POLITECNICO
MILANO 1863

Covid-19 Certificates

Systems and Methods for Big and Unstructured Data
SMBUD-Fall 2021

Amirhossein Zhalehmenrabi-10832946 Ghodrat Rezaei-10695039 Mahsa Meymari-10833116

Mahshid Shiri-10832947 Alireza Nadirkhanlou -10823835

Problem Specification

Problem Description

The goal is to manage certificates on tests and vaccines. The app is used for:

- ✓ Representing status of vaccinated people, it means it should shows how many doses a person got and all details about it.
- ✓ Representing test results of a person.
- ✓ Checking whether a result is valid or not, based on rules that can change at any time.
- ✓ Containing information about emergency contacts.
- ✓ Containing information about authorized bodies.

How we managed…

- *A person?* Each person is an instance with attributes corresponding for personal data.
- *Flexible number of doses with different brands?* Each dose is an instance, each person can have a list of vaccine doses that contains details about that injection. So, the length of list can vary and we can have as many doses as we want. Also, doses are independent, so it is not compulsory that the next dose match the previous ones.
- *Rules?* As the application is used for checking the validity of each dose, we have an instance to make it possible. The policy makers can set the months that each dose is valid. Each type of vaccine can have its specific validity for different doses.
- *Evolutions in rules?* Rules are set as independent instances, so at any time that policy makers demand, they can change validities. We check our output all the times before showing it to users, so if the result is not valid based on rules, it is not returned.

Hypothesis

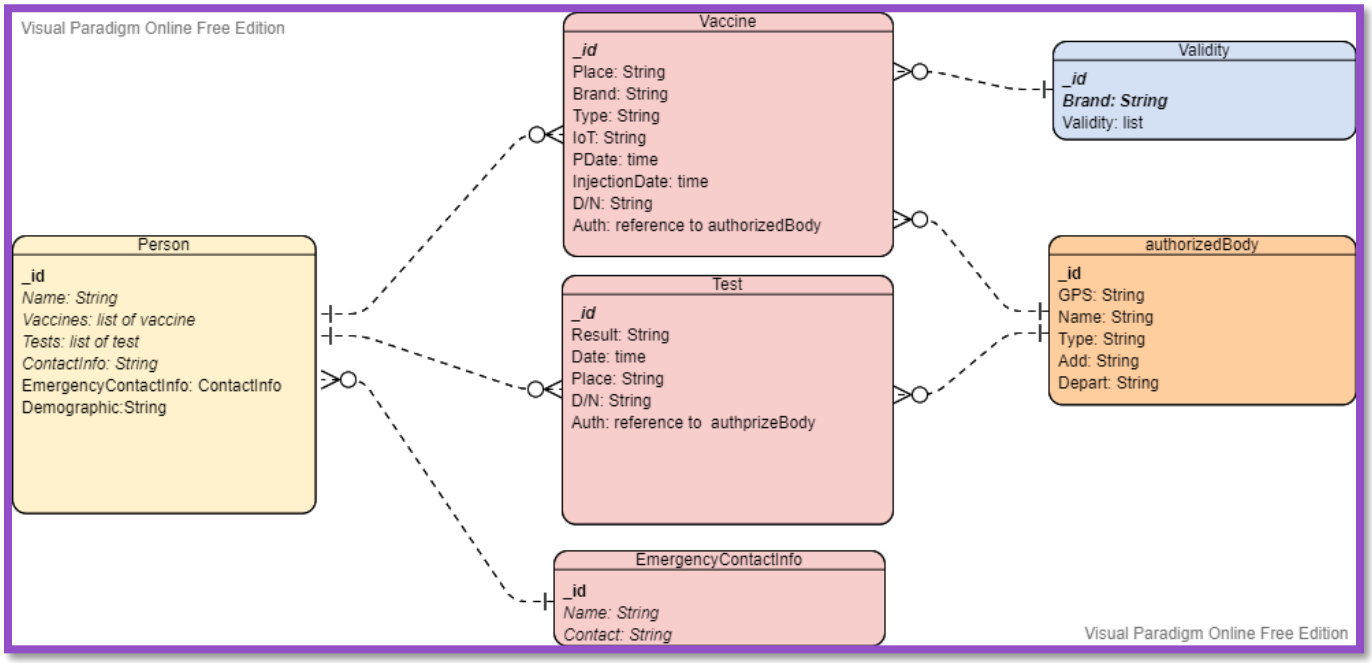
- Emergency contact is supported by saving one name with a contact number, not more.
- Authorized body for each vaccine's injection or tests are independent.
- Only one doctor or nurse name is saved as the attended person in test/vaccine.
- The validity of vaccines doses are in scale of months.

Diagrams & Dataset description

ER:

We used Visual Paradigm to produce ER model. In our model we have 6 entity:

1. AuthorizedBody: this entity contains details of authorized bodies only.
2. EmergencyContactInfo: this entity only contains a name and phone number. It is used for emergency calls.
3. Test: In a covid-19 test, we need to save the result, date and place that the test is taken, the attended doctor or nurse and the authorized body that issued this test. So, this entity has a many to one relation to Authorized body entity because a test can be issued by only one body and a body can issue many tests.
4. Vaccine: in this entity all details of vaccine is saved. Also, the information about injection is saved. As this is issued by an authorized body, it has a many to one relation with authorized body entity.
5. Person: which saves personal and contacts information and is related to entities that contains information on tests and vaccines. As a person can have as many tests and vaccines as wants or the rules forced, this relations are one to many. Also, this entity is in one to one relation to EmergencyContactInfo because a person can only has one person for contact on emergency situations.
6. Validity: This entity shows that how long different dose of vaccines are valid for different brands. For example, if the brand is set to "Astrazenca" then we have this list: [3, 9, and 10] which means that the first dose is valid for 3 months, second dose is valid for 9 months and the third dose is valid for 10 months.



Document:

From the above ER, we designed our documents. Test, vaccines and emergency contact information are implemented as embedded documents. However, to avoid redundancy, authorized bodies are implemented independently and are referenced.

Collection	Entity that mapped to document	Embedded documents	Reference document
AuthorizedBodies	Authorizedbody	----	----
People	Person	Vaccine, Tests, EmergencyContactInfo	authorizedBody which is referenced inside tests and vaccines that are embedded
Validities	Validity	----	----

Dataset description:

By using pymongo library, we accessed the datasets on the cloud, and by using Random library, we generated a dataset consists of 300 people on our documents. The number of people is flexible and can be scaled. Also, for referencing Authorized bodies, we defined a simple ID so that we can store the information in a simpler way. We need to access them using their IDs, so it was possible to use random IDs. However, this simpler IDs helps to troubleshoot the system in a faster way. An example of documents is represented in the table below.

Person


```
_id: ObjectId("61b3a7d9e3babadc62ad96fc")
Name: "Mary"
Vaccines: Array
  0: Object
      Place: "Room6"
      Brand: "Pfizer"
      Type: "XX"
      Iot: "WW"
      PDate: 2018-07-27T00:00:00.000+00:00
      InjectionDate: 2018-08-27T00:00:00.000+00:00
      D/N: "Doctor1"
      Auth: 4
      _id: ObjectId("61b3a7d9e3babadc62ad96f9")
  1: Object
      Place: "Room4"
      Brand: "Razi Cov Pars"
      Type: "XX"
      Iot: "WW"
      PDate: 2018-08-27T00:00:00.000+00:00
      InjectionDate: 2018-09-27T00:00:00.000+00:00
      D/N: "Doctor1"
      Auth: 2
      _id: ObjectId("61b3a7d9e3babadc62ad96fb")
Tests: Array
  0: Object
      Result: "Pos"
      Date: 2018-07-09T00:00:00.000+00:00
      Place: "Hospital"
      D/N: "Nurse1"
      Auth: 4
      _id: ObjectId("61b3a7d9e3babadc62ad96fa")
ContactInfo: "+39123456"
EmergencyContactInfo: Object
   Name: "Sue"
   ContactInfo: "+39654789"
Demographic: "YY"
```


Collections

LastBig

- AuthorizedBodies
- People
- Validities

Authorized body


```
_id: 1
GPS: "64.84097542726741,45.98639616964263"
Name: "Milano Laboratory"
Type: "Laboratory"
Add: "WW"
Depart: "EE"
```


Validity


```
_id: ObjectId("61b3a7d8e3babadc62ad96f4")
Brand: "Pfizer"
Validity: Array
   0: 3
   1: 6
   2: 8
```


Test


```
_id: ObjectId("61b3a7d9e3babadc62ad96fa")
Result: "Pos"
Date: 2018-07-09T00:00:00.000+00:00
Place: "Hospital"
D/N: "Nurse1"
Auth: 4
```


Vaccine


```
_id: ObjectId("61b3a7d9e3babadc62ad96f9")
Place: "Room6"
Brand: "Pfizer"
Type: "XX"
Iot: "WW"
PDate: 2018-07-27T00:00:00.000+00:00
InjectionDate: 2018-08-27T00:00:00.000+00:00
D/N: "Doctor1"
Auth: 4
```

Commands & Queries

Commands:

- *Add New Vaccine Record for the person who got vaccinated recently:*
  - ✓ **Input:** Name of person, Place where the person got his or her new vaccination, Vaccine Brand, Type, IOT and PDate.
  - ✓ **Output:** All properties of new vaccine are added to the Vaccines embedded document, the injection date is the day data is inserted in database.

  Ps: First the name of person is found in "Persons" collection. Then, mentioned properties of new vaccine record are added in the corresponding embedded document preserving also old vaccine records.
- *Add New Test Record for the person who gave new Test recently:*
  - ✓ **Input**: Name of person, Result of test (negative or positive), Date of test, Place of test and the person who performs testing that can be doctor or nurse.
  - ✓ **Output**: All properties of new Test are added to the Tests embedded document.

  Ps: As previous command, first the name of person is found in "Persons" collection. Then mentioned properties of new test record are added in the corresponding embedded document preserving also old test records.
- *Insert new Document in Persons collection:*
  - ✓ **Input**: All personal information including Name and Contact Info (phone number), Vaccination details (Place, Brand, Type, IOT, PDate), Test details (Result, Date, Place and person in charge of performing test), Emergency Contact Information details (emergency telephone number).
  - ✓ **Output**: Adding one new document having all sort of defined fields in data base with corresponding inserted values.

Queries:

1. *Returning Emergency Contact by name*
   - ✓ **Input**: name of the person.
   - ✓ **Output**: showing the emergency contact for the requested person.
2. *Percentage of each Vaccine Brand Used In vaccination process of society(Astrazeneca, Pfizer, J&J, Razi Cov Pars, Barekat and COVIran Barkat)*
3. *Returning number of each type of Authorized Body*
   - ✓ **Input**: the type of authorized body for example Hospital
   - ✓ **output**: number of authorized body in this type
4. *List of people who were vaccinated in a special authorized body*
   - ✓ **Input**: the name of an authorized body

- ✓ **Output**: showing the name of people who were vaccinated in requested authorized body.

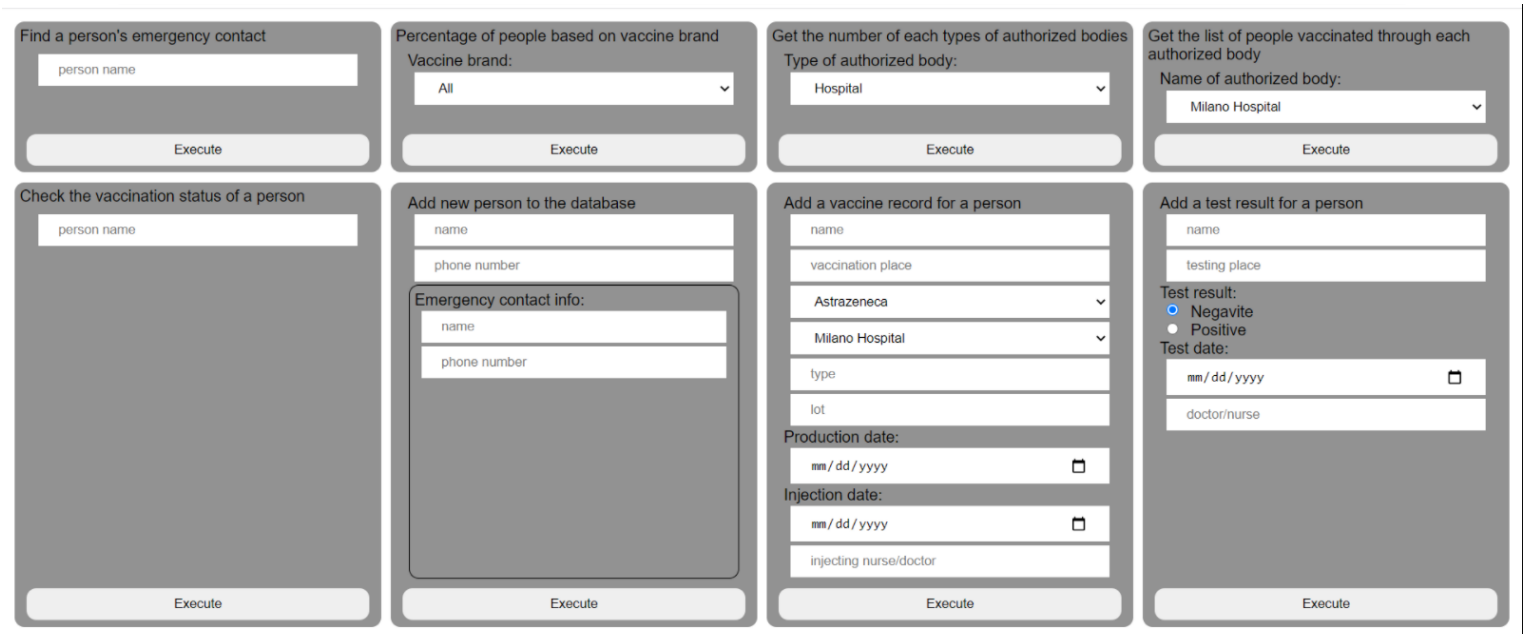
5. *Check if a person's vaccine is still valid or has been vaccinated at all*

Ps: In Validity collection, specific duration threshold was defined for each vaccine brand and which dose it is upon which vaccination is not valid. This threshold was set considering the health index and other indications of vaccines' performance.

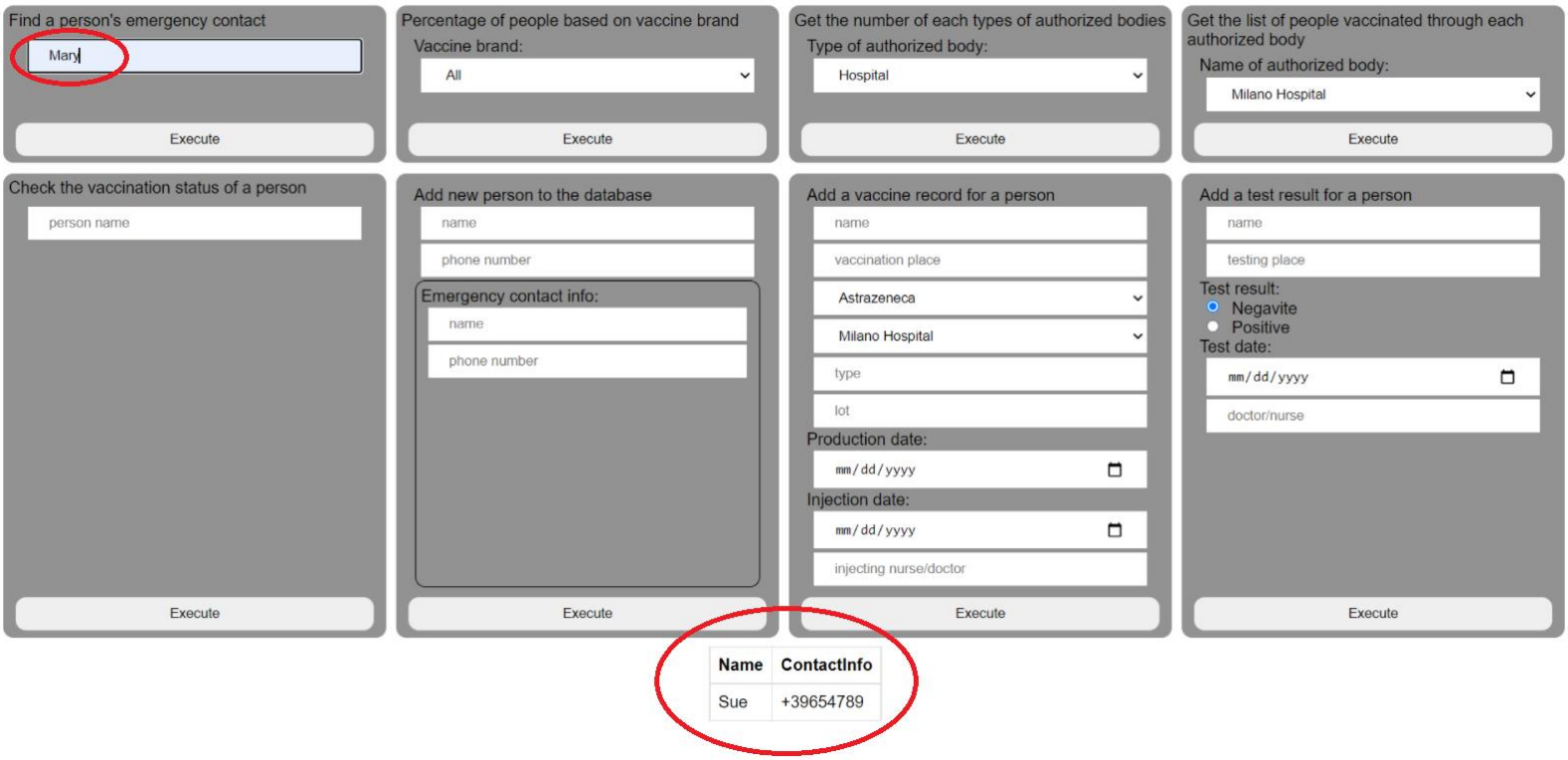
UI Description

UI

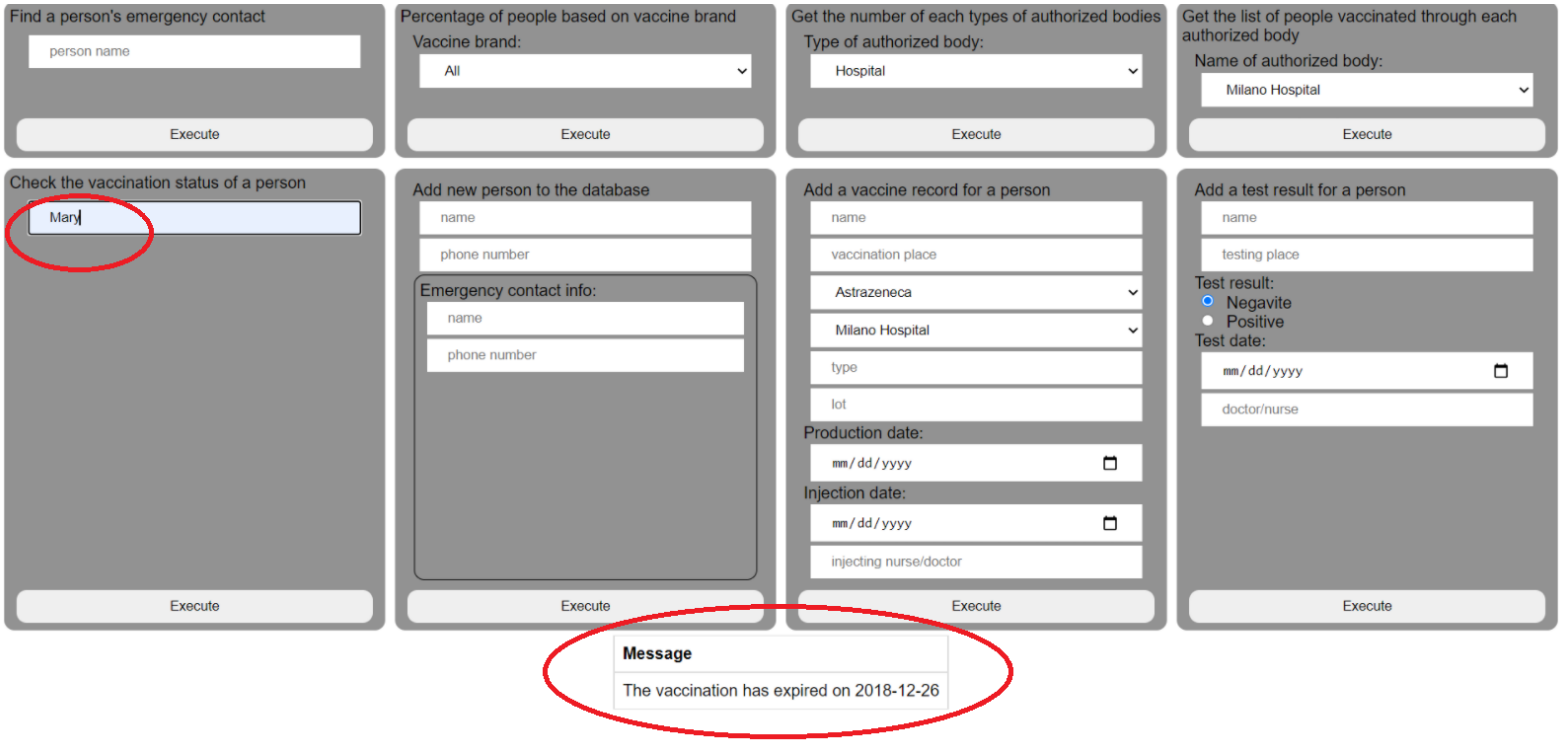
For the application, we decided to build a simple web-based dashboard using Flask as the backend framework, and the PyMongo library to connect to our MongoDB library. We implemented only the essential components of the web application that utilized the functions and queries we wrote for the exercise, and put the rest of our effort on the exercise itself. You can find a few screenshot's of the dashboard below.



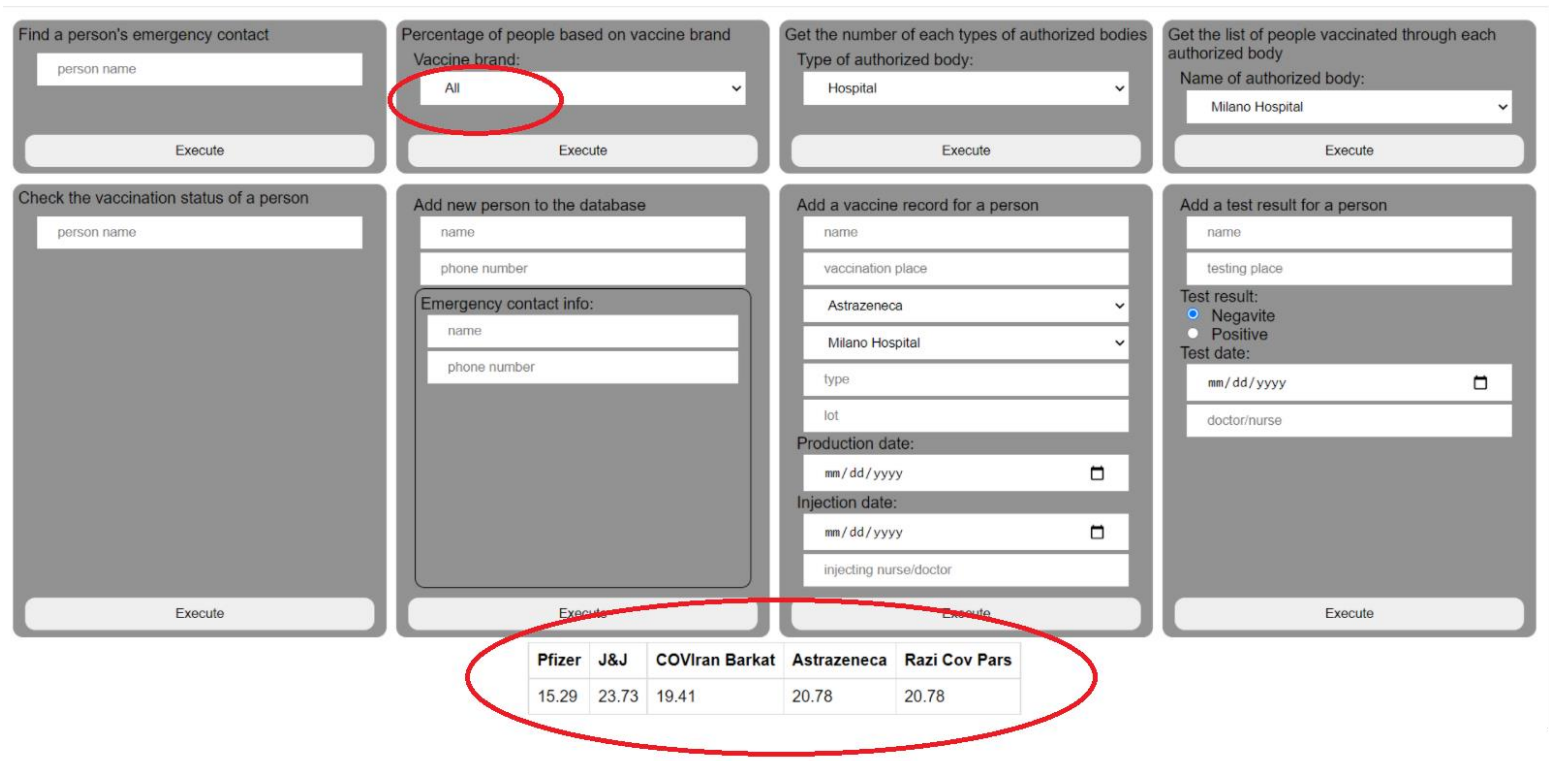
Dashboard



Emergency Contact Info



Vaccine status



Pfizer	J&J	COVIran Barkat	Astrazeneca	Razi Cov Pars
15.29	23.73	19.41	20.78	20.78

Percentage of people vaccinated with different brands