

# Crownpass: A Covid-19 Tracking System

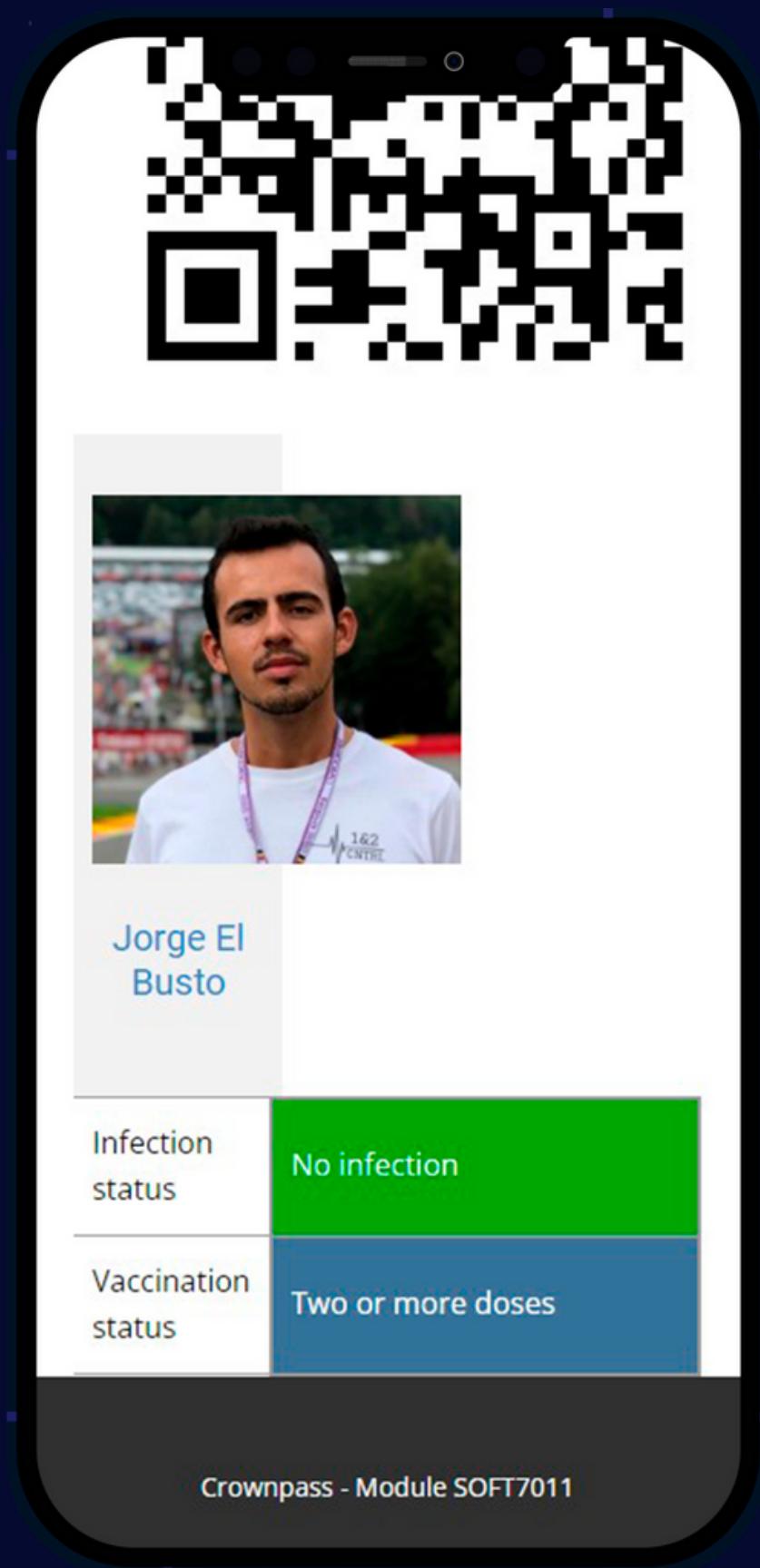
*Student 19179422*

*MSc in Advanced Computer Science  
Oxford Brookes University*

Module SOFT7011

# Design

- Designed with Nicepage.
- Compatible with mobile devices to fulfil platform compatibility requirements.
- Functionality implemented.



The image shows four computer monitors arranged in a 2x2 grid, each displaying a different screen of the Crownpass application.

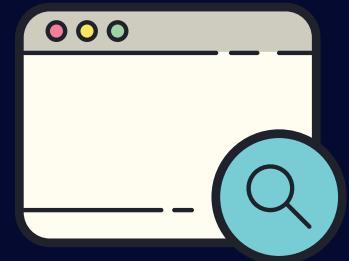
- Top Left Monitor:** Displays the "User tracing" section of the application. It features a search bar for "From" and "To" dates, followed by a table showing user movement history. The table includes columns for "user", "DATE", "VENUE", and "CHECK". The data shows four entries: 1. 2022/04/06, 10:26:33, The Randolph Hotel, In; 2. 2022/04/07, 10:26:56, The Randolph Hotel, Out; 3. 2022/04/19, 20:46:44, The Oxford Retreat, In; 4. 2022/04/19, 20:54:24, The Oxford Retreat, Out. The footer of the screen reads "Crownpass - Module SOFT7011".
- Top Right Monitor:** Displays the "Account details" section. It includes fields for "Crownpass ID" (set to 1), "Name" (Jorge El Busto), "Phone number" (+34 676 892 280), "E-Mail" (19179422@brookes.ac.uk), "Home address" (Flat H2 Postgraduate Centre, John Game Way, Oxford OX3 0FP), "Birth date" (1999/09/25), and "Gender" (Male). A "SAVE" button and a "Reset movement tracing" button are at the bottom.
- Bottom Left Monitor:** Displays a QR code and a small profile picture of Jorge El Busto. Below the image, his name is displayed. Under "Infection status", it says "No infection" (in green). Under "Vaccination status", it says "Two or more doses" (in blue).
- Bottom Right Monitor:** Displays the "Book your COVID-19 vaccination" section. It includes a "Brookes Medical Centre" dropdown, a "Date of vaccination" field set to "12/05/2022", a "Vaccine type" dropdown set to "Astrazeneca", and a "BOOK" button. The footer of the screen reads "Crownpass - Module SOFT7011".

# API and services



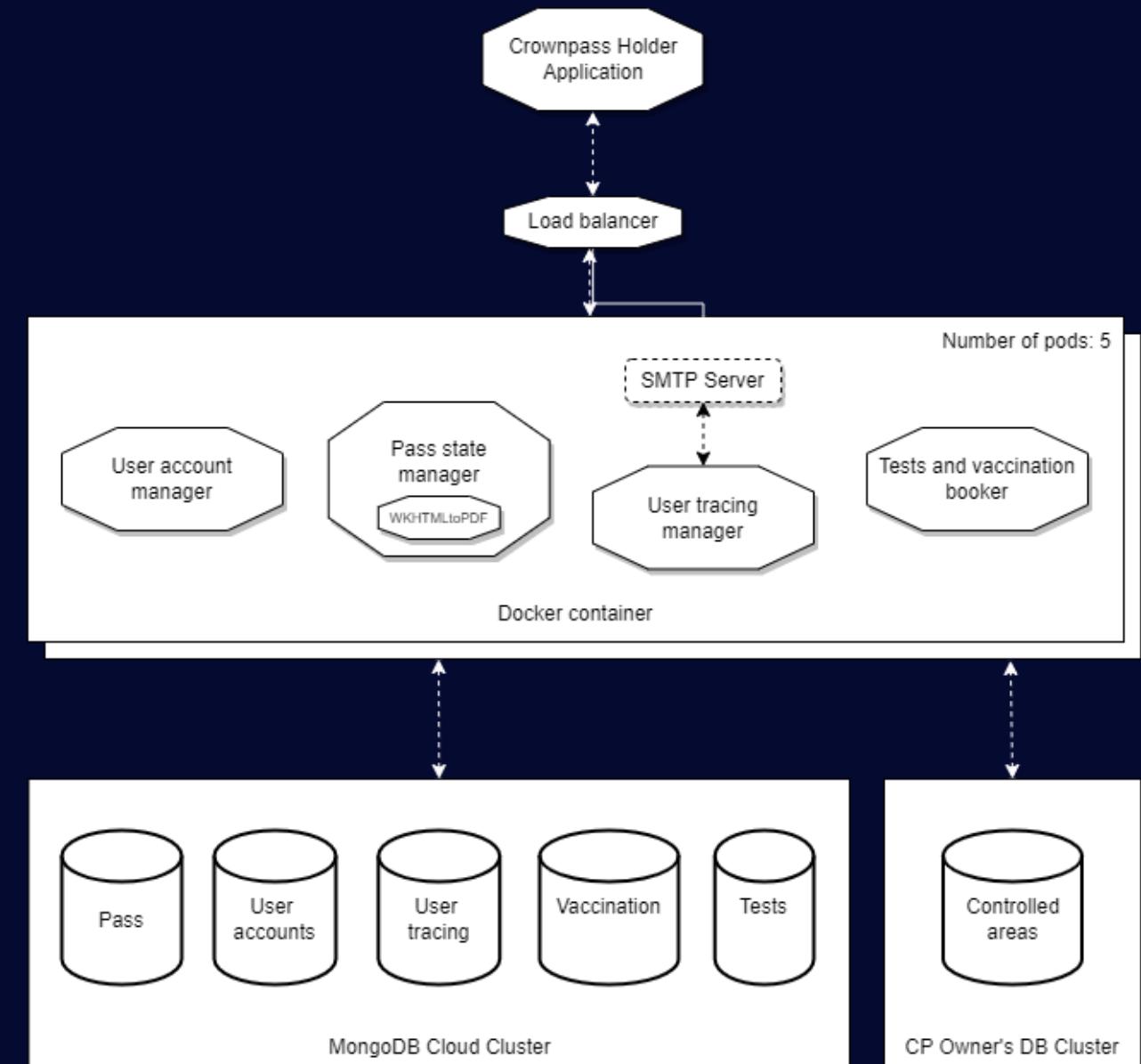
## Cloud storage

- MongoDB Atlas.
- Four Docker images wrapped in a container.
- Deployed to GKE.

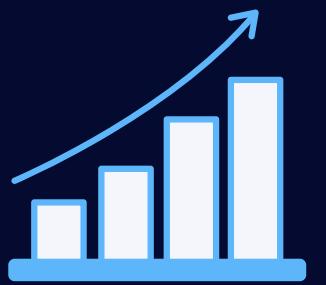


## Microservices

- Four applications
  - User account manager
  - Pass state manager
  - User tracing manager
  - Tests and vaccination booker



# Quality requirements

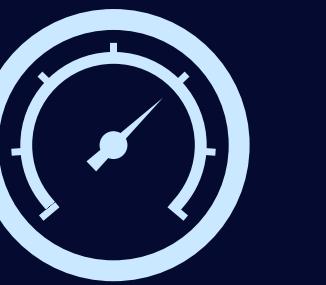


## Scalability and reliability

NoSQL databases - horizontally scalable.

Load balancing with five pods in GKE.

If one fails, others will continue operating.



## Performance

Unit tests ran to execute requests and measure the time.

Dynamically generated images and QRs to optimize file storage.



## Security and privacy

Unit tests ran to verify that only the Crownpass holder can access or modify his pass data.



# Implementation

## Programming

- Language: Python
- IDE: PyCharm
- Libraries: Flask, Flask-Mail, PyMongo, PDFKit, QRCode

## Deployment

- MongoDB
- Docker
- Google Kubernetes Engine

Cloud Platform SOFT7011-Jorge

Search Products, resources, docs (/)

Deployment details

CPU Memory Disk

Cluster: cp-test-cluster Namespace: default Labels: No labels set Logs: Container logs, Audit logs Replicas: 5 updated, 5 ready, 5 available, 0 unavailable Pod specification: Revision 1, containers: user-account-app, tracing-app, pass-app, booker-app

Active revisions

Revision	Name	Status	Summary	Created on	Pods running/Pods total
1	cp-holder-6f8657c6df	OK	user-account-app: gcr.io/soft7011-jorge/user_account_app:version1.1, tracing-app: gcr.io/soft7011-jorge/tracing_app:version1.1, pass-app: gcr.io/soft7011-jorge/pass_app:version1.1, booker-app: gcr.io/soft7011-jorge/booker_app:version1.1	2 May 2022, 21:38:24	5/5

Managed pods

Revision	Name	Status	Restarts	Created on
1	cp-holder-6f8657c6df-4z4nw	Running	0	2 May 2022, 21:38:24
1	cp-holder-6f8657c6df-jvswd	Running	0	2 May 2022, 21:38:25
1	cp-holder-6f8657c6df-vj8cp	Running	0	2 May 2022, 21:38:25
1	cp-holder-6f8657c6df-q5znc	Running	0	2 May 2022, 21:38:25
1	cp-holder-6f8657c6df-8x47q	Running	0	2 May 2022, 21:38:25

Exposing services

Name	Type	Endpoints
cp-holder	Load balancer	34.142.3.248:6000
cp-holder-k8s	Load balancer	35.234.152.237:5000

# Code

```
require File.expand_path("../spec_helper", __FILE__)
# Prevent database truncation if the test fails
abort("The Rails environment is running in production mode!") if Rails.env.production?
require 'spec_helper'
require 'rspec/rails'

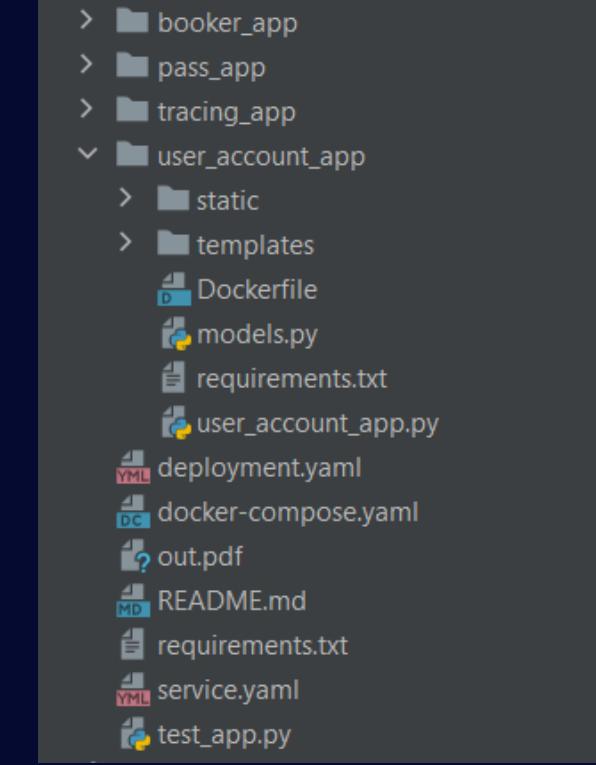
# Load support files
require 'capybara/rspec'
require 'capybara/rails'

# Load fixtures from the engine
if defined?(Rails::Engine)
  require 'factory_girl_rails'
end

# Add additional requires below this line
# Load factories
# require 'factory_girl'

# Requires supporting ruby files
# spec/support/ and its subdirectories
# run as spec files by default
# in _spec.rb will both be
# run twice. It is recommended
# to put actions in the
# action on the command line
# No results found for 'mongoid'
```

# Code



# Code and test report accessible in GitHub.

# Testing



## Unit testing

- Software and tools: PyTest
- Timestamps recorded before and after requests to measure elapsed time.
- Privacy requirements return 200 or 500.
- Features tested and returning expected responses.

## Integration testing

- CI/CD pipeline with GitHub Actions.
- All tests ran after each commit.
- Secrets encrypted.

## Bugs and errors

- Linting the code with Flake8

```
1 ► Run # stop the build if there are Python syntax errors or undefined names
10 0
11 ./test_app.py:1:1: F401 'sys' imported but unused
12 ./test_app.py:58:6: E231 missing whitespace after ','
13 ./test_app.py:59:22: E261 at least two spaces before inline comment
14 ./test_app.py:59:23: E262 inline comment should start with '#'
15 ./test_app.py:64:30: E261 at least two spaces before inline comment
16 ./test_app.py:64:31: E262 inline comment should start with '#'
17 ./test_app.py:76:46: E231 missing whitespace after ','
18 ./test_app.py:93:55: E231 missing whitespace after ','
19 ./test_app.py:131:41: E261 at least two spaces before inline comment
20 ./test_app.py:131:42: E262 inline comment should start with '#'
21 ./test_app.py:131:128: E501 line too long (156 > 127 characters)
22 ./test_app.py:153:46: E261 at least two spaces before inline comment
23 ./test_app.py:153:47: E262 inline comment should start with '#'
24 ./test_app.py:153:128: E501 line too long (154 > 127 characters)
25 ./test_app.py:155:38: E261 at least two spaces before inline comment
26 ./test_app.py:155:39: E262 inline comment should start with '#'
27 ./test_app.py:155:128: E501 line too long (130 > 127 characters)
28 ./test_app.py:165:6: E231 missing whitespace after ','
29 ./test_app.py:176:6: E231 missing whitespace after ','
30 ./test_app.py:233:72: W292 no newline at end of file
31 ./user_account_app/user_account_app.py:1:1: F401 'sys' imported but unused
32 ./user_account_app/user_account_app.py:5:1: E722 do not use bare 'except'
33 ./user_account_app/user_account_app.py:66:9: F841 local variable 'x' is assigned to but never used
34 ./booker_app/booker_app.py:1:1: F401 'sys' imported but unused
35 ./pass_app/models.py:63:1: W391 blank line at end of file
36 ./pass_app/pass_app.py:1:1: F401 'sys' imported but unused
37 ./pass_app/pass_app.py:5:1: E722 do not use bare 'except'
38 ./pass_app/pass_app.py:25:1: E722 do not use bare 'except'
```

```
def test_crownpassReg(clientUser):
    """
    QR-PF-01: Registration of Crownpass.
    The response time for each interactive operation in the process of registration
    for a Crownpass should be no more than 20 seconds.
    """
    start = time.time()
    response = clientUser.post("/register", data={
        "name": "Sample testing user",
        "infStatus": "No infection",
        "vacStatus": "Two or more doses",
        "phone": "000011112222",
        "email": "sampleuser@brookes.ac.uk",
        "home": "Home address",
        "birthdate": "1999/01/01",
        "gender": "Male",
        "photo": open('user_account_app/static/images/testuser.jpg', 'rb')
    }, follow_redirects=True)
    end = time.time() # The timestamp is recorded after processing the POST request.
    myclient = pymongo.MongoClient(os.environ.get('MONGO_CLIENT'))
    mydb = myclient["users"]
    mycol = mydb["users"]
    myquery = {"name": "Sample testing user"}
    mycol.delete_one(myquery) # The registered user is deleted after having created it.
    assert ((end - start) < 20 and response.status_code == 200)
```

✓  Lint with flake8

```
1 ► Run # stop the build if there are Python syntax errors or undefined names
10 0
11 ./user_account_app/user_account_app.py:4:1: E722 do not use bare 'except'
12 ./pass_app/pass_app.py:4:1: E722 do not use bare 'except'
13 ./pass_app/pass_app.py:24:1: E722 do not use bare 'except'
14 ./tracing_app/tracing_app.py:7:1: E722 do not use bare 'except'
15 4      E722 do not use bare 'except'
16 4
```

▼  Test with pytest

```
1 ► Run pytest
15 ===== test session starts =====
16 platform linux -- Python 3.9.12, pytest-7.1.2, pluggy-1.0.0
17 rootdir: /home/runner/work/19179422_CP_Holder/19179422_CP_Holder
18 collected 11 items
19
20 test_app.py ..... [100%]
21
22 ===== 11 passed in 27.01s =====
```

# Demo

Thank you