

# Room Capacity Limiter

## สมาชิก

Thanaphum Thepwan 6330223021  
Noppakorn Jiravaranun 6330258021  
Nopparuj Poonsubanan 6330261921  
Nanthicha Makjinda 6330282021

## เสนอ

อ.ดร.พิชญะ สิทธิอมร

รายงานนี้เป็นส่วนหนึ่งวิชา 2110366 Embedded System  
Laboratory  
ภาคการศึกษาปลาย ปีการศึกษา 2564

# สารบัญ

<b>1</b>	<b>Description</b>	<b>3</b>
<b>2</b>	<b>Device</b>	<b>3</b>
<b>3</b>	<b>Usage</b>	<b>3</b>
<b>4</b>	<b>Website</b>	<b>3</b>
4.1	Website Usage . . . . .	3
<b>5</b>	<b>Source Code</b>	<b>3</b>
<b>6</b>	<b>Roles and Responsibility</b>	<b>4</b>
6.1	Team Management . . . . .	4
6.1.1	Responsibility . . . . .	4
6.1.2	Plan . . . . .	4
6.2	System Architecture . . . . .	5
6.2.1	Responsibility . . . . .	5
6.2.2	Hardware Design . . . . .	5
6.2.3	Database Design . . . . .	5
6.2.4	API Design . . . . .	5
6.3	Embedded System Development . . . . .	6
6.3.1	Responsibility . . . . .	6
6.3.2	Hardware Connectivity . . . . .	6
6.3.3	Function of Hardware . . . . .	6
6.3.4	ESP8266 . . . . .	7
6.3.5	STM32 . . . . .	7
6.3.6	Operation of the system . . . . .	7
6.4	UI/UX Designer and Development . . . . .	8
6.4.1	Responsibility . . . . .	8
6.4.2	UI Design . . . . .	8
6.4.3	Web Development . . . . .	8

## 1 Description

อุปกรณ์นี้เป็นอุปกรณ์สำหรับจำกัดจำนวนนิสิต หรือคนภายในห้อง ซึ่งต้องมีการแตะบัตรนิสิต หรือบัตร RFID ก่อนเข้าสถานที่ต่างๆ โดยจะบอกว่าตอนนี้มีคนอยู่ในห้องนั้นๆ ก็คนที่เข้ากับการระบาดของไวรัส COVID-19 ซึ่งในปัจจุบันมีการรณรงค์การ Social Distancing และลดการสัมผัสซึ่งอุปกรณ์ที่ออกแบบสามารถตอบสนองการใช้งานในส่วนนี้ได้เป็นอย่างดี โดยสามารถลดการสัมผัสระหว่างบุคคลและสามารถทำให้การตรวจเช็คบัตรหรือคนที่อยู่ในห้องทำได้สะดวกขึ้น และยังประยุกต์หลักการของ IoT เข้ามาเพื่อช่วยให้สามารถแสดงผลการทำงานของอุปกรณ์ได้แบบ Real time

## 2 Device

อุปกรณ์ที่ใช้ประกอบด้วย

1. STM32 NUCLEO F411-RE
2. ESP8266
3. RFID-RC522
4. LCD Screen (LCD1602)

## 3 Usage

1. usage...

## 4 Website

<https://embedded-project.vercel.app>

### 4.1 Website Usage

1. usage...

## 5 Source Code

Source Code และเอกสารเกี่ยวกับ Project สามารถเข้าถึงได้ที่ <https://github.com/noppakorn/embedded-project>

## 6 Roles and Responsibility

### 6.1 Team Management

By: Thanaphum Thepwan

#### 6.1.1 Responsibility

- เสนอหัวข้อ Project ในการทำงาน
- ประสานงานกับทุกฝ่ายในกลุ่มในการกำหนดวันในการมาทำงานร่วมกัน การแบ่งงานในแต่ละส่วน
- จัดทำส่วนของการ Presentation
- ดูแลภาพรวมของ Project ให้เป็นไปตามเป้าหมาย

จากโจทย์ในการทำงานได้มาว่า Contactless Society ซึ่งจากการเป็น Team Management ซึ่งหัวข้อที่ได้เสนอไปก็คือ ทำระบบจำกัดจำนวนคนในห้องใช้ระบบ Checkin โดยใช้บัตรนิสิต หรือบัตร RFID ซึ่งเมื่อตกลงกันในกลุ่มแล้ว ได้กำหนดช่องทางการสื่อสารหลักเป็น Discord และได้แบ่งงานให้แต่ละคนดังนี้

1. Nanthicha: System Architecture  
(จัดหา sensor ต่างๆที่ต้องใช้ ออกแบบระบบว่าแต่ละอุปกรณ์เชื่อมต่อกันได้อย่างไร)
2. Noppakorn: Embedded System Development  
(การโปรแกรมลง Board)
3. Nopparuj: รับผิดชอบในส่วนของ UI/UX Designer and Development  
(ออกแบบพัฒนาเว็บไซต์)

#### 6.1.2 Plan

งาน	วันที่ดำเนินงาน
กำหนดหัวข้อโปรเจกต์จากโจทย์ที่ได้รับ	12/04/2022
เริ่มจัดทำ frontend	13/04/2022
ทำให้ส่วนของ frontend สามารถ fetch ข้อมูลจาก firebase ได้	18/04/2022
นำเสนอหัวข้อโปรเจกต์ให้อาจารย์อนุมัติ และได้กำหนดอุปกรณ์ที่ต้องใช้	20/04/2022
ทำงานร่วมกันครั้งแรก รับ ESP8266, RFID-RC522 และโปรแกรมทำให้ ESP8266 สามารถเชื่อม Internet ได้	27/04/2022
เชื่อมต่อกับเซนเซอร์ RFID-RC522 ให้สามารถอ่านบัตรได้	03/05/2022 ถึง 20/05/2022
ทำให้ข้อมูลที่อ่านได้จาก RFID-RC522 สามารถส่งขึ้นไป Cloud ได้	23/05/2022
ทำให้ Cloud Response data กลับมาที่ ESP8266 เพื่อที่จะนำข้อมูลไปแสดงบนหน้าจอ LCD	25/05/2022
เชื่อมต่อจอ LCD Screen เพื่อให้สามารถแสดงข้อมูลได้จาก Cloud	27/05/2022
จัดทำ Presentation และ Report	31/05/2022

## 6.2 System Architecture

By: Nanthicha Makjinda

### 6.2.1 Responsibility

1. Hardware Design: กำหนด sensor ที่ต้องใช้ในระบบทั้งหมด
2. Database Design: ออกแบบโครงสร้าง Database
3. API Design: ออกแบบโครงสร้าง API เพื่อการติดต่อของ ESP8266

### 6.2.2 Hardware Design

ระบบจะต้องมีรายละเอียดและสามารถทำงานได้ดังนี้

- สามารถแตะบัตรและอ่านข้อมูล id ของบัตรได้โดยใช้ sensor RFID Reader
- สามารถแสดงผลการ check in ให้ผู้ใช้งานทราบเมื่อมีการแตะบัตร โดยแสดงผลผ่าน LCD Screen
- สามารถใช้ internet ในการติดต่อกับ API เพื่อเพิ่มนักเรียนเข้าไปในห้องหรือเรียกดูข้อมูลนักเรียนได้โดยใช้ NodeMCU

### 6.2.3 Database Design

โครงสร้าง Database (Firestore Database) แบ่งออกเป็น 3 collections ได้แก่

- room  
เก็บข้อมูลนักเรียนที่ทำการแตะบัตร check in เข้าห้อง โดยในแต่ละ document ใน collection เก็บข้อมูลรหัสบัตร ชื่อ-นามสกุล และ เวลาที่ทำการ check in
- room-detail  
เก็บข้อมูล capacity ของห้อง
- students-name  
เก็บข้อมูลนักเรียน ชื่อ-นามสกุล รหัสบัตร โดยอ้างอิงกับบัตรนิสิตที่จะใช้ในการแตะ check in เข้าห้อง

### 6.2.4 API Design

การใช้งาน Embedded System กับ Firebase Firestore Database นั้นมีขั้นตอนการใช้งานที่ยุ่งยาก อีกทั้งยังต้องมี Boilerplate Code จำนวนมากในการใช้งาน และ Library ยังมี Feature น้อยกว่า Library ใน JS ecosystem เป็นอย่างมาก จึงมีการ Design API เพื่อให้การพัฒนาของ Embedded System สะดวกขึ้นอีกทั้งยังเป็นการลดภาระของ Embedded System ทำให้ระบบมีประสิทธิภาพและความยืดหยุ่นมากขึ้น

โดย Frontend มีการใช้งาน Next.js ซึ่งเป็น Frontend Web Development Framework ที่มี Feature Next API ทำให้การสร้าง API สามารถทำได้ง่าย และสามารถนำไป Host ที่ Vercel ในที่เดียวกับ Website ได้เลยโดยไม่ที่ค่าใช้จ่าย

API นั้นเขียนโดยภาษา TypeScript โดยเป็น REST API ซึ่งทำการ handle POST Request โดย Embedded System นั้นคือ ESP8266 นั้นสามารถส่ง POST Request มายัง API ที่ <https://embedded-project.vercel.app/api/user> โดยใน request ส่งข้อมูลในรูปแบบ JSON โดยประกอบด้วย field card id โดย เป็น string hexadecimal 10 หลัก จากนั้น API จะทำการตรวจสอบจาก database ว่ามีนิสิตคนนี้อยู่ในระบบหรือไม่ และทำการ Check in และ Check out ออกจากห้องเรียนนั้นๆ อีกทั้งยังมีการเช็คจำนวนนิสิตในห้องเรียน แล้วจึงตอบเป็น status ไปยัง ESP8266 เพื่อทำการประมวลผลและทำการแสดงผลต่อไป

Source Code ของ API อยู่ที่ <https://github.com/noppakorn/embedded-project/blob/main/frontend/pages/api/user.ts>

## 6.3 Embedded System Development

By: Noppakorn Jiravaranon

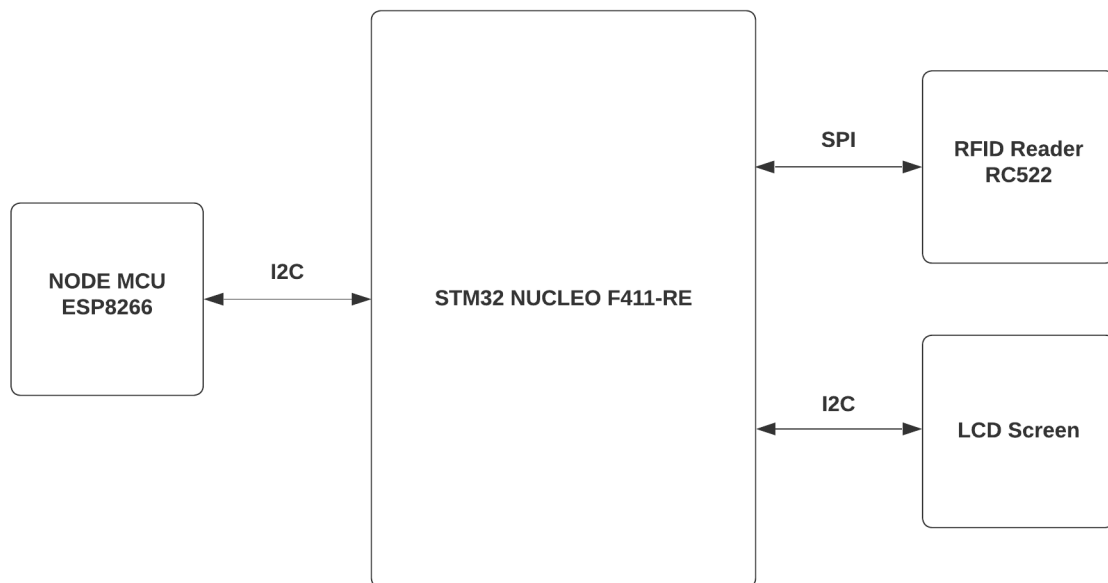
### 6.3.1 Responsibility

1. การเชื่อมต่อระหว่าง Sensor และ Hardware
2. การโปรแกรม STM32
3. การโปรแกรม ESP8266

### 6.3.2 Hardware Connectivity

Hardware ทั้งหมดเชื่อมต่อกันโดยมี STM32 NUCLEO F411-RE เป็นศูนย์กลาง โดยมีการเชื่อมต่อกันดังนี้

1. NodeMCU ESP8266 เชื่อมต่อกับ STM32 โดย I2C โดย ESP8266 เป็น master
2. RFID Reader RC522 เชื่อมต่อกับ STM32 โดย SPI
3. LCD Screen (LCD1602) เชื่อมต่อกับ STM32 โดย I2C



### 6.3.3 Function of Hardware

- STM32 NUCLEO F411-RE มีหน้าที่เป็นศูนย์กลางการเชื่อมต่อระหว่าง Hardware ต่างๆ และประมวลผลที่ได้รับจาก NodeMCU
- NodeMCU ESP8266 มีหน้าที่ติดต่อไปยัง API ที่ Host บน Vercel โดยทำการ ส่ง GET และ POST Request ไปยัง <https://embedded-project.vercel.app/api/user> โดยส่งเป็นรหัสของบัตรนิสิต และจะได้ Response เป็นผลของการ Check In หรือ Check Out และทำการประมวลผลในขั้นต้นเพื่อให้ง่ายต่อการนำข้อมูลไปใช้บน STM32
- RFID Reader RC522 มีหน้าที่อ่านบัตรนิสิตเพื่อส่งข้อมูลให้ STM32
- LCD Screen (LCD1602) มีหน้าที่แสดงผลการ Check In ให้ผู้ใช้ทราบ

#### 6.3.4 ESP8266

ในงานได้มีการใช้งาน ESP8266 ซึ่งได้ทำการโปรแกรมโดยใช้ Arduino IDE และมีการใช้งาน library **Wire.h** ในการใช้งาน I2C สำหรับการติดต่อกับ STM32 และ **ArduinoJson.h** ในช่วยประมวลผล Response จาก API

ESP8266 จะมีขั้นตอนการทำงานดังนี้

1. คอย monitor bus I2C ว่ามีการส่งจาก STM32 หรือไม่โดยจะ monitor จนกว่าจะมีการส่งข้อมูล ข้อมูลที่ส่งจาก STM32 จะเป็น card ID ซึ่งเป็น string ความยาว 10
2. เมื่อได้รับ string ครบถ้วนจะมีการส่ง POST Request ไปยัง API ที่ host ไว้ที่ Vercel
3. รอการตอบกลับจาก API ซึ่งจะได้การตอบกลับเป็น JSON โดยมีสถานะ http response code เป็น 200 หรือ 404 ถ้าเป็น response code นอกเหนือจากนี้จะส่ง response แจ้งไปยัง STM32 ว่าเกิด error ขึ้น
4. ถ้า response code เป็น 404 แสดงถึงการไม่มี card ID ในฐานข้อมูล ก็จึงส่ง "2" ไปยัง STM32 แสดงการไม่มี user
5. ถ้า response code เป็น 200 แสดงถึงการมี card ID ในฐานข้อมูล โดยข้อมูลในช่อง "status" ของ JSON มีค่าที่เป็นไปได้ 3 ค่า นั่นคือ checked\_in, checked\_out และ room\_full
6. จากนั้นทำการพิจารณาค่าที่ได้รับจาก API แล้วจึงส่ง "10" แสดงการ Check out หรือ "11" แสดงการ Check in ไปยัง STM32 โดยในสองกรณีข้างต้น จะมีการส่ง first name ของ user ไปด้วย กรณีห้องเต็มไม่สามารถ check in ได้ก็จะทำการส่ง "12" ไปยัง STM32 ตามด้วยการส่ง occupancy/capacity
7. เมื่อส่งข้อมูลเรียบร้อยแล้วจึงกลับไปทำงานขั้นตอนที่ 1

#### 6.3.5 STM32

ในงานนี้ได้มีการใช้ STM32 NUCLEO F411-RE เป็นศูนย์กลางการเชื่อมต่อ hardware ทั้งหมด โดยมีการต่อกับ ESP8266 (โดย I2C), RC522 (โดย SPI) และ LCD1602 (โดย I2C) โดย STM32 จะคอยรับข้อมูลจาก RC522 และส่งข้อมูลไปยัง ESP8266 เพื่อติดต่อ API และแสดงผลการทำงานผ่าน LCD1602

การเขียนโปรแกรมบน STM32 ใช้ STM32CubeIDE และใช้ภาษา C โดยมีการใช้ library ของ MFRC522 และ LCD1602 ที่เป็น open source software

1. คอยตรวจสอบบัตรเมื่อพบบัตรจึงส่ง Card ID ไปยัง ESP8266 ผ่าน I2C โดย Card ID เป็น string ความยาว 10
2. รอ response ซึ่งเป็นสถานะการทำงานจาก ESP8266
3. เมื่อได้รับสถานะ จึงแสดงข้อมูล ถ้าได้รับ สถานะ "10" และ "11" จะแสดงชื่อผู้เข้าใช้ ถ้าได้รับ "12" จะแสดงค่า occupancy ถ้าได้รับค่า "2" จะแสดงค่าว่าไม่มีผู้ใช้ ถ้าได้ค่าอื่นๆ จะแสดง error
4. เมื่อทำงานเสร็จสิ้นจึงกลับไปทำงานขั้นตอนที่ 1

#### 6.3.6 Operation of the system

1. STM32 คอยตรวจสอบบัตร ผ่าน RFID Reader RC522 เมื่อพบบัตรจึงส่ง Card ID ไปยัง ESP8266 ผ่าน I2C
2. เมื่อ ESP8266 ได้รับข้อมูลผ่าน I2C จึงส่งข้อมูลไปยัง api ที่ Vercel เมื่อได้รับข้อมูลกลับมา จึงส่งข้อมูลกลับไปยัง STM32
3. เมื่อ STM32 ได้รับข้อมูลกลับมาจึงประมวลผลที่ได้รับและแสดงผลผ่าน LCD Screen เพื่อให้ User ทราบ
4. เมื่อจบกระบวนการจึงกลับไปเริ่มในขั้นตอนแรก

## 6.4 UI/UX Designer and Development

By: Nopparuj Poonsubanan

### 6.4.1 Responsibility

1. UI/UX Design: กำหนด Feature ของเว็บและการใช้งานของ user ให้มีประสิทธิภาพ และใช้งานง่าย อีกทั้งยังมีการ Design ความสวยงามของ UI
2. Web Development นำ Design ที่กำหนดไว้ในขั้นตอน UI/UX Design มาทำเป็นเว็บไซต์

### 6.4.2 UI Design

แบ่งส่วนแสดงผลออกเป็น 3 ส่วน ได้แก่

1. ส่วนแสดงข้อมูลจำนวนนักเรียนภายในห้องและแก้ไข capacity ของห้อง
2. ส่วน search box ใช้ในการค้นหานักเรียนในห้อง
3. ส่วนที่แสดงข้อมูลรายละเอียดของนักเรียนแต่ละคนที่ check in เข้าห้อง โดยจะมีการแสดงเวลาที่นักเรียนอยู่ในห้องและมีปุ่มสำหรับการ check out

### 6.4.3 Web Development

Web Development นั้นได้ปฏิบัติงานโดยนำ Frontend Design ที่กำหนดไว้ในขั้นตอนการ Design UX/UI มาสร้างเป็น Single Page Application

ในงานนี้เลือกใช้ Next.js เป็น Frontend Web Development Framework เนื่องจากง่ายในการใช้งาน อีกทั้งยังมี Next API ที่ built-in ใน frontend อีกทั้งยังสามารถ Deploy ไปยัง Vercel ได้อย่างง่ายดาย

เว็บไซต์ใช้การดึงข้อมูลจาก Firebase มาแสดงผล การแก้ไขข้อมูลต่างๆ ใน Firebase ไม่ว่าจะเป็นการแก้ไขจาก Embedded System หรือการแก้ไขจากหน้าเว็บจะได้รับการอัปเดตโดยทันทีโดยไม่ต้องมี Interaction จาก user

โดยในเว็บไซต์นั้นจะมีรายชื่อนิสิตที่ทำการ Check in ด้วยการแตะบัตร โดยผู้ใช้งานเว็บไซต์จะสามารถส่ง Check out นิสิตจาก Dashboard ได้ อีกทั้งยังสามารถปรับปริมาณนิสิตที่อนุญาตให้เข้ามาใช้งานในห้องได้



