

# **MAPD 712**

## **Final Project Presentation**

**Patient Mobile App with React-native**

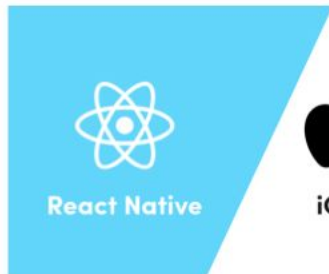
**Group 2**

**Quoc Phong Ngo – Student Id: 301148406**

**Feiliang Zhou – Student Id: 301216989**

# Description

- Design a React-native application for the health care providers.
- API Request methods: POST, GET, PUT, DELETE, calling REST API Service to Nodejs server.
- Built on iOS and Android platform.
- List of technology used:  
React-native, JSON, XML , Mongo DB, Postman...



iOS



Android



mongoDB®



# Mongo DB structures

+ Create Database						
Q NAMESPACES						
▼ patients-db						
patient-records						
patients						
tasks						
users						

patients-db						
DATABASE SIZE: 3.89KB INDEX SIZE: 144KB TOTAL COLLECTIONS: 4						
CREATE COLLECTION						
Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
patient-records	5	902B	181B	1	36KB	36KB
patients	5	889B	178B	1	36KB	36KB
tasks	7	756B	108B	1	36KB	36KB
users	9	1.41KB	160B	1	36KB	36KB

There are 4 collections:

- Users
- Tasks
- Patients
- Patient-records

# List of functions in App

## **User:**

- Register a new account, login, get information of a user.

## **Task:**

- Create a new task for user, delete a task, get all tasks of a user, get a task of a user.

## **Patients:**

- Get all patients, get information of a patient, create a new patient, delete a patient.

## **Patient Records (Clinical Records' patient):**

- Get a clinical records' patient, create a new clinical records' patient, delete a clinical records' patient, update a clinical records' patient.

# Register and Login

Username

Password

Position checkbox

← Register

## Register

User name

Password

Repeat your Password

Choose your position

☒ doctor ☐ nurse ☐ administrator

[Agree with Terms and policy](#)

[REGISTER](#)

Already hava an account?  
[Login](#)

← Login

## Login

Username

Password

[LOGIN](#)

Do not hava an account yet?  
[Register](#)

# Home page

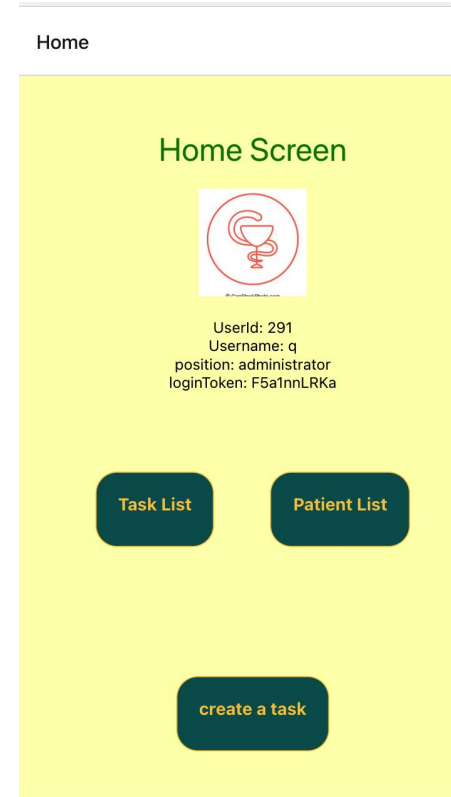
Basic information display

Task List

Patient List

Create a task:

(only display when the user is  
the position of administrator)



# Create a task

[←](#) Create a Task

Task for userId: 291

Task Name:

Task Time:

Priority: ☒ high ☐ low

Task Initial Status:

assigned	<input checked="" type="radio"/>
In Process	<input type="radio"/>
declined	<input type="radio"/>
closed	<input type="radio"/>

SUBMIT

# Task list, task detail and update task

← Your Task List

Task ID	Time	Task Name	Task Status
19eb443	2022-06-06T04:00:00.000Z	plant a flower	assigned
19edb28	2021-12-19T00:00:00.000Z	zzzz	Complted
1ba2d8f	2021-12-19T00:00:00.000Z	help others	inProcess

← About this Task

Task ID: 61ba2d8f1445f838c4710fc6

Task Name: help others

Time: 2021-12-19T00:00:00.000Z

Priority: normal

Task Status: inProcess

DELETE

UPDATE

TASK LIST

← Update this Task

Task ID: 61ba2d8f1445f838c4710fc6

Task Name:

Time:

Priority:

Task Status:

UPDATE

TASKLIST



# I. Patient

## I.I Add a new patient (POST): Host\_name:port/patients

POST http://127.0.0.1:5000/patients

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "firstName": "Duc",
3   ... "lastName": "Nguyen",
4   ... "age": 96,
5   ... "gender": "Male",
6   ... "healthInsuranceNo": "0111188866",
7   ... "phoneNo": "8888889",
8   ... "email": "duc@gmail.com"
9 }
```

Body Cookies Headers (8) Test Results

Status: 201 Created Time: 100 ms Size: 455 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "61b8b3bd09efc217f9fa55cd",
3   "firstName": "Duc",
4   "lastName": "Nguyen",
5   "age": 96,
6   "gender": "Male",
7   "healthInsuranceNo": "0111188866",
8   "phoneNo": "8888889",
9   "email": "duc@gmail.com",
10  "__v": 0
11 }
```

```
> {
  _id: ObjectId("61b8b3bd09efc217f9fa55cd"),
  firstName: "Duc",
  lastName: "Nguyen",
  age: 96,
  gender: "Male",
  healthInsuranceNo: "0111188866",
  phoneNo: "8888889",
  email: "duc@gmail.com",
  __v: 0
}
```

iPhone 12  
iOS 14.4

12:09

< Patient List Add New Patient

First Name:

Last Name:

Gender: ☒ Male ☐ Female

Age:

Health Insurance Number:

Phone Number:

Email:

Submit

RCTBridge required dispatch\_sync to loa...

# I. Patient

## I.II View all patients (GET): Host\_name:port/patients

http://127.0.0.1:5000/patients Save

GET ▼ http://127.0.0.1:5000/patients

Params Authorization Headers (6) Body Pre-request Script Tests Settings

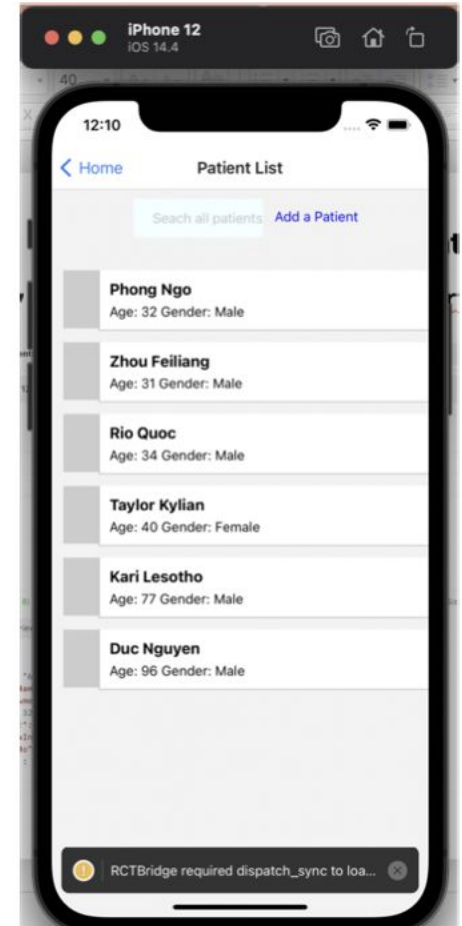
Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 87 ms Size: 1.34 KB

Pretty Raw Preview Visualize JSON ≡

```
1 {  
2     
3   "id": "61796547858e40cdd3eba1b9",  
4   "firstName": "Phong",  
5   "lastName": "Ngo",  
6   "age": 32,  
7   "gender": "Male",  
8   "healthInsuranceNo": "123456789",  
9   "phoneNo": "090111",  
10  "email": "phong@gmail.com"  
11 },  
12 {
```



# I. Patient

## I.III Get a patient (GET): Host\_name:port/patients/:patientId

GET `http://127.0.0.1:5000/patients/618c6394a5f3ac800f8bd9c4`

Params Authorization Headers (6) Body Pre-request Script Tests Settings

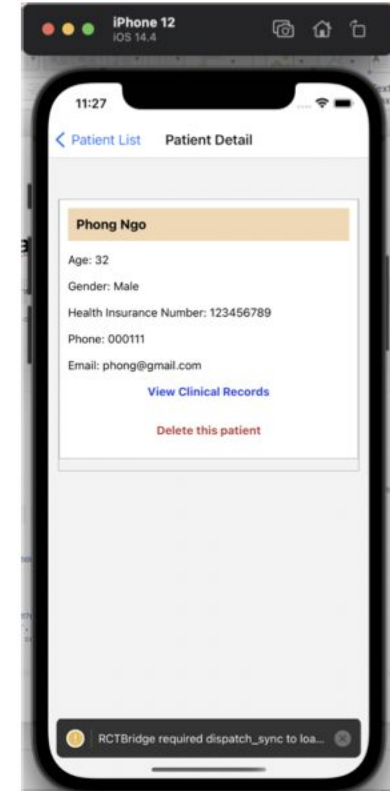
Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 55 ms Size: 453 B

Pretty Raw Preview Visualize JSON

```
1 {  
2     
3   "_id": "618c6394a5f3ac800f8bd9c4",  
4   "firstName": "Taylor",  
5   "lastName": "Kylia",  
6   "age": 40,  
7   "gender": "Female",  
8   "healthInsuranceNo": "87657892",  
9   "phoneNo": "4168370649",  
10  "email": "taylor@gmail.com"  
11 }  
12 }
```



# I. Patient

## I.IV Delete a patient (DELETE): Host\_name:port/patients/:patientId

**DELETE** ▼ <http://127.0.0.1:5000/patients/61b8b3bd09efc217f9fa55cd>

Params Authorization Headers (6) Body Pre-request Script Tests Settings

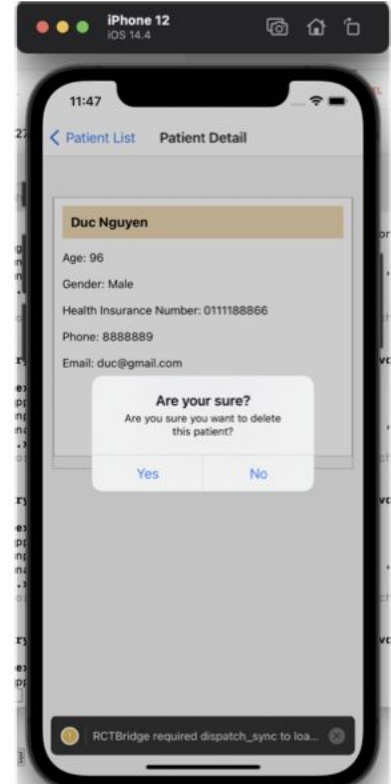
Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results 🌐 Status: 200 OK Time: 87 ms Size: 304 B

Pretty Raw Preview Visualize JSON ▼ 🔧

```
1
2  "msg": "delete patient successfully"
3
```



## II. Clinical Records

### II.I Get a clinical records' patient (GET): Host\_name:port/patients/:patientId/clinical-records

GET <http://127.0.0.1:5000/patients/61796547858e40cdd3eba1b9/clinical-records>

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

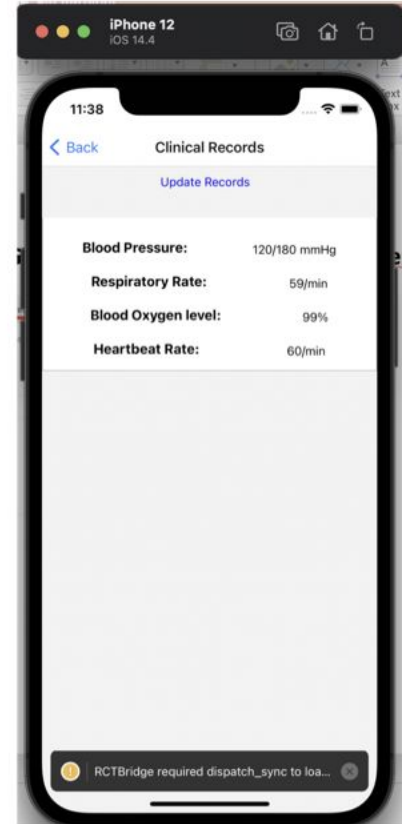
KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 53 ms Size: 451 B

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "_id": "618c22dc89ba3fcd3673b7a2",
4     "patientId": "61796547858e40cdd3eba1b9",
5     "bloodPressure": "120/180 mmHg",
6     "respiratoryRate": "59/min",
7     "bloodOxygenLevel": "99%",
8     "heartBeatRate": "60/min"
9   }
10 }
```



## II. Clinical Records

### II.II Create a clinical records' patient (POST): Host\_name:port/patients/:patientId/clinical-records

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** `http://127.0.0.1:5000/patients/61b8ca2f09efc217f9fa55ce/clinical-records`
- Body Type:** JSON
- Body Content:**

```
1 {
2   "patientId": "61b8ca2f09efc217f9fa55ce",
3   "bloodPressure": "130/180 mmHg",
4   "respiratoryRate": "60/min",
5   "bloodOxygenLevel": "95%",
6   "heartBeatRate": "89/min"
7 }
```
- Response Status:** 201 Created
- Response Time:** 60 ms
- Response Size:** 462 B
- Response Body (Pretty):**

```
1 {
2   "_id": "61b8ccf909efc217f9fa55cf",
3   "patientId": "61b8ca2f09efc217f9fa55ce",
4   "bloodPressure": "130/180 mmHg",
5   "respiratoryRate": "60/min",
6   "bloodOxygenLevel": "95%",
7   "heartBeatRate": "89/min",
8   "__v": 0
9 }
```

## II. Clinical Records API

### II.III Update a clinical records' patient (PUT):

Host\_name:port/patients/:patientId/clinical-records/:clinicalRecordsId

The screenshot shows a REST client interface with a PUT request to the URL `http://127.0.0.1:5000/patients/61b8ca2f09efc217f9fa55ce/clinical-records/61b8ccf909efc217f9fa55cf`. The request body is a JSON object with the following fields: `patientId`, `bloodPressure`, `respiratoryRate`, `bloodOxygenLevel`, and `heartBeatRate`. The response status is 200 OK, and the response body is a JSON object with a `msg` field indicating a successful update.

```
PUT http://127.0.0.1:5000/patients/61b8ca2f09efc217f9fa55ce/clinical-records/61b8ccf909efc217f9fa55cf
```

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "patientId": "61b8ca2f09efc217f9fa55ce",
3   "bloodPressure": "140/180 mmHg",
4   "respiratoryRate": "70/min",
5   "bloodOxygenLevel": "96%",
6   "heartBeatRate": "99/min"
7 }
```

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 107 ms Size: 327 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "update a clinical record by patientId successfully"
3 }
```

