.out.println( x:"enter the new Id:");
132 // Start Handel Exception
133 String newidString;
134 int newid;
135 newidString = s.nextInt();
136 try {
137     newid = Integer.parseInt(s:newidString);
138 }catch (Exception e){
139     System.out.println( x:"plz enter the integer value");
140     newid = s.nextInt();
141 }
142 // End Handel Exception
143 i.setId( id:newid);
144 System.out.println( x:"****information about Admin after Edit****");
145 this.display(i);
146 break;
147 case 3:
148 System.out.println( x:"enter the new password:");
149 String newpass = s.next();
150 i.setAdmin_password( admin_password:newpass);
151 System.out.println( x:"****information about Admin after Edit****");
152 this.display(i);
153 break;
154 case 4:
155 state = false;
156 break;
157 }
158 }
159 return;
160 }
161 }
162 System.out.println( x:" Admin unavailable");
163 }
```

Then we build a method to search about admin , first ask user to enter the admin id to search and use exception Handel when user enter a wrong value and check if the id is true the information about the admin will appear to user , else the id is not available in Array list will tell user this admin isn't available,

```

164
165 public void searchadmin(Scanner s) {
166     int c = 0;
167     System.out.println("plz enter the ID of Admin:");
168     // Start Handel Exception
169     String idString;
170     int id;
171     idString = s.next();
172     try {
173         id = Integer.parseInt(s:idString);
174     }catch (Exception e){
175         System.out.println("plz enter the integer value");
176         id =s.nextInt();
177     }
178     // End Handel Exception
179     for (Admin i : admindb) {
180         if (i.id == id) {
181             System.out.println("*****information about Admin*****");
182             this.display(i);
183             c++;
184             break;
185         } else {
186             System.out.println("Admin unavailable");
187         }
188     }
189 }

```

Then we build a method to delete admin from Array list , first we check if the array list is empty or no , in case no we ask user to enter the id for this admin then check if this admin in the array list case true this admin will removed from Array list , case false the system tell user this admin id is not found .

```

190
191 public void deleteadmin(Scanner s) {
192     if (admindb.isEmpty()) {
193         System.out.println("Not possible ,plz check Database");
194     } else {
195         System.out.println("plz enter the id of Admin:");
196         // Start Handel Exception
197         String idString;
198         int id;
199         idString = s.next();
200         try {
201             id = Integer.parseInt(s:idString);
202         }catch (Exception e){
203             System.out.println("plz enter the integer value");
204             id =s.nextInt();
205         }
206         // End Handel Exception
207         for (Admin i : admindb) {
208             if (id == i.id) {
209                 admindb.remove(i);
210                 System.out.println("Suceesfully Done ");
211                 return;
212             } else {
213                 System.out.println(" Not Found");
214             }
215         }
216     }
217 }
218
219
220 public void display(Admin i) {
221     System.out.println("username: ID : password: ");
222     System.out.println(i.getName() + " " + i.getId() + " " + i.getAdmin_password());
223 }
224
225

```

And a method to print the information that we use in other method .

Now this is the next class (sick) , We've been said that we have a multilevel inheritance as shown

{BBclass} → {person} → {Sick}

In this class(sick) that inherit from previous class (person), then we make an array list called (sickdb )that we will use in this class to store the information for this sick

First we built a method called (addSick) , in this method first ask user to enter the number of sick that he want to add in array list , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value , then we have (for loop) to enter the number of sick that the user need to add them in the array list , then we take an object from this class(sick )and start with entering the sick id ,

```
1 package BBS;
2 import java.util.ArrayList;
3 import java.util.Scanner;
4 public class sick extends person {
5     public static ArrayList<sick> sickdb = new ArrayList<>();
6     public sick() {
7     }
8
9     public void addSick(Scanner s) {
10        System.out.println("plz enter number of sick");
11        // Start Handel Exception
12        String numberString;
13        int number;
14        numberString = s.next();
15        try {
16            number = Integer.parseInt(s.numberString);
17        } catch (Exception e) {
18            System.out.println("plz enter the integer value");
19            number = s.nextInt();
20        }
21        // End Handel Exception
22        if (number == 0) {
23            System.out.println("error,enter number>0");
24        } else {
25            for (int i = 0; i < number; i++) {
26                sick a = new sick();
27                System.out.println("enter the ID of sick " + (i + 1) + ":" );
28                // Start Handel Exception
29                String idString;
30                int id;
31                idString = s.next();
32                try {
33                    id = Integer.parseInt(s.idString);
34                } catch (Exception e) {
35                    System.out.println("plz enter the integer value");
36                    id = s.nextInt();
37                }
38                // End Handel Exception
39            }
40        }
41    }
42 }
```

and check if this id had been entered to the array list , if no this id of sick will added to the array list .

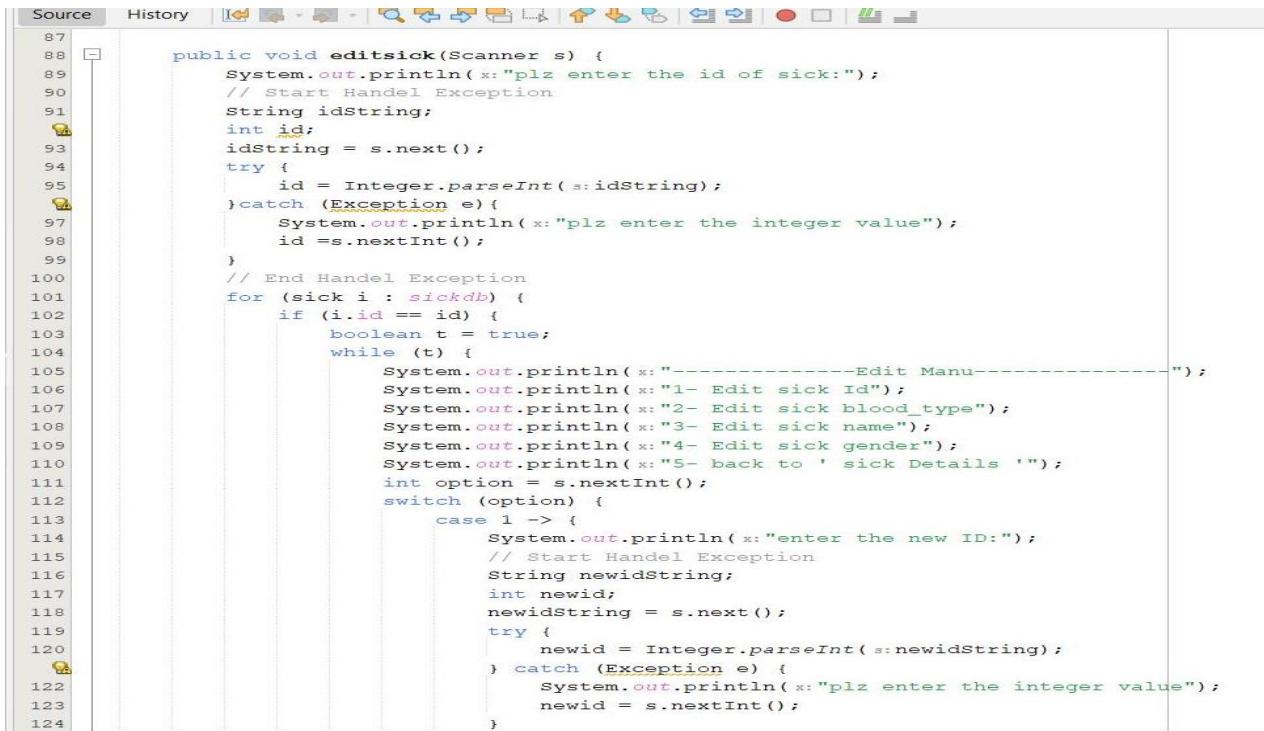
After then ask user to entering the name , gender and blood type of this sick , then add this sick in the array list and so on to each sick.

```
27 // Start Handel Exception
28 string idString;
29 int id;
30 idString = s.next();
31 try {
32     id = Integer.parseInt(s:idString);
33 }catch (Exception e){
34     System.out.println("plz enter the integer value");
35     id = s.nextInt();
36 }
37 // End Handel Exception
38 if (sick.sickdb.isEmpty()) {
39     a.setId(id);
40 } else {
41     for (sick e : sickdb) {
42         if (e.id == id) {
43             System.out.println("Error, this ID is exist");
44             return;
45         }
46     }
47 }
48 String name = s.next();
49 System.out.println("enter the gender of Admin " + (i + 1) + ":");
50 String gender = s.next();
51 System.out.println("enter the bloodtype of Admin " + (i + 1) + ":");
52 String type = s.next();
53 a.setId(id);
54 a.setGender(gender);
55 a.setName(name);
56 a.setBloodtype(bloodtype: type);
57 sick.sickdb.add(e:a);
58 }
59 }
60 }
61 }
62 }
63 }
```

Then we build a method to search about sick , first ask user to enter the sick id to search and use exception Handel when user enter a wrong value and check if the id is true the information about this sick will appear to user , else the id is not available in Array list will tell user this sick isn't available .

```
64 public void searchsick(Scanner s) {
65     System.out.println("plz enter the ID of sick:");
66     // Start Handel Exception
67     String idString;
68     int id;
69     idString = s.next();
70     try {
71         id = Integer.parseInt(s:idString);
72     }catch (Exception e){
73         System.out.println("plz enter the integer value");
74         id = s.nextInt();
75     }
76     // End Handel Exception
77     for (sick i : sickdb) {
78         if (i.id == id) {
79             System.out.println("*****information about Sick*****");
80             this.display(i);
81             return;
82         }
83     }
84     System.out.println("sick unavailable");
85 }
86 }
87 }
```

Then we build a method called(editsick) Then ask user to enter an id of the sick then check if this sick id is already added to Array list . in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value(id), if this id true an edit menu will appear to the user and some options and make a switch to an entered option in this switch case1 to till user to enter the sick id ,

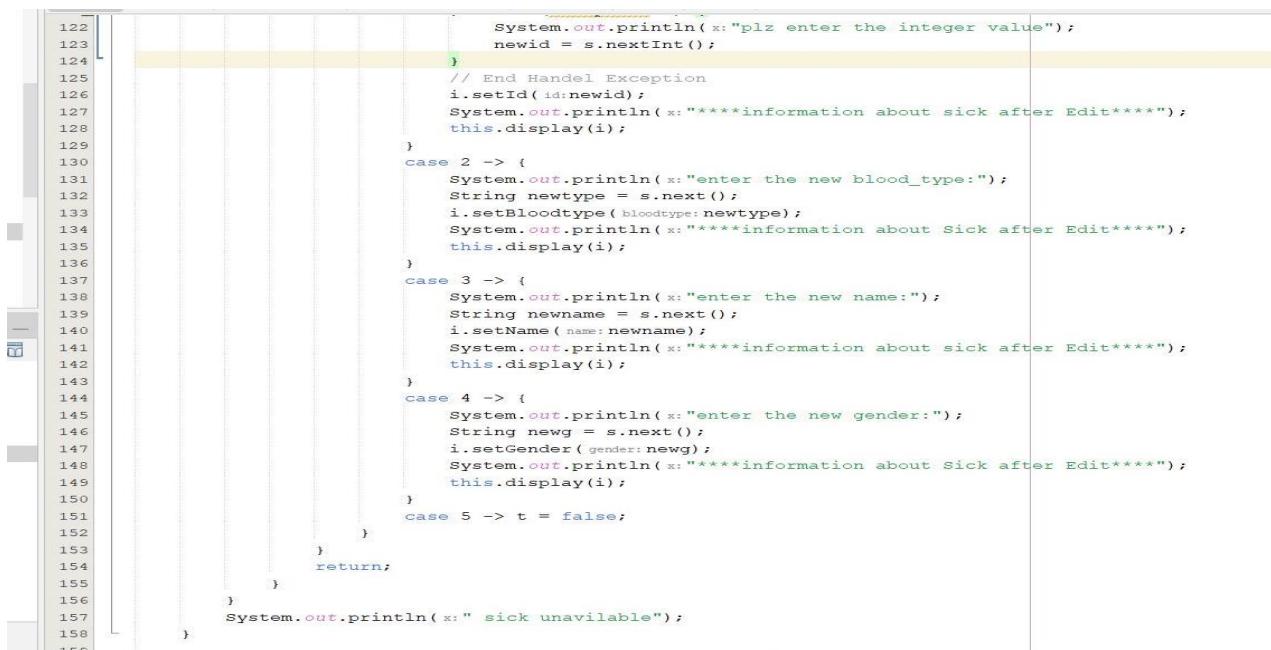


```

87
88     public void editsick(Scanner s) {
89         System.out.println("plz enter the id of sick:");
90         // Start Handel Exception
91         String idString;
92         int id;
93         idString = s.next();
94         try {
95             id = Integer.parseInt(s:idString);
96         } catch (Exception e) {
97             System.out.println("plz enter the integer value");
98             id = s.nextInt();
99         }
100        // End Handel Exception
101        for (sick i : sickdb) {
102            if (i.id == id) {
103                boolean t = true;
104                while (t) {
105                    System.out.println("-----Edit Manu-----");
106                    System.out.println("1- Edit sick Id");
107                    System.out.println("2- Edit sick blood_type");
108                    System.out.println("3- Edit sick name");
109                    System.out.println("4- Edit sick gender");
110                    System.out.println("5- back to ' sick Details '");
111                    int option = s.nextInt();
112                    switch (option) {
113                        case 1 -> {
114                            System.out.println("enter the new ID:");
115                            // Start Handel Exception
116                            String newidString;
117                            int newid;
118                            newidString = s.next();
119                            try {
120                                newid = Integer.parseInt(s:newidString);
121                            } catch (Exception e) {
122                                System.out.println("plz enter the integer value");
123                                newid = s.nextInt();
124                            }
125                        }
126                    }
127                }
128            }
129        }
130    }
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159

```

After edit the id to new id and add it , then print the information after editing .Case 2 to edit sick blood type and a message till user to enter the new blood type and then print the information after editing . Case 3 to edit sick name and a message till user to enter the new name and then print the information after editing . Case 4 to edit sick gender and a message till user to enter the new gender and then print the information after editing. When the id that has been entered is false there is a message to tell user that this sick is unavailable.



```

122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159

```

Then we build a method to delete sick from Array list , first we check if the array list is empty or no , in case no we ask user to enter the id for this sick then check if this sick in the array list ,

case true this sick will be removed from Array list and a message to till user that delete operation done successfully ,

case false the system tell user this sick id is not found .

The screenshot shows a Java code editor with two methods defined:

```
160
161     public void deletesick(Scanner s) {
162         if (sickdb.isEmpty()) {
163             System.out.println(" Not possible ,plz check Database");
164         } else {
165             System.out.println("plz enter the id of sick:");
166             // Start Handel Exception
167             String idString;
168             int id;
169             idString = s.next();
170             try {
171                 id = Integer.parseInt(s.idString);
172             } catch (Exception e) {
173                 System.out.println("plz enter the integer value");
174                 id = s.nextInt();
175             }
176             // End Handel Exception
177             for (sick i : sickdb) {
178                 if (id == i.id) {
179                     sickdb.remove(i);
180                     System.out.println("Suceesfully Done ");
181                     return;
182                 }
183             }
184             System.out.println(" sick unavailable");
185         }
186     }
187
188     public void display(sick i) {
189         System.out.println("name: ID : gender: bloodtype: ");
190         System.out.println(i.getName() + " "
191                           + i.getId() + " "
192                           + i.getGender() + " "
193                           + i.getBloodtype());
194     }
}
```

And a method to print the information that we use in other method .

Now this is the next class (Blood) , that inherit from class (BBclass) as shown

{BBclass} → {Blood}

In this class(Blood )that inherit from class (BBclass), then we define

an string( blood type)and an integer variable(available\_quantity).

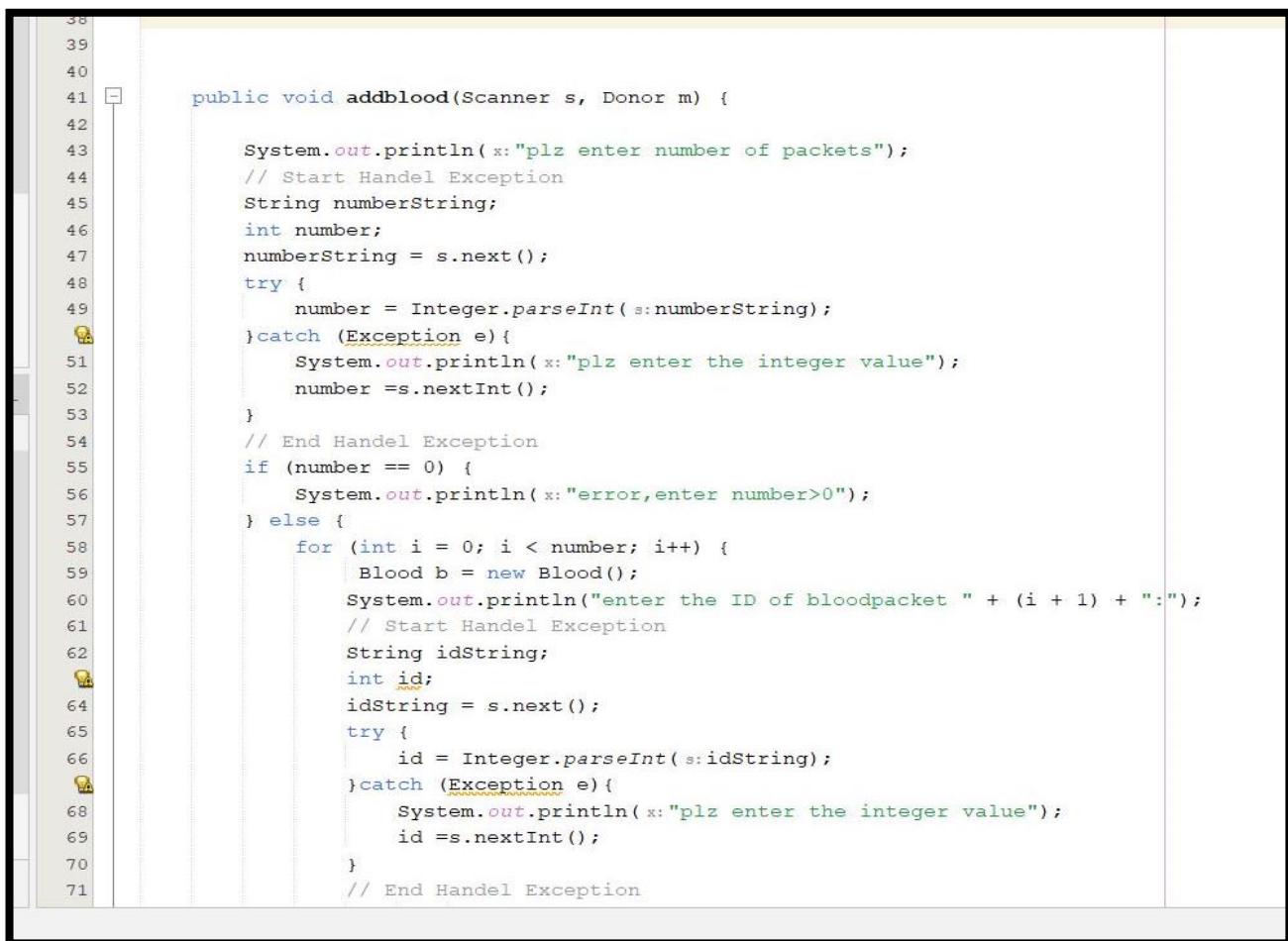
Then we invoke HashSet class , this class is very useful tool that allows you to store unique items and access them in constant time .No duplicate values stored.

And we make an array list called (blooddb )that we will use in this class to store the information for this Blood

```
2 import java.util.HashSet;
3 import java.util.ArrayList;
4 import java.util.Scanner;
5 public class Blood extends BBclass{
6
7     private String blood_type;
8     private int available_quantity;
9     static HashSet<String> AVblood = new HashSet<>();
10    public static ArrayList<Blood> blooddb = new ArrayList<>();
11
12    public Blood() {
13    }
14
15    public Blood(int blood_id, String blood_type, int available_quantity ) {
16        this .id = blood_id;
17        this.blood_type = blood_type;
18        this.available_quantity = available_quantity;
19        this.available_blood_packets = available_blood_packets;
20    }
21    public String getBlood_type() {
22        return blood_type;
23    }
24
25    public void setBlood_type(String blood_type) {
26        this.blood_type = blood_type;
27    }
28
29    public int getAvailable_quantity() {
30        return available_quantity;
31    }
32
33
34    public void setAvailable_quantity(int available_quantity) {
35        this.available_quantity = available_quantity;
36    }
37
38
39
```

First we built a method called (addblood) ,

this method takes two parameter(Scanner ,and an object from donor calss) and we will show why we take an object in this method , first ask user to enter the number of packets that he want to add in array list , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer number , and an is statement to check if this number is equal 0 or no ,then we have (for loop) to enter the number of packets that the user need to add them in the array .

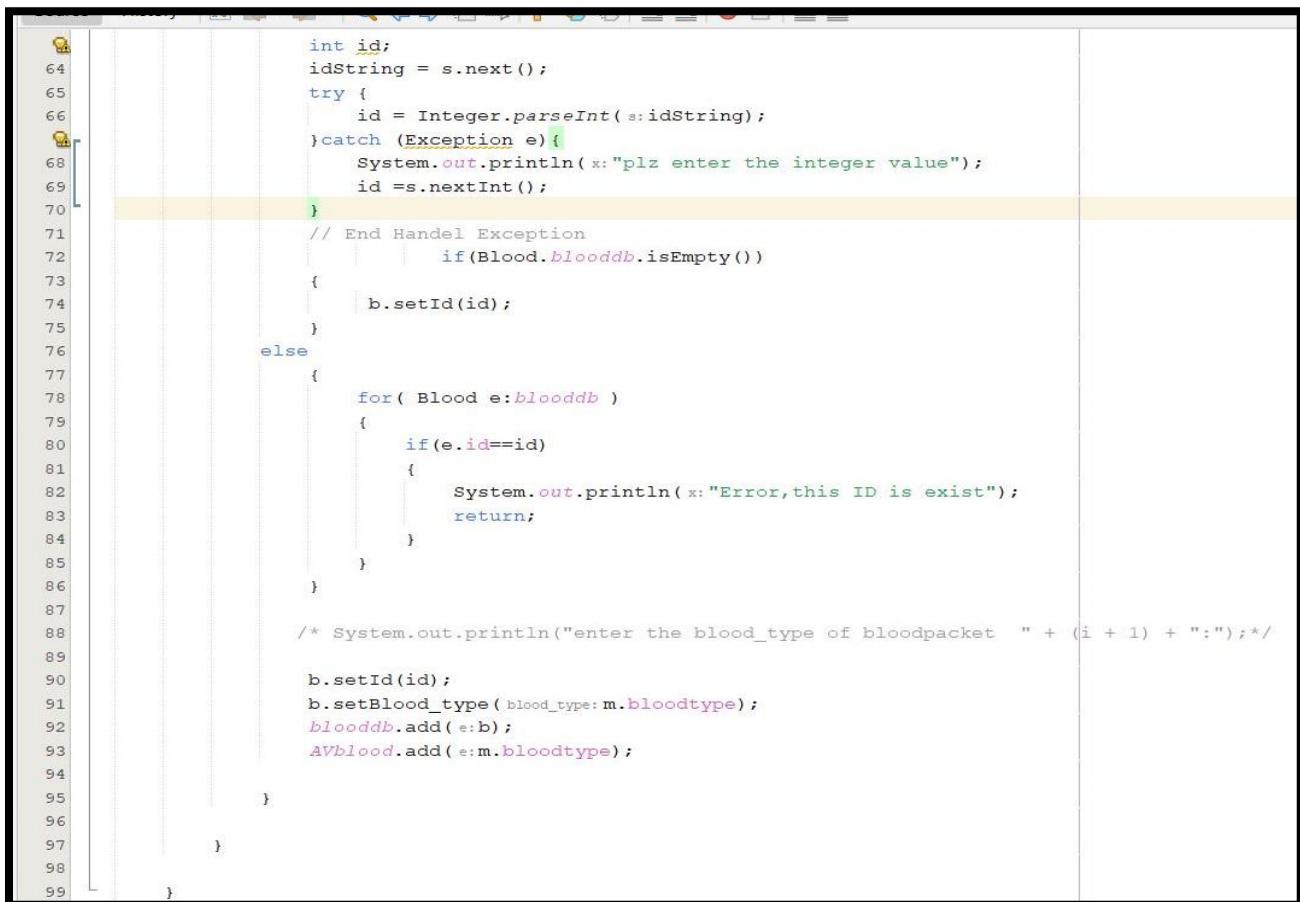


```
38
39
40
41     public void addblood(Scanner s, Donor m) {
42
43         System.out.println("plz enter number of packets");
44         // Start Handel Exception
45         String numberString;
46         int number;
47         numberString = s.nextInt();
48         try {
49             number = Integer.parseInt(s.next());
50         } catch (Exception e) {
51             System.out.println("plz enter the integer value");
52             number = s.nextInt();
53         }
54         // End Handel Exception
55         if (number == 0) {
56             System.out.println("error,enter number>0");
57         } else {
58             for (int i = 0; i < number; i++) {
59                 Blood b = new Blood();
60                 System.out.println("enter the ID of bloodpacket " + (i + 1) + ":");
61                 // Start Handel Exception
62                 String idString;
63                 int id;
64                 idString = s.next();
65                 try {
66                     id = Integer.parseInt(s.next());
67                 } catch (Exception e) {
68                     System.out.println("plz enter the integer value");
69                     id = s.nextInt();
70                 }
71                 // End Handel Exception
72             }
73         }
74     }
75 }
```

in the for we start with take an object from blood class and start to entering the id of the first packet in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value(id),

After entering the id then check if the array list is empty then this blood id is add to the array list ,else we have a for loop to loop on the array list if this id has entered before threr is a message to tell user that this id is already exist in array list.

Then add this packet to array list and store blood type in HashSet to prevent recurrence blood type in there a lot of doner has the same blood type.



The screenshot shows a Java code editor with the following code:

```
int id;
idString = s.next();
try {
    id = Integer.parseInt(s:idString);
} catch (Exception e){
    System.out.println(x:"plz enter the integer value");
    id =s.nextInt();
}
// End Handel Exception
if(Blood.blooddb.isEmpty())
{
    b.setId(id);
}
else
{
    for( Blood e:blooddb )
    {
        if(e.id==id)
        {
            System.out.println(x:"Error,this ID is exist");
            return;
        }
    }
}
/* System.out.println("enter the blood_type of bloodpacket " + (i + 1) + ":");*/
b.setId(id);
b.setBlood_type( blood_type:m.bloodtype);
blooddb.add( e:b );
AVblood.add( e:m.bloodtype );
}
}
}
```

Then we build a method called(editblood ) Then ask user to enter an id of the packet then check if this packet id is already added to Array list . in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value(id), if this id true an edit menu will appear to the user ,after that we define an integer variable (option ) . in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value and some options and make a switch to an entered option in this switch

```

100
101     public void editblood(Scanner s) {
102         System.out.println("plz enter the id of bloodpacket:");
103         // Start Handel Exception
104         String idString;
105         int id;
106         idString = s.next();
107         try {
108             id = Integer.parseInt(s:idString);
109         }catch (Exception e){
110             System.out.println("plz enter the integer value");
111             id =s.nextInt();
112         }
113         // End Handel Exception
114         for (Blood i : blooddb) {
115             if (i.id == id) {
116                 boolean t = true;
117                 while (t) {
118                     System.out.println("-----Edit Manu-----");
119                     System.out.println("1- Edit bloodpacket Id");
120                     System.out.println("2- Edit bloodpacket type");
121                     System.out.println("3- back to ' Blood Details'");
122                     // Start Handel Exception
123                     String optionString;
124                     int option;
125                     optionString = s.next();
126                     try {
127                         option = Integer.parseInt(s:optionString);
128                     }catch (Exception e){
129                         System.out.println("plz enter the integer value");
130                         option =s.nextInt();
131                     }
132                     // End Handel Exception
133                     switch (option) {
134                         case 1 -> {
135                             System.out.println("enter the new ID:");
136                             // Start Handel Exception
137                             String newidString;

```

case1 to till user to enter the packet id ,after edit the id to new id and add it , then print the information after editing .

Case 2 to edit the blood type , and tell user to enter the new blood type and add it to array list

Then print information after editing.

Case3 that the Boolean if is equal false ,then exit the switch .

When the id of the packet isn't in the array list . the program will tell user that this packet is unavailable.

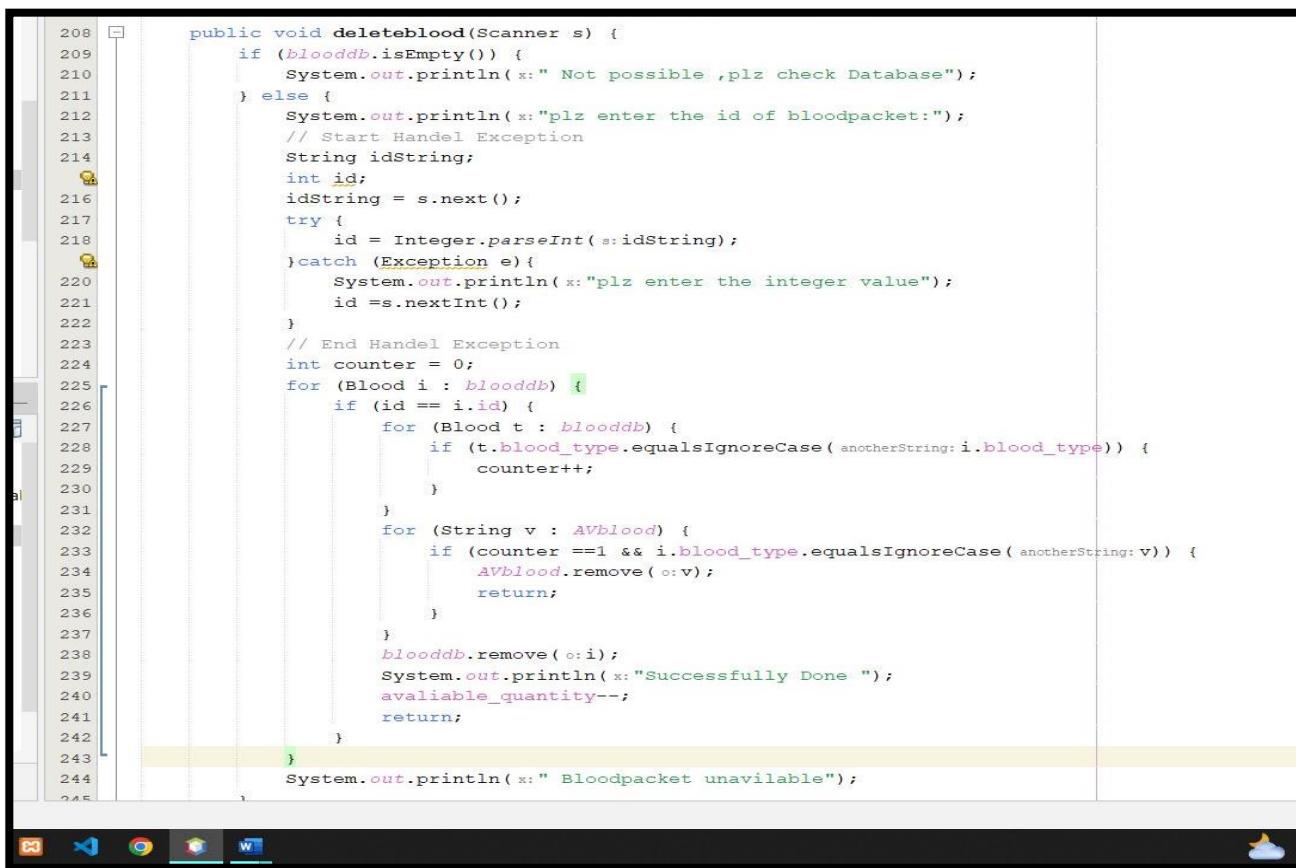
The screenshot shows a Java code editor with the following code:

```
144     newid = s.nextInt();
145
146     // End Handel Exception
147     i.setId( id:newid );
148     System.out.println( x:"****information about bloodpacket after Edit****");
149     this.display(i);
150 }
151 case 2 -> {
152     int c=0;
153     System.out.println( x:"enter the new type:");
154     String newtype = s.next();
155     for(Blood r: blooddb)
156     {
157         if(r.blood_type.equalsIgnoreCase( anotherString: i.blood_type))
158             {c++;}
159     }
160     System.out.println( x:c);
161     if(c==1)
162     { Blood.AVblood.remove( o:i.blood_type);
163     Blood.AVblood.add( e:newtype);}
164     else if(c>1)
165     {
166         Blood.AVblood.add( e:newtype);
167     }
168
169     i.setBlood_type( blood_type: newtype);
170     System.out.println( x:"****information about bloodpacket after Edit****");
171     this.display(i);
172 }
173 case 3 -> t = false;
174 }
175 }
176 return;
177 }
178 }
179 System.out.println( x:" bloodpacket unavailable");
180 }
```

Then we build a method to search about blood , first ask user to enter the packet id to search and use exception Handel when user enter a wrong value and check if the id is true the information about this packet will appear to user , else the id is not available in Array list will tell user this packet isn't available .

```
180
181
182
183     public void searchblood(Scanner s) {
184         System.out.println("plz enter the ID of bloodpacket:");
185         // Start Handel Exception
186         String idString;
187         int id;
188         idString = s.next();
189         try {
190             id = Integer.parseInt(s:idString);
191         }catch (Exception e){
192             System.out.println("plz enter the integer value");
193             id =s.nextInt();
194         }
195         // End Handel Exception
196         for (Blood i : blooddb) {
197             if (i.id == id) {
198
199                 System.out.println("*****information about Bloodpacket*****");
200             this.display(i);
201
202                 return;
203             }
204         }
205         System.out.println("bloodpacket unavailable");
206     }
207 }
```

Then we build a method to delete blood packet from Array list , first we check if the array list is empty or no , in case no we ask user to enter the id for this blood packet in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value(id) , then check if the id of the packet is in the array list then this packet will be removed „if this id is in correct the system tell user that this packet is unavailable.



```

208     public void deleteblood(Scanner s) {
209         if (blooddb.isEmpty()) {
210             System.out.println(" Not possible ,plz check Database");
211         } else {
212             System.out.println("plz enter the id of bloodpacket:");
213             // Start Handel Exception
214             String idString;
215             int id;
216             idString = s.next();
217             try {
218                 id = Integer.parseInt(idString);
219             } catch (Exception e) {
220                 System.out.println("plz enter the integer value");
221                 id = s.nextInt();
222             }
223             // End Handel Exception
224             int counter = 0;
225             for (Blood i : blooddb) {
226                 if (id == i.id) {
227                     for (Blood t : blooddb) {
228                         if (t.blood_type.equalsIgnoreCase(i.blood_type)) {
229                             counter++;
230                         }
231                     }
232                     for (String v : AVblood) {
233                         if (counter == 1 && i.blood_type.equalsIgnoreCase(v)) {
234                             AVblood.remove(v);
235                             return;
236                         }
237                     }
238                     blooddb.remove(i);
239                     System.out.println("Successfully Done ");
240                     available_quantity--;
241                     return;
242                 }
243             }
244             System.out.println(" Bloodpacket unavailable");
}

```

we built a method to display the HashSet Array and check if this set is empty , in case no this method will print the available packet in this set.

And a method to display the information about blood packet .

```
247
248     public void disAvblood() {
249         if (AVblood.isEmpty()) {
250             System.out.println(" unavailable packet ");
251         } else {
252             System.out.println(AVblood + "\n");
253         }
254     }
255     public void display(Blood i) {
256         System.out.println("Id:           type:      ");
257         System.out.println(i.getId() + "           "
258                           + i.getBlood_type() + "      ");
259     }
260
261
262
```

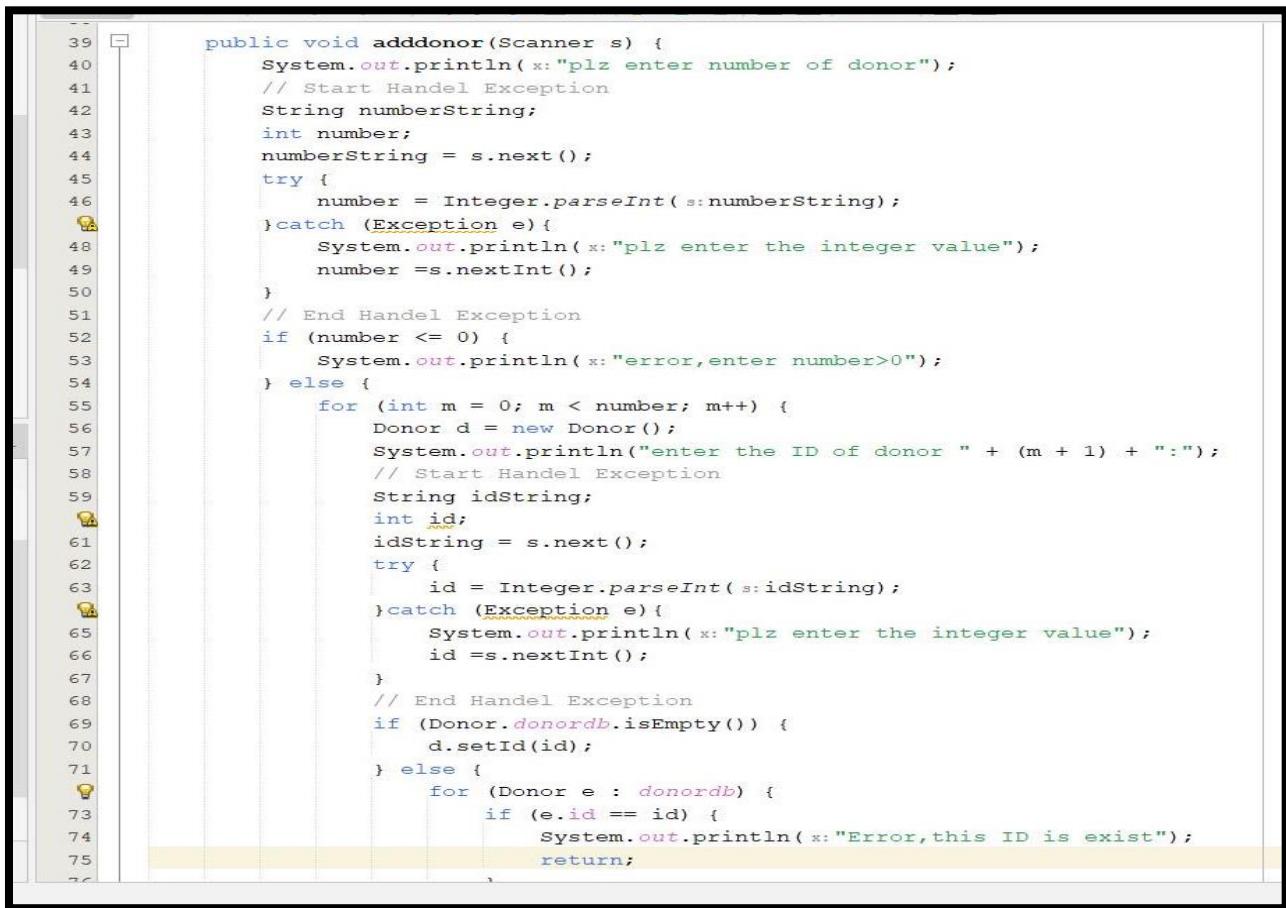
Now this is the next class (doner) , We've been said that we have a multilevel inheritance as shown

{BBclass} → {person} → {Doner}

In this class(doner) that inherit from previous class (person), then we define a, string (doner address) , double (doner age ) ,string(doner disease) , And String(doner mobile) , and make an array list called (donerdb )that we will use in this class to store the information for this donor , then invoke the constructors for this class ,setters and getters .

```
1 package BBS;
2 import java.util.ArrayList;
3 import java.util.Scanner;
4 public class Donor extends person{
5     private String donor_address;
6     private double donor_age;
7     private String donor_mobile;
8     private String donor_disease;
9     public static ArrayList<Donor> donordb = new ArrayList<>();
10    public Donor() {}
11    public Donor( String donor_address, double donor_age, String donor_mobile, String donor_disease) {
12        this.donor_address = donor_address;
13        this.donor_age = donor_age;
14        this.donor_mobile = donor_mobile;
15        this.donor_disease = donor_disease;
16    }
17    public String getDonor_address() {
18        return donor_address;
19    }
20    public void setDonor_address(String donor_address) {
21        this.donor_address = donor_address;
22    }
23    public double getDonor_age() {
24        return donor_age;
25    }
26    public void setDonor_age(double donor_age) {
27        this.donor_age = donor_age;
28    }
29    public String getDonor_mobile() {
30        return donor_mobile;
31    }
32    public void setDonor_mobile(String donor_mobile) {
33        this.donor_mobile = donor_mobile;
34    }
35    public String getDonor_disease() {
36        return donor_disease;
37    }
38    public void setDonor_disease(String donor_disease) {
39        this.donor_disease = donor_disease;
40    }
41 }
```

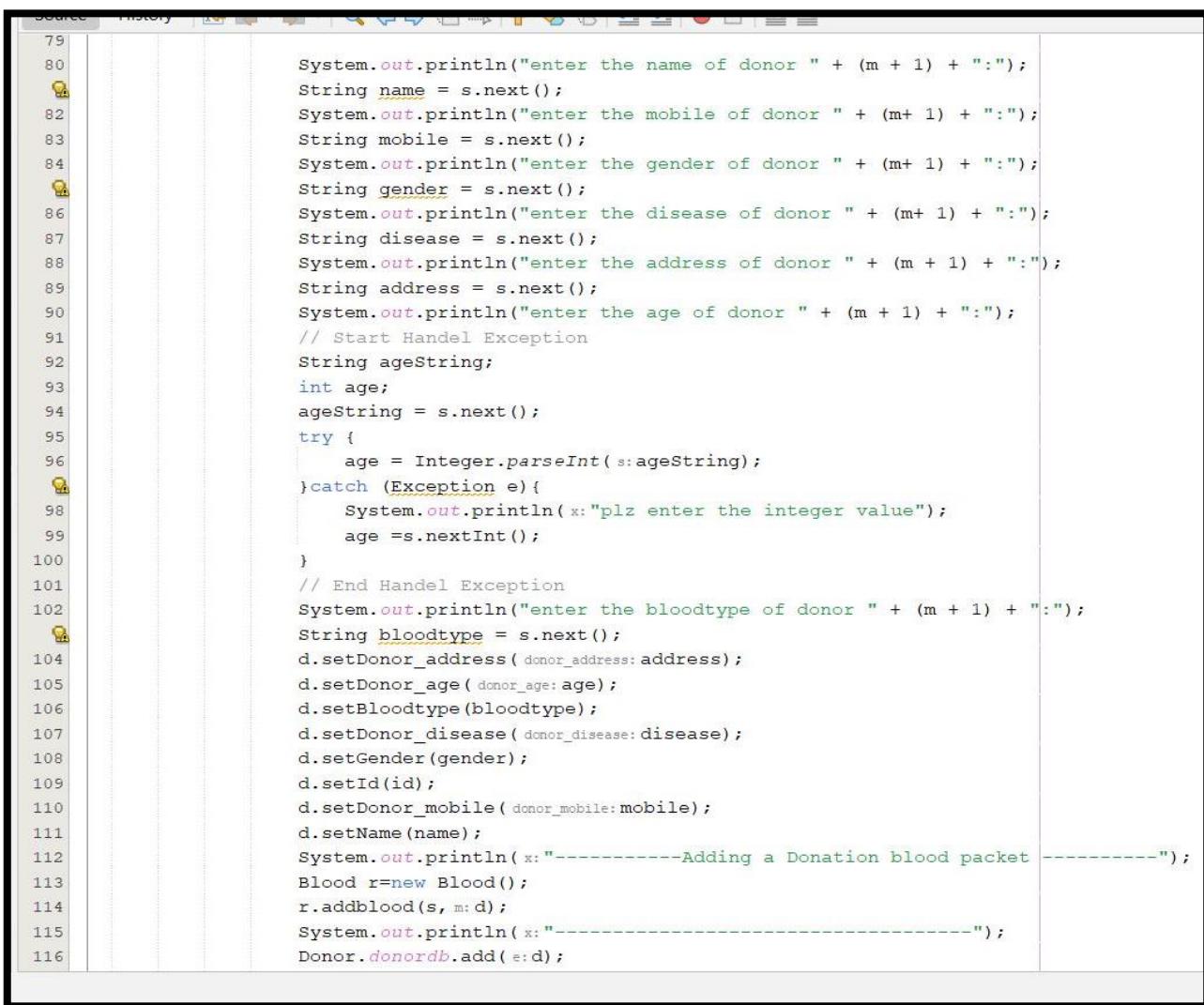
First we built a method called (adddoner) , in this method first ask user to enter the number of donor that he want to add in array list , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value , then we have (for loop) to enter the number of donor that the user need to add them in the array list , then we take an object from this class(donor )and start with entering the donor id , and check if this id had been entered to the array list , if no this id of donor will added to the array list .



```
39     public void adddonor(Scanner s) {
40         System.out.println("plz enter number of donor");
41         // Start Handel Exception
42         String numberString;
43         int number;
44         numberString = s.nextInt();
45         try {
46             number = Integer.parseInt(s.next());
47         } catch (Exception e) {
48             System.out.println("plz enter the integer value");
49             number = s.nextInt();
50         }
51         // End Handel Exception
52         if (number <= 0) {
53             System.out.println("error,enter number>0");
54         } else {
55             for (int m = 0; m < number; m++) {
56                 Donor d = new Donor();
57                 System.out.println("enter the ID of donor " + (m + 1) + ":");
58                 // Start Handel Exception
59                 String idString;
60                 int id;
61                 idString = s.next();
62                 try {
63                     id = Integer.parseInt(s.next());
64                 } catch (Exception e) {
65                     System.out.println("plz enter the integer value");
66                     id = s.nextInt();
67                 }
68                 // End Handel Exception
69                 if (Donor.donordb.isEmpty()) {
70                     d.setId(id);
71                 } else {
72                     for (Donor e : donordb) {
73                         if (e.id == id) {
74                             System.out.println("Error,this ID is exist");
75                             return;
76                         }
77                     }
78                 }
79             }
80         }
81     }
```

After then ask user to entering the name, mobile ,disease , address, and age of this donor , then Exception Handling start in the event that user enter the value of age any value expect an integer value ,then solve this error and a message to warned user to entering an integer value , After that a message to tell user to adding a donation blood packets ,then we take an object from blood class to be able to add this donor blood packets to array list.

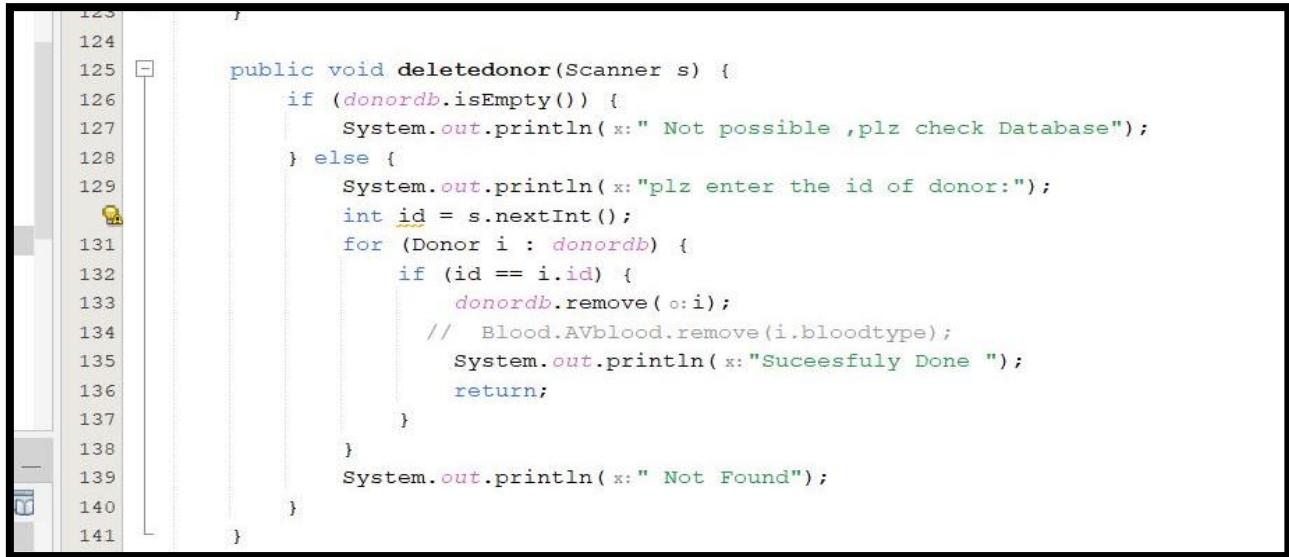
After entering all data about donor , add this data in the array list ,



```
79
80         System.out.println("enter the name of donor " + (m + 1) + ":" );
81         String name = s.next();
82         System.out.println("enter the mobile of donor " + (m+ 1) + ":" );
83         String mobile = s.next();
84         System.out.println("enter the gender of donor " + (m+ 1) + ":" );
85         String gender = s.next();
86         System.out.println("enter the disease of donor " + (m+ 1) + ":" );
87         String disease = s.next();
88         System.out.println("enter the address of donor " + (m + 1) + ":" );
89         String address = s.next();
90         System.out.println("enter the age of donor " + (m + 1) + ":" );
91         // Start Handel Exception
92         String ageString;
93         int age;
94         ageString = s.next();
95         try {
96             age = Integer.parseInt(s:ageString);
97         }catch (Exception e){
98             System.out.println( x:"plz enter the integer value");
99             age =s.nextInt();
100        }
101        // End Handel Exception
102        System.out.println("enter the bloodtype of donor " + (m + 1) + ":" );
103        String bloodtype = s.next();
104        d.setDonor_address( donor_address:address);
105        d.setDonor_age( donor_age:age);
106        d.setBloodtype(bloodtype);
107        d.setDonor_disease( donor_disease:disease);
108        d.setGender(gender);
109        d.setId(id);
110        d.setDonor_mobile( donor_mobile:mobile);
111        d.setName(name);
112        System.out.println( x:"-----Adding a Donation blood packet -----");
113        Blood r=new Blood();
114        r.addblood(s, m:d);
115        System.out.println( x:"-----");
116        Donor.donordb.add( e:d);
```

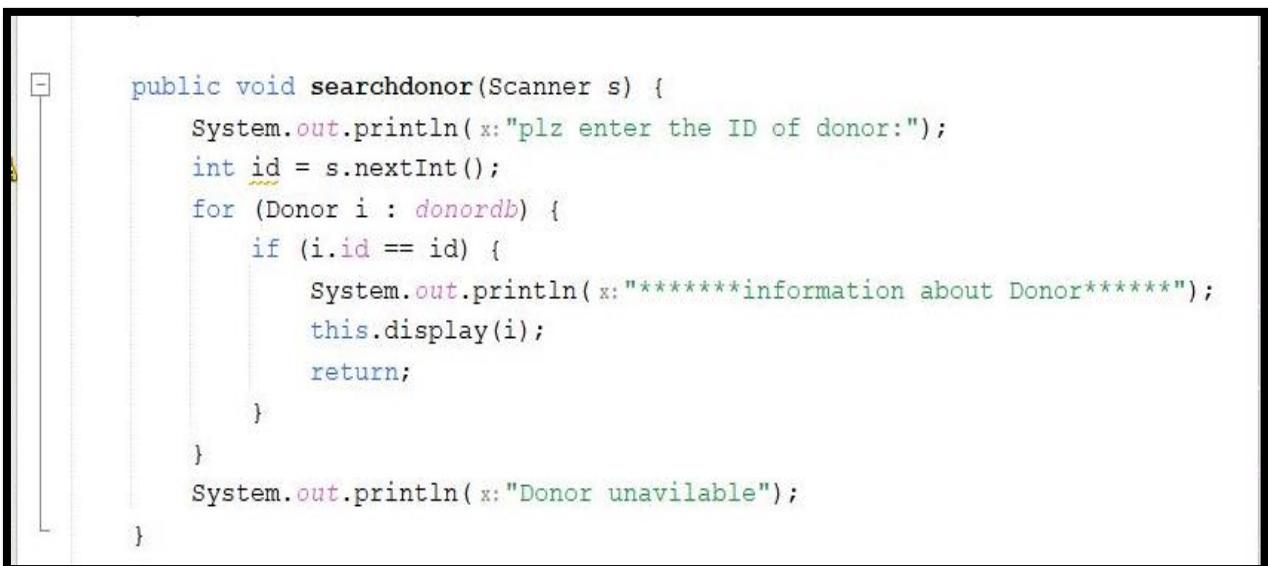
All of these operation will repeated if user need to enter a lot of one donor .

Then we build a method to delete donor from array list , first we check if the array list is empty or no , in case no we ask user to enter the id for this donor then check if this donor in the array list , case true this donor will be removed from array list , case false the system tell user this donor id is not found .



```
123
124
125     public void deletedonor(Scanner s) {
126         if (donordb.isEmpty()) {
127             System.out.println(" Not possible ,plz check Database");
128         } else {
129             System.out.println("plz enter the id of donor:");
130             int id = s.nextInt();
131             for (Donor i : donordb) {
132                 if (id == i.id) {
133                     donordb.remove(i);
134                     // Blood.AVblood.remove(i.bloodtype);
135                     System.out.println("Suceesfully Done ");
136                     return;
137                 }
138             }
139             System.out.println(" Not Found");
140         }
141     }
```

Then we build a method to search about donor , first ask user to enter the donor id to search and check if the id is true the information about this donor will appear to user , else the id is not available in array list, program will tell user this donor isn't available .



```
public void searchdonor(Scanner s) {
    System.out.println("plz enter the ID of donor:");
    int id = s.nextInt();
    for (Donor i : donordb) {
        if (i.id == id) {
            System.out.println("*****information about Donor*****");
            this.display(i);
            return;
        }
    }
    System.out.println("Donor unavailable");
}
```

Then we build a method called(editdoner) Then ask user to enter an id of the donor then check if this donor id is already added to arraylist . in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value(id), if this id true an edit menu will appear to the user ,

```
155
156     public void editdonor(Scanner s) {
157         System.out.println("plz enter the id of donor:");
158         // Start Handel Exception
159         String idString;
160         int id;
161         idString = s.next();
162         try {
163             id = Integer.parseInt(s:idString);
164         }catch (Exception e){
165             System.out.println("plz enter the integer value");
166             id =s.nextInt();
167         }
168         // End Handel Exception
169         for (Donor i : donordb) {
170             if (i.id == id) {
171                 boolean t = true;
172                 while (t) {
173                     System.out.println("-----Edit Manu-----");
174                     System.out.println("1- Edit donor name");
175                     System.out.println("2- Edit donor id");
176                     System.out.println("3- Edit donor bloodtype");
177                     System.out.println("4- Edit donor disease");
178                     System.out.println("5- Edit donor age");
179                     System.out.println("6- Edit donor address");
180                     System.out.println("7- back to 'donor Details '");
181                     // Start Handel Exception
182                     String optionString;
183                     int option;
184                     optionString = s.next();
185                     try {
186                         option = Integer.parseInt(s:optionString);
187                     }catch (Exception e){
188                         System.out.println("plz enter the integer value");
189                         option =s.nextInt();
190                     }
191                     // End Handel Exception
192                     switch (option) {
```

After that we define an integer variable (option) . in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value . then we make a switch to an entered option in this switch .

case1 to till user to enter the new doner name, and add it ,then print information after editing.

Case2 to till user to enter the new doner id ,and add it ,then print information after editing

.Case3 to till user to enter the new doner blood type ,and add it ,then print information after editing.

Case4 to till user to enter the new doner blood type ,and add it ,then print information after editing

```
192         switch (option) {
193             case 1 -> {
194                 System.out.println("enter the new name:");
195                 String newname = s.next();
196                 i.setName(name: newname);
197                 System.out.println("****information about Donor after Edit****");
198                 this.display(i);
199             }
200             case 2 -> {
201                 System.out.println("enter the new id:");
202                 // Start Handel Exception
203                 String newidString;
204                 int newid;
205                 newidString = s.next();
206                 try {
207                     newid = Integer.parseInt(s:newidString);
208                 } catch (Exception e) {
209                     System.out.println("plz enter the integer value");
210                     newid = s.nextInt();
211                 }
212                 // End Handel Exception
213                 i.setId(id: newid);
214                 System.out.println("****information about Donor after Edit****");
215                 this.display(i);
216             }
217             case 3 -> {
218                 System.out.println("enter the new bloodtype:");
219                 String newbloodtype = s.next();
220                 i.setBloodtype(bloodtype: newbloodtype);
221                 System.out.println("****information about Donor after Edit****");
222                 this.display(i);
223             }
224             case 4 -> {
225                 System.out.println("enter the new disease:");
226                 String newdisease = s.next();
227                 i.setDonor_disease(donor_disease: newdisease);
228                 System.out.println("****information about Donor after Edit****");
229                 this.display(i);
```

Case5 to till user to enter the new doner age and add it ,then print information after editing.

Case 6 to till user to enter the new doner address, and add it ,then print information after editing.

After the end of all cases , When the id that has been entered is false there is a message to tell user that this donor is unavailable.

```
Source History |             
230 }  
231 case 5 -> {  
232     System.out.println(x:"enter the new age:");  
233     double newage = s.nextDouble();  
234     i.setDonor_age(donor_age:newage);  
235     System.out.println(x:"****information about Donor after Edit****");  
236     this.display(i);  
237 }  
238 case 6 -> {  
239     System.out.println(x:"enter the new address:");  
240     String newaddress = s.next();  
241     i.setDonor_address(donor_address:newaddress);  
242     System.out.println(x:"****information about Donor after Edit****");  
243     this.display(i);  
244 }  
245         case 7 -> t = false;  
246     }  
247 }  
248 return;  
249 }  
250 }  
251 System.out.println(x:" Donor unavailable");  
252 }  
253  
254 public void display(Donor i) {  
255     System.out.println(x:"name:      age :      gender:      address:      Bloodtype:      disease:");  
256     System.out.println(i.getName() + "      " +  
257         + i.getDonor_age() + "      " +  
258         + i.getGender() + "      " +  
259         + i.getDonor_address() + "      " +  
260         + i.getBloodtype() + "      " +  
261         + i.getDonor_disease());  
262 }  
263 }  
264 }
```

And a method to print the information that we use in other method .

Now this is the next class (Blood request ) , that inherit from (BBclass) shown

{BBclass} → {Blood Request}

In this class(doner) we define a,string (request date) ,string(request blood type) , And an integer(blood Quantity) , and make an array list called (reqdb )that we will use in this class to store the information for this request , then invoke the constructors for this class ,setters and getters .

```
1 package BBS;
2 import static BBS.sick.sickdb;
3 import java.util.ArrayList;
4 import java.util.Scanner;
5 public class BloodRequest extends BBclass {
6
7     private String request_date;
8     private String request_bloodtype;
9     private int blood_quantity;
10    public static ArrayList<BloodRequest> reqdb = new ArrayList<>();
11
12    public BloodRequest() {
13
14    }
15    public BloodRequest(int request_number, String request_date, String request_bloodtype, int blood_quantity) {
16        this.id = request_number;
17        this.request_date = request_date;
18        this.request_bloodtype = request_bloodtype;
19        this.blood_quantity = blood_quantity;
20    }
21    public String getRequest_date() {
22        return request_date;
23    }
24    public void setRequest_date(String request_date) {
25        this.request_date = request_date;
26    }
27    public String getRequest_bloodtype() {
28        return request_bloodtype;
29    }
30    public void setRequest_bloodtype(String request_bloodtype) {
31        this.request_bloodtype = request_bloodtype;
32    }
33    public int getBlood_quantity() {
34        return blood_quantity;
35    }
36    public void setBlood_quantity(int blood_quantity) {
37        this.blood_quantity = blood_quantity;
38    }
}
```

And we import the class sick and array list (sickdb) that will be use here in this class.

First we built a method called (addrequest) , in this method first ask user to enter the number of requests that he want to add in array list , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value , then we have (for loop) to enter the number of requests that the user need to add them in the array list ,then we take an object from this class(Blood request) and start with entering the request id , and check if this id had been entered to the array list ,program will tell user that this id is exist , if no the array list is empty this id of request will added to the array list .

```

Source History
39     public void addrequest(Scanner s) {
40         System.out.println("plz enter number of request");
41         // Start Handel Exception
42         String numberString;
43         int number;
44         numberString = s.next();
45         try {
46             number = Integer.parseInt(s:numberString);
47         }catch (Exception e){
48             System.out.println("plz enter the integer value");
49             number =s.nextInt();
50         }
51         // End Handel Exception
52         if (number <= 0) {
53             System.out.println("error,enter number>0");
54         } else {
55             for (int i = 0; i < number; i++) {
56                 BloodRequest a = new BloodRequest();
57                 System.out.println("enter the ID of Request " + (i + 1) + ":");
58                 // Start Handel Exception
59                 String numString;
60                 int num;
61                 numString = s.next();
62                 try {
63                     num = Integer.parseInt(s:numString);
64                 }catch (Exception e){
65                     System.out.println("plz enter the integer value");
66                     num =s.nextInt();
67                 }
68                 // End Handel Exception
69                 if (BloodRequest.reqdb.isEmpty()) {
70                     a.setId(id:num);
71                 } else {
72                     for (BloodRequest e : reqdb) {
73                         if (e.id == num) {
74                             System.out.println("Error,this ID is exist");
75                             return;
76                         }
77                     }
78                 }
79             }
80         }
81     }
82 }
```

Then entering the request date , After that ask user to enter the id of the sick that we has import in this class , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value , then we check if the sick array list is empty ,System will tell user that this sick not found

else we have a loop on array list (sickdb) have been entered is equal the id that exist in the array list then will print the require blood type and then set this blood type request in the array list , and then ask user to entering the number of required packet

```
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

    }

}

System.out.println("enter the date of Request " + (i + 1) + ":");

String date = s.next();
System.out.println("plz enter the ID of sick:");
// Start Handel Exception
String idString;
int id;
idString = s.next();
try {
    id = Integer.parseInt(idString);
} catch (Exception e) {
    System.out.println("plz enter the integer value");
    id = s.nextInt();
}
// End Handel Exception
int u=0;
if( sickdb.isEmpty())
{ System.out.println(" No Sick Found"); return; }

for (sick l : sickdb) {

    if (l.id == id) {
        System.out.println(" the required blood_type " + ":" + l.bloodtype);
        String type = l.bloodtype;
        a.setRequest_bloodtype(request_bloodtype:type);
        u++;
        break;
    }
}

if(u==0)
{
    System.out.println(" No Sick Found"); return;
}

System.out.println("enter the number of required packet " + (i + 1) + ":");

// Start Handel Exception
String quanString;
```

after entering the data about request then add it in the array list(reqdb).And check if the HashSet (AVblood) contains the request blood type of the object that we take it from blood request class , in case true then will print that the request is accepted , in case false then print this blood type of request isn't available.

```
107
108
109
110
111
112     System.out.println(x:" No Sick Found"); return;
113
114
115
116
117
118     String quanString;
119
120     int quan;
121
122     quanString = s.next();
123
124     try {
125         quan = Integer.parseInt(x:quanString);
126     }catch (Exception e){
127         System.out.println(x:"plz enter the integer value");
128         quan =s.nextInt();
129     }
130
131
132
133
134 }
```

Then we build a method to delete request from array list , first we check if the array list is empty or no , in case no we ask user to enter the id for this request then check if this request in the array list , case true this request will be removed from array list , case false the system tell user this request id is not found .

```

137
138     public void deleterequest(Scanner s) {
139         if (reqdb.isEmpty()) {
140             System.out.println(" Not poosible ,plz check Database");
141         } else {
142             System.out.println("plz enter the ID of bloodpacket:");
143             // Start Handel Exception
144             String numString;
145             int num;
146             numString = s.next();
147             try {
148                 num = Integer.parseInt(s:numString);
149             }catch (Exception e){
150                 System.out.println("plz enter the integer value");
151                 num =s.nextInt();
152             }
153             // End Handel Exception
154             for (BloodRequest i : reqdb) {
155                 if (num == i.id) {
156                     reqdb.remove(i);
157                     System.out.println("Suceesfully Done");
158                     return;
159                 }
160             }
161         }
162     }
163 }
164

```

Then we build a method called(editrequest) Then ask user to enter an id of the request then check if this id is already added to array list . in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value(id), if this id true an edit menu will appear to the user ,

```

164
165     public void editrequest(Scanner s) {
166         System.out.println("plz enter the ID of bloodRequest:");
167         // Start Handel Exception
168         String numString;
169         int num;
170         numString = s.next();
171         try {
172             num = Integer.parseInt(s:numString);
173         }catch (Exception e){
174             System.out.println("plz enter the integer value");
175             num =s.nextInt();
176         }
177         // End Handel Exception
178         for (BloodRequest i : reqdb) {
179             if (i.id == num) {
180                 boolean t = true;
181                 while (t) {
182                     System.out.println("-----Edit Manu-----");
183                     System.out.println("1- Edit ID");
184                     System.out.println("2- Edit bloodtype ");
185                     System.out.println("3- Edit date ");
186                     System.out.println("4- Edit quantity ");
187                     System.out.println("5- back to ' Bloodrequest Details' ");
188                     // Start Handel Exception
189                     String optionString;
190                     int option;
191                     optionString = s.next();
192                     try {
193                         option = Integer.parseInt(s:optionString);
194                     }catch (Exception e){
195                         System.out.println("plz enter the integer value");
196                         option =s.nextInt();
197                     }
198                     // End Handel Exception

```

After that we define an integer variable (option) . in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value . then we make a switch to an entered option in this switch .

case1 to till user to enter the new request id , and add it ,then print information after editing.

Case2 to till user to enter the new type ,and add it ,then print information after editing

.Case3 to till user to enter the new date ,and add it ,then print information after editing.

```
193         option = Integer.parseInt(s.optionString);
194     }catch (Exception e){
195         System.out.println("plz enter the integer value");
196         option = s.nextInt();
197     }
198 // End Handel Exception
199 switch (option) {
200     case 1 -> {
201         System.out.println("enter the new ID:");
202         // Start Handel Exception
203         String newnumString;
204         int newnum;
205         newnumString = s.next();
206         try {
207             newnum = Integer.parseInt(s.newnumString);
208         } catch (Exception e) {
209             System.out.println("plz enter the integer value");
210             newnum = s.nextInt();
211         }
212         // End Handel Exception
213         i.setId(id:newnum);
214         System.out.println("****information about Request after Edit****");
215         this.display(i);
216     }
217     case 2 -> {
218         System.out.println("enter the new type:");
219         String newtype = s.next();
220         i.setRequest_bloodtype(request_bloodtype:newtype);
221         System.out.println("****information about Request after Edit****");
222         this.display(i);
223     }
224     case 3 -> {
225         System.out.println("enter the new date:");
226         String date = s.next();
227         i.setRequest_date(request_date:date);
228         System.out.println("****information about Request after Edit****");
229         this.display(i);
230     }
}
```

After the end of all cases , When the id that has been entered is false there is a message to tell user that this doner is unavailable.

```
case 4 -> {
    System.out.println("enter the new date:");
    // Start Handel Exception
    String qunString;
    int qun;
    qunString = s.next();
    try {
        qun = Integer.parseInt(s.qunString);
    } catch (Exception e) {
        System.out.println("plz enter the integer value");
        qun = s.nextInt();
    }
    // End Handel Exception
    i.setBlood_quantity(blood_quantity:qun);
    System.out.println("****information about Request after Edit****");
    this.display(i);
}
case 5 -> t = false;
}
}
return;
}
}
System.out.println(" bloodpacket unavailable");
```

Then we build a method to search about request first ask user to enter the request id to search and check if the id is true the information about this request will appear to user , else the id is not available in array list, program will tell user this request isn't available

```
256
257     public void searchrequest(Scanner s) {
258
259         System.out.println("plz enter the ID of bloodRequest:");
260         // Start Handel Exception
261         String numString;
262         int num;
263         numString = s.next();
264         try {
265             num = Integer.parseInt(s.nextLine());
266         }catch (Exception e){
267             System.out.println("plz enter the integer value");
268             num = s.nextInt();
269         }
270         // End Handel Exception
271         for (BloodRequest i : reqdb) {
272             if (i.id == num) {
273                 System.out.println("*****information about BloodRequest*****");
274                 this.display(i);
275
276                 return;
277             }
278         }
279         System.out.println("bloodrequest unavailable");
280     }
281
282     public void display(BloodRequest i) {
283         System.out.println("ID:          date:          type:          quantity:      ");
284         System.out.println(i.id + "          "
285                             + i.getRequest_date() + "          "
286                             + i.getRequest_bloodtype() + "          " + i.getBlood_quantity()
287                         );
288     }
289
290 }
291
```

And a method to print the information that we use in other method

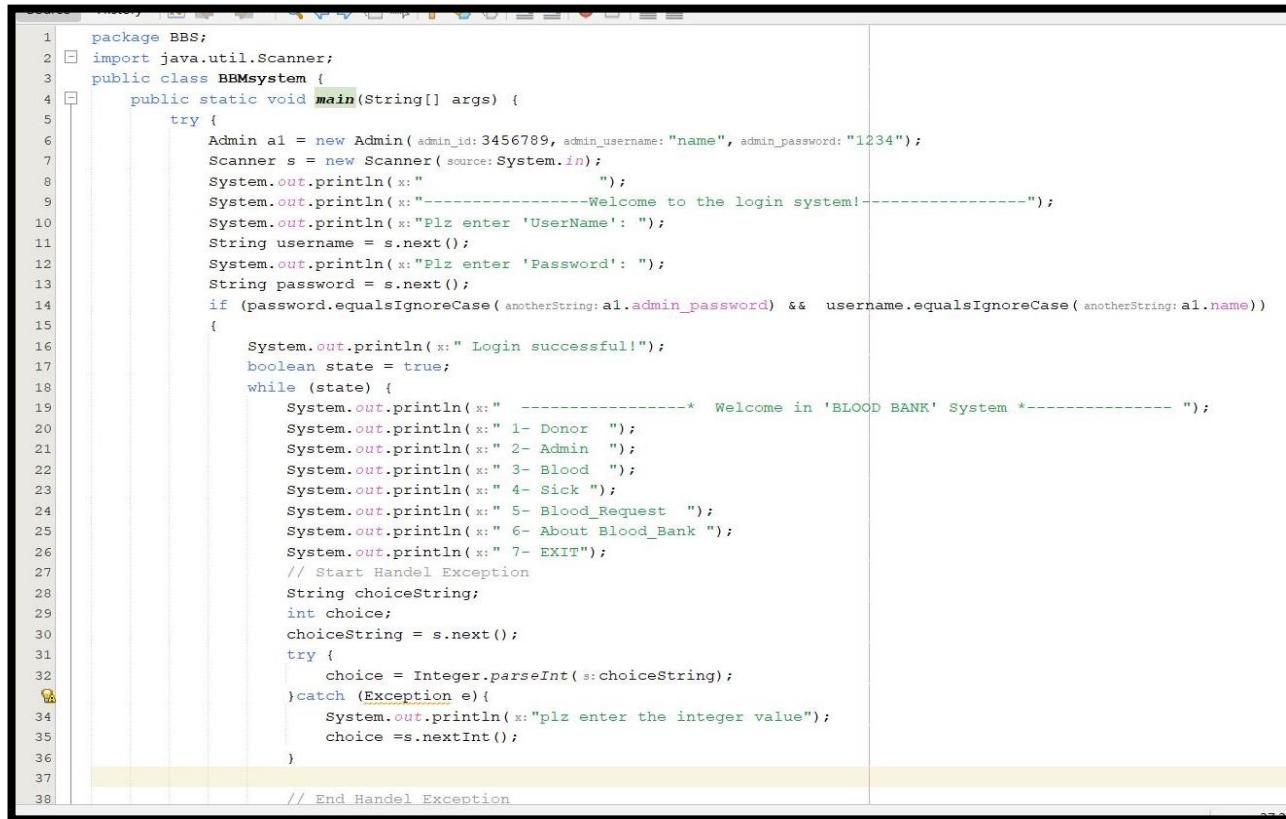
Now this Class(Blood Bank)

In this class we define an integer (Blood Bank id) , string(Blood Bank email) , string(Blood Bank location ) ,and a string (Blood Bank website) and five them their initial values , then invoke the constructors for this class ,setters and getters .

Then we built a method to display all information about Blood Bank .

```
Source History |  1 package BBS;
2 public class BloodBank {
3     int BloodBank_id=987644;
4     String BloodBank_email= "KSABloodbank99@gmail.com";
5     String BloodBank_location="KSA,Riyadh";
6     String BloodBank_website="https://www.kfshrc.edu.sa/en/home/giving/blood";
7     public BloodBank() {}
8     public BloodBank(int BloodBank_id, String BloodBank_email, String BloodBank_location, String BloodBank_website) {
9         this.BloodBank_id = BloodBank_id;
10        this.BloodBank_email = BloodBank_email;
11        this.BloodBank_location = BloodBank_location;
12        this.BloodBank_website = BloodBank_website;
13    }
14    public int getBloodBank_id() {
15        return BloodBank_id;
16    }
17    public void setBloodBank_id(int BloodBank_id) {
18        this.BloodBank_id = BloodBank_id;
19    }
20    public String getBloodBank_email() {
21        return BloodBank_email;
22    }
23    public void setBloodBank_email(String BloodBank_email) {
24        this.BloodBank_email = BloodBank_email;
25    }
26    public String getBloodBank_location() {
27        return BloodBank_location;
28    }
29    public void setBloodBank_location(String BloodBank_location) {
30        this.BloodBank_location = BloodBank_location;
31    }
32    public String getBloodBank_website() {
33        return BloodBank_website;
34    }
35    public void setBloodBank_website(String BloodBank_website) {
36        this.BloodBank_website = BloodBank_website;
37    }
38    public void displayBBInfo()
39    {
40        System.out.println( " Bloodbank ID : " + this.getBloodBank_id());
41        System.out.println( " Bloodbank Location : " +this.getBloodBank_location());
42        System.out.println( " Bloodbank Website : " +this.getBloodBank_website());
43        System.out.println( " Bloodbank Email : " +this.getBloodBank_email());
44    }
45}
```

Now in this class (BBMsystem) , we use the main method and start with take an object from Admin class , and gave it this value (An id ,username and password ) . then we define a scanner to receive data from user .and there is a welcome message and ask user to entering the admin username and also his password then check if this username and password that user entered is equal to the data that we entered when we define an object from Admin , while true The login is successful , while these username and password is wrong the program will stop ,then print that there is an error



```

1 package BBS;
2 import java.util.Scanner;
3 public class BBMsystem {
4     public static void main(String[] args) {
5         try {
6             Admin a1 = new Admin(admin_id: 3456789, admin_username: "name", admin_password: "1234");
7             Scanner s = new Scanner(source: System.in);
8             System.out.println(x: " " );
9             System.out.println(x: "-----Welcome to the login system!-----");
10            System.out.println(x: "Plz enter 'UserName': ");
11            String username = s.next();
12            System.out.println(x: "Plz enter 'Password': ");
13            String password = s.next();
14            if (password.equalsIgnoreCase(anotherString: a1.admin_password) && username.equalsIgnoreCase(anotherString: a1.name))
15            {
16                System.out.println(x: " Login successful!");
17                boolean state = true;
18                while (state) {
19                    System.out.println(x: " -----* Welcome in 'BLOOD BANK' System *----- ");
20                    System.out.println(x: " 1- Donor ");
21                    System.out.println(x: " 2- Admin ");
22                    System.out.println(x: " 3- Blood ");
23                    System.out.println(x: " 4- Sick ");
24                    System.out.println(x: " 5- Blood_Request ");
25                    System.out.println(x: " 6- About_Blood_Bank ");
26                    System.out.println(x: " 7- EXIT");
27                    // Start Handel Exception
28                    String choiceString;
29                    int choice;
30                    choiceString = s.next();
31                    try {
32                        choice = Integer.parseInt(s:choiceString);
33                    } catch (Exception e){
34                        System.out.println(x: "plz enter the integer value");
35                        choice = s.nextInt();
36                    }
37                    // End Handel Exception
38                }
39            }
40        }
41    }
42 }

```

When the login is successful ,then we define a Boolean state and we have a loop as the state is equal true The system will start with welcome message to Blood Bank System , and a list of choice will appear to user all of these choices is arranged by integer values , then we define an integer variable (choice) , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value of this choice .

After that we have a switch to this choice ,

Case 1 about donor , we take an object (d) from donor , then we define a Boolean state and we have a loop as the state is equal true a list of donor details will appear to user and a message to ask user to entering his option and we define an integer variable (option1) , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value of this option .

After that we have a switch to this option ,

```
case 1 -> d.adddonor(s);

case 2 -> d.deletedonor(s);

case 3 -> d.editdonor(s);

case 4 -> d.searchdonor(s);

case 5 -> stat = false;
```

then the loop will stop if the option that user entered is (5) .

```
39         switch (choice) {
40             case 1 -> {
41                 Donor d = new Donor();
42                 boolean stat = true;
43                 while (stat) {
44                     System.out.println("-----* Donor Details *-----");
45                     System.out.println("1- add_donor");
46                     System.out.println("2- delete_donor");
47                     System.out.println("3- edit donor");
48                     System.out.println("4- search_donor");
49                     System.out.println("5- Exit");
50                     System.out.println(" Enter your option :");
51                     // Start Handel Exception
52                     int option1;
53                     choiceString = s.next();
54                     try {
55                         option1 = Integer.parseInt(s.choiceString);
56                     } catch (Exception e) {
57                         System.out.println("plz enter the integer value");
58                         option1 = s.nextInt();
59                     }
60                     // End Handel Exception
61                     switch (option1) {
62                         case 1 -> d.adddonor(s);
63                         case 2 -> d.deletedonor(s);
64                         case 3 -> d.editdonor(s);
65                         case 4 -> d.searchdonor(s);
66                         case 5 -> stat = false;
67                     }
68                 }
69             }
```

Case 2 about admin , we take an object (a) from admin , then we define a Boolean (sta) and we have a loop as the state is equal true a list of admin details will appear to user and a message to ask user to entering his option and we define an integer variable (option2) , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value of this option .

After that we have a switch to this option ,

```
case 1 -> a.addadmin(s);  
case 2 -> a.deleteadmin(s);  
case 3 -> a.editadmin(s);  
case 4 -> a.searchadmin(s);  
case 5 -> sta = false;
```

then the loop will stop if the option that user entered is (5) .

```
70 | case 2 -> {  
71 |     Admin a = new Admin();  
72 |     boolean sta = true;  
73 |     while (sta) {  
74 |         System.out.println("-----* Admin Details *-----");  
75 |         System.out.println("1- add_admin");  
76 |         System.out.println("2- delete_admin");  
77 |         System.out.println("3- edit_admin");  
78 |         System.out.println("4- search_admin");  
79 |         System.out.println("5- Exit");  
80 |         System.out.println(" Enter your option :");  
81 |         // Start Handel Exception  
82 |         String option2String;  
83 |         int option2;  
84 |         option2String = s.next();  
85 |         try {  
86 |             option2 = Integer.parseInt(s.option2String);  
87 |         } catch (Exception e) {  
88 |             System.out.println("plz enter the integer value");  
89 |             option2 = s.nextInt();  
90 |         }  
91 |         // End Handel Exception  
92 |         switch (option2) {  
93 |             case 1 -> a.addadmin(s);  
94 |             case 2 -> a.deleteadmin(s);  
95 |             case 3 -> a.editadmin(s);  
96 |             case 4 -> a.searchadmin(s);  
97 |             case 5 -> sta = false;  
98 |         }  
99 |     }  
100| }
```

Case 3 about blood , we take an object (b) from blood class , then we define a Boolean (st) and we have a loop as the state is equal true a list of blood details will appear to user and a message to ask user to entering his option and we define an integer variable (option2) , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value of this option .

After that we have a switch to this option ,

```
case 1 -> b.deleteblood(s);  
case 2 -> b.editblood(s);  
case 3 -> b.searchblood(s);
```

case 4 -> to print the available blood packets by invoking the method that we have been built in blood class (disAVblood ).

case 5 -> to print the available Quantity blood packets by invoking the built in method (.size()) , Blood.blooddb.size()

```
case 6 -> st = false;
```

then the loop will stop if the option that user entered is (6)

```
101 | case 3 -> {  
102 |     Blood b = new Blood();  
103 |     boolean st = true;  
104 |     while (st) {  
105 |         System.out.println("-----* Blood Details *-----");  
106 |         System.out.println("1- delete blood_packet");  
107 |         System.out.println("2- edit blood_packet");  
108 |         System.out.println("3- search blood_packet");  
109 |         System.out.println("4- view Available 'blood packets' TYPE ");  
110 |         System.out.println("5- view Available 'blood packets' Quantity ");  
111 |         System.out.println("6- Exit");  
112 |         System.out.print(" Enter your option :");  
113 |         // Start Handel Exception  
114 |         String option2String;  
115 |         int option2;  
116 |         option2String = s.nextLine();  
117 |         try {  
118 |             option2 = Integer.parseInt(option2String);  
119 |         } catch (Exception e) {  
120 |             System.out.println("plz enter the integer value");  
121 |             option2 = s.nextInt();  
122 |         }  
123 |         // End Handel Exception  
124 |         switch (option2) {  
125 |             case 1 -> b.deleteblood(s);  
126 |             case 2 -> b.editblood(s);  
127 |             case 3 -> b.searchblood(s);  
128 |             case 4 -> {  
129 |                 System.out.println("Available 'blood packet' Type : ");  
130 |                 b.disAVblood();  
131 |             }  
132 |             case 5 ->  
133 |                 System.out.println("Available 'blood packet' quantity: " + Blood.blooddb.size());  
134 |             case 6 -> st = false;  
135 |         }  
136 |     }  
137 | }
```

Case 4 about sick , we take an object (s1) from sick class , then we define a Boolean (p) and we have a loop as the (p) is equal true a list of sick details will appear to user and a message to ask user to entering his option and we define an integer variable (option2) , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value of this option .

After that we have a switch to this option ,

```
case 1 -> s1.addsick(s);  
case 2 -> s1.deleteSick(s);  
case 3 -> s1.editSick(s);  
case 4 -> s1.searchSick(s);  
case 5 -> p = false;
```

then the loop will stop if the option that user entered is (5)

```
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
  
    case 4 -> {  
        sick s1 = new sick();  
        boolean p = true;  
        while (p) {  
            System.out.println("-----* Sick Details *-----");  
            System.out.println("1- add_sick");  
            System.out.println("2- delete_sick");  
            System.out.println("3- edit_sick");  
            System.out.println("4- search_sick");  
            System.out.println("5- Exit");  
            System.out.println("Enter your option :");  
            // Start Handel Exception  
            String option1String;  
            int option1;  
            option1String = s.nextLine();  
            try {  
                option1 = Integer.parseInt(option1String);  
            } catch (Exception e) {  
                System.out.println("plz enter the integer value");  
                option1 = s.nextInt();  
            }  
            // End Handel Exception  
            switch (option1) {  
                case 1 -> s1.addsick(s);  
                case 2 -> s1.deleteSick(s);  
                case 3 -> s1.editSick(s);  
                case 4 -> s1.searchSick(s);  
                case 5 -> p = false;  
            }  
        }  
    }
```

Case 5 about blood request , we take an object (r) from blood request class , then we define a Boolean (u) and we have a loop as the (p) is equal true a list of blood request details will appear to user and a message to ask user to entering his option and we define an integer variable (option2) , in the event that user entered any value that is not an integer The Exception Handling will start to solve this error and warned user to enter an integer value of this option .

After that we have a switch to this option ,

case 1 -> r.addrequest(s); case 2 -> r.deleterequest(s); case 3 -> r.editrequest(s);  
case 4 -> r.searchrequest(s); case 5 -> u = false

then the loop will stop if the option that user entered is (5)

```
170
171
172
173
174     case 5 -> {
175         BloodRequest r = new BloodRequest();
176         boolean u = true;
177         while (u) {
178             System.out.println("-----* Request Details *-----");
179             System.out.println("1- add_request");
180             System.out.println("2- delete_request");
181             System.out.println("3- edit_request");
182             System.out.println("4- search_request");
183             System.out.println("5- Exit");
184             System.out.print(" Enter your option :");
185             // Start Handel Exception
186             String option1String;
187             int option1;
188             option1String = s.nextLine();
189             try {
190                 option1 = Integer.parseInt(option1String);
191             } catch (Exception e) {
192                 System.out.println("plz enter the integer value");
193                 option1 = s.nextInt();
194             }
195             // End Handel Exception
196             switch (option1) {
197                 case 1 -> r.addrequest(s);
198                 case 2 -> r.deleterequest(s);
199                 case 3 -> r.editrequest(s);
200                 case 4 -> r.searchrequest(s);
201                 case 5 -> u = false;
202             }
203         }
204     }
205 }
```

Case 6 -> about Blood bank class , we take an object from this class ,then w invoke that use to display information

Case 7-> when the state that we define above is equal false the loop will stop and logout from the switch.

```
201         case 6 -> {
202             BloodBank bb = new BloodBank();
203             bb.displayBBInfo();
204         }
205         case 7 -> {
206             state = false;
207             System.out.println(" Thanks,we wish a speedy recovery");
208         }
209     }
210 }
211 } else {
212     System.out.println("Invalid username or password.");
213 }
214 }catch (Exception ex){
215     System.out.print("Error!");
216 }
217 }
218 }
219 }
```

All this cases will done if the log in to the system is successful , if the login is incorrect the system will stop and warn user that is Error .