SMBUD 2021 - Project work 2

Aman Gabba - 10793117
Andrea Cerasani - 10680486
Giovanni Demasi - 10656704
Pasquale Dazzeo - 10562130
Vlad Marian Cimpeanu - 10606922

POLITECNICO
MILANO 1863

Contents

1 Introduction

1.1 Problem Specification

The aim of this project was to design a 'query document data structure' in MongoDB for supporting a certification application for COVID-19. The database must register all the necessary information about the users including their tests and vaccination status in order to know if a certain person can be considered 'harmless' and so able for instance to visits some public spaces. The application using this database will be able to exploit all the data coming from test results and vaccination status.

1.2 Hypothesis

The assumptions taken into account are the following:

- The recovery document has been introduced in the document DB to distinguish the usual negative test result to the one did after an infection to consider a certain person healed from Covid-19. Using this document can be computationally useful because otherwise, to know if a person has a recovery certificate, it would be necessary to scan all the tests and check the relative results and dates to deduce an infection and a consequent healing. It has been also assumed that the recovery certificate doesn't get automatically released, but it is registered manually by whom of competence after a specific iter. The recovery certificate was not expressly required by the problem specification but it has been introduced because it can be used as support to calculate the expiration date of the certificate.

- Authorized bodies has been implemented as an autonomous collection and a pointer has been used in the document Place instead of a sub-document because they are few and they are repeated a big amount of times. Even if redundancy can be accepted in MongoDB, the information contained in this document are not widely used, for this reason having multiple copies of them could be inefficient.

- It has been assumed that the database only contains documents related to the Italian state. Thanks to this the government rules are well known and well defined and it allows to have the expiration date directly in the certificate document since rules changes rarely happen. In this last case the update would be a little time consuming but this choice is widely justified considering the fact that checking the validity of a certificate is one of the main reason of our database and having it directly in the document allow to access it instantaneously without any computation.

- As said in the problem description, the designed database would be mainly used to check the validity of a determined certificate. For this reason it has been decided to make a collection of certificates (with possibly more than one associated to the same person) instead of making a single one for

every person containing all the relevant data. This choice allows to make the certificate access faster since they are indexed by UIC and it anyway allows to reconstruct the Covid history of a given person by looking for all his/her certificates also indexed by fiscal code. In conclusion, this choice allows to have the same database that would be obtained with the other approach but simplify the queries for which the database would be mainly used.

- The following database has been designed with the awareness of existence of the Covid certification, so the structure thought has been taken into consideration thanks to its efficency in the most used cases. If the same database would have to be designed without knowing the existence of the certification adding it later maybe it would have make to a different and adapted, and probably also less efficient, design.

2 ER diagram

The designed ER diagram contains the following entities: Person, Place, Authorized body, Certificate, Event and Sanitary operator. Nurse and Doctor have been designed as sub-entities of Medical Stuff, Vaccine and Swab have been designed as sub-entities of Certificate and the Recovery entity is related to a swab.

3 Document diagram

The designed Document diagram is shown below. Certificate and Authorized body are represented as collections of documents.

The certificate document doesn't represent an unique person but it is specific for a determined event. It can have one between Recovery, Test and Vaccination as sub-document and these last two have both Place and Sanitary operators as sub-documents which contain all the information related to the place and the doctor/nurse involved in the specific event.

The Recovery document has been implemented to make the recovery certificate validation easier and faster. Otherwise, doing this validation through the scan of the tests information would have been more complex due to the possibility of a person to do an unlimited amount of tests. With this design choice the time complexity of validate a recovery certificate has been reduced.

4 Dataset description

The Dataset has been built by making a script in Python, using some useful packages like: random-italian-person that automatically generates people with random attributes (this package has been modified due to our necessity); the official PyMongo driver for Python for the communication with the Database since it allows to make queries directly from Python.

It has been also used a CSV document, downloaded directly from the official Italian Government github repository, of all the vaccination centres (due to the limited scale of the to make database only some regions have been taken into consideration).

A CSV file for vaccines information has been built by hand due to necessity with the relevant information.

5 Queries and Commands

5.1 Queries

5.1.1 Query 1

Query description...

```
CODE
```

5.1.2 Query 2

Query description...

```
CODE
```

5.1.3 Query 3

Query description...

```
CODE
```

5.1.4 Query 4

Query description...

```
CODE
```

5.1.5 Query 5

Query description...

```
CODE
```

5.2 Commands

5.2.1 Command 1

Command description...

```
CODE
```

5.2.2 Command 2

Command description...

```
CODE
```

5.2.3 Command 3

Command description...

```
CODE
```

6 Generator database

The `"MongoDB-assignment/main.py"` file is responsible of generating a random database. In order to correctly run the generator, it is mandatory to... This script exploits the following packages: `pandas, numpy, PyMongo and python-codicefiscale`.

6.1 Place generation

The CSV file `"datasets/locations.csv"` has been used to generate all the places related to tests and vaccinations. Due to database scale limitation the generation of places has been limited to the Lombardy region.

6.2 Certificate generation

In order to consistently generate people certificates...

6.3 Sanitary operator generation

Sanitary operators information are randomly generated by using our adapted version of the random italian person package.

6.4 Authorized body generation

...

7 Application description

Goal of the application `"GUI/App.py"` is to give some information about the Covid-19 Certification. The user can use the application in order to:

- Given information about a specific people, print all his valid certifications.
- Given a UCI, check if the related certificate is still valid, who it belongs to and its expiration date according to the actual government regulation.

8 User guide

The application is meant to work on pc or macOS. As the application has been entirely developed in Python, it is not operative system dependent.
In order to correctly run the application it is mandatory to...
The user must check to have installed correctly in the virtual environment the following pacakges: `tkinter`, `pandas`, `PyMongo`, `numpy` and `python-codicefiscale`.

9 Conclusion

Some interesting conclusions can be drawn from the development of this project: Document databases are, if well designed, easy to use and really scalable.

...

10 References and Sources

- Random-italian-person package: https://pypi.org/project/random-italian-person
- PyMongo package: https://docs.mongodb.com/drivers/pymongo/
- Tkinter package: https://docs.python.org/3/library/tkinter.html
- Italian Government repository: https://github.com/italia/covid19-opendata-vaccini