Name: Deep Patel
Dal ID: B00865413

Final Project Documentation

Overview

The program is used to implement functionality to alert user device in the circumstances when they got in contact with any positive tested individual. TheThe program mainly provides few functionalities like:

- record contacts
- record device's test hash
- synchronize all that data with government
- test agencies can set result of test hash into database
- program provides alert if device got in contact with any positive tested person
- find number of gatherings on particular date.

Files and External Data

The program contains some java files as mentioned below;

- Government.java – Class that manages all database functionalities like:
    - Storing contacts into Government database.
    - Agencies can store test results into database
    - Government can also fetch gathering information from stored data
- MobileDevice.java – Class manages all the contacts until all data got synchronized.
    - Store contacts information locally till data got synchronized
    - Stores device's test Hash values
    - Synchronize all data with government database.
- DeviceContacts.java – This is just class to represent contacts details like contacted device's testHash, contact date, contact duration whenever contacts happens.
    - Constructor to create object and store data
    - Getter and setters to get and set data.

The program also requires some external files to access its functionalities. External required files must simple text file. Following files are required;

- configurationFile.txt(For MobileDevice) – Files will contain device information like network address and device name in a specific format as given below;
    - address=<Address>
    - deviceName=<Name>
- configurationFile.txt(For Government) – This file will contain government database URL, username and password to access government database. All credentials are required and must be in given formate;
    - database=<URL>
    - user=<username>
    - password=<password>

The program uses Government Database as a storage for that one sql file is also required.

- GovernmentDB.sql – SQL file containing database creation and all table creation sql statements.

Data Structures:

There are many possible data structures that can be used for this project, here I chose some data structures as below;

- Map<String, Map<Integer,DeviceContacts>> – To store individuals contacts information until data is synchronized using method call to synchronizeData().
    - In map the key could be individual's hash code: unique identity string by which different individuals can be differentiated.
    - The Values associated with specific key could be the Map<Integers,DeviceContacts> where key is date and value is Class used for representing contact information of device.
        - The date in number of days from 1st January 2020 to identify on which date individual get in contact with user.
        - If data is of the same day twice then duration will be added instead of storing same value again.

- ArrayList<Integer> – List is used to store the different test results of the user using the unique testHash provided by the test Lab to identify that test result belongs to some particular person where integer value will be alpha numeric value

- In findGathering() method many data structures were used like:
    - Map<Integer, List<Integer>> - where key is hashCode of one device and List<Integer> is number of contacted device's hash code.
    - List<Integer>- List of integers to store number of set values, number of connections, number of max possible connection.
    - List<Float> - to get the density of each paired connection.

Assumption

You may assume that,

- Configurationfile in mobileDevice class is a correct path of file containing correct device information in a proper format. Format will be in the form as below;
    - address=123hbsdf
    - deviceName=abc's Phone
- In record contact individual comes in as a parameter is a correct hashcode of contacted device.
- Configuration file in government class is the correct path of the database credential file in the correct format as given below;
    - database=<URL>
    - user=<username>
    - password=<password>
- There could be a case that a person get in contact with device user even before 14 days but person's result get in the next few days which is positive, but as its possible that device user could be asymptotic and for that purpose the period to be considered will be up to 20 days just to be on the safe side.

Key Algorithms and design element

The program uses some algorithms to insert data into the data structure, remove data from data structure and find data from the used data structures. Besides that, there are some algorithms used to store data into some particular object.

Government class also uses some algorithm of graph to implement adjacency list using Maps and lists as Map<Integer, List<Integer>> to create a structure like Graph to implement functionality of finding gatherings.

Implementation of the program is as given below using following design:

- MobileDevice.java class contains following methods:
    - MobileDevice(configurationFile,governmentTracer)
        - Constructor of the MobileDevice to get device information from configuration file.
        - Constructor will call openConfigurationFile() method and store device information
        - governmentTracer will keep track of the government object with government database credentials.
    - openConfigurationFile(configurationFile)
        - Open the configuration file which will be in a specific format as provided in assumption.
        - Reads device's network address and device's name from configuration file and stores it locally.
    - recordContact(individual,date,duration)—
        - Store the individual's data(hashCode as a string) into the map's key.
        - Value for that key will also be a Map<Integer,DeviceContacts> where key will be a date and value is another class.
        - In the case of contacts on the same day duration will be added into the previous duration.
    - positiveTest(testHash)—
        - Method will get device's test hash and store that test hash into an arrayList to keep track of all test results.
    - synchronizeData()—
        - Method will generate hash code from the combination of device's network address and device's name and convert it into a string for passing it to the government as an initiator.
        - Method will also call getXmlFormattedString()- to get formatted string of contacts which will work as xml.
        - The method will then pass the initiator and this xml stored in formatted string will be passed to the Government class's mobileContact() method.

- getXmlFormattedString() –
  - This method will generate a string in a way its formatted as xml file and return that formatted string after careful construction of xml as a string.
  - The xml generated by program will be in a format of one main root node(government_data) and two Node Lists for device_test and contact_info .
  - Generated Xml will be in the given format(Sample xml file):

```
<government_data>
        <device_tests>
                <test>asd123dfg</test>
        </device_tests>
        <contact_info>
                <contact>
                        <contact_hash>3118</contact_hash>
                        <date>344</date>
                        <duration>10</duration>
                </contact>
                <contact>
                        <contact_hash>3211</contact_hash>
                        <date>344</date>
                        <duration>10</duration>
                </contact>
                <contact>
                        <contact_hash>3056</contact_hash>
                        <date>344</date>
                        <duration>10</duration>
                </contact>
        </contact_info>
</government_data>
```

- Government.java class contains following methods:
  - Government(configurationFile)—
    - Constructor will call the openConfigurationFile() method and store database credentials to access the government database to get data and add data into database.
  - openConfigurationFile(configurationFile)—
    - Supportive private method to read credential data from configurationFile in a specific format as shown in assumption.
  - mobileContact(initiator, contactInfo)—
    - This method will first get initiator as a deviceHashCode to store in into government database.
    - contactInfo is xml as a formatted string. To read xml I use xml parsers like DocumentBuilderFactory, DocumentBuilder and Document to get xml Data from a formatted string into the form of Document.

- This method will establish the connection using private method startConnection().
- Then that document object, connection object and deviceHashCode will be passed to private method syncToDatabase() to store all contacts and test hash data into government database.
- After storing contacted data method can get the list of recent contacts by calculating difference between contact dates and current date if the difference is less than 20 days(as a precautionary period) and stored them as a recently contacted individuals.
- After getting those contacts in given time period we can get those devices testHash which are stored into the database recently in the period of 14 days and results are positive(true).
- Using those results method will call checkForAlert() method- which will check if alert is necessary or not and return true or false accordingly.

o checkForAlert(conn, contactedPositiveIndividual, positiveTestHash)-
- Method will check into the database for testHash stores in the list of positiveTestHash and any contact record related to that testHash.
- If exist then also checks weather that record already stored inside alertinfo table.
- If there is no record of that contact in alertinfo table then this method will add that record into that table and return true for alerting the mobile device.

o getCurrentDate()-
- Method will return the current date in number as a count of days from the beginning of the year 2020(1st January 2020).

o syncWithDatabase(conn, document, deviceHashCode) –
- This is a helper method to mobileContact() method which fetches xml formatted data and store that data into government database using conn as connection to government database.
- First it will get element from the document by which method can access actual data stored in xml.
- I will check if same record already stored into mobiledevices table by using method checkTableForDataBySQL().
- Using same method for storing test Hash it will check with testinfo table if any testHash like that exist or not and only let add testHash if any match found in testinfo table.
- After adding data for device hash and test hash using prepared statements method will also store all contacts into the database as well into contacts table.

o checkTableForDataBySQL(conn,bCheck,sqlQuary)-
- Method will simply run quary using given connection and return true if any data found using that quary, and returns false if any data found using that quary.

- recordTestResults(testHash, date, result)-
  - First this method will check into testinfor table for any already stored testHash using checkTableForDataBySQL() method.
  - If no data stored with given testHash new data will be provided to database using prepared statements.

- startConnection()-
  - This private method simply establishes connection and returns same connection as needed through out the program.

- findGathering(date, minSize, minTime, density)-
  - After checking that date is greater than zero, minSize is greater than 1, minTime is greater than zero and density also between 0 and 1 method will get all the contacts on the given date from the database.
  - From that fetched contacts method will form two lists containing information as a pair. Initiator devices will be on the device list and contacted devices will be on the contactlist (Only unique pairs will be stored).
  - Using those unique pars from both lists method will call createAdjecencyList() to create adjList as a Map<Integer,List<Integer>> where key is device's HashCode and value is of each device's hash code that got in contact.
  - Using adjacency list method will calculate list of left node and right nod which works as a pair combinedly. Using list of map keys method will iterate through each element and get count of sets, connections and max connections. Using all that into calculation method will also calculate density of the connection by unique pairs from two different tables.
  - After finding all that if set size is greater then minSize of gathering and density is also greater then min density provided method will count all exceeded gatherings and return as integer.

Database Design

The Database of government is consist of four table. Four tables are given below;

- Database: GovernmentDatabase
- Tables:
    - mobiledevices – Table contains mobile device's hash code with unique testHash value (deviceHash can have multiple different testHashes)
    - The table consist following fields like:
        - id – auto incremented int and primary key of table
        - deviceHash – Varchar(255),
          hashCode of each unique mobile device
        - testHash – Varchar(255)
          unique test Code which is unique per user

    - contacts – Table contains contact information on which device contacted with which device with contact date and contact duration.
    - The table consist following fields like:
        - id – auto incremented int and primary key
        - deviceHash – Varchar(255),
          hashCode of mobile device
        - contactHash - Varchar(255),
          hashCode of contacted mobile device
        - contactDate – int,
          Date of contact in number from 1st Jan 2020
        - contactDuration – int,
          Duration of contact in minutes

    - testinfo – Table to store each unique testHash with the information of test date and test result. This table is updated by test agencies.
    - The table consist following fields like:
        - id – auto incremented int and primary key
        - testHash – Varchar(255),
          unique test Code which is unique per user
        - testDate – int,
          Date of test in number from 1st Jan 2020
        - testresult- Varchar(50),
          result associated with uniques testHash
        -
    - alertinfo – Table contains all alerted contact details.
    - The table consist following fields like:
        - id – auto incremented int and primary key
        - deviceHash – Varchar(255),
          hashCode of mobile device
        - contactHash - Varchar(255),
          hashCode of contacted mobile device
        - contactDate – int,
          Date of contact in number from 1st Jan 2020

Limitations

- All the recorded and alerted contacts are store forever and are not getting removed from database which could lead to memory issues.
- It is not possible to store multiple test results of the same mobile device on a single day.
- Using government's gathering method we can't say that it provides totally accurate result due to several affecting factors.