

HO CHI MINH CITY, UNIVERSITY OF TECHNOLOGY  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEER



## **Application Based Internet of Things Report**

*Student:* Trần Việt Hoàng

*ID:* 1852145

HỒ CHÍ MINH CITY

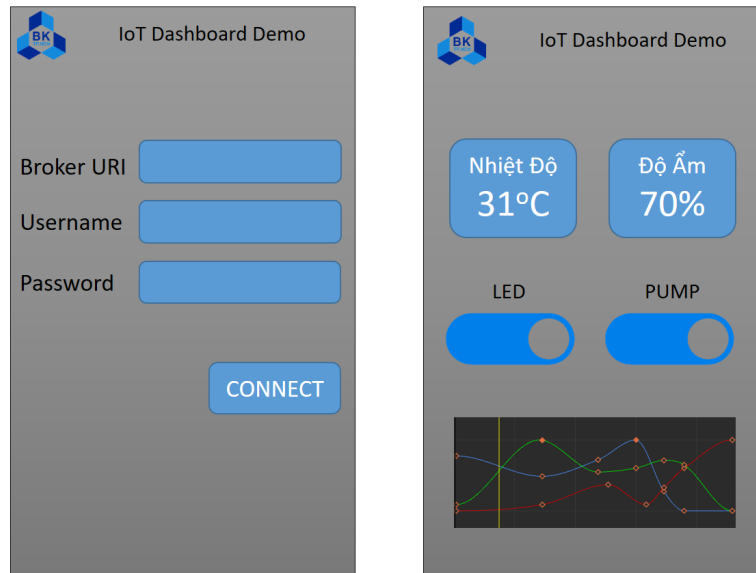


## Content

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Requirements</b>	<b>2</b>
2.1	Screen 1 . . . . .	2
2.2	Screen 2 . . . . .	2
2.3	Advance UI elements . . . . .	3
<b>3</b>	<b>Report</b>	<b>3</b>
3.1	Screen 1 . . . . .	3
3.2	Error screen . . . . .	4
3.3	Screen 2 . . . . .	5
3.4	Github link . . . . .	8

## 1 Introduction

In this second LAB, students are proposed to create a simple dashboard using Unity 3D editor. Basically, the dashboard has 2 screens (GUIs), as depicted following:



Hình 1: *Software mockup GUI*

The source code of the second LAB is also required to publish in your Github. The details are described in the next section of this report.

## 2 Requirements

### 2.1 Screen 1

There are 3 input fields, including the broker of the server, username and password. Using Thingsboard server, the broker should be demo.thingsboard.io, the username is your access token and the password is empty.

The values of these fields can be set by default (in the design phase, by setting the text property of the input component).

When the CONNECT button is pressed, the app will connect to the server. If there is an error, a simple textview can be used to display this error. Otherwise, the second UI is launched.

The testing account for this app **bkiot** and **12345678** for the username and password. The broker URI is **mqttserver.tk**. The default port is **1883**.

### 2.2 Screen 2

The app needs to subscribe to the following topic

`/bkiot/STUDENT_ID/status`

in order to receive the current values of sensors (e.g. temperature and humidity) and update these values on textviews. Students can change the information according to their use cases. However, at least 2 different information of the sensors are required.

Two toggle buttons are required to controll two different devices (e.g. a simple LED or a PUMP). When the button is clicked, the data is published to

`/bkiot/STUDENT_ID/led`

for the LED and

`/bkiot/STUDENT_ID/pump`

for the PUMP.

The data for each button is a json string, described as follow:

- {"device":"LED","status":"ON"}
- {"device":"PUMP","status":"OFF"}

## 2.3 Advance UI elements

The end of the second is an example of an advance eleement in Unity3D. It can be a graph, a gauge or a map. This part is the extra point in this lab.

## 3 Report

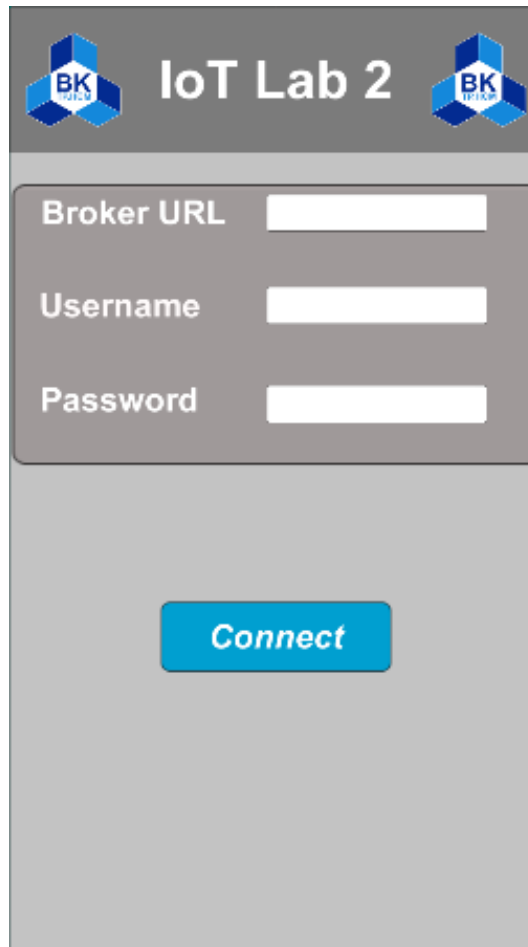
### Software Version:

- Unity: 2020.3.31f
- Python: 3.8.5

The Python file is used to publicize the information about temperature and humidity into the mqttserver. The code is quite similar to the lab 1 code because I reused most of it, just changing some minor details.

### 3.1 Screen 1

The first screen is proposed to be initially displayed when opening the app. The display resolution I chose is 1080x1920, which is suitable for most smartphones used nowadays.



The image shows a login interface for 'IoT Lab 2'. At the top, there is a header bar with a blue and white logo on the left, the text 'IoT Lab 2' in the center, and another logo on the right. Below the header, there is a dark gray rounded rectangle containing three input fields: 'Broker URL', 'Username', and 'Password'. Each field has a white text input area. Below this rectangle, on a light gray background, is a blue button with the word 'Connect' in white text.

Figure 1: Log in screen

### 3.2 Error screen

After filling in the information, it will be checked. if the provided information is wrong the Error screen will appear.

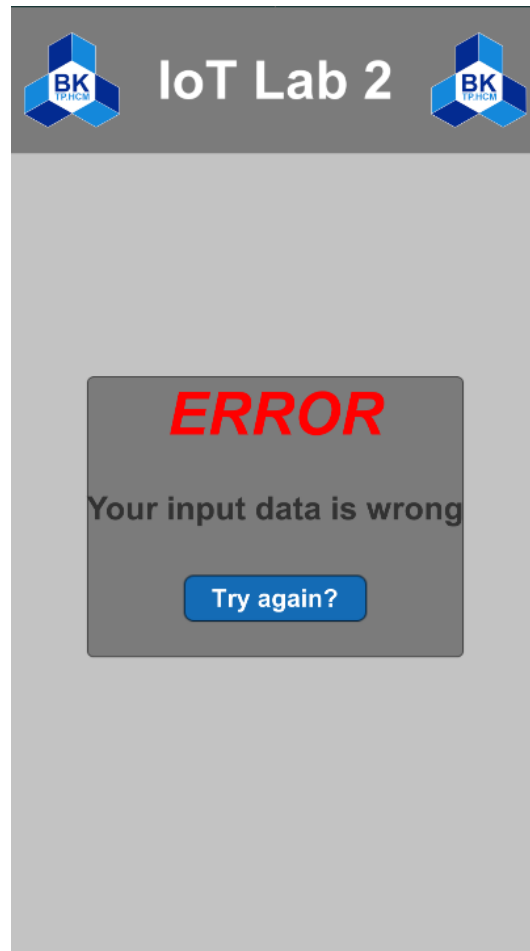


Figure 2: Error screen

User can click on the Try again? button in order to get back to the Log in screen.

### 3.3 Screen 2

In case of providing correct information, the second screen will appear. In this screen, if you have already run the python code provided, the information about temperature and humidity will be shown in the temp box and humid box respectively, there's data will be updated every 10 seconds.

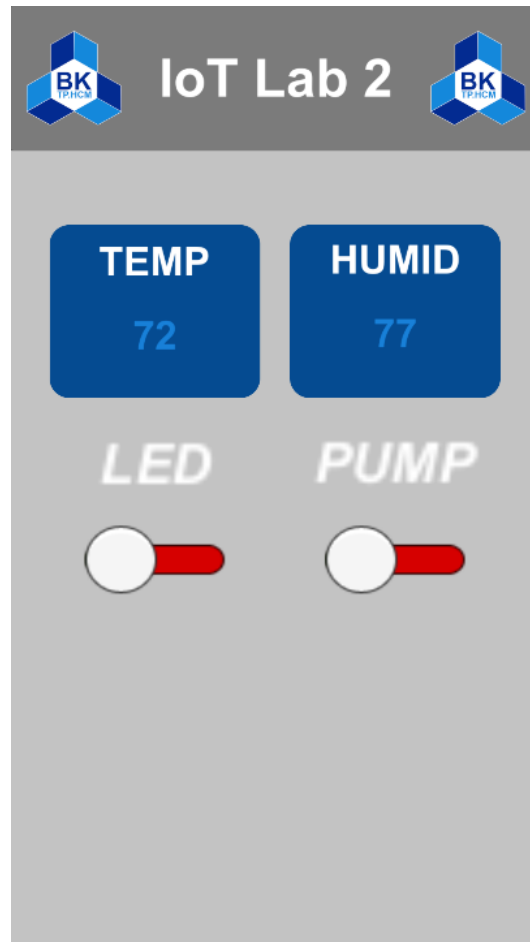


Figure 3: Second screen

If you haven't run the python code, both of them will display 0 all the time.

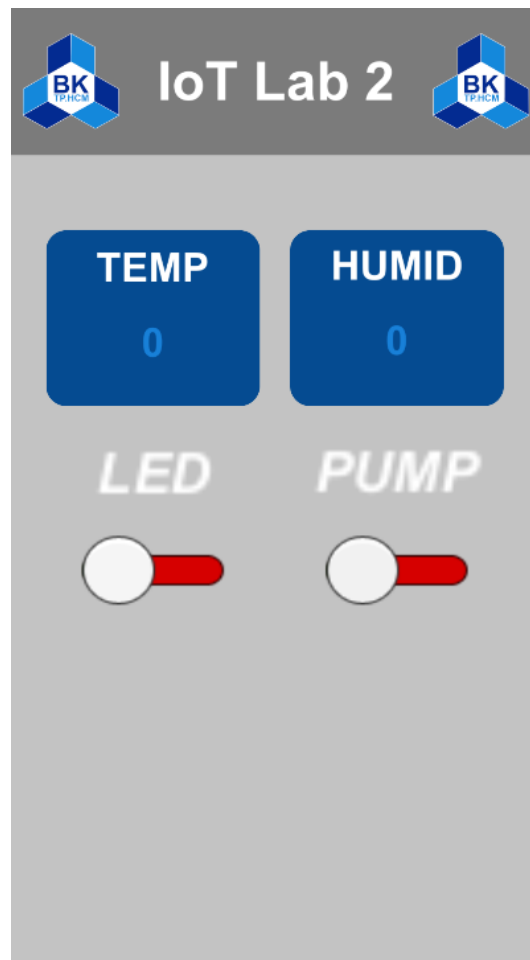


Figure 4: Second screen when not receiving information

I also use 2 sliders to publicize the status of Led and Pump. In default mode, the handle will be on the left-hand side and the slide area will have red color when pulling the handle to the right the slide area will have green color. It will have a message sending to the mqttserver every time it has any changes in the status of the slider.



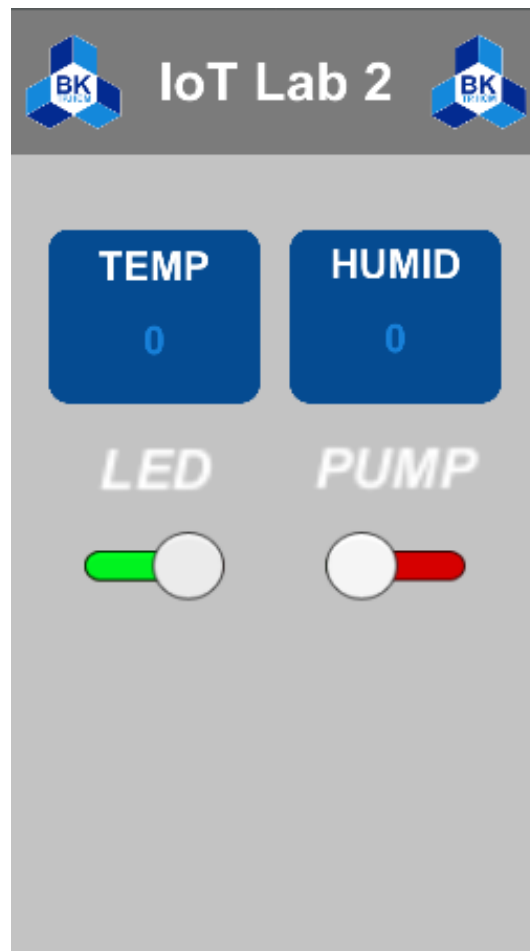


Figure 5: Led toggle in on mode

```
/bkiot/1852145/pump 0 {"device":"PUMP","status":"OFF"}
```

```
/bkiot/1852145/pump 0 {"device":"PUMP","status":"ON"}
```

```
/bkiot/1852145/led 0 {"device":"LED","status":"OFF"}
```

```
/bkiot/1852145/led 0 {"device":"LED","status":"ON"}
```

Figure 6: json string send to mqttserver

### 3.4 Github link

[https://github.com/hoangviettran/IoT\\_Lab\\_2.git](https://github.com/hoangviettran/IoT_Lab_2.git)