Assignment No 2

Important Note:

- For all the questions, you have to use Visual Studio and upload all of the files of the solution.
- Create separate solutions files for each question
- Every Question should be named as "k<Year><StudentID>_<QuestionNumber> for instance: k060195_Q1
- Your Assignment should have proper Objected Oriented Programming, Data Structures and no hardcoding
- Last Date for Submission: **April 25, 2020 (11:55 PM)**

Question 1: Write a small Windows Form application to maintain heart rate (bpm) records. Your application can perform the following activities:

a) User can add Patient's personal information (Name, Date of Birth, Gender, Email)
b) User can add Patient's heart rate (bpm) for a specific time (should be accurate till seconds)
c) User can enter multiple heart rate records for a patient without having to close and restart the application.
d) The output will be saved in an XML file (PatientDetails_yyyy_mm_dd.xml). Only one file will be created per day.

Sample output file:

```
<Patients>
  <Patient name = "" DOB = "" gender = "" email = "">
    <bpm></bpm>
    <time></time>
    <confidence></confidence>
  </Patient>
<Patients>
```

Note: Confidence level is a hardcoded value which will always be 0 when being inserted from windows application whereas it may be different if it is generated from another source.

Question 2: Write a windows service which will read the xml files generated from the question 1 from a configurable path after every 5 minutes and perform the following activities:

a) Create folder with the name of Patient (Considering Name to be unique identifier without any special characters). (Example: C:\Murtaza)
b) Create another folder within the Name folder as "user-profile". (Example: C:\Murtaza\user-profile\)
c) Generate a File to store patient's personal information in JSON format as "User-Profile.json". (Example: C:\Murtaza\user-profile\user-profile.json)
d) Create another folder within the Name folder as "user-detail". (Example: C:\Murtaza\user-detail\)

e) Within this folder, you will create multiple files (1 file per day) and store User's heart rate records in json format. Each file would have the naming convention heart_rate-<YYYY-MM-DD>.json. For instance, a user has heart rates of two days then he would have following files:
C:\Murtaza\user-detail\heart_rate-2020-03-14.json
C:\Murtaza\user-detail\heart_rate-2020-03-15.json

Following is the sample format for User-Profile.json:
{"Name":"<Full Name>","gender":"<gender>","age":<age>,"email": "<email>"}

Following is the sample format for heart_rate json file:

```
[{ "dateTime" : "12/26/16 20:02:42",
  "value" : {
    "bpm" : 70,
    "confidence" : 0 }
}
,{
  "dateTime" : "12/26/16 20:02:45",
  "value" : {
    "bpm" : 70,
    "confidence" : 0 }
}]
```

Question 3: In this question, you have to create another Windows Service which will read multiple heart rate json files every 10 minute and consolidate into one json file per user. Every 10 minute, it will check for any new file, in case there are any new heart rate files, it will merge into the existing one.

Question 4: Create two windows service:

a) First windows service will take the input from question 3 every 15 minutes and generate two XML files containing the following details:
   - File 1: UserChart.xml
     - This file will have each user's following detail:
       - Name
       - Email
       - Highest heart rate (till date)
       - Average heart rate (till date)
       - Lowest heart rate (till date)
       - Target Heart Rate Range
     - Sample output:
       - <Users>
         <User name = "" email = "">
         <High></High>
         <Average></Average>
         <Low></Low>
         <Range></Range>
         </User>
         </Users>
   - File 2: ConsolidatedChart.xml

- This file will have consolidated information:
  - Highest heart rate per age group
  - Average heart rate per age group
  - Lowest heart rate per age group
- Sample output:
  - *<Patients>*
    <AgeGroup value= "1">
    <High></High>
    <Average></Average>
    <Low></Low>
    </AgeGroup>
    <AgeGroup value= "2">
    <High></High>
    <Average></Average>
    <Low></Low>
    </AgeGroup>
    </Patients>

b) Second windows service will send email to the user every 15 minutes if there are any modifications in the results of UserChart.xml

Age Groups:

- Group 1: 0 – 10
- Group 2: 11 – 20
- Group 3: 21 – 30
- Group 4: 31 – 40
- Group 5: 41 – 50
- Group 6: 51 – 60
- Group 7: 61 – 70
- Group 8: 71 – 80

Heart Rate Range Calculation:
Maximum heart rate is calculated by subtracting age from 220. For a 30-year-old person, for example: 220 – 30 = 190.

The target heart rate range would be between 50 and 85 percent of the maximum heart rate:

- 50 percent: 190 x 0.50 = 95 bpm
- 85 percent: 190 x 0.85 = 162 bpm

Rules for Marking:

It should be clear that your assignment will not get any credit if:

- The Assignment is submitted after due date.
- The Submitted Assignment does not open or file is corrupted
- No assignment will be accepted through email unless slate is not working at the submission deadline.
- Copied amongst students or from another source