Name:-Ninad Nitin Shukla
Banner Id:B00863694
Email Id:- nn320259@dal.ca

Assumptions

1. The database name is given in the form of a full link
   Eg:-jdbc:mysql://db.cs.dal.ca:3306/nnshukla
2. User and password are alphanumeric
3. Configfile is a string that gives the address of the file
4. Database used is already created
5. Duration is a positive integer
6. People meeting on the same date and for the same duration will be stored only once
7. However if the duration is different even if the date is same it will be stored again
8. Connection with the database is good.
9. While Synchronizing data internet connection is good.
10. All the tables in the database are already created before running the code.
11. All the errors are displayed on the console.

Explanation

Government Class Constructor:-

This will be the first object that will be instantiated as even the constructor of MobileDevice needs it. Constructor of the Government class. It takes input as the address of configFile which will have the user, password and Database (jdbc:mysql://db.cs.dal.ca:3306/nnshukla). These credentials can be used to access the database of the Government. If any of them are null it will be displayed on the console that the user knows.

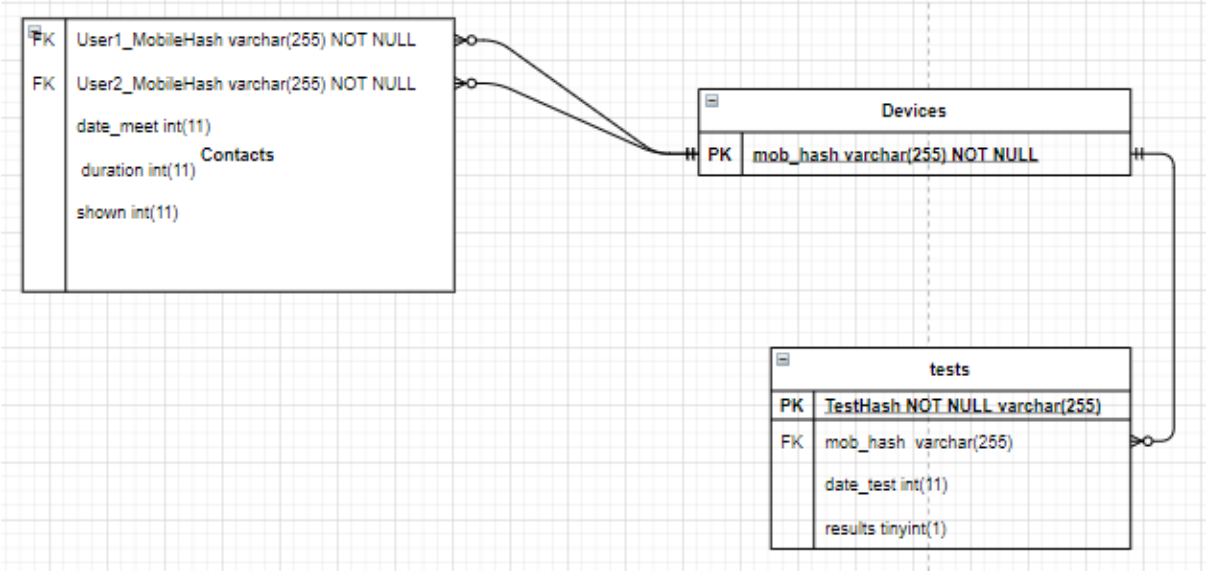
MobileDevice Class Constructor:-

Constructor off MobileDevice. It takes two inputs first is the path of the configFile second is the object of Government class. Through this object methods of Government class can be invoked in class MobileDevice. However, if any of the user, deviceName is absent or could not be retrieved for some reason then no other method will work too. This is done to avoid sending incomplete data to the government database. And the user will be notified that his data could not be taken due to some error.

Schema of the Government Database:-

It contains three tables Contacts, Devices and tests.The table Contacts consists of the entry of meeting between two users at a date and for a duration and contains the field shown which will indicate if user1 was shown positive by user 2 or not. The Primary Key of Contacts is the Composite key(User1_MobileHash,User2_MobileHash,date_meet,duration). So even if two users A and B meet at two points in a day it will be stored as long as they are meeting

for different duration. This was done for find gathering. As it might be possible that two people met at two different times and at one time they met a common third person and at other time they did not this means if A and B met C at 9 am for 10 minutes and A met B for 15 minutes at the same time in the gathering however A met B for 20 minutes at 1 pm too but this time they did not meet C. The table tests stores TestHashes for particular mob_hashes and the test taken at a date and the result of that test. The mob_hash field is allowed to be null in case the testhash reaches the government before it is sent to the user through positiveTest.Testhashes are primary key hers. The table Devices only have unique MobileHashes which are the primary key for this table. This table is made just to ensure that the other two tables are connected through foreign key constraints.



recordContact in mobileDevice()

This method takes in the hash of the individual, date and the duration of the contact. and stores it in the mobile Device.This is done in the form of an ArrayList(contact_List) of a class called Contacts which will store all this information.The class Contact uses setters and getters to obtain the private information of that contact . Please note that it does not store it in the government database but in the local memory of the mobileDevice. The anticipated error that can occur in this method may be due to wrong or out of bound input parameters so they have been handled. And in case of error the data will not be stored.

positiveTest(String testHash) in MobileDevice

In this method an alphanumeric testHash will be sent by the testing Agency. And it will be stored in the mobileDevice initially for that person till his mobile Synchronizes.

boolean synchronizeData()

This method is responsible for returning to a user if he has come in contact with someone who tested positive in 14 days of their contact. This method is responsible for sending class by calling the MobilDevice all the data to the Government through mobileContact method.The initiator is sent in the form of a Hash which is produced by SHA-256 and contactInfo string which sends data in the form of an XML string. The Xml is in the form:-

```
<contactInfo>
        <testHash>
            <test>cov123</test>
            <test>cov234</test>
        <testHash>
        <contact_List>
            <contact>
                <individual>rj</individual>
                <Date>134</Date>
                <Duration>45</Duration>
            </contact>
        </contact_List>
</contactInfo>
```

mobileContact

This method is called by SynchronizeData It has two input initiators and contactInfo .And it returns whether the initiator came in contact with someone positive in the last 14 days or not.ContactInfo also contains testHashes. It is first checked that if the government has already got these in the form of recordTestResult.If they have got then the mobile hash of the initiator is put in the row of corresponding testhash.However, if the testhash is not already available with the government then it is added to the hashmap testing. It also adds the data abou who came in contact with whom in the Contacts table.

recordTestResult

This method will be directly called by the object of Government class date is the date on which the testing is done and result is the result of the test. There can be two cases here either the MobileDevice has already sent this testHash to the government class through mobileContact. In that case we can make the entry in tests table of the testHash and the the mobileHash of that and the true result(False result will not be stored)and it is removed from the testing hashMap. The other case is that the mobile device user is yet to send the testHash. In that case we store the testHash and the information regarding that in the hashMap testing which has a key of testHash and the value is the object of class testing. Here we will store the mobile Hash as null as we are yet to know who this testHash belongs to. And then we make an entry in the tests table where the mob_Hash field is kept as null until the mob_Hash whose test it is sends it.

findGathering

This method is responsible for finding the gathering on a particular date for at least some minimum duration.It returns the number of Large gatherings only. So in this method method I have taken each pair like a and b and then the intersection of all the members that came in contact with a and b is taken even a and b is included to this group after that it is checked that how many members in this group came in contact with each other.Only groups larger than minSize are considered. The total number of contacts is c and m is the maximum number of contact that can occur in all the members so m=n*(n-1)/2 where n is the size of

the group. If c/m>density then only it is reported back as a large gathering Please note that sub-gatherings will not be recorded here. This means that {a,b,c,d} and {a,b,c} won't be considered different.However {a,b,c,d} and {a,b,e,f} will be considered different this is because consider a situation in which a,b meet with c,d and then on the same day they met with e,f or any other bigger gathering so it should be considered.

Testing Plan

1. Input Validation:-

- If in MobileDevice class either the file is not found or the user or Device name is not given. User will be notified on the console. If the user still tries to run other MobileDevice methods he will not be able to do so.Example method positiveHash will report “TestHash cannot be recorded”. Similarly Syncdata will report False. And record contact will also not take place.
- In Government class if file is not found / username,database name or password is not present or are given wrong Error will be printed on the console
- Record Contacts:-In case of negative duration entry will not be made. In case of negative date it will be considered as Days before january 1 2021
- If testHash is null then it should be handled too
- For findGathering if density<=0,minTime<=0 or minSize<0 return 0.

2. Data Flow:-

- If recordTestResult is called before positiveTest for a person B person and data synchronized and he comes in contact with A within 14 days, it should be recorded in the database. And A will get output as True
- If positiveTest is called before recordTestResult for a person B person and data synchronized and he comes in contact with A within 14 days, it should be recorded in the database. And A will get output as True
- If person B is called positiveHash, recordTestResult is called for person B. However, no synchronizeData is called .Then he meets person A(recordContact happens) thenA.synchroniz data is called it will return False. This is because without B.synchronizeData B will not be considered positive. If after B.synchronizedata(). A.synchronizeData is called it will return true.
- Ensuring that only new data can be reported so if A gets true as an output of syncData because of B then it should not get true again because of B it can however get it positive because of other new contact.

- Findgathering after recordContacts.,Before RecordContacts and for two different dates after recordContacts.

3. Boundary cases:-

- For findGathering if density=0,minTime=0 or minSize=0.