Rapport de stage :

Projet **B**ig **D**ata **S**anté et **E**nvironnement dans la ville de Nice en partenariat avec l'IMREDD

MDBS France

Juillet 2017 – Septembre 2017

Tuteur MBDS : Gabriel MOPOLO-MOKE

Responsable scientifique : Serge MIRANDA

Etudiant :	Membres du jury :
Mlle Irina GRIGOREVA	M. Gabriel MOPOLO
	M. Serge MIRANDA

Résumé

Ce document est le rapport du stage sur la suite du projet Big Bridge – SE : Big Data Santé et Environnement, réalisé au sein du MBDS en partenariat avec l'IMREDD.

L'objectif de ce projet est de développer un outil logiciel utilisant les approches Big Data permettant de trouver des corrélations entre des données environnementales et les données sur la santé, en utilisant des outils d'analyse de données issue de plateformes Open Source.

Le but de ce stage est d'enrichir le «Data Lake» du système avec les données personnelles de smartwatch (le profil de l'utilisateur et l'information de l'activité cardiaque) pour l'utiliser pour la recherche individuelle de l'effet de la pollution de l'air sur la santé humaine.

Les données utilisées pour la recherche ont été collectées par le service AirPaca et Fitbit API entre juillet 2017 et septembre 2017.

La réalisation de ce projet a été faite avec les outils suivants : Java / JEE, Android SDK, R, Oracle SQL et NoSQL (Hadoop HDFS, Oracle NoSQL).

Mots clés : Big Data, base de données, Oracle, Apache, Hadoop, Hive, HDFS, NoSQL, DWH, SQL, Web, Java, Android, JSF, JEE, EJB, R, IMREDD, Fitbit.

Abstract

This document is the report of the internship on the follow-up of the Big Bridge project - SE: Big Data Health and Environment, realized within the MBDS in partnership with the IMREDD.

The objective of this project is to develop a software tool using Big Data approaches to find correlations between environmental data and health data, using analysis tools derived from Open Source platform.

The aim of the internship is to enrich the “Data Lake” of the system with the personal data from smartwatch (user's profile and the information of cardiac activity) to use it for individual research of air pollution's effect on human's health.

The data used for the research were collected by the AirPaca service and the Fitbit API between July 2017 and September 2017.

The realization of this tool has been made using Java/JEE, Android SDK, R language, Oracle SQL and NoSQL databases (Hadoop HDFS, Oracle NoSQL).

Key words: Big Data, database, Oracle, Apache, Hadoop, HDFS, Hive, NoSQL, DWH, SQL, Web, Java, Android, JSF, JEE, EJB, R, IMREDD, Fitbit.

List of figures and tables

List of Acronyms and Abbreviations

MBDS – Mobiquité, Bases de Donées et intégration de Systèmes

IMREDD - Institut Méditerranéen du Risque, de l'Environnement et du Développement Durable

API - Application Programming Interface

HDFS - Hadoop Distributed File System

DWH - Data Warehouse

SQL - Structured Query Language

JSF - JavaServer Faces

JEE - Java Platform, Enterprise Edition

EJB - Enterprise Java Beans

SDK - Software Development Kit

REST - Representational State Transfer

BMI – Body Mass Index

Table of Contents

1 General Introduction

The most important mechanism for management decisions aimed at improving air quality and reducing the negative impact of environmental factors on the human body is to carry out a health risk assessment of the population. Methodology for assessing the health risk is an element of mathematical modeling of the causal relationships between environmental factors and health, under their influence in the specific conditions of time and area.

At the present time, there is a large number of studies on the impact of air pollution on human health. Studies in various geographical areas have shown associations of respiratory symptoms and conditions with long-term exposure to total suspended particulates (TSP) and SO2 (1-7), to particulate matter (8-10), to black smoke (11), and to NO2 (7). Furthermore, studies of hospital admissions and mortality studies point to an association of short- and long-term exposure to air pollution with symptoms that are related both to pulmonary and to cardiac diseases (12-19).

Thus, it is important to investigate a correlation between the level of air pollution and human health in the city of Nice.

The mission of current internship is to develop a software product using the Big Data approach to enrich the “Data Lake” of the Big Bridge SE project with the personal data from smartwatch (user's profile and the information of cardiac activity) to use it for individual research of air pollution's effect on human's health, as well as the formation of personal recommendations.

Plan of report:

Chapter 1: General Introduction — the reasons of project implementation, a short summary of the sphere, the internship’s mission

Chapter 2: Project Presentation — Big Bridge project for master MBDS, project partners, project’s objective and goals

Chapter 3: State of Art — current state of the projects, studying the correlation of environment and health

Chapter 4: Envisaged solution — deliverables of the internship’s project

Chapter 5: Project Organization — information about the planning of work and risk plan

Chapter 6: Personal and real-time environmental data descriptions — description of the content and volume of the source data

Chapter 7: Project Architecture and Data Management — detailed description of the architecture of the system

Chapter 8: The Algorithm of Generating the Data — description of the Java application for generating the data

Chapter 9: The Web Service — description of the Java Spring application for manipulating the data using REST API

Chapter 10: Big Bridge - SE Android Application — description of the Android Application

Chapter 11: Data Analysis with the R Language — description of the used data analysis methods with R language and the results

Chapter 12: General Conclusion — current results of the internship, plan of the continuation of research

2 Project Presentation

2.1 Presentation of the Master MBDS and the Big bridge project

Since more than 26 years Master MBDS form project managers to the information services of the future in symbiosis with the industrial world. Strong industrial links signed in 1990 with all major global technology information (Oracle, Sun, IBM/Informix, Computer Associates, Microsoft, Sybase, CNAF, Amadeus, Unilog, etc.) and then from 1999 telecommunications (Cegetel, Siemens, Lucent, Intel, 3Com, Nokia, etc.) and large users (Amadeus, SFM, Crédit Agricole, GMF, etc.) to prototype future wireless online services. Training of students by practice in the future of information technologies. Construction of a single strategic vision of the information technology market in Europe and around the world with the MBDS inscription in a clear vision of development in future.

The Nice Sophia Antipolis University is a university located in Nice, France and neighboring areas. It was founded in 1965 and is organized in eight faculties, two autonomous institutes and an engineering school.

The project Big Bridge is a generic Big Data project in the MBDS, which will enable students involved experience of practical approaches Big Data market and including bridges between the management of structured and unstructured data as well as data analysis tools using the Open Source platforms. The main databases publishers (IBM, Microsoft, Oracle, etc.), major IT companies (ATOS, CAPGEMINI, Sopra, etc.) and major key accounts (AIR France, Amadeus, HP, etc.) offer Big Data solutions for manage unstructured and structured production data by DBMS. The goal of the MBDS Big Bridge project is propose Big Data projects to demonstrators with publishers, IT services companies and / or large accounts that wish. These projects are spread out on concrete issues such as healthcare, government data, consumer, security, insurance, environment, sports, etc. The different solutions are generally structured around open source ecosystem Hadoop, NoSQL databases (such as MongoDB), SQL databases and various strategies of data analysis (data analytics / data science with the use of the language open source R).

In this document, there is presented a continuation of the project, called Big Bridge SE, around the Big Data of Oracle solution and the Open Source Hadoop/Map Reduce platforms and R language.

2.2 Presentation of the project partner IMREDD

IMREDD is the Mediterranean Institute for Risk, Environment and Sustainable Development. The IMREDD is a new form of cooperation between research, business and the territory in the areas of green technology and intelligent city (Smart City).

IMREDD's mission is to stimulate research, create initial and continuing training courses on environment and sustainable development, and promote expertise and innovation in these fields. It pursues a triple mission:

- conduct and promote the scientific, technological, economic, social and human scientific and technological research and training activities of sustainable development;
- impulse the logics of platforms necessary for mutualized research and development activities;
- promote the valorization of previous activities by helping to identify and support innovative new entrepreneurs and by facilitating collaborative networks between existing actors, such as laboratories, companies, local authorities and associations.

The role of IMREDD in this project is to assist in preparing specifications and obtaining data from the Central Hospital of Nice and the pollution data.

2.3 Presentation of the subject and project goals

The objective of this project is to design and develop the software tool, using experience of Big Data market approaches including bridges between the management of data and data analysis tools using open source platforms, to search the correlations between environmental and health data, as well as using the personal data for this goal.

The aim of the internship is to use the individual data to make the personal recommendations based on real-time data of air pollution and to analyze the existence of air pollution's effect on human's health.

The data used for the research were collected by the AirPaca service and the Fitbit API between July 2017 and September 2017.

3 State of Art

3.1 Criteria for comparing

- Analysis of medical data
- Analysis of environment data
- Open-source solution
- Big Data
- Real time analysis

3.2 Existing projects

Today, there are not a lot of solutions in both spheres at the same time: medicine and pollution analysis. But there are projects, which can analyze one of this sphere.

- *Air Paca*

Air PACA is the Association Approved by the Ministry for the Environment for Monitoring Air Quality in the region Provence-Alpes-Côte d'Azur (AASQA).

Link: http://www.airpaca.org/

- *Open Air*

The Open Air project is a Natural Environment Research Council (NERC) knowledge exchange project that aims to provide a collection of open-source tools for the analysis of air pollution data. These pages provide some background information to the project. The project is also supported by Defra. The project is led by the Environmental Research Group at King's College London, supported by the University of Leeds.

Technologies: R language

Link: http://www.openair-project.org/

- *IBM Analytics (Watson)*

Watson Analytics offers you the benefits of advanced analytics without the complexity. A smart data discovery service available on the cloud, it guides data exploration, automates predictive analytics and enables effortless dashboard and infographic creation. You can get answers and new insights to make confident decisions in minutes—all on your own.

Technologies: Cloud based service

Link: https://www.ibm.com/analytics/watson-analytics/us-en/

- *OHDSI*

The Observational Health Data Sciences and Informatics (or OHDSI, pronounced "Odyssey") program is a multi-stakeholder, interdisciplinary collaborative to bring out the value of health data through large-scale analytics. All our solutions are open-source. OHDSI has established an international network of researchers and observational health databases with a central coordinating center housed at Columbia University.

Technologies:

✓ Atlas (a web-based integrated platform for database exploration, standardized vocabulary browing, cohort definition, and population-level analysis)

✓ Achilles (a standardized database profiling tool for database characterization and data quality assessment)

✓ Calypso (an analytical component for clinical study feasibility assessment)

✓ KnowledgeWebBase (an experimental user interface for exploration of data present in the LAERTES evidence base)

Link: http://www.ohdsi.org/

- *Easy Med Stat*

Medical Statistic Analysis

Technologies: PHP

Link: http://www.easymedstat.com/

- *FreeMED*

FreeMED is an open-source, old-as-dirt EMR. Founded in 1999, it's one of the longest-running open source EMRs out there. It boasts over 81,000 downloads and implementation in everything from small private practices to large government hospitals.

Technologies: Web service

Link: http://freemedsoftware.org/

- *REMITT*

REMITT is a revolutionary medical information translation and transmission system, which is primarily used for preparing and submitting medical billing data.

Technologies:

✓ Written using Java 1.6 / J2EE application standard.

✓ MySQL-database backed operation.

✓ Full REST/SOAP interface with WSDL.

✓ Supports processing X12 835 remittance information and pushing it back to an EMR/PM system via SOAP callbacks.

✓ Web interface to allow configuration per user, testing of individual plugins, etc.

✓ JUnit testing using JUnitEE with web interface for full regression and functionality testing.

✓ File scooper support for pulling remittance and other claim data from clearinghouses.

✓ Scriptable claim submission using Javascript scripting for clearinghouses.

✓ Fully database-backed filestore for claims, remittance and processing data with audit/processing trail.

Link: http://remitt.org/

Existing solutions do not make a complex analysis of medical and pollution data at the same time. They just analyze one sphere: or medicine, or pollution data. Also, some of existing solutions are commercial and not open source.

3.3 Comparing projects

Criteria Project	Analysis of medical data	Analysis of environment data	Open-source solution	Big Data	Real time analysis
Air Paca		✓	✓	✓	✓
Open Air		✓	✓	✓	
OHDSI	✓			✓	
Easy Med Stats	✓			✓	
FreeMED	✓		✓	✓	
REMITT	✓		✓	✓	
Big Bridge - SE	✓	✓	✓	✓	✓

Table 1 Comparing the solutions

4 Existing and envisaged solution

4.1 Existing solution

Big Bridge SE is the software tool, which use Big Data approach and Open-source solutions to make the effective management and analysis of health and environment data. At the moment, the project includes:

- static data on patients of the hospital, as well as static data on air pollution during the period from January 2014 to December 2016 – it is stored in Hadoop HDFS with access through external tables;
- R script to search for data correlation (linear regression, predictive methods, charts);
- the web application for visualization of research results.

4.2 Envisaged solution

The envisaged solution suggests enriching the existing solution with personal real-time data, as well as real-time data on air pollution, allowing for an individual analysis of data and making personal recommendations.

4.3 Deliverables

1. Databases: loading and managing data.

2. Analysis script on R language for prediction, classification and visualization.

3. Android application for using the FitBit API to get the profile data and making personal recommendations.

4. Java Application for generating the data.

5. The Java Web Service for manipulating the data.

5 Project Organization

5.1 Used Project method

For managing the project, the Agile method was used.

The meetings with the manager of the project were held every Friday for a duration of 30 minutes. There were identified the current tasks and difficulties.

5.2 Project team and member's role

The Project Manager: the mentor of the Master 2 MBDS Mr. Mopolo
The Developer: Irina Grigoreva
Medical Consultant (assistance in developing the concepts for the analysis): Sergey Gorianin

5.3 Used tools in the project

- *IDE:* IntelliJ IDEA 2017.2.3, Eclipse 4.6.0 (Neon), Android Studio 2.2
- *Web Application Server:* GlassFish 4.1.2
- *Modeling of schemes:* Draw.io
- *Emulator:* Genymotion
- *Smartwatch:* Fitbit Charge 2
- *Environment:* Oracle DWH 12g
- *Databases:* Oracle NOSQL, HBASE, Oracle SQL
- *Programming languages:* Java/JEE, R
- *Version Control System:* Git (Bitbucket)

5.4 Configuration management

Configuration management is done using Dropbox. Project folder directory, named "ProjetBigdataImredd2017" has been created containing the subdirectories for each type of documents. Each member deposits its files in its directory. Source code is available in the Dropbox. Folder description:

BigDataAndroidApp — the workspace with the source code of the Android application;

BigDataWebService — IntelliJ IDEA workspace with the source code of the web service;

BigDataGenAlg — Eclipse IDEA workspace with the source code of the script for generating data;

StatisticR2 — RStudio workspace with the source code of the program performs the calculating the classification and prediction methods;

Report — all the reports for project with annexes and the presentation.

The CamelCase notation was used to denote the variable names. The name of the variable is the purpose to which data they refer. Variable from the example refers to the type of style for a block with personalized recommendations.

5.5 Risk Plan

Risk	Solution
Current non-existence of enough personal data	Using the algorithm for generating the data and filling the database
No external connections on the MBDS cluster	Using the local machine for developing
The problem with obtaining specifications for current data	Choosing the other possible way to use the system
Loss of time waiting for data and specifications	To use this time for study Big Data approaches and similar projects
Server of MBDS cluster crash	Working on local virtual machines

Table 2 Risk plan

5.6 Project planning

Release "The enriching the Data Lake with the personal data and its analysis"

Sprint 1

Objective of the sprint: define the most effective architecture, configure the smartwatch, create all the necessary database tables, implement the Android application with the requests for manipulating data and the algorithm of making recommendations; writing an algorithm for generating personal data using Java.

Task	Description	Duration	Planned Start Date	Planned Finish Date	Actual Start Date	Actual Finish Date
1.1	Configure the smartwatch and FitBit developer's account, define the architecture of the system, creating the Android application	6 days	24.07.17	31.07.17	24.07.17	31.07.17
1.2	Create and include to the system the web service for manipulating the data from Android app and Oracle NoSQL database; add the methods for downloading the real-time air pollution data from AirPaca service.	3 days	07.08.17	09.08.17	07.08.17	11.08.17
1.3	Implement the requests for manipulating data and working with FitBit API on the side of Android app. Configure the database.	10 days	31.07.17	11.08.17	31.07.17	15.08.17
1.4	Write the algorithms for generating personal data	10 days	14.08.17	25.08.17	14.08.17	25.08.17

Table 3 List of tasks of Sprint 1

Sprint 2

Objective of the sprint: implementing the data analysis with R Language

Task	Description	Duration	Planned Start Date	Planned Finish Date	Actual Start Date	Actual Finish Date
2.1	Implementation of analysis with profile data	1 day	18.08.17	18.08.17	18.08.17	18.08.17
2.2	Implementation of displaying profiles (ordering data, using group operations)	2 days	21.08.17	22.08.17	18.08.17	18.08.17
2.3	Implementation of making prediction on group of risk using classification	5 days	22.08.17	28.08.17	21.08.17	28.08.17
2.4	Research on the relation between the quality of air and personal cardiac activity (using generated data)	2 days	24.08.17	25.08.17	28.08.17	28.08.17

Table 4 List of tasks of Sprint 2

The full project plan is presented in Annexes.

5.7 Project Budget

The project budget includes the purchase of the FitBit smartwatch - 112,99 €.

6 Personal and real-time environmental data descriptions

6.1 Personal Data Description

For the personal part of the research was used the individual profile data from the FitBit profile as well as the information about daily cardiac activity. This real-time data is available at the FitBit API and presented as a response for the GET-request in the JSON format, which is converted and saved to the database through the web service. The data structure is presented in Table 5 and Table 6.

Field	Description
User Id	*Unique sequence of symbols (generated by FitBit after the registration)*
Weight	*Weight*
Height	*Height*
Age	*Age*
Date of birth	*Date of birth*
Full name	*Full name, specified while the registration*
Gender	*Gender*
Indicator of smoking	*Boolean field, indicates the presence/absence of bad habits for person*
Indicator of drinking alcohol	*Boolean field, indicates the presence/absence of bad habits for person*

Table 5 The Structure of Personal Profile Data

Field	Description
User Id	*Foreign key to user's profile*
Date	*Date of measuring*
Calories (out of range zone)	*Calories that were spent during the heart rate from 30 to 94 beats per minute*
Minutes (out of range zone)	*Total time of the heart rate from 30 to 94 beats per minute*
Calories (fat burn zone)	*Calories that were spent during the heart rate from 94 to 132 beats per minute*
Minutes (fat burn zone)	*Total time of the heart rate from 94 to 132 beats per minute*
Calories (cardio zone)	*Calories that were spent during the heart rate from 132 to 160 beats per minute*
Minutes (cardio zone)	*Total time of the heart rate from 132 to 160 beats per minute*
Calories (peak zone)	*Calories that were spent during the heart rate from 160 to 220 beats per minute*
Minutes (peak zone)	*Total time of the heart rate from 160 to 220 beats per minute*

Table 6 The Structure of Personal Cardiac Data

6.2 Real-time environmental Data description

For the environmental part of this research was used the real-time data about the air pollution in Nice from the API of service AirPaca (20). This data is presented as a response for the GET-request in the JSON format, which is converted and saved to the database through the web service. The data structure is presented in Table 7.

Field	Description
Date	*Date of measuring*
City	*City of measuring*
Postal code	*The postal code of the city*
Value	*Level of pollution as a number*
Quality	*The pollution level, expressed in one of three grades (Bon/Médiocre/Moyen)*

Table 7 The Structure of Environmental Data

7 Big Bridge Project Architecture and Data Management

7.1 Objective

The objective of Big Bridge Project Architecture and Data Management is:

✓ understanding the Oracle Linux Server architecture;

✓ understanding the Oracle Big Data SQL solution;

✓ understanding different types of data storing;

✓ understanding the Big Bridge mechanism;

✓ acquisition of data (Oracle NoSQL, Apache Hive, HDFS) for our reference Big Bridge project;

✓ access to data from the database DWH through access drivers (Hadoop Big Data SQL access driver, Hive Big Data SQL access driver);

✓ analysis with tools beyond data analytics tools (such as open language source R).

7.2 System's architecture

The scheme below contains the architecture of data streams and data storing in the project:

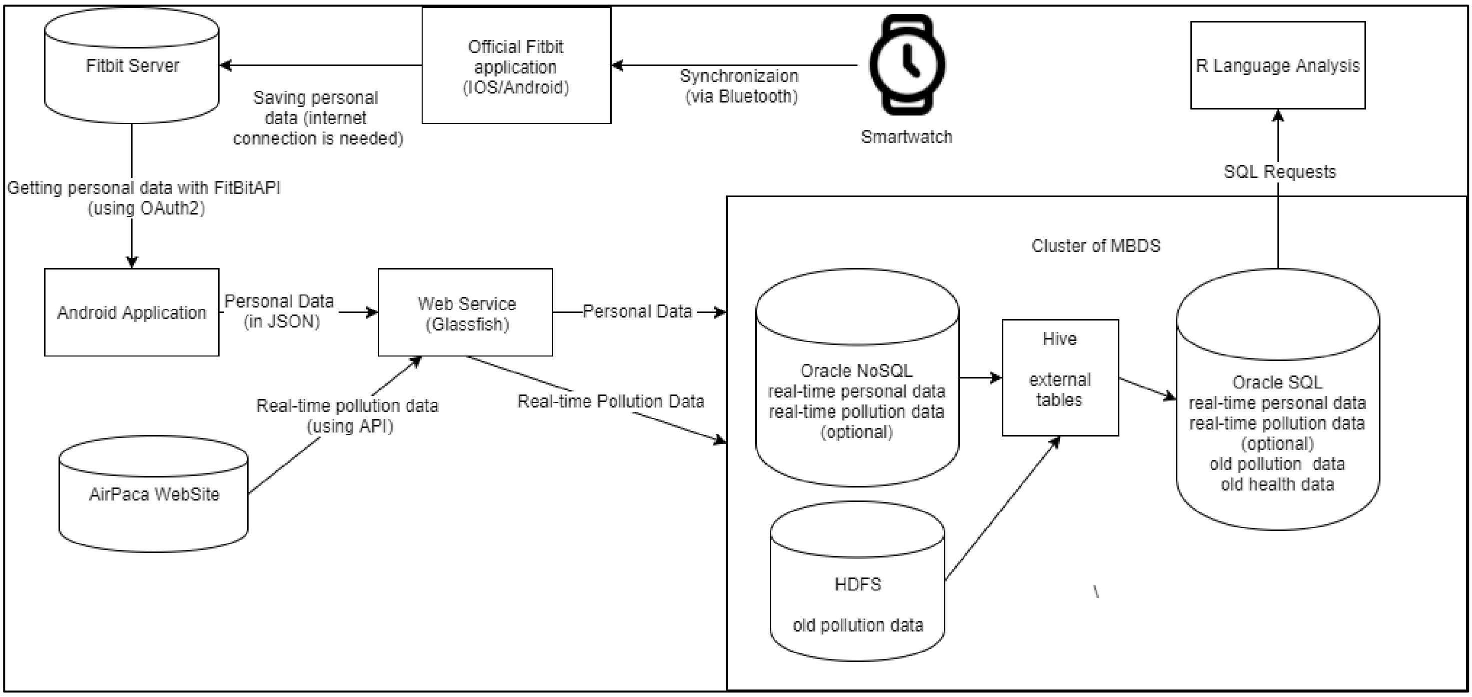

Fitbit Server
Official Fitbit application (IOS/Android)
Synchronizaion (via Bluetooth)
Smartwatch
R Language Analysis
Saving personal data (internet connection is needed)
Getting personal data with FitBitAPI (using OAuth2)
SQL Requests
Cluster of MBDS
Android Application
Personal Data (in JSON)
Web Service (Glassfish)
Personal Data
Oracle NoSQL real-time personal data real-time pollution data (optional)
Hive external tables
Oracle SQL real-time personal data real-time pollution data (optional) old pollution data old health data
Real-time pollution data (using API)
Real-time Pollution Data
AirPaca WebSite
HDFS old pollution data

Figure 1 The architecture of the system

As it can be seen, the continuation of the project is the adding new data sources as a smartwatch and the AirPaca API service. New data is real-time and was integrated to the system through the web service running on Glassfish server and stored in Oracle NoSQL Database. To get the heart-rate data from the smartwatch it's necessarily to synchronize the device with the official application, so after it will be possible to use it from external applications (in this case – from Android application). The Hadoop Hive is used in the system as a bridge between NoSQL Databases and SQL Database, so it's possible to access the data with R Language from external Oracle SQL tables.

The heart-rate information is obtained with the GET-requests from an Android Application, and then it's sent to the web service to be parsed and saved to the database. In this step, the following link is used:

```
https://api.fitbit.com/1/user/-/activities/heart/date/today/1d.json
```

Example of response:

```
{
  "activities-heart": [
    {
      "dateTime": "2015-08-04",
      "value": {
        "customHeartRateZones": [],
        "heartRateZones": [
          {
            "caloriesOut": 740.15264,
            "max": 94,
            "min": 30,
            "minutes": 593,
            "name": "Out of Range"
          },
          {
            "caloriesOut": 249.66204,
            "max": 132,
            "min": 94,
            "minutes": 46,
            "name": "Fat Burn"
          },
          {
            "caloriesOut": 0,
            "max": 160,
            "min": 132,
            "minutes": 0,
            "name": "Cardio"
          },
          {
            "caloriesOut": 0,
            "max": 220,
            "min": 160,
            "minutes": 0,
            "name": "Peak"
          }
        ],
        "restingHeartRate": 68
      }
    }
  ]
}
```

The real-time air pollution data is obtained with the GET-request directly from this service. In this step, the following link is used:

```
http://api.airpaca.org/communes/06088/indices?date=yesterday&api_token=token"
```

Example of response:

```
{
"commune": "NICE",
"code_insee": "06088",
"mentions_legales": "http://www.airpaca.org/mentions-legales",
"indices":
{
"date": "2017-09-06",
"valeur": 58,
"couleur_html": "#FFFF00",
"qualificatif": "Moyen"
}
}
```

The data, which was used in the research before, is still stored in HDFS (environmental data) and in Oracle SQL (health data) and available from external tables. The logical data model is presented on Figure 2.

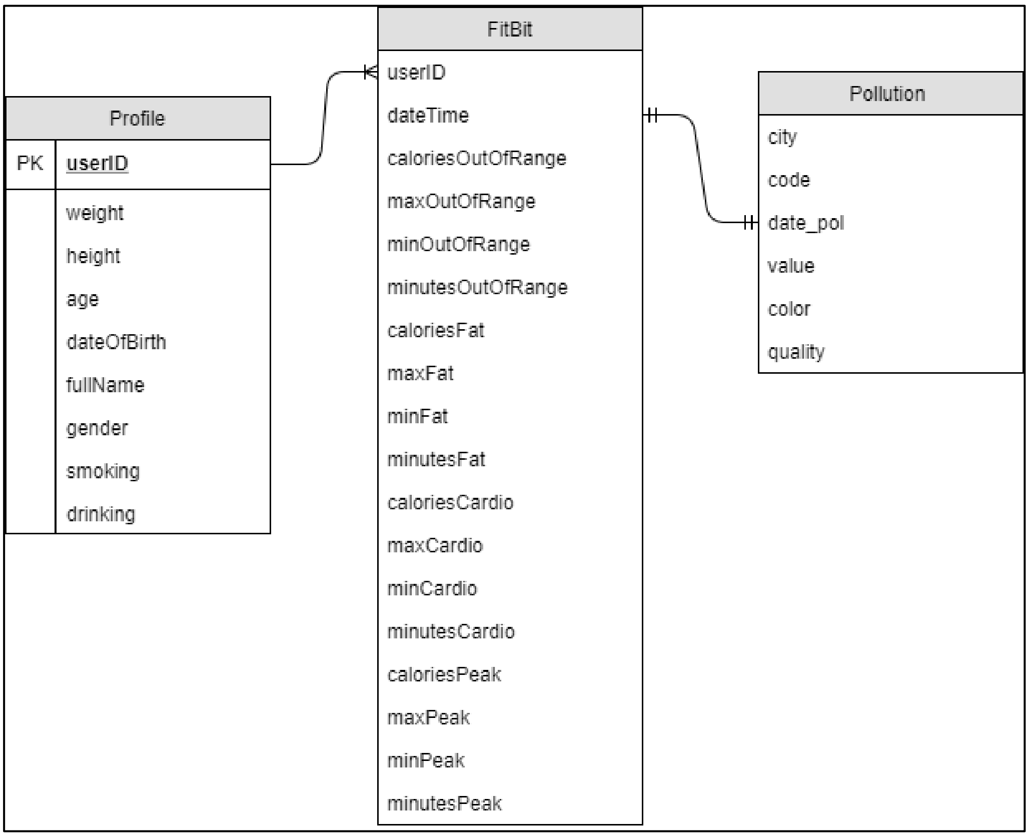


Figure 2 The logical data model

7.3 Managing personal data around Oracle NoSQL Database in the Oracle Big Data Environment

7.3.1 Creating NoSQL tables

To create the tables in Oracle NoSQL database the following commands were used:

- for table with the information about profiles:

```
    execute 'create table profile (userID string, weight string, height string, age string,
dateOfBirth string, fullName string, gender string, smoking string, drinking string, primary
key(userID))'
```

- for table with everyday heart-rate data:

```
    execute 'create table fitbit (userID string, dateTime string, caloriesOutOfRange string,
maxOutOfRange string, minOutOfRange string,  minutesOutOfRange string, caloriesFat string,
maxFat string, minFat string, minutesFat string, caloriesCardio string, maxCardio string,
minCardio string, minutesCardio string, caloriesPeak string, maxPeak string, minPeak string,
minutesPeak string, primary key(shard(userID), dateTime))'
```

How it can be seen from the commands, in the table for profiles the user id is used like a primary key, and it connects the users with their heart-rate information. For cardiac data table there are user id and date of measuring used as a primary key.

To import generated data (which is contained in JSON-files) to these tables is used the command “put”:

```
    put table" -name tableName -file file.json
```

7.3.2 Creating Oracle NoSQL external tables in Hive

After connecting to the Hadoop Hive, to create the external tables in the Hive environment the following commands were used:

- for table with the information about profiles:

```
    create external table profile_hive_ext (userID string, weight string, height string, age
string, dateOfBirth string, fullName string, gender string, smoking string, drinking string)
    stored by 'oracle.kv.hadoop.hive.table.TableStorageHandler'
    TBLPROPERTIES (
    "oracle.kv.kvstore" = "kvstore",
    "oracle.kv.hosts" = "bigdatalite.localdomain:5000",
    "oracle.kv.hadoop.hosts" = "bigdatalite.localdomain/127.0.01",
    "oracle.kv.tableName" = "profile"
```

```
        );
```

- for table with everyday heart-rate data:

```
        create external table fitbit_hive_ext (userID string, dateTime string, caloriesOutOfRange
string, maxOutOfRange string, minOutOfRange string,  minutesOutOfRange string, caloriesFat
string, maxFat string, minFat string, minutesFat string, caloriesCardio string, maxCardio string,
minCardio string, minutesCardio string, caloriesPeak string, maxPeak string, minPeak string,
minutesPeak string)
        stored by 'oracle.kv.hadoop.hive.table.TableStorageHandler'
        TBLPROPERTIES (
        "oracle.kv.kvstore" = "kvstore",
        "oracle.kv.hosts" = "bigdatalite.localdomain:5000",
        "oracle.kv.hadoop.hosts" = "bigdatalite.localdomain/127.0.01",
        "oracle.kv.tableName" = "fitbit"
        );
```

7.3.3 Creating Hive external tables in Oracle SQL Database

To have a connection to the tables, located in Oracle NoSQL databases and created on the Hive, from Oracle SQL it’s necessarily to execute next commands:

- for table with the information about profiles:

```
        create table profile_hive_ext (userID varchar2(100), weight varchar2(100), height
        varchar2(100), age varchar2(100),
        dateOfBirth varchar2(100), fullName varchar2(100), gender varchar2(100), smoking
        varchar2(100), drinking varchar2(100))
        organization external (type oracle_hive
        default directory oracle_bigdata_config
        access parameters (
        com.oracle.bigdata.tablename = bigdataprojectdb.profile_hive_ext)) reject limit
        unlimited;
```

- for table with everyday heart-rate data:

```
        create table fitbit_hive_ext (userID varchar2(100), dateTime varchar2(100),
        caloriesOutOfRange varchar2(100), maxOutOfRange varchar2(100),
        minOutOfRange varchar2(100),  minutesOutOfRange varchar2(100), caloriesFat
        varchar2(100), maxFat varchar2(100), minFat varchar2(100), minutesFat varchar2(100),
        caloriesCardio varchar2(100), maxCardio varchar2(100), minCardio varchar2(100),
        minutesCardio varchar2(100), caloriesPeak varchar2(100), maxPeak varchar2(100),
        minPeak varchar2(100), minutesPeak varchar2(100))
```

```
organization external (type oracle_hive
default directory oracle_bigdata_config
access parameters (
com.oracle.bigdata.tablename = bigdataprojectdb.fitbit_hive_ext)) reject limit unlimited;
```

7.4 Managing environmental data around Oracle NoSQL Database in the Oracle Big Data Environment

7.4.1 Creating NoSQL table

To create the table for storing environmental real-time data about air pollution in Oracle NoSQL database the following command was used:

```
execute 'create table pollution (city string, code string, date_pol string, value integer, color string, quality string, primary key(shard(date_pol), city, code, quality))'
```

How it can be seen from the command, in this table a shard primary key contains date of measuring, city, postal code and quality of air.

7.4.2 Creating Oracle NoSQL external table in Hive

After connecting to the Hadoop Hive, to create the external table in the Hive environment the following commands were used:

```
create external table pollution_hive_ext (city string, code string, date_pol string, value bigint, color string, quality string)
stored by 'oracle.kv.hadoop.hive.table.TableStorageHandler'
TBLPROPERTIES (
"oracle.kv.kvstore" = "kvstore",
"oracle.kv.hosts" = "bigdatalite.localdomain:5000",
"oracle.kv.hadoop.hosts" = "bigdatalite.localdomain/127.0.01",
"oracle.kv.tableName" = "pollution"
);
```

7.4.3 Creating Hive external table in Oracle SQL Database

To have a connection to the table, located in Oracle NoSQL database and created on the Hive, from Oracle SQL it’s necessarily to execute next command:

```
create table pollution_hive_ext (city varchar2(100), code varchar2(100), date_pol varchar2(100), value NUMBER(19),
color varchar2(100), quality varchar2(100))
```

```
organization external (type oracle_hive
default directory oracle_bigdata_config
access parameters (
com.oracle.bigdata.tablename = bigdataprojectdb.pollution_hive_ext)) reject limit unlimited;
```

8 The Algorithm of Generating the Data

In the current conditions of insufficient amount of personal data, it was decided to use an algorithm which generates user profiles and their daily cardiac activity according to certain rules. To implement this algorithm, the Java language was used.

8.1 Generation of profile data

Generation of the user profile data proceeds using class Random from the java.util library, as well as for generating the names and surnames (male and female) the appropriate files in CSV format are used. The user's gender and bad habits are generated in the logical variable (yes / no), and such indicators as height and weight are generated taking into account the person's gender and the formula for calculating the normal weight. The date of birth is generated using the class RandomDateOfBirth and following next command:

```
	long ms = -946771200000L + (Math.abs(rnd.nextLong()) % (70L * 365 * 24 * 60 * 60 * 1000));
```

To set a unique user Id it is used a random sequence of 8 characters which contains letters of the alphabet and numbers. This script generates a JSON file, which can then be loaded into the database after.

8.2 Generation of heart-rate data

To generate data on the cardiac activity day-by-day of users, a list of pre-generated profiles is needed. A list of all users is obtained from the database, and using the “for” cycle and random number generator - the indicators of cardiac activity are generated, according to the intervals that satisfy the medical patterns. The measurement date is set in the HearthActivity class. This script generates a JSON file, which can then be loaded into the database after.

9 The Web Service

In order to make it possible to transfer personal data between the Android application and the NoSQL database which is located on the server, it was decided to develop a REST service using the Java programming language and the Spring framework. To implement the service, were used such libraries as oracle.kv to access the KvStore and org.json for parsing JSON objects. For mapping web requests, the GetMapping annotation of Spring Framework is used. This service is running on the web application server Glassfish.

When accessing the address “/airpaca”, the service applies to the AirPaca API for getting the actual data on air pollution, parses the response in JSON format and then saves new row to the database, using next commands:

```
    KVStore store = null;

    if (store == null) {
        try {
            /* Connect to Oracle NoSQL DB and Store */
            store =
                    KVStoreFactory.getStore(new KVStoreConfig(KVSTORE_NAME,
KVSTORE_URL));

            /* -- Call the table API -- */
            TableAPI tbl = store.getTableAPI();

            /* -- Select table from Store -- */
            Table pollution_table = tbl.getTable("pollution");

            /* -- Create new row with values -- */
            Row pol_row = pollution_table.createRow();

            /* -- Set values for new row: "fieldName", value -- */
            pol_row.put("city", city);
            pol_row.put("code", code_insee);
            pol_row.put("date_pol", date);
            pol_row.put("value", value);
            pol_row.put("color", color);
            pol_row.put("quality", quality);

            /* -- Insert row to table -- */
            tbl.put(pol_row, null, null);
        } catch (Exception e) {
            System.err.println("ERROR: Please make sure Oracle NoSQL
Database is up and running at '" +
                KVSTORE_URL + "' with store name as: '" + KVSTORE_NAME +
                "'");
            e.printStackTrace();
        }
    }
```

In the case of personal data, the Android application, applying to the web service, transmits as parameters the information necessary for saving to the Profile table or to the Fitbit table (the current cardiac activity of the user). Addressing proceeds at the addresses “/profile” and “/fitbit”, respectively. Example of code for applying parameters from a request:

```
    @GetMapping("/profile")
public String Profile(Model m, @RequestParam("userID") String userID,
                      @RequestParam("weight") String weight,
                      @RequestParam("height") String height,
                      @RequestParam("age") String age,
                      @RequestParam("dateOfBirth") String dateOfBirth,
                      @RequestParam("fullName") String fullName,
                      @RequestParam("gender") String gender,
                      @RequestParam("smoking") String smoking,
                      @RequestParam("drinking") String drinking) {
```

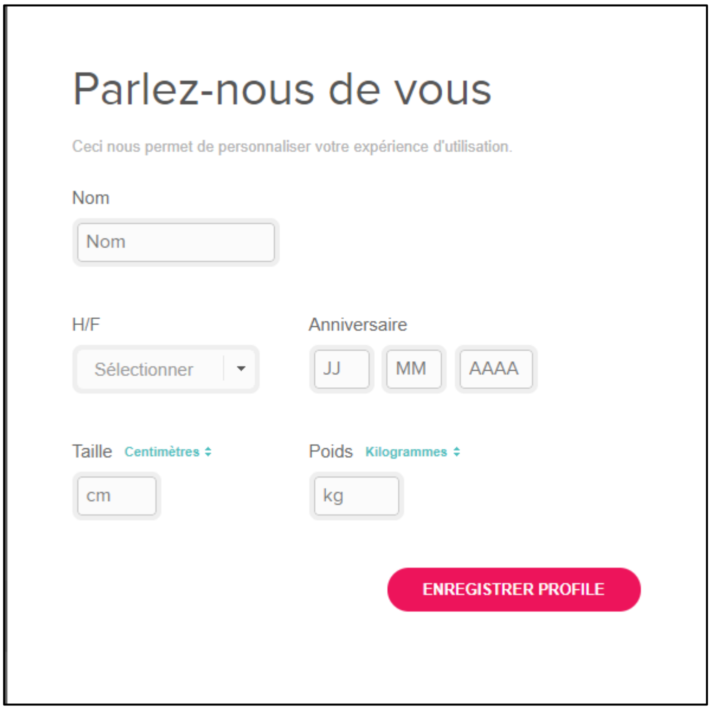
Thus, in the system, this web service performs the following functions:

- saving the real-time data on air pollution to the database, when applying the relevant address (using AirPaca API);
- saving the profile data of user to the database when receiving them from the Android application;
- saving the data of user’s real-time cardiac activity to the database when receiving them from the Android application.

10 Big Bridge - SE Android Application

10.1 Creating a new profile

To access the android application of the system and obtain data about the profile and cardiac activity, user must be authorized through the FitBit account. To create it, he can use the official FitBit application and the device itself (smartwatch). The interface for creating the profile is shown in Figure 3.



Parlez-nous de vous

Ceci nous permet de personnaliser votre expérience d'utilisation.

Figure 3 Creation a new FitBit profile

Thus, when creating an account (for authorization it is needed an email and password), the user specifies his gender, height, weight, full name, and date of birth.

10.2 Using the Android Application

The Android application of the Big Bridge - SE project is intended primarily for interaction with the user and forming up recommendations. Its functionality includes authorization via FitBit account using OAuth 2 standard, obtaining the personal data using FitBit API, obtaining real-time air pollution data using AirPaca API, as well as identifying a risk group of users and creating individual recommendations based on this. The application was tested using the Genymotion emulator. Next, consider the application classes and their functions.

The application includes three activities:

1. MainActivity.java is responsible for authorizing the user through the FitBit account. In accordance with the FitBit rules, following the link is made through the CustomTabsIntent class:

```
   String url = "https://www.fitbit.com/oauth2/authorize?" +
           "response_type=token" +
           "&client_id=228NLH" +
           "&expires_in=2592000" +
   "&scope=activity%20nutrition%20heartrate%20location%20nutrition%20profile
   %20settings%20sleep%20social%20weight" +
           "&redirect_uri=fitbittester://logincallback" +
           "&prompt=login";
customTabsIntent.launchUrl(this, Uri.parse(url));
```

This link contains the client id, which is preliminary obtained during the registration of the developer account, as well as all the necessary permissions that the user should allow when the application is first launched. An example of this activity is shown on Figure 4.

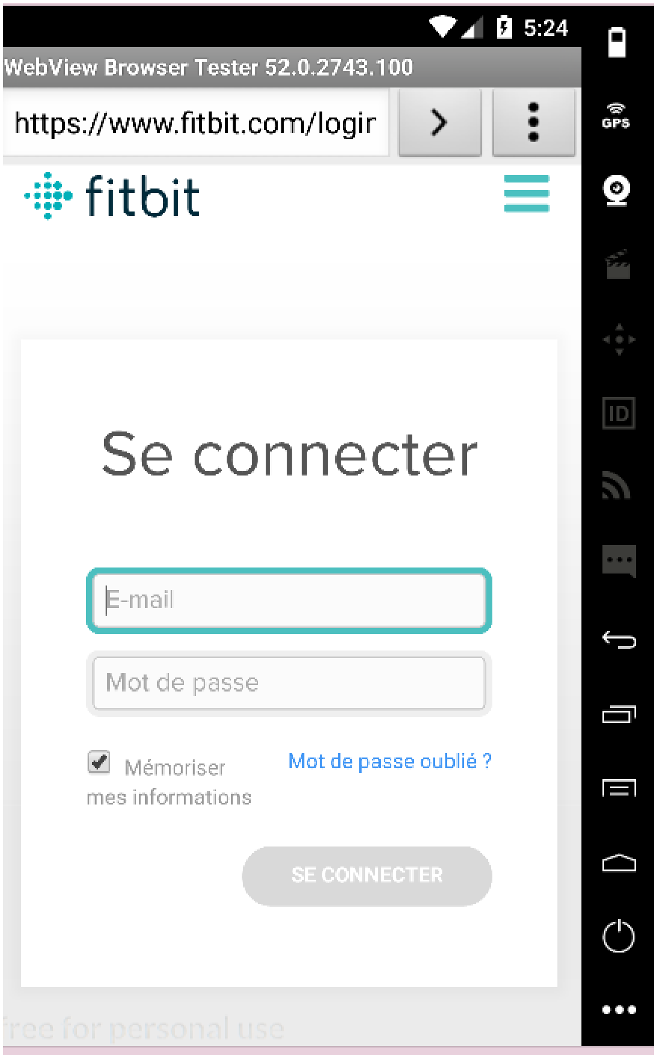
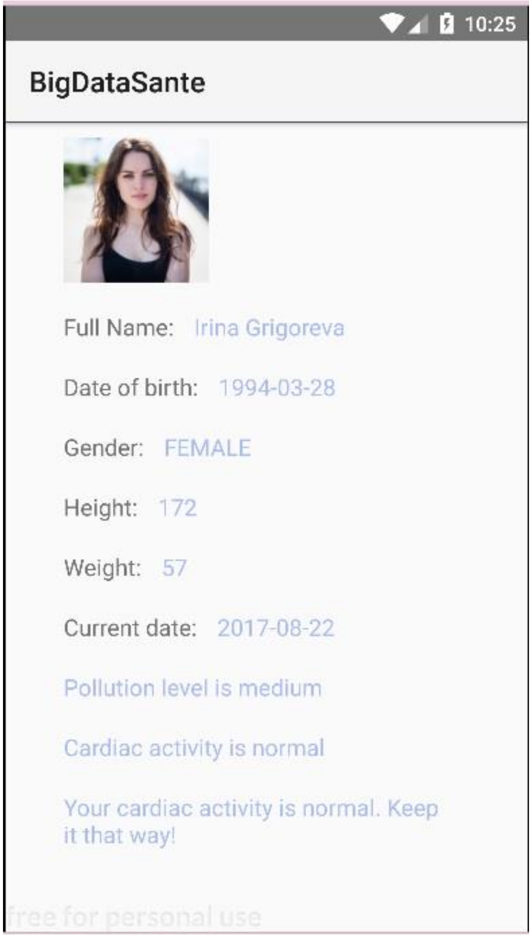


Figure 4 Autorization Activity

1. TestActivity.java is designed for checking the authorization, as well as saving all necessary data, such as access token and user id. This activity redirects the user to the user activity.
2. UserActivity.java is responsible for receiving data on the cardiac activity of the user, calls all the necessary methods of data transmission to the web service and generating the recommendations, and also serves as a form for displaying the personal data. An example of this activity is shown on Figure 5.



BigDataSante

Full Name: Irina Grigoreva

Date of birth: 1994-03-28

Gender: FEMALE

Height: 172

Weight: 57

Current date: 2017-08-22

Pollution level is medium

Cardiac activity is normal

Your cardiac activity is normal. Keep it that way!

Figure 5 The user activity of the Android application

The application also contains model classes for the user profile, information on cardiac activity and air pollution, to simplify the working with JSON objects. To load the user's avatar from the URL and to formulate recommendations, the classes-services ImageLoadTask and RecommendationsGenerator are used, respectively. The JSONMaker class is used to parse objects received in response to the GET requests, as well as to form a set of parameters for applying the web service. For launching the GET requests the library Volley is used.

The recommendations are formulated in accordance with the concept of the project prepared earlier and takes into account the user's risk group and the current level of air pollution. Fragment of the code for determining the risk group:

```
public int getARiskLevel(UserProfile userProfile) {
     int level = 0;
     double bmi = 0.0;
     bmi = (Integer.parseInt(userProfile.getWeight()) *10000)/
(Integer.parseInt(userProfile.getHeight()) * 172);
     if ((userProfile.getSmoking().equalsIgnoreCase("no")) ||
(userProfile.getDrinking().equalsIgnoreCase("no"))) {
         if ((bmi >= 18.5) && (bmi <= 24.5)) {
             if (Integer.parseInt(userProfile.getAge()) <= 45)
                 return 0;
```

```
            else if (Integer.parseInt(userProfile.getAge()) <= 60)
                return 1;
        } else if ((bmi <= 18.5) || ((bmi >= 25) && (bmi <= 30)))
            return 1;
        else return 2;
    } else {
        if (Integer.parseInt(userProfile.getAge()) <= 30)
            return 0;
        else if (Integer.parseInt(userProfile.getAge()) <= 50)
            return 1;
        else return 2;
    }

    return level;
}
```

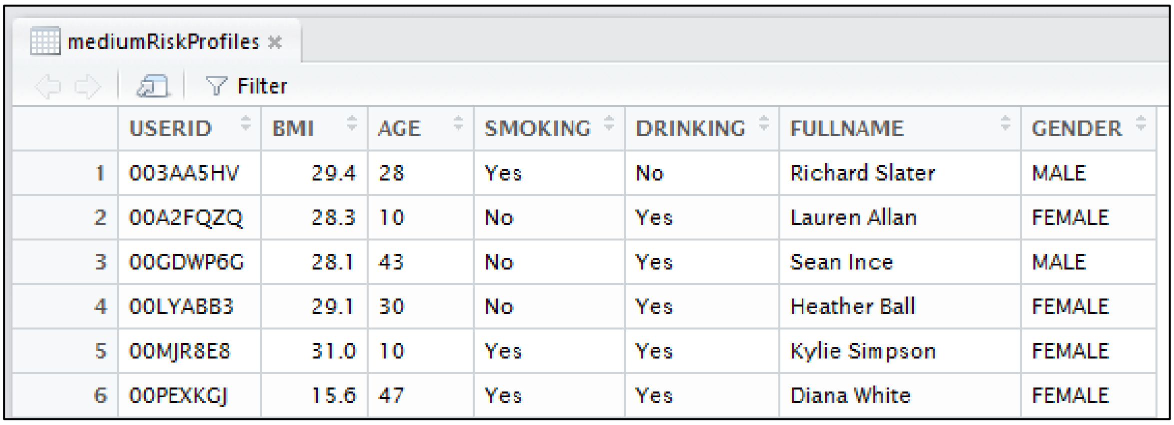
Thus, the sequence of actions when launching the android application is as follows:

- authorization of the user through the FitBit account, obtaining his basic personal information;
- request to the FitBit API for information about daily cardiac activity by user id;
- request to the AirPaca API for the information of today's level of air pollution;
- the definition of the risk group of the user in accordance with his basic information;
- the formation of recommendations for the user, taking into account his heart activity and the level of air pollution.

11 Data Analysis with the R Language

11.1 Preparing the data for analysis

To prepare the generated personal data for the analysis with R Language, there was formed the data frames, which contain the information about users with a high, medium and low risk of cardiovascular and pulmonary diseases (the rules for grouping are based on the conception and presented in Annexes). For this, the queries to the external tables of the Oracle SQL database were used. These queries are presented in Annexes. An example of one of these data frames is shown in Figure 6.



	USERID	BMI	AGE	SMOKING	DRINKING	FULLNAME	GENDER
1	003AA5HV	29.4	28	Yes	No	Richard Slater	MALE
2	00A2FQZQ	28.3	10	No	Yes	Lauren Allan	FEMALE
3	00GDWP6G	28.1	43	No	Yes	Sean Ince	MALE
4	00LYABB3	29.1	30	No	Yes	Heather Ball	FEMALE
5	00MJR8E8	31.0	10	Yes	Yes	Kylie Simpson	FEMALE
6	00PEXKGJ	15.6	47	Yes	Yes	Diana White	FEMALE

Figure 6 The data frame of profiles

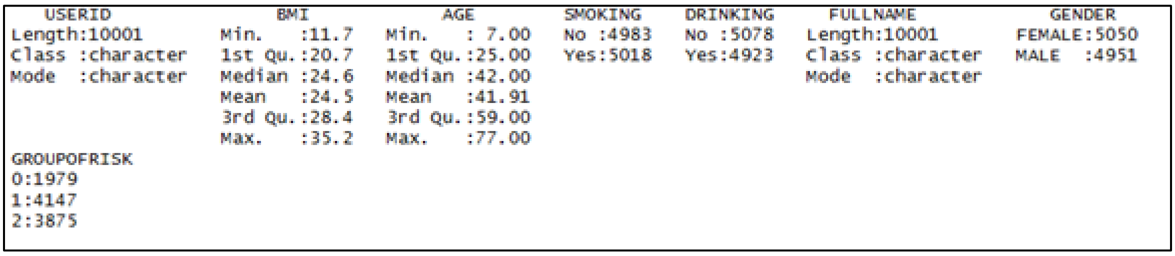
Thus, each of frames contains the user's id, his calculated BMI (Body Mass Index), age, indicator of smoking or drinking alcohol, full name and sex.

Next, the group of risk field was included and all data frames were merged into one – profileData:

```
> highRiskProfiles$GROUPOFRISK <- 2
> mediumRiskProfiles$GROUPOFRISK <- 1
> lowRiskProfiles$GROUPOFRISK <- 0
> profileData<-rbind(highRiskProfiles,mediumRiskProfiles,lowRiskProfiles)
```

11.2 The analysis of profile data

Figure 6 shows the analysis of the common data frame of profiles in accordance with the factors.



```
    USERID                 BMI              AGE          SMOKING     DRINKING       FULLNAME              GENDER
 Length:10001       Min.   :11.7   Min.   : 7.00   No :4983   No :5078   Length:10001       FEMALE:5050
 Class :character   1st Qu.:20.7   1st Qu.:25.00   Yes:5018   Yes:4923   Class :character   MALE  :4951
 Mode  :character   Median :24.6   Median :42.00                         Mode  :character
                    Mean   :24.5   Mean   :41.91
                    3rd Qu.:28.4   3rd Qu.:59.00
                    Max.   :35.2   Max.   :77.00
 GROUPOFRISK
 0:1979
 1:4147
 2:3875
```

Figure 7 The factor analysis of profile data

The plot which shows the distribution of risk groups is presented in Figure 8.

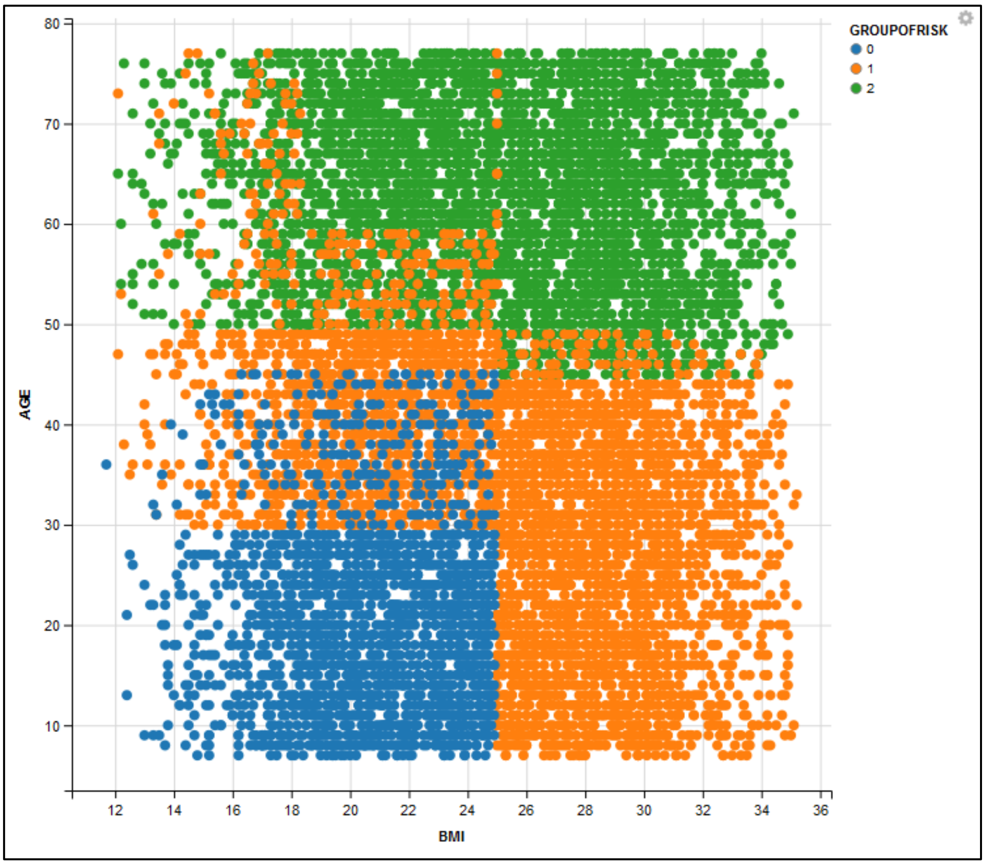


Figure 8 The distribution of risk groups

11.3 The analysis of relations between air pollution and personal data

To create a data frame with day-by-day air pollution data, it's used the command:

```
> realPollutionData <- dbGetQuery(conn_oracle, "select date_pol, quality, value from
pollution_hive_ext")
```

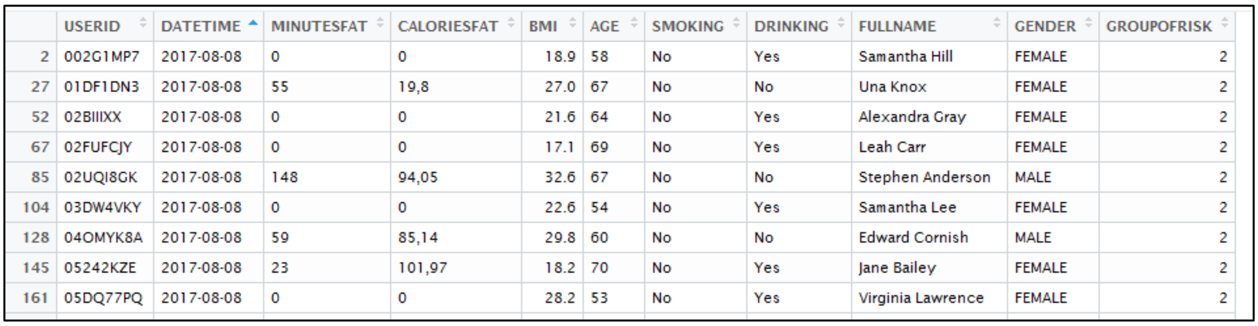
To create a data frame with day-by-day personal cardiac data, it's used the command:

```
> cardiacData<-dbGetQuery(conn_oracle, "select userid, datetime, minutesfat, caloriesfat from
fitbit_hive_ext")
```

Next, the information on cardiac activity may be merged with the profile information by user id, using the following command:

```
> newFrame <- merge(x = cardiacData, y = profileData, by = "USERID", all = TRUE)
```

The sample data of newFrame is presented in Figure 9.



	USERID	DATETIME	MINUTESFAT	CALORIESFAT	BMI	AGE	SMOKING	DRINKING	FULLNAME	GENDER	GROUPOFRISK
2	002G1MP7	2017-08-08	0	0	18.9	58	No	Yes	Samantha Hill	FEMALE	2
27	01DF1DN3	2017-08-08	55	19,8	27.0	67	No	No	Una Knox	FEMALE	2
52	02BIIIXX	2017-08-08	0	0	21.6	64	No	Yes	Alexandra Gray	FEMALE	2
67	02FUFCJY	2017-08-08	0	0	17.1	69	No	Yes	Leah Carr	FEMALE	2
85	02UQI8GK	2017-08-08	148	94,05	32.6	67	No	No	Stephen Anderson	MALE	2
104	03DW4VKY	2017-08-08	0	0	22.6	54	No	Yes	Samantha Lee	FEMALE	2
128	04OMYK8A	2017-08-08	59	85,14	29.8	60	No	No	Edward Cornish	MALE	2
145	05242KZE	2017-08-08	23	101,97	18.2	70	No	Yes	Jane Bailey	FEMALE	2
161	05DQ77PQ	2017-08-08	0	0	28.2	53	No	Yes	Virginia Lawrence	FEMALE	2

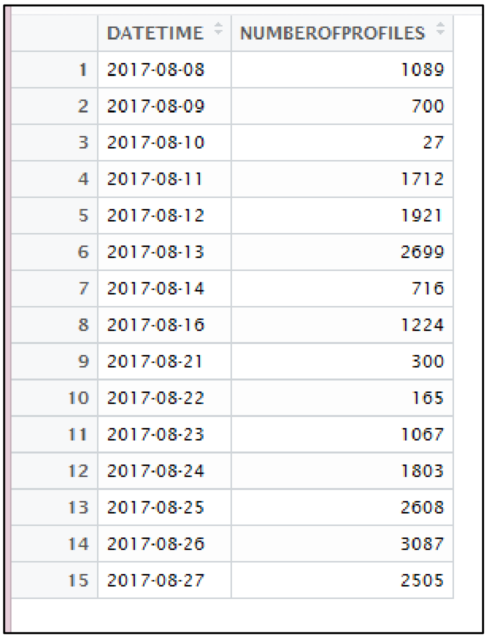
Figure 9 The daily cardiac data merged with profile data

Now, having all the data about the cardiac activity of users, and also about the level of air pollution, one can check their connection by visualizing this data. But first it is necessary to make a filter for profiles with a high risk of pulmonary and cardiac abnormalities, since they can hypothetically be susceptible to air quality. For this purpose, the library "dplyr" may be used:

```
> library(dplyr)

> cardiacDataForHighRiskProfiles <- filter(newFrame, GROUPOFRISK==2)
```

Next, it's needed to create a data frame which contains the date information and the number of people who have abnormal heart activity at this time. An example of this data frame shown in the Figure 10.



	DATETIME	NUMBEROFPROFILES
1	2017-08-08	1089
2	2017-08-09	700
3	2017-08-10	27
4	2017-08-11	1712
5	2017-08-12	1921
6	2017-08-13	2699
7	2017-08-14	716
8	2017-08-16	1224
9	2017-08-21	300
10	2017-08-22	165
11	2017-08-23	1067
12	2017-08-24	1803
13	2017-08-25	2608
14	2017-08-26	3087
15	2017-08-27	2505

Figure 10 Data frame of profiles with abnormal cardiac activity

To visualize the relation of air quality and people with heart rhythm disturbances, combine these data by date:

```
> newFrame2 <- merge(x = realPollutionData, y = dataSet, by = "DATETIME", all = TRUE)
```

The final plot of the newFrame2, constructed with the library “ggvis”, is shown in Figure 11.

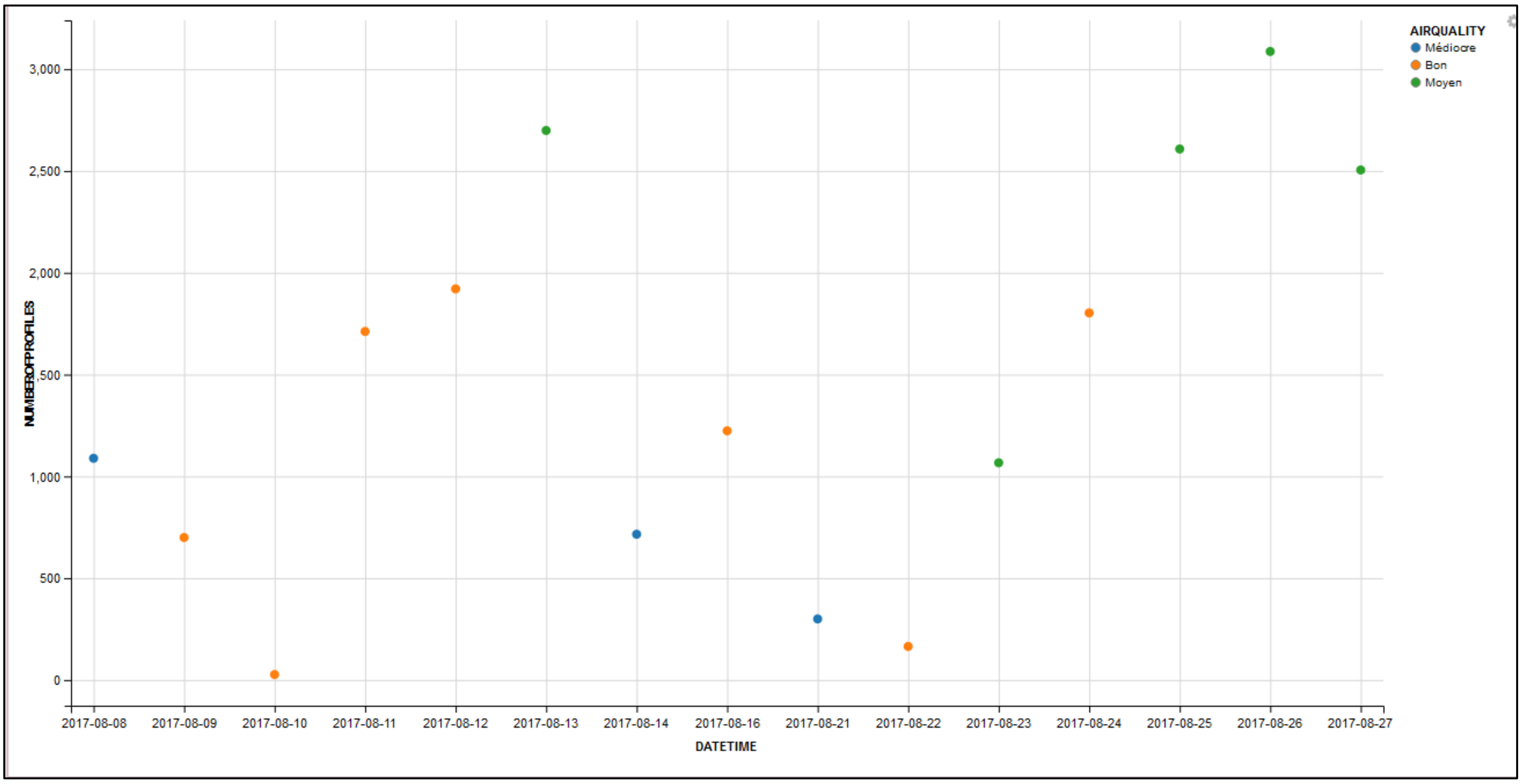


Figure 11 Profiles with abnormal cardiac activity - Quality of air Chart

In this chart, it can be seen how the number of people with abnormal cardiac activity and with a high risk of cardiac and pulmonary diseases and more susceptible to environmental influences changes, depending on air quality.

11.4 Classification and prediction on profile data

The features of R Language may also be used to classify profiles from a database, in this case by a risk group, for use it for prediction. First, summarize the distribution of classes in percentages:

```
> percentage <- prop.table(table(profileData222$GROUPOFRISK)) * 100
> cbind(freq=table(profileData222$GROUPOFRISK), percentage=percentage)
  freq percentage
0 1979   19.78802
1 4147   41.46585
2 3875   38.74613
```

The largest number of profiles in current data frame belong to the middle class of risk, the smallest - to the low. To make the prediction method, it’s necessary to create a list of 80% of the rows in the original dataset to use it for training (the remaining 20% will be used for prediction):

```
validation_index <- createDataPartition(profileData$GROUPOFRISK, p=0.80, li
st=FALSE)
```

```
validation <- profileData[-validation_index,]
profileData <- profileData[validation_index,]
```

Then - using linear and non-linear algorithms for learning:

```
# a) linear algorithms
set.seed(7)
fit.lda <- train(GROUPOFRISK~., data=profileData, method="lda", metric=metr
ic, trControl=control)
# b) nonlinear algorithms
# CART
set.seed(7)
fit.cart <- train(GROUPOFRISK~., data=profileData, method="rpart", metric=m
etric, trControl=control)
# kNN
set.seed(7)
fit.knn <- train(GROUPOFRISK~., data=profileData, method="knn", metric=metr
ic, trControl=control)
```

To compare their effectiveness, we use the resampling function:

```
> results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn))
> summary(results)

Call:
summary.resamples(object = results)

Models: lda, cart, knn
Number of resamples: 10

Accuracy
       Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
lda  0.8541  0.8571 0.8644 0.8649  0.8688 0.8851    0
cart 0.9039  0.9124 0.9306 0.9260  0.9365 0.9510    0
knn  0.9151  0.9220 0.9249 0.9259  0.9293 0.9410    0

Kappa
       Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
lda  0.7708  0.7752 0.7864 0.7872  0.7927 0.8192    0
cart 0.8483  0.8615 0.8903 0.8831  0.8999 0.9227    0
knn  0.8658  0.8772 0.8813 0.8831  0.8887 0.9073    0
```

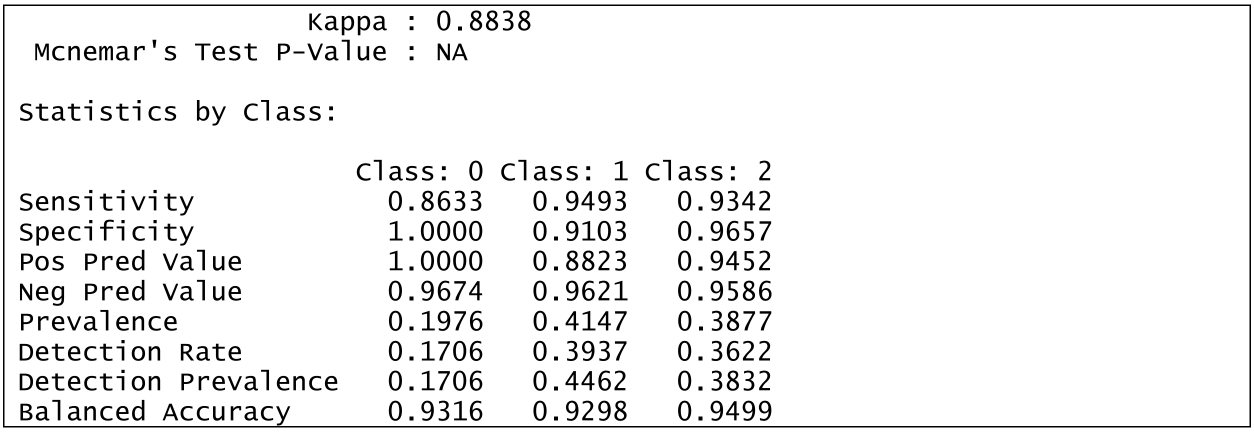
It can be seen that the CART method showed the greatest accuracy, so it can be used for prediction. Then one can estimate the skill of CART on the validation dataset:

```
> predictions <- predict(fit.cart, validation)
> confusionMatrix(predictions, validation$GROUPOFRISK)
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2
         0 341   0   0
         1  54 787  51
         2   0  42 724

Overall Statistics

               Accuracy : 0.9265
                 95% CI : (0.9141, 0.9375)
    No Information Rate : 0.4147
    P-Value [Acc > NIR] : < 2.2e-16
```



```
                   Kappa : 0.8838
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.8633   0.9493   0.9342
Specificity            1.0000   0.9103   0.9657
Pos Pred Value         1.0000   0.8823   0.9452
Neg Pred Value         0.9674   0.9621   0.9586
Prevalence             0.1976   0.4147   0.3877
Detection Rate         0.1706   0.3937   0.3622
Detection Prevalence   0.1706   0.4462   0.3832
Balanced Accuracy      0.9316   0.9298   0.9499
```

Thus, the accuracy of the prediction method is 95%, and it can be used to determine the group of risk of profile.

12 General Conclusion

The result of this internship is an implemented software, using the Big Data approach and Open Sours tools, which allows to manage and analyze data (including real-time data).

This software enriches the Data Lake of the Big Bridge – SE project with personal data and it can be used for collecting and analyzing the data, coming from smartwatch, as well as daily data on air pollution. These functions make the studies on the influence of air pollution on human health more personalized and broad. At the moment, the system was tested on the generated data, the analysis performed on real user data will be more correct.

The Android application developed as part of an internship can help users, especially those with a high risk of lung and cardiovascular disease, to monitor their health and receive recommendations depending on daily air quality.

The tasks set for the internship are fulfilled. For further work on the project, it is possible to implement the analysis on real data, as well as connect new data sources to the system.

Reference and bibliography

1. Holland W. W., Reid D. D. The urban factor in chronic bronchitis. Lancet
2. PAARC: Groupe Cooperative/Lelouche J Pollution atmosphérique et affections respiratoires chroniques ou à répétition. Bull. Eur. Physiopathol. Respir
3. Schenker M. B., Samet J. M., Speizer F. E., Gruhl J., Batterman S.Health effects of air pollution due to coal combustion in the Chestnut Ridge region of Pennsylvania: results of cross-sectional analysis in adults. Arch. Environ. Health
4. Euler G. L., Abbey D. E., Magie A. R., Hodlkin J. E.Chronic obstructive pulmonary disease symptom effects of long term cumulative exposure to ambient levels of total suspended particulates and sulfur dioxide in California Seventh-Day Adventist residents. Arch. Environ. Health
5. Portney P., Mullahy J.Urban air quality and respiratory disease. Reg. Sci. Urban Econ.
6. Schwartz J. Particulate air pollution and chronic respiratory disease. Environ. Res.
7. Forsberg, B., N. Stjernberg, and S. Wall. 1997. Prevalence of respiratory and hyperreactivity symptoms in relation to levels of criteria air pollutants in Sweden. Eur. J. Public Health 7/3:291–296.
8. Abbey D. E., Lebowitz M. D., Mills P. K., Petersen F. F., Beeson W. L., Burchette R. J.Long-term ambient concentrations of particulates and oxidants and development of chronic disease in a cohort of nonsmoking California residents. Inhal. Toxicol.
9. Abbey, D. E., B. E. Ostro, F. Petersen, and R. J. Burchette. 1995. Chronic respiratory symptoms associated with estimated long-term ambient concentrations of fine particulates less than 2.5 microns in aerodynamic diameter (PM2.5) and other air pollutants. J. Exp. Anal. Environ. Epidemiol. 5/2:137–159.
10. Abbey D. E., Hwang B. L., Burchette R. J.Estimated long term ambient concentrations of PM10 and development of respiratory symptoms in a nonsmoking population. Arch. Environ. Health
11. Scarlett J. F., Griffiths J. M., Strachan D. P., Anderson H. R.Effect of ambient levels of smoke and sulphur dioxide on the health of a national sample of 23-year-old subjects in 1981. Thorax
12. Schwartz J., Dockery D. W.Increased mortality in Philadelphia associated with daily air pollution concentrations. Am. Rev. Respir. Dis.

13. Spix C., Heinrich J., Dockery D., Schwartz J., Volksch G., Schwinkowski K., Collen C., Wichmann H. E.Air pollution and daily mortality in Erfurt, East Germany, 1980–1989. Environ. Health Perspect.
14. Dockery D., Pope A., Xu X., Spengler J. D., Ware J. D., Fay M. E., Ferris B. J., Speizer F. E.An association between air pollution and mortality in six U.S. cities. N. Engl. J. Med.
15. Touloumi G., Pocock S. J., Katsouyanni K., Trichopoulos D.Short-term effects of air pollution on daily mortality in Athens—a time-series analysis. Int. J. Epidemiol.
16. Schwartz J.Air pollution and daily mortality: a review and meta-analysis. Environ. Res.
17. Pope A., Thun M., Namboodiri M., Dockery H. D. W., Evans J. S., Speizer F. E., Heath C. W.Particulate air pollution as a predictor of mortality in a prospective study of U.S. adults. Am. J. Respir. Crit. Care Med.
18. Schwartz J., Morris R.Air pollution and hospital admissions for cardiovascular disease in Detroit, Michigan. Am. J. Epidemiol.
19. Burnett R., Dales R., Krewski D., Vincent R., Dann T., Brook J. Associations between ambient particulate sulfate and admissions to Ontario Hospitals for cardiac and respiratory diseases. Am. J. Epidemiol.
20. AirPaca. Association de surveillance de la qualité de l'air agréée par le ministère de l'environnement – URL – http://www.airpaca.org/
21. Wikipedia. R (programming language) – URL – https://en.wikipedia.org/wiki/R_(programming_language)
22. Wikipedia. Correlation and dependence – URL – https://en.wikipedia.org/wiki/Correlation_and_dependence
23. Wikipedia. Scrum (software development) – URL – https://en.wikipedia.org/wiki/Scrum_(software_development)
24. G. Mopolo-Moké, course Big Data et les SGBDs NoSQL, 2016
25. N. Pasquier, Data Analytics & Mining course, 2016
26. B. Renaut, Introduction à Hadoop & MapReduce course, 2015
27. S. Miranda, course Introduction stratégique des Bases de Données à Big Data, 2015

Annexes

Annex 1: Project Plan

Annex 2: Grouping profiles by risk level

These groups are formed based on: gender, ages, bad habits, physical activity, and body mass index.

Descriptions of these groups are on the table below.

Group of risk	Low	Middle	High

Description	Gender: male, female;
	Ages: less or equals 45;
	Bad habits: none;
	Physical activity: normal or high level;
	Body mass index: 18.5 – 24.9
		Gender: male, female;
		Ages: if there are no bad habits, the ages are in range from 46 to 60 years.
		In case of bad habits existence, the ages are in range from 30 to 50 years.
		Physical activity: normal or low level.
		Body mass index: less than 18.5 and from 25 to 30.
			Gender: male, female
			Ages: if user has no bad habits and his body mass index is from 18.5 to 24.9, the ages are from 60 and older. Main case of most diseases is the age of user.
			In case of bad habits existence, ages are in range from 50 and older.
			In case of high body mass index (more than 25) ages are in rage from 50 and older.
			In case of body mass index more than 25, also user has bad habits, the ages are in range from 45 and older.
			Physical activity is on low level.

Annex 3: SQL queries for forming the R Language data frames

Selecting the high-risk profiles:

```
select userid, bmi, age, smoking, drinking, fullname, gender
                    from (
                    select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking,
drinking, fullname, gender from profile_hive_ext)
                    where bmi>=18.5 and bmi<=24.9 and age>=60)
                    UNION
```

```
			(select userid, bmi, age, smoking, drinking, fullname, gender
			from (
			select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
			where (smoking='Yes' or drinking='Yes') and age>=50)
			UNION
			(select userid, bmi, age, smoking, drinking, fullname, gender
			from (
			select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
			where bmi>25 and age>=50)
			UNION
			(select userid, bmi, age, smoking, drinking,fullname, gender
			from (
			select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
			where bmi>25 and (smoking='Yes' or drinking='Yes') and age>=45;
```

Selecting the medium-risk profiles:

```
select userid, bmi, age, smoking, drinking, fullname, gender from
(
 (select userid, bmi, age, smoking, drinking, fullname, gender
 from (
 select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext) )
 MINUS (
 (select userid, bmi, age, smoking, drinking, fullname, gender
 from (
 select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
 where bmi>=18.5 and bmi<=24.9 and age>=60)
 UNION
 (select userid, bmi, age, smoking, drinking, fullname, gender
 from (
 select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
 where (smoking='Yes' or drinking='Yes') and age>=50)
 UNION
 (select userid, bmi, age, smoking, drinking, fullname, gender
 from (
 select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking,
fullname, gender from profile_hive_ext)
 where bmi>25 and age>=50)
 UNION
 (select userid, bmi, age, smoking, drinking, fullname, gender
 from (
 select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking,
fullname, gender from profile_hive_ext)
 where bmi>25 and (smoking='Yes' or drinking='Yes') and age>=45)))
 where (age>=46 or bmi >=25) or ((smoking='Yes' or drinking='Yes') and age>=30)
```

Selecting the low-risk profiles:

```
select userid, bmi, age, smoking, drinking, fullname, gender
from (select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking,
fullname, gender from profile_hive_ext))
                    MINUS
                    (select userid, bmi, age, smoking, drinking, fullname, gender from
                    ((select userid, bmi, age, smoking, drinking, fullname, gender
                    from (
                    select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking,
drinking, fullname, gender from profile_hive_ext) )
                    MINUS ((select userid, bmi, age, smoking, drinking, fullname, gender
                    from (select ROUND(weight/(height*height)*10000,1) as bmi, age, userid,
smoking, drinking, fullname, gender from profile_hive_ext)
                    where bmi>=18.5 and bmi<=24.9 and age>=60)
                    UNION
                    (select userid, bmi, age, smoking, drinking, fullname, gender
                    from (select ROUND(weight/(height*height)*10000,1) as bmi, age, userid,
smoking, drinking, fullname, gender from profile_hive_ext)
                    where (smoking='Yes' or drinking='Yes') and age>=50)
                    UNION
                    (select userid, bmi, age, smoking, drinking, fullname, gender
		from (select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
		where bmi>25 and age>=50)
		UNION
		(select userid, bmi, age, smoking, drinking, fullname, gender
		from (select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
		where bmi>25 and (smoking='Yes' or drinking='Yes') and age>=45)))
		where (age>=46 or bmi >=25) or ((smoking='Yes' or drinking='Yes') and age>=30))
MINUS ((select userid, bmi, age, smoking, drinking, fullname, gender
		from (
		select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
		where bmi>=18.5 and bmi<=24.9 and age>=60)
		UNION
		(select userid, bmi, age, smoking, drinking, fullname, gender
		from (select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
		where (smoking='Yes' or drinking='Yes') and age>=50)
		UNION
		(select userid, bmi, age, smoking, drinking, fullname, gender
		from (select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
		where bmi>25 and age>=50)
		UNION
		(select userid, bmi, age, smoking, drinking, fullname, gender
		from (select ROUND(weight/(height*height)*10000,1) as bmi, age, userid, smoking, drinking, fullname, gender from profile_hive_ext)
		where bmi>25 and (smoking='Yes' or drinking='Yes') and age>=45);
```