Project report - MongoDB

Introduction

Second delivery of the project work for systems and methods for big and unstructured data.
Members of the group 5:

- Alice Brugnoli, 10608045.
- Ginevra Iorio, 10607321.
- Francesco Gonzales, 10614882.
- Leonardo Caponi, 10618575.
- Andrea Ceresetti, 10633584.

Problem Specification

The task of the project is to store the certificates of vaccination/testing and the authorized bodies that can deliver them, designing, storing and querying data on a NoSQL DB.

In our database we had to record:

- Who is vaccinated when and where (first and second doses, with which type of vaccine).
- The kind of vaccine issued (brand, type, lot, production date).
- The list of tests taken, when and where, and the respective results.
- Doctors and nurses attending the vaccination/tests.
- Demographic details of the vaccinated person.
- Emergency contact person and details.

The description of the authorized bodies includes the full description of the organization and location with all the relevant details.

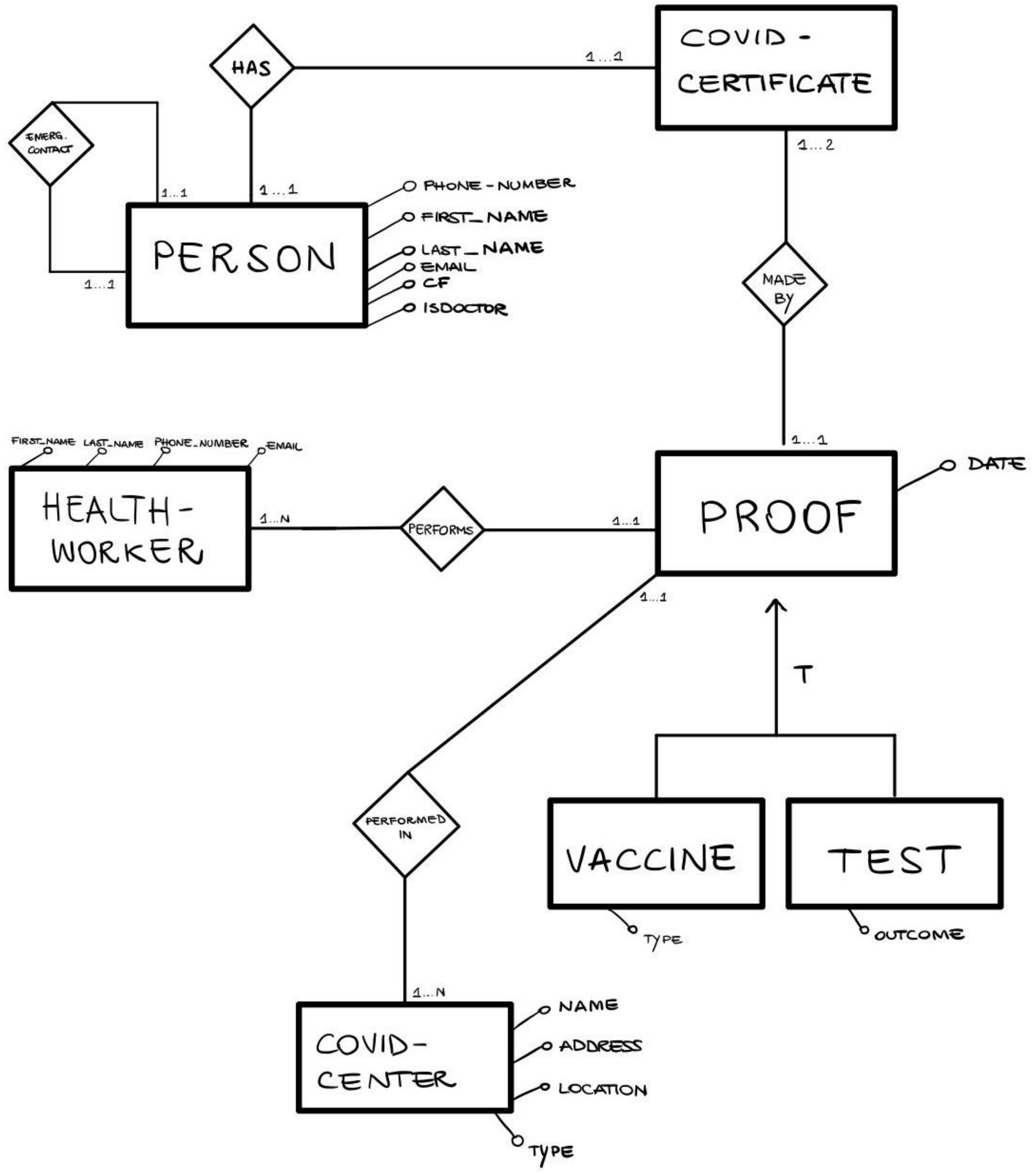
Hypothesis

- Nurses and doctors can both perform vaccines and tests.
- Every person in the database has at least one vaccine or one test.
- Many reads on certificates, few writes, therefore optimize for reads.

ER diagram

PERSON: For every certificate we have a person. Every person has an emergency contact.
PROOF: Every certificate has at least one vaccination or one covid - test.

HEALTH - WORKER: This is the authorized body that can perform a vaccine or a covid - test. They're persons and doctors.
COVID - CENTER: The place where persons can go to test their positivity or to go get a vaccine.



Dataset description

The dataset was created in 2 steps. In the first step using mockaroo [1], a free online service, we created 3 json files: people.json, tests.json and vaccines.json. Then using a python script we generated a new json file, certificates.json. The script creates certificates by joining a person to a test list and / or vaccine list. Below there is a template of a document contained in the final database.

```
"_id": {
  "$oid": "61b5dd931220b929bed85ace"
 },
 "person": {
  "_id": {
   "$oid": "61b4d863fc13ae5f7e000a28"
  },
  "first_name": "Alvina",
  "last_name": "Filpo",
  "email": "afilpo0@nps.gov",
  "phone_number": "236-332-8895",
  "CF": "PDNTED62M21N300R",
  "is_doctor": false,
  "emergency_contact": {
   "first_name": "Esra",
   "last_name": "Mitkcov",
   "phone_number": "374-580-8318",
   "email": "emitkcov5r@themeforest.net"
  }
 },
 "tests": [
  {
   "_id": {
    "$oid": "61b4d6f3fc13ae2c6600073d"
   },
   "outcome": false,
   "date": {
    "$date": "2021-03-07T14:38:23Z"
   },
   "covid_center": {
    "name": "Conn, Gottlieb and Emmerich",
    "address": "08 Florence Crossing",
    "center_type": "Pharmacy",
    "location": {
     "type": "Point",
     "coordinates": [
      12.1235658,
      37.7795618
     ]
    }
   },
   "health_worker": {
    "first_name": "Dinnie",
    "last_name": "Tunniclisse",
    "phone_number": "815-203-3399",
    "email": "dtunniclisse8f@barnesandnoble.com"
   }
  }
 ],
 "vaccines": [ {
   "_id": {
    "$oid": "61b4d8d2fc13ae5ca2000758"
   },
   "brand": "Moderna",
   "date": {
    "$date": "2021-02-04T10:26:43Z"
   },
   "covid_center": {
    "name": "Kerluke-Wehner",
    "address": "2 Grayhawk Parkway",
    "center_type": "Pharmacy",
    "location": {
     "type": "Point",
     "coordinates": [
      34.9255083,
      64.1478236
     ]
    }
   },
   "health_worker": {
    "first_name": "Laural",
    "last_name": "Nash",
    "phone_number": "325-825-5666",
    "email": "lnashaf@shinystat.com"
   }
  }
 ]
}
```

Queries & Commands

Queries:

- Return a valid certificate (the vaccination certificate expires after 9 months, the test certificate expires after 2 days).

```
db.certificates.findOne({"$or":
[
 {"tests":
   {"$elemMatch": {
       "outcome": false,
       "date": {"$gt": new Date(ISODate().getTime() - 2*24*60*60*1000)}
    }}},
 {"vaccines":
  {"$elemMatch": {
       "date": {"$gt": new Date(ISODate().getTime() - 9*30*24*60*60*1000)}
   }}}
]})
```

Frame of the output::

```
{ _id: ObjectId("61b62ca908e0a1767ed25386"),
  person:
   { _id: ObjectId("61b4d863fc13ae5f7e000a28"),
     first_name: 'Alvina',
     last_name: 'Filpo',
     email: 'afilpo0@nps.gov',
     phone_number: '236-332-8895',
     CF: 'PDNTED62M21N300R',
     is_doctor: false,
     emergency_contact:
      { first_name: 'Saundra',
        last_name: 'Bottrill',
        phone_number: '183-533-6034',
        email: 'sbottrill2u@geocities.jp' } },
  tests:
   [ { _id: ObjectId("61b4d6f3fc13ae2c660004c8"),
       outcome: true,
       date: 2021-10-19T03:06:09.000Z,
       covid_center:
        { name: 'Gusikowski and Sons',
          address: '43175 Eastlawn Park',
          center_type: 'Vaccination Hub',
          location: { type: 'Point', coordinates: [ 50.52546,
42.6645851 ] } },
       health_worker:
        { first_name: 'Deloria',
          last_name: 'Vannozzii',
          phone_number: '248-511-6919',
          email: 'dvannozzii1h@rakuten.co.jp' } },
     { _id: ObjectId("61b4d6f3fc13ae2c6600068e"),
       outcome: false,
       date: 2021-05-28T05:01:55.000Z,
       covid_center:
        { name: 'Prosacco, Goyette and Hickle',
          address: '19 Beilfuss Center',
          center_type: 'Hospital',
          location: { type: 'Point', coordinates: [ -23.4500163,
-51.70451 ] } },
       health_worker:
        { first_name: 'Channa',
          last_name: 'Frany',
          phone_number: '485-713-3596',
          email: 'cfrany7d@linkedin.com' } },
```

- Find people with a valid test (not vaccine) certificate on October 24, 2021.

```
db.certificates.find({"tests":
   {"$elemMatch": {
      "outcome": false,
      "date": {"$gt": new Date(ISODate("2021-10-24") - 2*24*60*60*1000),
                "$lt": new Date(ISODate("2021-10-24"))}
    }}
}, {"person.first_name": 1, "person.last_name": 1})
```

```
{ _id: ObjectId("61b62ca908e0a1767ed2539d"),
  person: { first_name: 'Kiah', last_name: 'Ogilby' } }

{ _id: ObjectId("61b62ca908e0a1767ed2546a"),
  person: { first_name: 'Perle', last_name: 'Archbold' } }
```

- Find a person whose certificate is invalid on November 4, 2021.

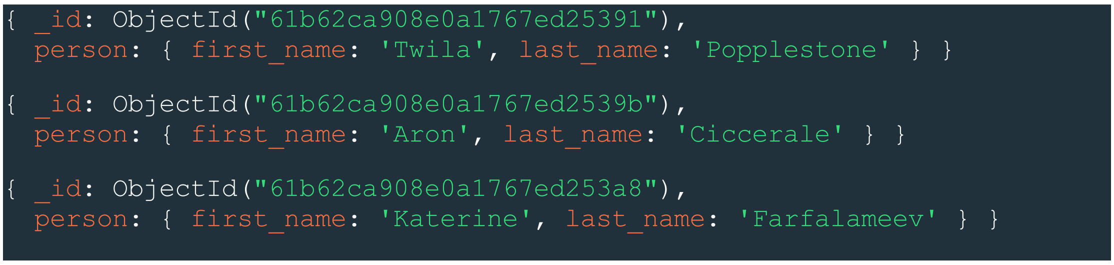
```
db.certificates.findOne({"$nor":
[
 {"tests":
   {"$elemMatch": {
      "outcome": false,
      "date": {"$gt": new Date(ISODate("2021-11-04") - 2*24*60*60*1000)}
    }}},
 {"vaccines":
  {"$elemMatch": {
      "date": {"$gt": new Date(ISODate("2021-11-04") - 9*30*24*60*60*1000)}
   }}}
 ]},
 {"person.first_name": 1, "person.last_name": 1})
```

```
{ _id: ObjectId("61b62ca908e0a1767ed25388"),

  person: { first_name: 'Mirabel', last_name: 'Alyutin' } }
```

- Find unvaccinated doctors.

```
db.certificates.find({"$and" : [
{"person.is_doctor" : true},
{"vaccines" : {"$size" : 0}}]
},
{"person.first_name": 1, "person.last_name": 1})
```

Frame of the output:

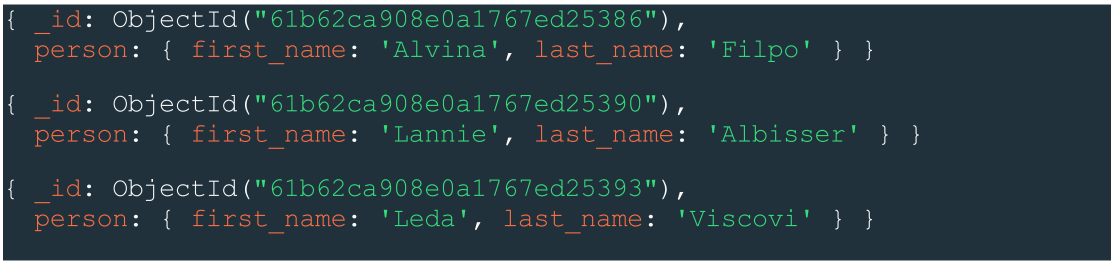


```
{ _id: ObjectId("61b62ca908e0a1767ed25391"),
  person: { first_name: 'Twila', last_name: 'Popplestone' } }

{ _id: ObjectId("61b62ca908e0a1767ed2539b"),
  person: { first_name: 'Aron', last_name: 'Ciccerale' } }

{ _id: ObjectId("61b62ca908e0a1767ed253a8"),
  person: { first_name: 'Katerine', last_name: 'Farfalameev' } }
```

- Find people that have done both doses in a Vaccination Hub.

```
db.certificates.find({"$and": [
 {"vaccines" : {"$size": 2}},
 {"vaccines.0.covid_center.type": {"$nin": ["Pharmacy", "Hospital"]}},
 {"vaccines.1.covid_center.type": {"$nin": ["Pharmacy", "Hospital"]}}
 ]},
 {"person.first_name": 1, "person.last_name" : 1}
)
```

Frame of the output:

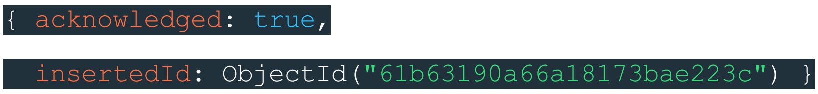


```
{ _id: ObjectId("61b62ca908e0a1767ed25386"),
  person: { first_name: 'Alvina', last_name: 'Filpo' } }

{ _id: ObjectId("61b62ca908e0a1767ed25390"),
  person: { first_name: 'Lannie', last_name: 'Albisser' } }

{ _id: ObjectId("61b62ca908e0a1767ed25393"),
  person: { first_name: 'Leda', last_name: 'Viscovi' } }
```

Commands:

- Create a new vaccination certificate.

```
db.certificates.insertOne({
  "person":{
    "_id": ObjectId(),
    "first_name": "Pippo",
    "last_name": "Sowlo",
    "email": "pipposowlo@gmail.com",
    "phone_number": "930-462-2255",
    "CF": "PPSSSS77W22G101D",
    "is_doctor": false,
    "emergency_contact": {
      "first_name": "Selinda",
      "last_name": "Bursnall",
      "phone_number": "431-808-8091",
      "email": "sbursnall3n@jalbum.net"
    }},
    "vaccines": [{
     "_id": ObjectId(),
     "brand": "Johnson&Johnson",
     "date": ISODate("2021-12-05"),
     "covid_center": {
           "name": "Conn, Gottlieb and Emmerich",
           "address": "08 Florence Crossing",
           "center_type": "Pharmacy",
          "location": {
                     "type": "Point",
                     "coordinates": [
                     51.1134675,
                     28.0490173 ]
          }
      },
    "health_worker": {
           "first_name": "Laural",
           "last_name": "Nash",
           "phone_number": "325-825-5666",
           "email": "lnashaf@shinystat.com"
    }
   }]
})
```



```
{ acknowledged: true,
  insertedId: ObjectId("61b63190a66a18173bae223c") }
```

- Add a new dose to a certificate.

```
db.certificates.updateOne(
          {"person.CF": "RGCFKS39S94H798B"},
          {"$push":
                    { "vaccines": {
                     "_id": ObjectId() ,
                     "brand": "Pfizer",
                     "date":  ISODate("2021-12-07"),
                     "covid_center": {
                                        "name": "D'Amore-Veum",
                                        "address": "463 Shasta Lane",
                                        "center_type": "Vaccination Hub",
                              "location": {
                                        "type": "Point",
                                        "coordinates": [
                                        35.2897617,
                                        137.1605839 ]
                              }
                     },
                    "health_worker": {
                              "first_name": "Laural",
                              "last_name": "Nash",
                              "phone_number": "325-825-5666",
                        "email": "lnashaf@shinystat.com"
                    }
            }
}})
```

```
{ acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

- Add a new test to a certificate.

```
db.certificates.updateOne(
        { "person.CF": "PDNTED62M21N300R" },
        { "$push": {
                "tests": {
                "_id": ObjectId(),
                "outcome": false,
                "date":  ISODate("2021-12-06"),
                "covid_center": {
                                "name": "D'Amore-Veum",
                                "address": "463 Shasta Lane",
                                "center_type": "Vaccination Hub",
                        "location": {
                                "type": "Point",
                                "coordinates": [
                                35.2897617,
                                137.1605839 ]
                        }
                },
               "health_worker": {
                        "first_name": "Laural",
                        "last_name": "Nash",
                        "phone_number": "325-825-5666",
                        "email": "lnashaf@shinystat.com"
               }}
        }
})
```

```
{ acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

- Update an emergency contact.

```
db.certificates.updateOne(
        {"person.CF": "TBWOEL51O31J468Z"},
        {"$set": {"person.emergency_contact.phone_number": "333-333-333"}}
)
```

```
{ acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

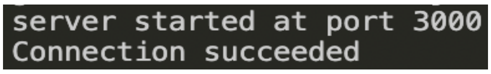
UI description

In order to develop the User Interface we used different tools:

- Express[4], a web application framework for Node.js, for the backend;
- Mongoose[5] as a driver for MongoDB;
- Handlebars[6] as a template engine for the front end.

User Guide

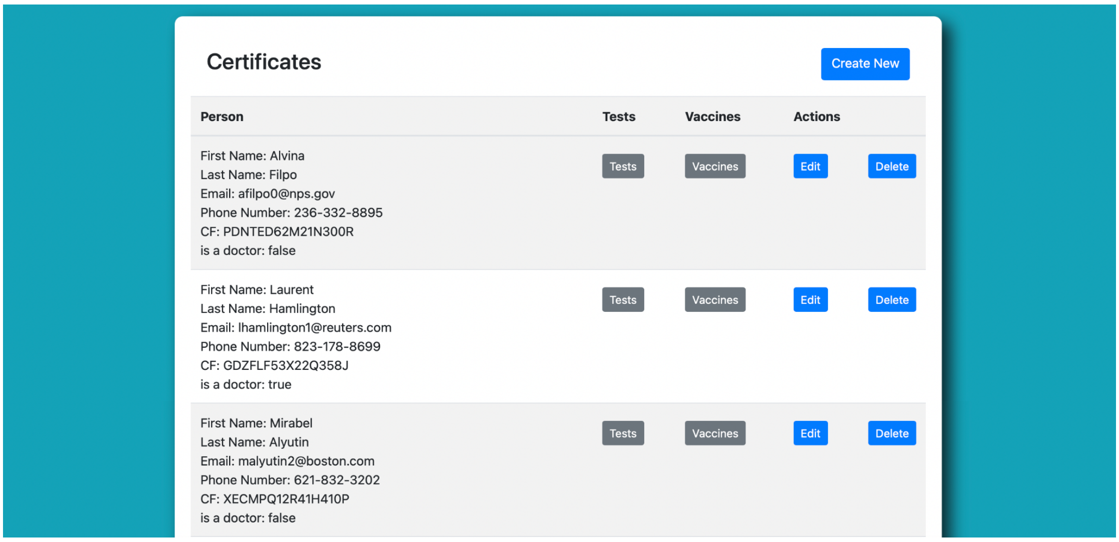
In order to run the UI you need to run the index.js file, located inside the UI folder.
This will start the server at port 3000.



```
server started at port 3000
Connection succeeded
```

Now you can visit http://localhost:3000 to see the database.

In the main page of the User Interface you can see the list of all the COVID certificates in the database, with a detail of the people's information.

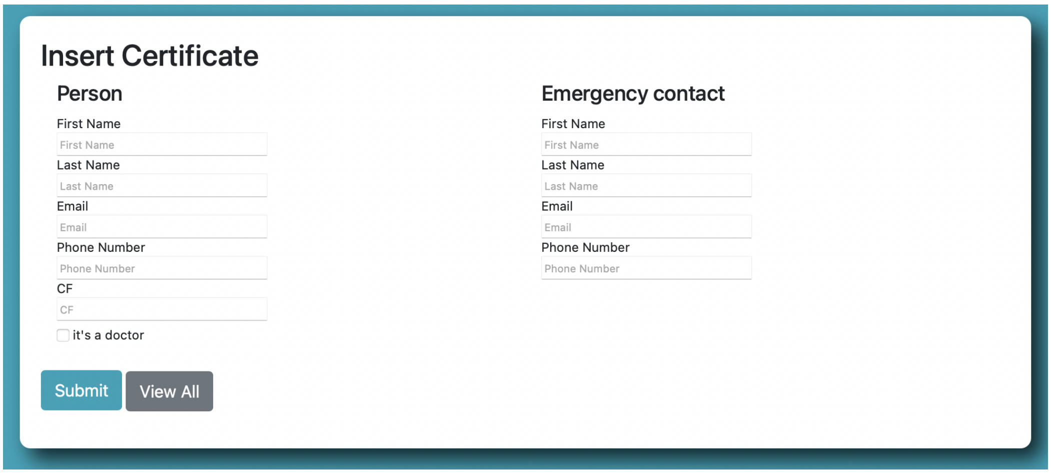


Certificates

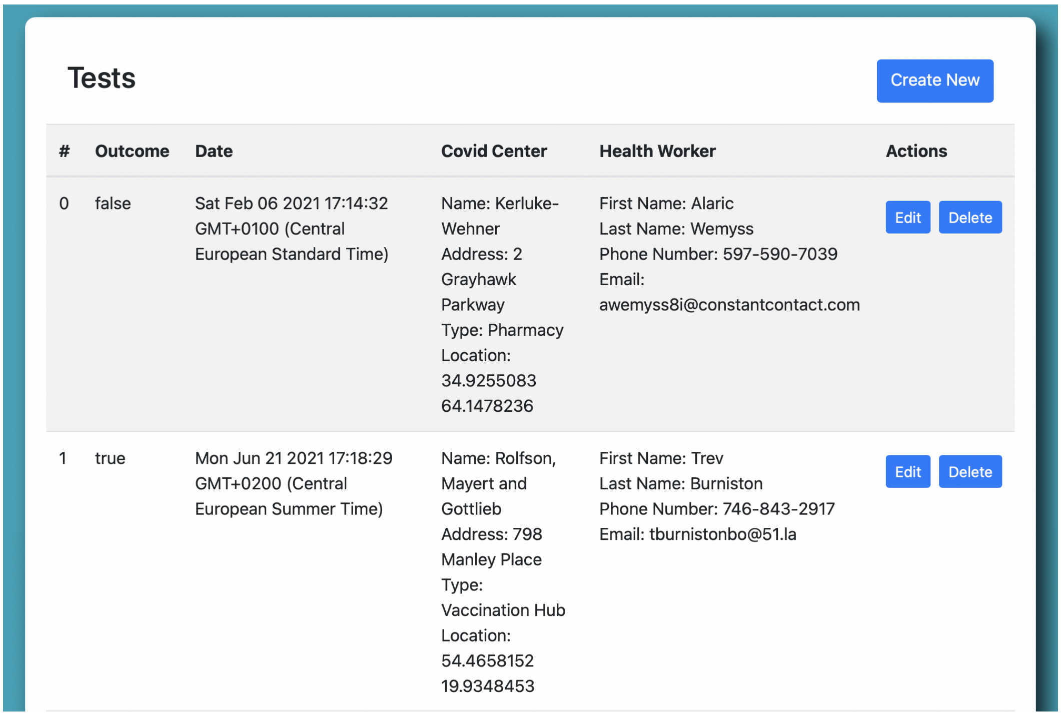
Person	Tests	Vaccines	Actions	
First Name: Alvina Last Name: Filpo Email: afilpo0@nps.gov Phone Number: 236-332-8895 CF: PDNTED62M21N300R is a doctor: false	Tests	Vaccines	Edit	Delete
First Name: Laurent Last Name: Hamlington Email: lhamlington1@reuters.com Phone Number: 823-178-8699 CF: GDZFLF53X22Q358J is a doctor: true	Tests	Vaccines	Edit	Delete
First Name: Mirabel Last Name: Alyutin Email: malyutin2@boston.com Phone Number: 621-832-3202 CF: XECMPQ12R41H410P is a doctor: false	Tests	Vaccines	Edit	Delete

From here, you have different options:

- You can add a new certificate clicking on "Create New" on the button on the top right of the screen;



- You can edit or delete a certificate using the action buttons;
- You can view someone's tests' or vaccines' lists.



#	Outcome	Date	Covid Center	Health Worker	Actions
0	false	Sat Feb 06 2021 17:14:32 GMT+0100 (Central European Standard Time)	Name: Kerluke-Wehner Address: 2 Grayhawk Parkway Type: Pharmacy Location: 34.9255083 64.1478236	First Name: Alaric Last Name: Wemyss Phone Number: 597-590-7039 Email: awemyss8i@constantcontact.com	Edit Delete
1	true	Mon Jun 21 2021 17:18:29 GMT+0200 (Central European Summer Time)	Name: Rolfson, Mayert and Gottlieb Address: 798 Manley Place Type: Vaccination Hub Location: 54.4658152 19.9348453	First Name: Trev Last Name: Burniston Phone Number: 746-843-2917 Email: tburnistonbo@51.la	Edit Delete

The tests and vaccines web pages also have the options to add a new one to the list, edit or delete an existing one.

References and sources

[1] Mockaroo https://www.mockaroo.com/
[2] Github repo https://github.com/fulcus/mongodb-covid19-db
[3] Mongodb docs https://docs.mongodb.com/
[4] Express https://expressjs.com/
[5] Mongoose https://mongoosejs.com

[6] Handlebars https://handlebarsjs.com