GROUP 5

Question:2

Ticket web4: For few Users address is not appearing correctly.

Incorrect output:

```
User{fName='Nikita', age=35, lName='John', address=Address{streetName='Saint Mark',
```

Error in code:

```
User user2=new User();
user2.setfName("Nikita");
user2.setlName("John");
user2.setAge(35);
user2.setSalary(2500.00);
Address address2=new Address();
address2.setCity("Ottawa");
address2.setCountry("CA");
address2.setProvince("Ontario");
address2.setStreetName("Saint Matheiu");
user2.setAddress(address);
```

Error was here in last line user2.setAddress(address) because address2 is object of user 2 and address was object of user1,so we have to use address2 instead of address.

Correction in code:

```
address2.setStreetName("Saint Matheiu");
user2.setAddress(address2);
```

Correct Output:

```
User{fName='Nikita', age=35, lName='John', address=Address{streetName='Saint Matheiu',
```

Variable name is salaryAccumulatedUser2 for salary of user 2 but in System.out.println the variable entered was salaryAccumulatedUser1 so that's why output was of user 1 not user 2.

```
Double salaryAccumulatedUser2= UserDetails.SalaryAccumulated(user2);
System.out.println(salaryAccumulatedUser1);
```

Correction in output:

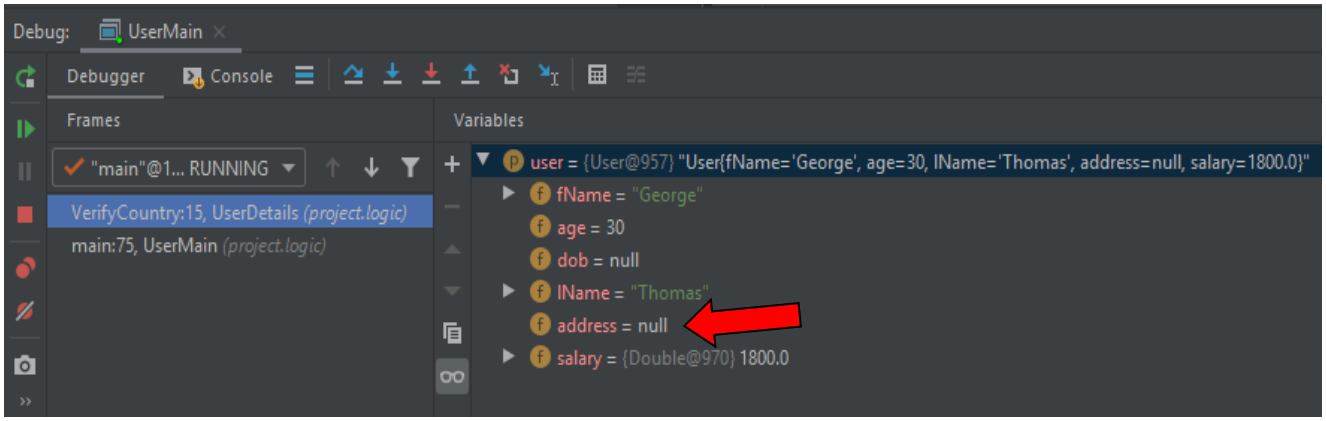
```
Double salaryAccumulatedUser2= UserDetails.SalaryAccumulated(user2);
System.out.println(salaryAccumulatedUser2);
```

Ticket Web 5: Not able to check if user is Canadian'

Error: Error was here because the code for user3 was incomplete because it was trying to verify user3 that user3 is Canadian or not but there were no setters of address fields for user3 in the code.

```
Exception in thread "main" java.lang.NullPointerException
    at project.logic.UserDetails.VerifyCountry(UserDetails.java:15)
    at project.logic.UserMain.main(UserMain.java:55)
```

Error Log: After putting debugging point at Line 75 in UserMain.java,then in the error log we come to know that in VerifyCountry() function address is null means it not getting address of user3.



Code Incomplete:

```
User user3=new User();
user3.setfName("George");
user3.setlName("Thomas");
user3.setAge(30);
user3.setSalary(1800.00);
```

Correction in error:

```
User user3=new User();
user3.setfName("George");
user3.setlName("Thomas");
user3.setAge(30);
user3.setSalary(1800.00);
Address address3=new Address();
address3.setCity("Vancouver");
address3.setCountry("CA");
address3.setProvince("BC");
address3.setStreetName("Van Street");
user3.setAddress(address3);
```

```
//TODO : WEB-5
Boolean isCanadian= UserDetails.VerifyCountry(user3);
System.out.println(UserDetails.FullName(user3)+" is "+isCanadian);
```

Correct Output:

```
58807.63029186222
George Thomas is true
```

New Requirement by Business :

RQ1: User will no longer need to manually enter the age , instead user will enter the date of birth.

```
// TODO : RQ - 1
//private int age;
private Dob dob;
```

```
public Dob getDob() {
    return dob;
}
public void setDob(Dob dob) {
    this.dob = dob;
}
```

```java
public class Dob {
    private int date;
    private int month;
    private int year;

    public int getDate() {
        return date;
    }

    public void setDate(int date) {
        this.date = date;
    }

    public int getMonth() {
        return month;
    }

    public void setMonth(int month) {
        this.month = month;
    }
```

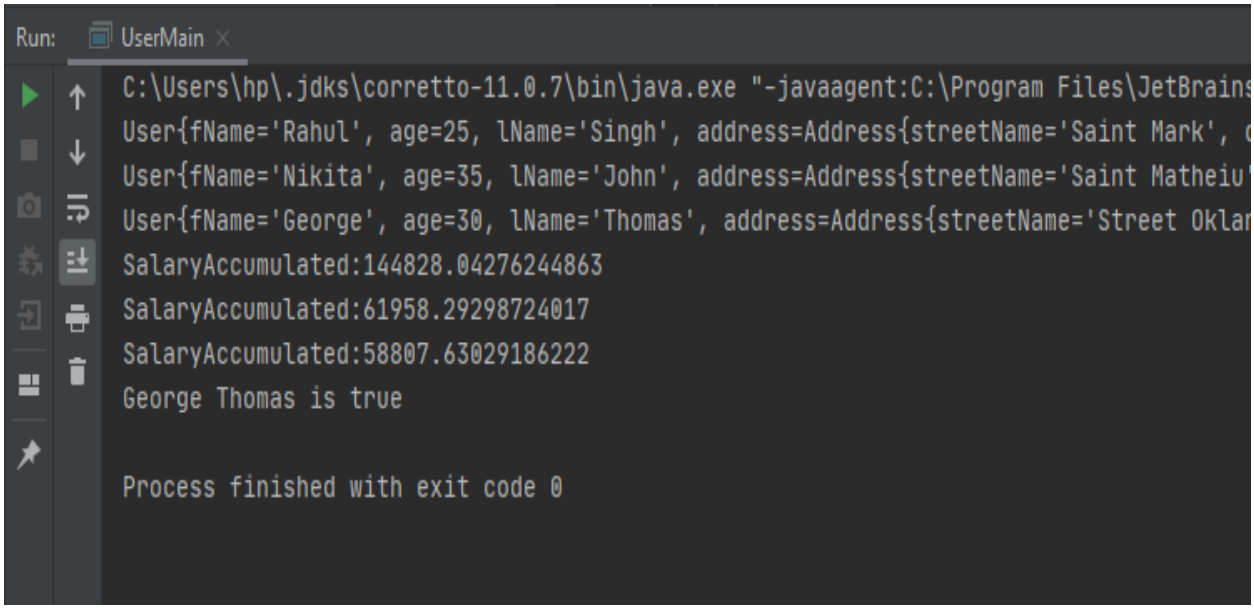
Dob.java file for Date of Birth.We have created dob.java file for date,month and year of user.Setters and getters are set in order to set and get date,month and year

In UserDetails.java file SalaryAccumulated() function now instead of getAge(),we will use getYear() from date of birth then storing the year in Birthyear variable, then we have used the logic originalAge=2020-Birthyear to calculate the age of user.

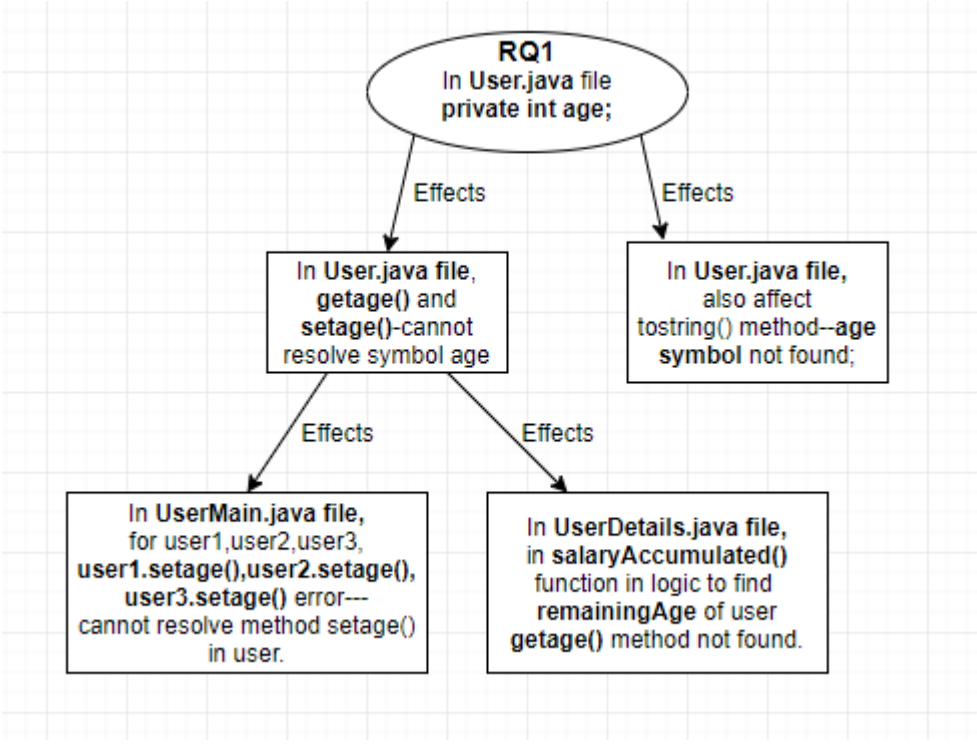
```java
public static Double SalaryAccumulated(User user){
    int BirthYear=user.getDob().getYear();
    int originalAge=0,remainingAge=0;
    double salaryHike=0,totalSalary=0;
    originalAge=2020-BirthYear;
    remainingAge=55-originalAge;
```

Now the logic of remaining age is also changed,instead of getage(),original age variable is used to get the remaining age of user.

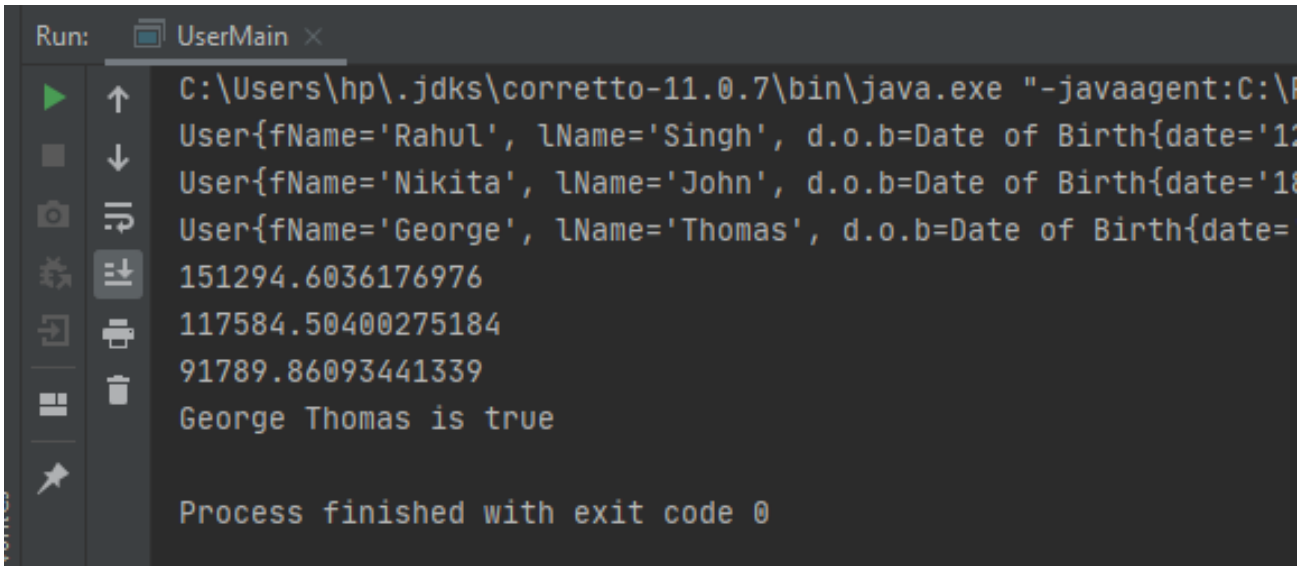
Output Before Doing RQ1:



Impact Analysis of RQ1:



New Output After Doing RQ1:



RQ2: Update the logic of salaryAccumulated based on province. Details : The age of retirement of QA - 55 and ON-58.

We have applied if statements in order to get the salaryAccumulated calculated based on province.

```
// TODO : RQ - 2
public static Double SalaryAccumulated(User user){
    String province=user.getAddress().getProvince();
    int BirthYear=user.getDob().getYear();
    int originalAge=0,remainingAge=0;
    double salaryHike=0,totalSalary=0;
    originalAge=2020-BirthYear;
    salaryHike=user.getSalary();
    if(province=="QC"){
        remainingAge=55-originalAge;
    }
    else if(province=="ON"){
        remainingAge=58-originalAge;
    }
    else{
        remainingAge=61-originalAge;
    }
```

Impact Analysis of RQ2:

When Updating the logic of salaryAccumulated based on province,there will no as such major impact on the output or code,we just have to change code a little bit,just have to apply if & else if statements and the salaryAccumulated will be different for users based on their province because now the retirement age is not same for every user,earlier before RQ2 the retirement age was 55 for all users regardless of their province but now After RQ2 for user who belong to QC-55 Retirement Age,for ON users Retirement Age will be 58 so due to this remaining Age will be changed so according to that salaryAccumulated will be different.

New Output After Doing RQ2: