



**ใบอนุญาตโครงการปริญญานิพนธ์**  
**สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาคอมพิวเตอร์**  
**คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร**

**ชื่อปริญญานิพนธ์** ระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ : กรณีศึกษาการเข้าและออกงานของเจ้าหน้าที่ภาควิชาคอมพิวเตอร์  
Automatic photo recording for attendance checking system: A case study of Department of Computing employees.

**ผู้จัดทำ** นางสาวกรณลินล์ เชวงทรัพย์

**ปีการศึกษา** ปีการศึกษา 2561

โครงการปริญญานิพนธ์นี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตาม  
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

..... ประธานกรรมการสอบ

( ผู้ช่วยศาสตราจารย์ ดร.คทา ประดิษฐ์วงศ์ )

..... กรรมการสอบ

( ผู้ช่วยศาสตราจารย์ ดร.ภิญโญ แท้ประสาธสิทธิ์ )

..... อาจารย์ที่ปรึกษาและกรรมการสอบ

( อาจารย์ อภิเชก หงษ์วิทยากร )



ระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ :  
กรณีศึกษาการเข้าและออกงานของเจ้าหน้าที่ภาควิชาคอมพิวเตอร์  
Automatic photo recording for attendance checking system:  
A case study of Department of Computing employees.

กรณีนลินล์ เสงวทรพพ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาในหลักสูตรวิทยาศาสตรบัณฑิต  
สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาคอมพิวเตอร์  
คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร  
ปีการศึกษา 2560

**ชื่อปริญญาบัตร** ระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ : กรณีศึกษาการเข้าและออกงานของเจ้าหน้าที่ภาค  
วิชาคอมพิวเตอร์  
Automatic photo recording for attendance checking system: A case study of Department  
of Computing employees.

**ผู้จัดทำ** นางสาวกรณีนลินธ์ เสงฆ์ทรัพย์

**อาจารย์ที่ปรึกษา** อาจารย์ อภิเชก หงษ์วิทยากร

**ชื่อปริญญา** วิทยาศาสตรบัณฑิต (เทคโนโลยีสารสนเทศ)

**ปีการศึกษา** 2561

## บทคัดย่อ

การบันทึกเวลาเข้าออกงานของพนักงานเป็นเรื่องที่สำคัญซึ่งถูกใช้กันโดยทั่วไปทั้งภาครัฐและเอกชน ปัจจุบันการบันทึกเวลามีหลากหลายวิธีทั้งแบบดั้งเดิมคือใช้กระดาษและแบบอัตโนมัติที่ใช้เทคโนโลยีเข้าช่วย ดังนั้นจึงได้พัฒนาระบบ IoT ขึ้นสำหรับบันทึกเวลาเข้าออกงานของพนักงานโดยใช้ Raspberry Pi ซึ่งเป็นคอมพิวเตอร์ขนาดเล็ก ใช้ร่วมกับกล้องและหน้าจอสัมผัสสำหรับถ่ายภาพพนักงานทั้งขาเข้าและขาออก หลังจากถ่ายภาพแล้วภาพถ่ายจะถูกบันทึกและถูกแจ้งเตือนในไลน์กลุ่มของพนักงาน สมาชิกในกลุ่มไลน์สามารถแท็กตนเองหรือแท็กสมาชิกคนอื่น ๆ ได้ และหลังจากแท็กภาพถ่ายแล้ว ข้อมูลต่าง ๆ จะถูกบันทึกลงฐานข้อมูลหลักทันที นอกจากนี้ยังสามารถเรียกดูประวัติการเข้างานและออกงานได้อีกด้วย ส่วนของการพัฒนาซอฟต์แวร์จะใช้ LINE APIs เพื่อให้ทำงานร่วมกับอุปกรณ์ฝังตัวที่ใช้ Raspberry Pi ภายใต้สภาวะการทำงานแบบ IoT

**คำสำคัญ** : อินเทอร์เน็ตทุกสรรพสิ่ง, บันทึกเวลาอัตโนมัติ, บอทอินเทอร์เน็ตทุกสรรพสิ่ง

**Keyword** : Internet of Things, Automatic time stamping, LIFF-LINE, Bot IoT

## กิตติกรรมประกาศ

โครงการปริญญานิพนธ์เรื่องระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ กรณีศึกษาการเข้าและออกงานของเจ้าหน้าที่ภาควิชาคอมพิวเตอร์ ฉบับนี้จะไม่สำเร็จลุล่วงหากไม่มีบุคคลที่ให้คำแนะนำและความช่วยเหลือ สนับสนุนโครงการปริญญานิพนธ์ตั้งแต่เริ่มต้นจนเสร็จสมบูรณ์

ขอขอบพระคุณ อาจารย์ อภิเชก หงษ์วิทยากร รองหัวหน้าภาควิชาคอมพิวเตอร์และอาจารย์ประจำคณะวิทยาศาสตร์ ภาควิชาคอมพิวเตอร์ ที่กรุณารับเป็นอาจารย์ที่ปรึกษาผู้ทำการวิจัย ขอขอบคุณคณะกรรมการสอบได้แก่ (1) ผู้ช่วยศาสตราจารย์ ดร.คทา ประดิษฐ์วงศ์ (2) ผู้ช่วยศาสตราจารย์ ดร.ภิญโญ แท้ประสาทสิทธิ์ ที่คอยเสนอแนะข้อแก้ไขเพื่อปรับปรุงและพัฒนางานวิจัยให้มีความสมบูรณ์มากยิ่งขึ้น และขอขอบพระคุณ อาจารย์ เสฐลัทธิ์ รอดเหตุภัย ที่ให้คำแนะนำและปรึกษาเกี่ยวกับ Server

ขอขอบคุณ นายคมสันต์ สมวงษ์ และนายณัฐภัทร พลังอิสระ ที่คอยให้คำปรึกษาเกี่ยวกับกระบวนการถ่ายรูปและเพื่อน ๆ ที่คอยช่วยเหลือและแบ่งปันความรู้เกี่ยวกับบอท LINE

สุดท้ายนี้ขอขอบพระคุณบิดา มารดา ที่ส่งเสียข้าพเจ้าให้เรียนมาถึงบัดนี้ งานวิจัยชิ้นนี้ขอมอบเป็นเครื่องบูชา บิดา มารดา ครูอาจารย์ ผู้ให้ความรู้และทางเดินในอนาคตแก่ข้าพเจ้า ขอขอบคุณ

กรณีนลินต์ เชวงทรัพย์

# สารบัญ

หน้า

บทคัดย่อ .....	ก
กิตติกรรมประกาศ .....	ข
สารบัญ.....	ค
สารบัญตาราง.....	ฉ
สารบัญรูปภาพ.....	ช
บทที่ 1 บทนำ .....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ .....	1
1.3 ลักษณะและขอบเขต .....	1
1.4 อุปกรณ์และเครื่องมือ .....	2
1.4.1 ฮาร์ดแวร์.....	2
1.4.2 ซอฟต์แวร์.....	2
1.5 ผลที่คาดว่าจะได้รับ .....	2
บทที่ 2 ผลงานที่เกี่ยวข้อง .....	3
2.1 การนับคนเข้าออกด้วย Raspberry Pi.....	3
2.2 ระบบลงเวลาทำงานพนักงาน .....	5
2.3 TimeMint App ลงเวลาพนักงานด้วยมือถือ .....	7
บทที่ 3 ทฤษฎีและความรู้ที่เกี่ยวข้อง .....	9
3.1 Internet of Things (IoT) คืออะไร.....	9
3.2 ความรู้เบื้องต้นเกี่ยวกับ API .....	10
3.3 มารู้จักกับ LINE Front-end Framework (LIFF) .....	10
3.4 Raspberry Pi คืออะไร.....	13
3.5 มารู้จัก W3School .....	14
3.6 ความแตกต่างระหว่าง HTTP กับ HTTPS .....	15
3.7 การทำงานของ Messaging API LINE.....	15
3.8 การรับและส่งข้อมูลจาก LINE Platform.....	16
3.9 LINE Notify .....	17
3.10 ความรู้เบื้องต้นเกี่ยวกับ GitHub .....	18
บทที่ 4 ขั้นตอนและแผนการดำเนินงาน .....	21
4.1 รวบรวมข้อมูลและกำหนดขอบเขต.....	21
4.2 ศึกษาทฤษฎีและความรู้ที่เกี่ยวข้อง.....	21

## สารบัญ (ต่อ)

	หน้า
4.3 ออกแบบและพัฒนาหน้า Web Application .....	21
4.4 พัฒนาระบบ สร้างฐานข้อมูลในการจัดเก็บข้อมูลสมาชิก สร้าง Web Application.....	21
4.5 ทดสอบระบบ.....	21
4.6 จัดทำเอกสารและคู่มือประกอบโครงการ.....	21
4.7 แผนการดำเนินงาน.....	22
บทที่ 5 การวิเคราะห์และการออกแบบ.....	23
5.1 โครงสร้างหลักของระบบงาน (Process Hierarchy) .....	23
5.1.1 การทำงานของผู้ใช้งานกับตัวอุปกรณ์ .....	23
5.1.2 การทำงานของบอทและระบบการจัดการแท็กภาพถ่ายก่อนจัดเก็บลงฐานข้อมูล .....	23
5.2 ผังงาน (Flow Chart) .....	30
5.2.1 ผังการทำงานโดยรวมของระบบ .....	30
5.2.2 ผังการทำงานของผู้ใช้กับบอทในการจัดการกระบวนการแท็กภาพถ่ายที่ถูกถ่าย .....	31
5.2.3 ผังการทำงานเมื่อผู้ใช้ต้องการดูรายงานการเข้างานออกงาน .....	32
5.3 Use case Diagram .....	33
5.4 การออกแบบฐานข้อมูลและ Data Dictionary .....	34
5.4.1 ไดอะแกรมแสดงความสัมพันธ์ระหว่างผู้ใช้งานกับช่วงเวลาเข้าและขาออกงาน .....	34
5.4.2 Data Dictionary .....	35
บทที่ 6 ผลการดำเนินงาน ข้อจำกัด และข้อเสนอแนะ.....	37
6.1 ผลการดำเนินงาน .....	37
6.1.1 การทำงานของเว็บแอปพลิเคชันของตัวอุปกรณ์.....	37
6.1.2 การทำงานของเว็บแอปพลิเคชันบนไลน์กลุ่ม .....	37
6.2 ข้อจำกัด.....	37
6.3 ข้อเสนอแนะ .....	37
บรรณานุกรม .....	39
ภาคผนวก ก คู่มือการติดตั้งระบบ.....	41
ก.1 การติดตั้งระบบปฏิบัติการลงบน Raspberry Pi3 Model B.....	41
ก.2 การตั้งค่าเบื้องต้นให้กับ Raspberry Pi3 Model B .....	42
ก.3 การติดตั้ง PiCamera และ OpenCV.....	44
ก.4 การติดตั้งไลบรารีที่จำเป็นกับระบบงาน.....	47
ก.5 การสมัครและใช้งานบอทไลน์.....	49

## สารบัญ (ต่อ)

	หน้า
ก.6 การทดลองสร้างบอทด้วยการใช้ ngrok .....	54
ก.7 การใช้ LINE LIFF .....	59
ก.8 การสมัครใช้งานไลน์ Notify .....	63
ภาคผนวก ข คู่มือการใช้งาน .....	66
ข.1 ส่วนของเว็บแอปพลิเคชันบนตัวอุปกรณ์ Raspberry Pi3 .....	66
ข.2 ส่วนของเว็บแอปพลิเคชันบนมือถือที่แสดงบนไลน์ .....	69

## สารบัญตาราง

	หน้า
ตารางที่ 5. 1 ผู้ใช้งาน .....	35
ตารางที่ 5. 2 บันทึกเวลาทำงานขาเข้า .....	35
ตารางที่ 5. 3 บันทึกเวลาทำงานขาออก.....	35



## สารบัญรูป

	หน้า
รูปที่ 2.1 การจับภาพบุคคลเดินเข้าห้อง.....	3
รูปที่ 2.2 การขีดเส้นอ้างอิง กันเส้นระหว่างภายในและนอกห้อง .....	4
รูปที่ 2.3 การจับภาพบุคคลและนับจำนวนบุคคลเข้าออกห้อง.....	4
รูปที่ 2.4 ตัวอย่างอุปกรณ์ OneDee.....	5
รูปที่ 2.5 ตัวอย่างอุปกรณ์ OneDee.io .....	5
รูปที่ 2.6 ตัวอย่างการใช้งานแอปพลิเคชันของพนักงาน.....	5
รูปที่ 2.7 ภาพแอปพลิเคชันการเข้าออกงาน .....	6
รูปที่ 2.8 ภาพแอปพลิเคชันเวลาการเข้าออกงาน .....	6
รูปที่ 2.9 หน้าตัวอย่างพิกัดสถานที่ในระบบของ TimeMint.....	7
รูปที่ 2.10 หน้าวันทำงานในระบบของ TimeMint .....	7
รูปที่ 3.1 Internet of Things .....	9
รูปที่ 3.2 NETPIE .....	9
รูปที่ 3.3 Enhance Chat Bot ของ LIFF LINE.....	11
รูปที่ 3.4 Useful Extension.....	11
รูปที่ 3.5 ตัวอย่างการตอบกลับของบอทไลน์แบบ Carousel .....	12
รูปที่ 3.6 Business Use Case.....	12
รูปที่ 3.7 ตัวอย่างการวาดและส่งภาพ GIF ของLIFF LINE .....	12
รูปที่ 3.8 ตัวอย่าง GIPHY ของ LINE.....	13
รูปที่ 3.9 เกม Chess บน LIFF LINE.....	13
รูปที่ 3.10 เกม Jungle Pang บน LIFF LINE .....	13
รูปที่ 3.11 ภาพ Raspberry Pi .....	14
รูปที่ 3.12 W3school .....	14
รูปที่ 3.13 ภาพความแตกต่างของ HTTP และ HTTPS .....	15
รูปที่ 3.14 ภาพอธิบายการรับส่งของบอทด้วย Messaging API.....	16
รูปที่ 3.15 ภาพการสร้างบอท .....	16
รูปที่ 3.16 ภาพกลุ่มของ LINE Notify .....	17
รูปที่ 3.17 ภาพการแจ้งเตือนมายังกลุ่ม Fluffy F01 ในLINE .....	17
รูปที่ 3.18 ภาพ GitHub .....	18
รูปที่ 3.19 ภาพอธิบายการ Commit ของ GitHub.....	18
รูปที่ 3.20 ภาพอธิบายการ Push ของ GitHub.....	19
รูปที่ 3.21 ภาพการ Pull ของ GitHub.....	19
รูปที่ 4.1 ภาพแผนดำเนินงาน .....	22

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 5.1 ภาพสถาปัตยกรรมของระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ.....	24
รูปที่ 5.2 อุปกรณ์ Raspberry Pi ต่อกับกล้อง.....	25
รูปที่ 5.3 ภาพจอ Touch Screen.....	25
รูปที่ 5.4 ภาพหน้าเว็บการเข้าออกงานบนหน้าจอ.....	25
รูปที่ 5.5 ภาพการแจ้งเตือนภาพถ่ายและเวลา.....	26
รูปที่ 5.6 ภาพการคุยโต้ตอบของผู้ใช้กับบอทบนมือถือ iPhone.....	26
รูปที่ 5.7 ภาพการคุยโต้ตอบของผู้ใช้กับบอทบนมือถือ Sony.....	27
รูปที่ 5.8 เป็นภาพหน้าเว็บที่มีภาพถ่ายที่ยังไม่ได้แท็ก.....	27
รูปที่ 5.9 เป็นภาพหน้าเว็บที่มีชื่อเล่นของผู้ใช้งาน.....	28
รูปที่ 5.10 เป็นภาพหน้าเว็บให้ผู้ใช้งาน.....	28
รูปที่ 5.11 ภาพการเก็บภาพถ่ายลงฟลashedของภาพถ่าย.....	28
รูปที่ 5.12 ภาพการบันทึกข้อมูลการแท็กภาพถ่ายเข้าฐานข้อมูล.....	29
รูปที่ 5.13 ภาพการบันทึกข้อมูลการแท็กภาพถ่ายออกฐานข้อมูล.....	29
รูปที่ 5.14 ภาพ Flow Chart ภาพรวมของระบบ.....	30
รูปที่ 5.15 ภาพ Flow Chart กระบวนการแท็กภาพถ่ายของผู้ใช้กับบอท.....	31
รูปที่ 5.16 ภาพ Flow Chart กระบวนการดูรายงานการเข้างานออกงานของผู้ใช้กับบอท.....	32
รูปที่ 5.17 ภาพ Use case Diagram ของระบบ.....	33
รูปที่ 5.18 ภาพความสัมพันธ์ของผู้ใช้งานกับช่วงเวลาเข้างานและขาออกงาน.....	34
รูปผนวกที่ ก.1 การดาวน์โหลด Raspbian.....	41
รูปผนวกที่ ก.2 ภาพโปรแกรม Etcher สำหรับแฟลชไฟล์.....	41
รูปผนวกที่ ก.3 ภาพไฟเตอร์ที่เก็บ Raspbian ที่ดาวน์โหลดมา.....	41
รูปผนวกที่ ก.4 Image File ของ Raspbian ที่ดาวน์โหลดมา.....	41
รูปผนวกที่ ก.5 ภาพการแฟลช Image File ลง SD Card.....	42
รูปผนวกที่ ก.6 ภาพการเลือก Raspberry Pi Configuration.....	42
รูปผนวกที่ ก.7 เลือกที่ System และทำการตั้งชื่อให้กับ Hostname ของ Raspberry Pi.....	42
รูปผนวกที่ ก.8 เลือกที่ Interfaces ทำการตั้งค่า Enable และ Disable.....	43
รูปผนวกที่ ก.9 เลือกที่ Localisation ทำการตั้งค่า Locale , Time Zone และ Keyboard.....	43
รูปผนวกที่ ก.10 ภาพการตั้งค่า Raspberry Pi.....	44
รูปผนวกที่ ก.11 ภาพการเปลี่ยนรหัส Raspberry Pi ใหม่.....	44
รูปผนวกที่ ก.12 ภาพติดตั้ง cryptography , pycrypto และ paramiko.....	47
รูปผนวกที่ ก.13 ภาพติดตั้ง pysftp.....	47

## สารบัญรูป (ต่อ)

	หน้า
รูปผนวกที่ ก.14 ภาพติดตั้ง Flask.....	48
รูปผนวกที่ ก.15 ภาพโค้ดสำหรับรันและกำหนด Port ของ Flask.....	48
รูปผนวกที่ ก.16 ภาพการรัน Flask บน Raspberry Pi.....	48
รูปผนวกที่ ก.17 ภาพการเปิดเว็บแอปพลิเคชันที่ใช้ Flask บน Raspberry Pi.....	49
รูปผนวกที่ ก.18 ภาพการเข้าสู่ระบบเพื่อใช้บริการไลน์ในการสร้างบอท.....	49
รูปผนวกที่ ก.19 ภาพการสร้าง Provider สำหรับเก็บบอทไลน์.....	50
รูปผนวกที่ ก.20 ภาพการเริ่มสร้างบอทไลน์โดยเลือก Massaging API.....	50
รูปผนวกที่ ก.21 ภาพการสร้างบอทไลน์โดยกำหนดค่าที่ Channel Setting.....	51
รูปผนวกที่ ก.22 ภาพเกี่ยวกับโค้ดที่ทักทายของบอทไลน์.....	52
รูปผนวกที่ ก.23 ภาพการกำหนดค่าให้ Use Webhooks โดยเลือก Enabled.....	52
รูปผนวกที่ ก.24 ภาพการใส่ URL ของบอทไลน์ที่ Webhook URL.....	52
รูปผนวกที่ ก.25 ภาพการกำหนดค่า Enabled ให้กับ Allow bot to join group chart.....	53
รูปผนวกที่ ก.26 ภาพ QR-Code ของบอทไลน์.....	53
รูปผนวกที่ ก.27 ภาพการคุยกับบอทของผู้ใช้งาน.....	53
รูปผนวกที่ ก.28 ภาพการดาวน์โหลด Node js.....	54
รูปผนวกที่ ก.29 ภาพการเช็คเวอร์ชันของ Node js และ npm.....	54
รูปผนวกที่ ก.30 ภาพการแก้ไข Channel Access Token และ Channel Secret ของบอทไลน์.....	55
รูปผนวกที่ ก.31 ภาพการเริ่มใช้งานบอทไลน์ที่ดาวน์โหลดมา.....	55
รูปผนวกที่ ก.32 ภาพการทำงานของบอทบนหน้าเว็บ.....	55
รูปผนวกที่ ก.33 ภาพการติดตั้งและตั้งค่าเพื่อใช้ ngrok บน PC.....	56
รูปผนวกที่ ก.34 ภาพการรัน ngrok เพื่อเชื่อมกับบอทไลน์.....	56
รูปผนวกที่ ก.35 ภาพการตรวจสอบ Webhook URL ของบอท.....	56
รูปผนวกที่ ก.36 ภาพการรัน ngrok บน PC.....	57
รูปผนวกที่ ก.37 ภาพการตอบกลับของบอทกับผู้ใช้.....	57
รูปผนวกที่ ก.38 ภาพสถานะการรับส่งค่าผ่าน Message API LINE.....	58
รูปผนวกที่ ก.39 ภาพ Log ของบอทไลน์.....	58
รูปผนวกที่ ก.40 ภาพบอทไลน์ที่ถูกสร้างขึ้น.....	59
รูปผนวกที่ ก.41 ภาพการให้บริการ LIFF LINE.....	59
รูปผนวกที่ ก.42 ภาพสร้าง LIFF ใหม่.....	60
รูปผนวกที่ ก.43 ภาพ LIFF URL ของ Add Member LIFF ที่สร้างขึ้น.....	60
รูปผนวกที่ ก.44 ภาพการกำหนดคีย์เวิร์ดเกี่ยวกับการเพิ่มสมาชิกใหม่ให้กับบอทไลน์.....	60
รูปผนวกที่ ก.45 ภาพโค้ดฟังก์ชันที่เขียนขึ้น เพื่อให้บอทตอบกลับผู้ใช้ในรูปแบบ Flex Message.....	61

## สารบัญรูป (ต่อ)

### หน้า

รูปผนวกที่ ก.46 ภาพโค้ดบอทไลน์ .....	62
รูปผนวกที่ ก.47 ภาพการตอบกลับของบอทในรูปแบบ Flex Message .....	62
รูปผนวกที่ ก.48 ภาพหน้าเว็บแอปพลิเคชันการสมัครสมาชิกใหม่ .....	63
รูปผนวกที่ ก.49 ภาพเข้าสู่ระบบ LINE Notify .....	63
รูปผนวกที่ ก.50 ภาพการออก Access Token เพื่อใช้บริการของ LINE Notify.....	64
รูปผนวกที่ ก.51 ภาพการตั้งชื่อ Token และเลือกกลุ่มที่ต้องการแจ้งเตือน .....	64
รูปผนวกที่ ก.52 ภาพ Token ที่ออกให้ของ LINE Notify.....	65
รูปผนวกที่ ก.53 ภาพการแจ้งเตือนครั้งแรกของ LINE Notify.....	65
รูปผนวกที่ ก.54 ภาพการแจ้งเตือนเวลาและคีย์เวิร์ดจากตัวอุปกรณ์ Raspberry Pi.....	65
รูปผนวกที่ ข.1 ภาพหน้าจอ Raspberry Pi การเข้าไปยังที่อยู่ของไฟล์ appCam.py .....	66
รูปผนวกที่ ข.2 ภาพการรันไฟล์ appCam.py เพื่อใช้ Flask Framework .....	66
รูปผนวกที่ ข.3 ภาพ URL ที่สามารถเข้าถึงหน้าเว็บแอปพลิเคชันที่รัน Flask Framework .....	66
รูปผนวกที่ ข.4 หน้าหลักของเว็บแอปพลิเคชันบนตัวอุปกรณ์ Raspberry Pi.....	67
รูปผนวกที่ ข.5 หน้าวีโอสตรีมบนเว็บแอปพลิเคชันเพื่อให้ผู้ใช้ทำการถ่ายภาพตนเองบน Raspberry Pi .....	67
รูปผนวกที่ ข.6 หน้ายืนยันภาพถ่ายที่ถูกถ่ายจากตัวอุปกรณ์ Raspberry Pi.....	68
รูปผนวกที่ ข.7 ภาพการ Alert เมื่อผู้ใช้กดปุ่ม ยกเลิก บนเว็บแอปพลิเคชันที่แสดงบนจอ Raspberry Pi .....	68
รูปผนวกที่ ข.8 หน้าบอกสถานะของกระบวนการถ่ายภาพเพื่อบันทึกเวลาเข้าออกงาน.....	68
รูปผนวกที่ ข.9 ภาพการแจ้งเตือน เวลา และคีย์เวิร์ดที่ถูกส่งมาจากตัวอุปกรณ์ Raspberry Pi.....	69
รูปผนวกที่ ข.10 การตอบกลับของบอทไลน์เพื่อแท็กภาพถ่าย.....	69
รูปผนวกที่ ข.11 การตอบกลับของบอทไลน์เพื่อเช็คเวลาเข้าออกงาน .....	69
รูปผนวกที่ ข.12 การตอบกลับของบอทไลน์เพื่อสมัครสมาชิกใหม่.....	70
รูปผนวกที่ ข.13 การตอบกลับของบอทไลน์เพื่อแท็กภาพถ่ายและเช็คเวลาเข้าออกงาน .....	70
รูปผนวกที่ ข.14 การตอบกลับของบอทไลน์เมื่อผู้ใช้พิมพ์คีย์เวิร์ดเพื่อแท็กภาพถ่าย .....	71
รูปผนวกที่ ข.15 หน้าเว็บแอปพลิเคชันภาพที่ยังไม่ถูกแท็กเจ้าของภาพ.....	72
รูปผนวกที่ ข.16 หน้าเว็บแอปพลิเคชันหน้าสมัครสมาชิกใหม่เข้าสู่ระบบและการ Alert .....	73
รูปผนวกที่ ข.17 หน้าเว็บแอปพลิเคชันหน้าเข้าสู่ระบบและการ Alert เมื่อผู้ใช้กรอกข้อมูลไม่ถูกต้อง .....	73
รูปผนวกที่ ข.18 หน้าเว็บแอปพลิเคชันหน้าแท็กเจ้าของภาพถ่าย.....	74
รูปผนวกที่ ข.19 หน้าเว็บแอปพลิเคชันหน้ายืนยันการแท็กเจ้าของภาพถ่าย.....	74
รูปผนวกที่ ข.20 หน้าเว็บแอปพลิเคชันหน้าภาพถ่ายที่ยังไม่ถูกแท็กเจ้าของภาพ.....	75
รูปผนวกที่ ข.21 การตอบกลับของบอทไลน์เมื่อผู้ใช้พิมพ์คีย์เวิร์ดเพื่อเช็คเวลาเข้าออกงาน .....	75
รูปผนวกที่ ข.22 หน้าเว็บแอปพลิเคชันหน้าเข้าสู่ระบบและการ Alert เมื่อผู้ใช้กรอกข้อมูลไม่ถูกต้อง.....	76
รูปผนวกที่ ข.23 หน้าเว็บแอปพลิเคชันหน้าเลือกวันที่ต้องการดูรายงานการเข้าออกงาน .....	76

## สารบัญรูป (ต่อ)

	หน้า
รูปผนวกที่ ข.24 หน้าเว็บแอปพลิเคชันหน้ารายชื่อผู้ใช้งานเพื่อดูรายงานเวลาเข้างานออกงาน .....	77
รูปผนวกที่ ข.25 หน้าเว็บแอปพลิเคชันรายงาน ช่วงเวลาในการเข้าออกงานของผู้ใช้งาน .....	77
รูปผนวกที่ ข.26 หน้าเว็บแอปพลิเคชันช่วงเวลาเข้าออกงานของผู้ใช้งาน .....	78
รูปผนวกที่ ข.27 ภาพวันที่ ช่วงเวลาขาเข้า ช่วงเวลาขาออก และคลิกเพื่อดูภาพถ่าย .....	78
รูปผนวกที่ ข.28 หน้าเว็บแอปพลิเคชันเวลาและภาพถ่ายของผู้ใช้งาน.....	78

# บทที่ 1 บทนำ

## 1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันการเข้าออกงานขององค์กร โรงงานและบริษัท มีการกำหนดระยะเวลาในการเข้างานและออกงานของพนักงานทั้งภาครัฐและเอกชน เพื่อตรวจสอบว่าพนักงานมาเข้างานตรงตามเวลาและไม่มาทำงานสาย ไม่ใช่แค่การเข้างานเพื่อทำงานเท่านั้น การเช็คชื่อมาเรียนของนักเรียนในโรงเรียนถือว่าเป็นเรื่องที่สำคัญเพราะโรงเรียนต้องตรวจสอบจำนวนของนักเรียนที่มาเรียนในแต่ละวัน เพื่อที่จะนำไปจัดการกับอะไรหลาย ๆ อย่าง แต่การจัดการนั้นยังคงเป็นแบบเดิมคือยังคงเป็นใบกระดาษ หรือเป็นระบบที่ต้องจัดการด้วยมือ เช่น การเช็คชื่อการมาเรียนของนักเรียน การเซ็นชื่อหรือตอกบัตรเพื่อเข้าและออกงาน เป็นต้น

ดังนั้นจะใช้อุปกรณ์ Internet of Things (IoT) มาช่วยในการจัดการทำระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ โดยตัวอุปกรณ์มีต้นทุนที่ไม่สูงมาก ซึ่งจะใช้ตัว Raspberry Pi ในการจัดการควบคุมตัวเซนเซอร์ต่าง ๆ เช่น การถ่ายภาพพนักงานในการเข้างานตอนเช้าและก่อนกลับ ระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพจะส่งไปบันทึกไว้ในฐานข้อมูลพร้อมกับเวลา เพื่อนำมาตรวจสอบการมาทำงานของพนักงาน และจะใช้ LINE Front-end Framework (LIFF) ในการเชื่อมต่อกับ Web Application เพื่อแสดงภาพถ่ายที่ถูกถ่ายในวันนั้น ๆ

## 1.2 วัตถุประสงค์

- 1) ระบบบันทึกเวลาเข้าออกด้วยการถ่ายภาพ
- 2) เพื่อศึกษาการพัฒนา Web Application ลงบน LINE Front-end Framework (LIFF)
- 3) เพื่อความสะดวกสบายของผู้ใช้งานในการตรวจสอบการมาทำงานของพนักงาน

## 1.3 ลักษณะและขอบเขต

โครงงานวิจัยนี้เป็นการนำอุปกรณ์มาเชื่อมต่อกัน สามารถสั่งการหรือควบคุมการทำงานได้ซึ่งเรียกว่า “Internet of Things” มีพื้นฐานอยู่บนระบบฝังตัว หรือสมองกลฝังตัว(Embedded System) โดยจะนำมาประยุกต์ทำระบบบันทึกเวลาเข้าออกด้วยการถ่ายภาพ ซึ่งจะใช้ Raspberry Pi ที่เป็นบอร์ดไมโครคอนโทรลเลอร์มีโมดูลสื่อสารไร้สาย ทำงานร่วมกันเป็น Internet of Things มาควบคุมการถ่ายภาพจากตัวเซนเซอร์โดยจะใช้ Raspberry Pi Camera Module ในการถ่ายภาพเจ้าหน้าที่ของภาควิชาในเวลาเข้าทำงานและก่อนกลับบ้านในแต่ละวัน จากนั้นจะส่งภาพถ่ายและเวลาบันทึกลงในฐานข้อมูลสำรองนั้นคือจะไปแสดงบน LINE และหลังจากนั้นเมื่อมีการจับคู่ภาพว่าเป็นใครแล้วจะส่งรูปที่ถ่ายไปพร้อมกับเวลา นำไปบันทึกลงฐานข้อมูลหลักที่อยู่บน Server

## 1.4 อุปกรณ์และเครื่องมือ

### 1.4.1 ฮาร์ดแวร์

#### 1) Raspberry Pi3 Model B

- หน่วยประมวลผลกลาง Broadcom BCM2837 64-bit ARM Quad-core  
ความเร็ว 1.2 GHz
  - หน่วยความจำสำรองขนาด 1 GB (SDRAM) แชนจ์กับ GPU
  - หน่วยความจำภายนอก Micro-SD 1 ช่อง
  - พิน GPIO 40 ขา
  - ช่อง USB 2.0 ทั้งหมด 4 ช่อง
  - ช่องต่อจอแสดงผล HDMI 1 ช่อง
  - ช่องต่อสายแพกล้อง Camera interface (CSI) 1ช่อง

#### 2) Raspberry Pi Camera v.2 Module

- ขนาด 25x24x9 มม.
- ความละเอียดภาพ 8 เมกะพิกเซล

#### 3) จอภาพ

#### 4) สาย HDMI

#### 5) แป้นพิมพ์

#### 6) เมาส์

### 1.4.2 ซอฟต์แวร์

#### 1) ระบบปฏิบัติการ Linux

#### 2) ระบบปฏิบัติการ Raspbian

#### 3) ระบบปฏิบัติการ Window 10

#### 4) Python

#### 5) OpenCV

#### 6) Atom IDE

## 1.5 ผลที่คาดว่าจะได้รับ

- 1) ระบบสามารถบันทึกเวลาและถ่ายภาพได้ และส่งไปบันทึกลงฐานข้อมูลได้
- 2) สามารถดึงข้อมูลที่เก็บไว้ในฐานข้อมูลขึ้นมาแสดงผลบนหน้า Web Application
- 3) สามารถนำ Web Application ที่พัฒนามาประยุกต์ไปใช้กับ LINE Front-end Framework (LIFF) ในการแสดงภาพถ่ายของพนักงานบนภาคีวิชาคอมพิวเตอร์

## บทที่ 2 ผลงานที่เกี่ยวข้อง

โครงการระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ (Automatic photo recording for attendance checking system) เป็นโครงการที่ประยุกต์โดยพัฒนาอุปกรณ์ที่สามารถเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้ มาประยุกต์กับการจัดการทางด้านเวลาพัฒนาเป็นระบบขึ้นมาเพื่อเป็นตัวช่วยในการอำนวยความสะดวกแก่ผู้ใช้งาน

### 2.1 การนับคนเข้าออกด้วย Raspberry Pi [1]

การนับจำนวนคนในการเข้าและออกห้องด้วยการใช้ Raspberry Pi เป็นโปรแกรมสำหรับใช้นับจำนวนคนสุทธิที่อยู่ในห้อง จากการเข้าห้อง และ การออกห้อง โดยใช้ OpenCV เป็นหลักและเชื่อมต่อข้อมูลไปแสดงยัง NETPIE โดยจะติดตั้งกล้องไว้ด้านบนของประตู ขนานกับพื้นโลก การติดกล้องไว้บริเวณด้านบนดังรูปที่ 2.1 นั้นเนื่องจากมุมตีกว่ามุมอื่น ๆ สามารถเห็นทุกคนที่ผ่านเข้าออกได้ และแทบจะไม่มีโอกาสจะเดินบังกันเลย



รูปที่ 2.1 การจับภาพบุคคลเดินเข้าห้อง

ในการจัดการเข้าออกเบื้องต้นนั้น โปรแกรมจะทำการขีดเส้นอ้างอิง (Reference Line) กันเส้นระหว่างภายในห้อง และข้างนอก ดังรูปที่ 2.2 สำหรับเงื่อนไขในการจัดการเข้าออกนั้น ครั้งแรกที่กรอบปรากฏอยู่ภายนอกห้อง แล้วภายหลังเคลื่อนที่ตัดเส้นอ้างอิงขึ้นมาอยู่เหนือเส้นอ้างอิง จะถือว่า “เข้าห้อง” ส่วนหากครั้งแรกที่กรอบปรากฏอยู่ภายในห้อง แล้วภายหลังเคลื่อนที่ตัดเส้นอ้างอิง ลงมาอยู่ใต้เส้นอ้างอิง จะถือว่า “ออกจากห้อง” ดังรูปที่ 2.2 และ 2.3





รูปที่ 2.2 การขีดเส้นอ้างอิง กั้นเส้นระหว่างภายในและนอกห้อง



รูปที่ 2.3 การจับภาพบุคคลและนับจำนวนบุคคลเข้าออกห้อง

#### ข้อดีของระบบ

- สามารถนับจำนวนคนสุทธิตั้งแต่ออกในห้องพักได้
- สามารถตรวจจับว่าเป็นคนได้ทั้งขาเข้าและขาออก แม่นยำคิดเป็น 80.75%

#### ข้อเสียของระบบ

- สามารถนับจำนวนคนที่อยู่ภายในห้องพักเท่านั้น
- ไม่สามารถตรวจจับศีรษะคนได้ หากตรวจจับได้เพียงบางส่วนเท่านั้น

## 2.2 ระบบลงเวลาทำงานพนักงาน [2]



รูปที่ 2.4 ตัวอย่างอุปกรณ์ OneDee

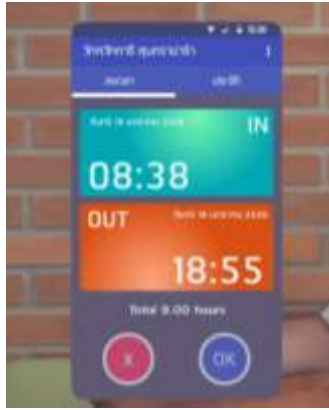
ระบบลงเวลาทำงานของพนักงานของ OneDee เป็นเครื่องลงเวลาพนักงานผ่านโทรศัพท์มือถือหรือต่อกับบัตรผ่านมือถือ Beacon โดยอุปกรณ์จะต้องติดตั้ง ณ ที่ทำงาน และพนักงานต้องดาวน์โหลดแอปพลิเคชันเพื่อลงเวลาการมาทำงานและออกงาน ซึ่งจะลงเวลาทำงานผ่านตัวอุปกรณ์ โดย “เครื่องลงเวลา OneDee” จะมีให้พนักงานลงเวลาเข้างานและออกงานได้ 4 แบบ คือ 1) ผ่าน Bluetooth พนักงานจะต้องอยู่ในระยะรัศมี 3 เมตร 2) ผ่านเครือข่าย Wi-Fi พนักงานจะต้องอยู่ในระยะรัศมี 15 เมตร 3) ผ่าน GPS พนักงานจะต้องอยู่ในระยะรัศมี 80 เมตร 4) ผ่าน QR Code เมื่อพนักงานลงเวลามาทำงานเรียบร้อยแล้ว ระบบลงเวลาทำงานของพนักงานจะรายงานผลออนไลน์ทันทีผ่านแอปพลิเคชันให้สามารถเรียกดูเวลาที่พนักงานเข้างานและออกงาน



รูปที่ 2.5 ตัวอย่างอุปกรณ์ OneDee.io



รูปที่ 2.6 ตัวอย่างการใช้งานแอปพลิเคชันของพนักงาน



รูปที่ 2.7 ภาพแอปพลิเคชันการเข้าออกงาน



รูปที่ 2.8 ภาพแอปพลิเคชันเวลาการเข้าออกงาน

#### ข้อดีของระบบ

- สามารถบันทึกเวลาทำงานของพนักงานได้ทั้งเวลาเมื่อเข้างานและเลิกงาน
- สามารถเรียกดูเวลาทำงานของพนักงานทางออนไลน์ผ่านแอปพลิเคชันได้ทันที

#### ข้อเสียของระบบ

- ผู้ใช้จะต้องติดตั้งแอปพลิเคชันลงบนมือถือก่อนถึงจะสามารถใช้บริการได้
- ผู้ใช้จะต้องมีการเรียนรู้ก่อนใช้งานระบบ

## 2.3 TimeMint App ลงเวลาพนักงานด้วยมือถือ [3]

เป็นแอปพลิเคชันต่อบัตรผ่านมือถือโดยผู้ใช้ต้องดาวน์โหลดมาติดตั้งในเครื่องก่อนใช้งาน ซึ่งในการบันทึกเวลาพนักงานในการเข้าออกงานนั้น TimeMint App จะมีการบันทึกเวลาเข้าออกได้ 3 รูปแบบ โดย 1) บันทึกการเข้าออกแบบสแกนป้าย QR CODE โดยหลังจากสแกนแล้วจะเปิดให้ถ่ายภาพผู้ใช้งาน 2) บันทึกการเข้าออกงานด้วยพิกัด โดยหลังจากหาพิกัดตำแหน่งที่ผู้ใช้งานอยู่แล้วจะเปิดให้ถ่ายภาพผู้ใช้งาน 3) บันทึกการเข้าออกด้วย Beacon ผ่าน Bluetooth ซึ่งผู้ใช้ไม่ต้องถ่ายภาพเพื่อบันทึกเวลาในการเข้าออกงาน นอกจากนี้ในแอปพลิเคชันผู้ยังสามารถระบุประเภทการลาโดยระบุวันที่จะลาได้



รูปที่ 2.9 หน้าตัวอย่างพิกัดสถานที่ในระบบของ TimeMint



รูปที่ 2.10 หน้าวันลางานในระบบของ TimeMint

#### **ข้อดีของระบบ**

- สามารถบันทึกเวลาทำงานของพนักงานได้ทั้งเวลาเมื่อเข้างานและเลิกงาน
- สามารถเรียกดูสรุปเวลาทำงานของพนักงานทางออนไลน์ผ่านแอปพลิเคชันได้
- สามารถกำหนดวันลาได้โดยจะต้องระบุวันที่ในปฏิทินและประเภทที่ผู้ใช้งานลา

#### **ข้อเสียของระบบ**

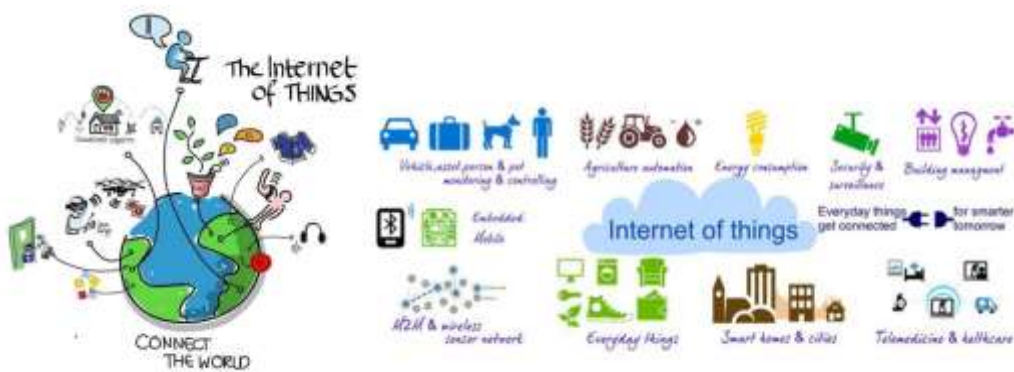
- ผู้ใช้จะต้องติดตั้งแอปพลิเคชันลงบนมือถือก่อนถึงจะสามารถใช้บริการได้

## บทที่ 3 ทฤษฎีและความรู้ที่เกี่ยวข้อง

ในบทนี้จะแสดงถึงรายละเอียดของทฤษฎี และความรู้ที่เกี่ยวข้องเพื่อนำมาใช้ในการพัฒนาโครงการดังต่อไปนี้

### 3.1 Internet of Things (IoT) คืออะไร

Internet of Things (IoT) หรือ “อินเทอร์เน็ตในทุกสิ่ง” หมายถึง การที่สิ่งต่าง ๆ ถูกเชื่อมโยงทุกสิ่งทุกอย่างเข้าสู่โลกอินเทอร์เน็ต ทำให้มนุษย์สามารถสั่งการ ควบคุมใช้งานอุปกรณ์ต่าง ๆ ผ่านทางเครือข่ายอินเทอร์เน็ต เช่น การสั่งเปิด/ปิด อุปกรณ์เครื่องใช้ไฟฟ้า รถยนต์ โทรศัพท์มือถือ เครื่องมือสื่อสาร เครื่องใช้สำนักงาน เครื่องมือทางการแพทย์ เครื่องจักรในโรงงานอุตสาหกรรม อาคาร บ้านเรือน เครื่องใช้ในชีวิตประจำวันต่าง ๆ ผ่านเครือข่ายอินเทอร์เน็ต เป็นต้น [4]



รูปที่ 3.1 Internet of Things

IoT เรียกอีกอย่างว่าเป็น M2M (Machine to Machine) คืออุปกรณ์ต่าง ๆ เชื่อมต่อเข้าไว้ด้วยกัน โดยช่วยให้สื่อสารกันผ่านระบบอินเทอร์เน็ต จากการคาดการณ์ ในปี ค.ศ.2020 สิ่งต่าง ๆ จะสามารถเชื่อมต่อกันได้ด้วยระบบ IoT ซึ่งจะส่งผลให้ผู้บริโภคทั่วไปจะเริ่มคุ้นเคยกับเทคโนโลยีที่ทำให้สามารถควบคุมสิ่งของต่าง ๆ ได้จากที่ไหนก็ได้



รูปที่ 3.2 NETPIE

ปัจจุบันได้มีการแบ่งกลุ่ม Internet of Things ออกตามตลาดไว้เป็น 2 กลุ่มดังนี้ [5]

- 1) Industrial IoT แบ่งจาก Local Network ที่มีหลายเทคโนโลยีที่แตกต่างกันในโครงข่าย Sensor Nodes โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะเชื่อมต่อแบบ IP network เพื่อเข้าสู่อินเทอร์เน็ต

- 2) Commercial IoT แบ่งจาก Local Communication ที่เป็น Bluetooth หรือ Ethernet (wired or wireless) โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะสื่อสารภายในกลุ่ม Sensor Nodes เดียวกันเท่านั้นหรือเป็นแบบ Local Devices เพียงอย่างเดียวอาจไม่ได้เชื่อมสู่อินเทอร์เน็ต

### 3.2 ความรู้เบื้องต้นเกี่ยวกับ API [6]

Application Programming Interface (API) เปรียบเสมือนสะพานที่เชื่อมต่อช่องทางการเข้าถึงต่างๆทั้งหมด (Channel Access) ไว้ด้วยกัน เป็นระบบบริการข้อมูลระหว่าง Client Side กับ Server Side ซึ่ง API ทำหน้าที่รับคำสั่งจากฝั่ง Client ยกตัวอย่างเช่น Web App, Mobile App, Desktop App เป็นต้น เมื่อมีคำสั่งใด ๆ API จะรับคำสั่งนั้น ๆ ไปประมวลผลและสรุปเป็นข้อมูลที่ตรงกับความต้องการและส่งข้อมูลเหล่านั้นตอบกลับไปที่ส่วนของ Client API ถือว่าเป็นกลุ่มของฟังก์ชัน ขั้นตอน หรือคลาสที่ระบบปฏิบัติการ (OS) หรือผู้ให้บริการสร้างขึ้นมาเพื่อรองรับการเรียกขอข้อมูลจากโปรแกรมอื่น ๆ ทั้งนี้ API สามารถใช้งานได้กับภาษาในการเขียนโปรแกรมที่รองรับเท่านั้น ซึ่งมันจะถูกจัดทำให้อยู่ในรูปแบบ Syntax หรือ Element ที่สามารถนำไปใช้ได้อย่างสะดวกสบาย

นอกจากนี้ API เป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น หรือเชื่อมการทำงานเข้ากับระบบปฏิบัติการได้ ยกตัวอย่างเช่น Twitter มีหลายเว็บไซต์ที่มีการเชื่อมโยงข้อมูลกับ Twitter ทั้งเป็นการอ่านข้อมูลและส่งข้อมูล [7]

#### ประโยชน์ของ API

1. สามารถรับส่งข้อมูลข้าม Server ได้
2. ไม่จำเป็นต้องเข้าหน้าเว็บหลัก ก็มีข้อมูลของเว็บหลัก จากเว็บที่ดึง API ซึ่งจะแบ่งเป็น
  - API ที่ขึ้นกับภาษา (Language - dependent API) คือ เอพีไอที่สามารถเรียกใช้จากโปรแกรมที่เขียนขึ้นด้วยภาษาเพียงภาษาใดภาษาหนึ่ง
  - API ที่ไม่ขึ้นกับภาษา (Language - independent API) คือ เอพีไอที่สามารถเรียกได้จากโปรแกรมหลายๆภาษา

### 3.3 มารู้จักกับ LINE Front-end Framework (LIFF) [8]

LIFF เป็น WebView ของ Line ที่ช่วยในการเชื่อมต่อระหว่าง Chat และ Web ได้อย่างมีประสิทธิภาพ แนนอนทั้ง HTML JS CSS สามารถทำงานรวมกันได้อย่างแนนอน ซึ่งแบ่ง LIFF สามารถแบ่งเป็น 4 โหมดดังต่อไปนี้

- 1) Compact Mode จะแสดงขึ้นมา 50% ของตัวแชท Line สามารถออกแบบฟอร์มเล็กๆได้อย่างง่ายดาย โดยหลัก ๆ มักจะนำไปทำ Input assistance, Simple form, Validation เช่น การทำ OTP, Member ID เป็นต้น
- 2) Tall Mode จะแสดงขึ้นมา 75% ของตัวแชท Line มักใช้ในการแสดงข้อมูล ใช้ทำ Product Catalog และ Long form for registration เป็นต้น
- 3) Full Mode จะแสดงขึ้นมา 100% หรือเต็มหน้าจอ ส่วนมากใช้ทำ Full Web Application หรือ Web utility เป็นต้น

- 4) Cover Mode จะแสดงขึ้นมา 100% หรือเต็มหน้าจอแต่จะไม่มี Title Bar ส่วนมากไว้ใช้ทำเกมส์ (Html5 Game)

#### ข้อดีของ LIFF [9]

- 1) Enhance Chat Bot โดยปกติใน Message จะไว้รับส่งข้อความ สติ๊กเกอร์ ยังไม่สามารถทำอะไรหลาย ๆ อย่างได้ แต่พอใช้ LIFF ซึ่งเป็น Bot ที่ช่วยในการนำทาง (Navigate) ก็จะสามารถทำอะไรหลาย ๆ อย่างได้มากขึ้นได้ เช่น การทำเกี่ยวกับโลจิสติกส์ ส่งจองตั๋วหนัง หรือทำเกี่ยว IoT มาควบคุมของภายในบ้านได้ เป็นต้น



รูปที่ 3.3 Enhance Chat Bot ของ LIFF LINE

- 2) สามารถทำ Useful Extension เพิ่มต่อจาก LINE ได้



รูปที่ 3.4 Useful Extension

- Business Use Case ยกตัวอย่างสมมุติว่าเราจะทำ e-Commerce ขึ้นมาสักอันบน LINE เราก็อาจจะไปสร้าง Bot ขึ้นมาตัวหนึ่งใช้ Template Message ประเภท Carousel เพื่อให้ผู้ใช้ทำการเลือกสินค้า จากนั้นตอนที่ชำระเงินเราสามารถ LIFF เป็นตัวกลางในการจบการขายโดยไม่ต้องออกจากห้องแชต





รูปที่ 3.5 ตัวอย่างการตอบกลับของบอทไลน์แบบ Carousel



รูปที่ 3.6 Business Use Case

- Chat Utility สามารถสร้าง App สารพัดประโยชน์ (Utility) ให้อยู่ภายในห้องแชตได้ เช่น App ส่งรูปภาพชนิด GIF หรือ App ที่ให้เราวาดรูปแล้วส่งให้เพื่อนได้

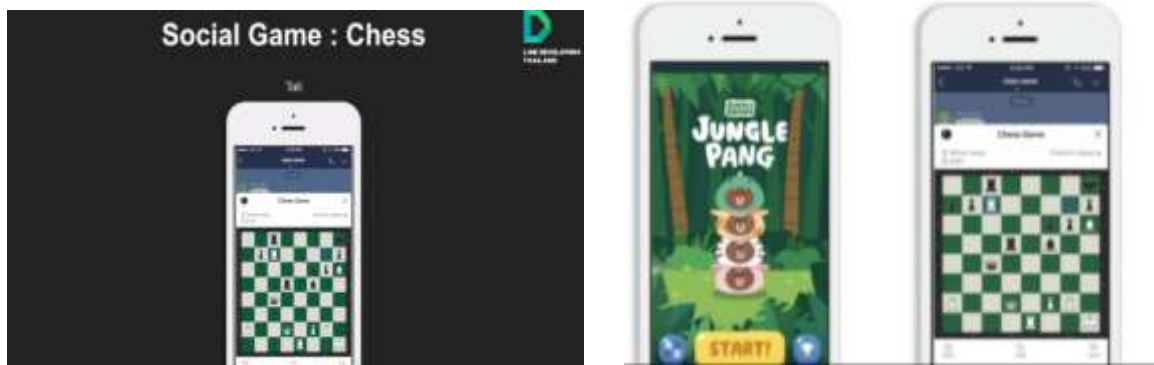


รูปที่ 3.7 ตัวอย่างการวาดและส่งภาพ GIF ของLIFF LINE



รูปที่ 3.8 ตัวอย่าง GIPHY ของ LINE

- Game สามารถสร้างเกมส์ HTML เพื่อเล่นแข่งกับเพื่อนในห้องแชตได้ ยกตัวอย่าง เกมส์ Jungle Pang



รูปที่ 3.9 เกม Chess บน LIFF LINE

รูปที่ 3.10 เกม Jungle Pang บน LIFF LINE

- 3) Reachable End Users ด้วย LINE มีผู้ใช้งานมากกว่า 4.2 ล้านคน ทำให้ผู้ใช้สามารถใช้ระบบที่พัฒนาขึ้นจากการแชร์ได้ง่าย

### 3.4 Raspberry Pi คืออะไร

Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็กที่จัดอยู่ในกลุ่มคอมพิวเตอร์แบบฝังตัว (Embedded Computer) เป้าหมายเพื่อส่งเสริมการเรียนการสอนวิทยาศาสตร์คอมพิวเตอร์ขั้นพื้นฐาน ในปัจจุบันเป็นที่นิยมมากขึ้นสำหรับการทำงานในด้านต่าง ๆ [10] ยกตัวอย่างเช่น ใช้เป็นคอมพิวเตอร์ส่วนบุคคล เพื่อดูหนัง ฟังเพลง เล่นอินเทอร์เน็ตได้ ใช้เป็นเว็บเซิร์ฟเวอร์สำหรับเปิดเว็บไซต์ขนาดเล็ก ใช้ควบคุมการทำงานของหุ่นยนต์และเครื่องจักร ใช้ควบคุมอุปกรณ์ไฟฟ้าอิเล็กทรอนิกส์ต่าง ๆ ได้ และใช้ในการพัฒนางานที่เป็นอุปกรณ์ Internet of Things ได้ เช่น ใช้ทำระบบเปิด/ปิดหลอดไฟ หรือเครื่องใช้ไฟฟ้าแบบอัตโนมัติ ใช้ทำประตูไฟฟ้า ล็อค/ปลดล็อคด้วยรีโมท หรือสั่งงานด้วยเสียง ใช้ทำสถานีวัดอุณหภูมิความชื้นในอากาศ ใช้ทำระบบกล้องวงจรปิด และยังใช้ทำระบบตรวจจับใบหน้าได้อีกด้วย [11]

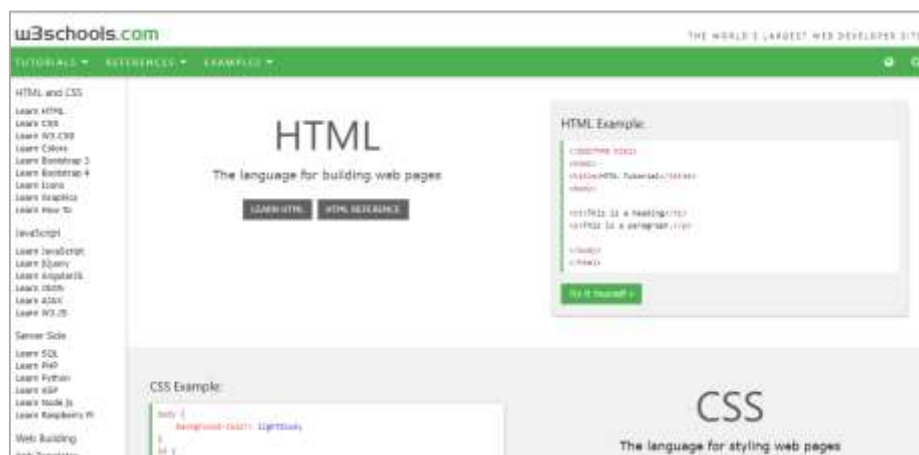


รูปที่ 3.11 ภาพ Raspberry Pi

ก่อนที่จะใช้ Raspberry Pi จะต้องมีระบบปฏิบัติการก่อน โดยระบบปฏิบัติการที่นิยมใช้กัน คือ ระบบปฏิบัติการ Raspbian เพราะเป็นระบบปฏิบัติการที่ถูกสนับสนุนโดยตรงจากทาง Raspberry Pi Foundation ซึ่ง Raspbian เป็นระบบปฏิบัติการที่อยู่ในตระกูลลินุกซ์ แต่นอกเหนือระบบปฏิบัติการ Raspbian แล้วยังมีระบบปฏิบัติการอื่น ๆ ให้เลือกใช้ได้ เช่น Noobs, Raspberry Pi Desktop (สำหรับ PC และ Mac), Third Party Operating System Images (Windows10 IoT Core, OSMC, RISC OS, Weather Station, PINET, Ubuntu Mate, Snappy Ubuntu Core) เป็นต้น [12] การควบคุมอุปกรณ์อิเล็กทรอนิกส์ จะใช้ GPIO (General Purpose Input-Output) ในการควบคุมซึ่งเป็นส่วนที่เอาไว้ต่อสัญญาณ Input/Output เข้ากับวงจรอิเล็กทรอนิกส์ภายนอกได้

### 3.5 มารู้จัก W3School [13]

เป็นแหล่งข้อมูลสำหรับนักพัฒนาเว็บไซต์ที่รวบรวมความรู้ในการพัฒนาเอาไว้ ซึ่ง W3School มีคำแนะนำและข้อมูลอ้างอิงหลากหลายภาษาที่ช่วยให้พัฒนาเว็บไซต์ได้สะดวกสบายขึ้น เช่น HTML, CSS, JavaScript, PHP, jQuery, W3.CSS และ Bootstrap เป็นต้น



รูปที่ 3.12 W3school

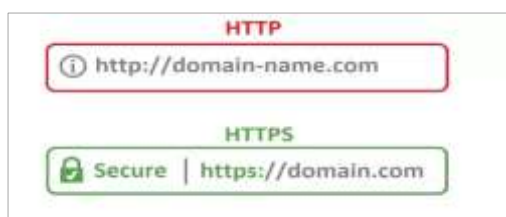
### 3.6 ความแตกต่างระหว่าง HTTP กับ HTTPS

HTTP (Hypertext Transfer Protocol) เป็นโพรโทคอลสื่อสารสำหรับการแลกเปลี่ยนสารสนเทศผ่านเครือข่ายอินเทอร์เน็ต โดยการทำงานหลัก ๆ จะใช้ในการรับส่งเอกสารในรูปแบบ World Wide Web (WWW) ไปยัง Web Browser เช่น Firefox, Google Chrome, Safari, Opera และ IE Microsoft Internet Explorer เป็นต้น เพื่อเรียกดูข้อมูลหรือเว็บเพจ โดยจะใช้โพรโทคอล HTTP ร้องขอไปยังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์จะส่งข้อมูลที่ต้องการมาให้ และแสดงผลออกทางหน้าจอ ในการแลกเปลี่ยนข้อมูลกันระหว่าง Server และ Client ของ World Wide Web (Server) จะส่งข้อมูลแบบ Clear text การส่งข้อมูลแบบนี้ ไม่ได้ทำการเข้ารหัส ทำให้สามารถถูกดักจับและอ่านข้อมูลได้ง่าย [14]

HTTPS (Hypertext Transfer Protocol Secure) เป็นโพรโทคอลสื่อสารสำหรับการแลกเปลี่ยนสารสนเทศผ่านเครือข่ายอินเทอร์เน็ตที่ช่วยรักษาความสมบูรณ์ถูกต้องของข้อมูลผู้ใช้และเก็บข้อมูลไว้เป็นความลับระหว่างคอมพิวเตอร์ของผู้ใช้กับเว็บไซต์ [15] โดยจะระบุถึงการเชื่อมต่อแบบปลอดภัย โดยในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต ข้อมูลที่ทำการส่งได้ถูกเข้ารหัสเอาไว้ โดยใช้ Asymmetric Algorithm ซึ่งถ้าถูกดักจับได้ก็ไม่สามารถที่จะอ่านข้อมูลนั้นได้รู้เรื่อง โดยข้อมูลนั้นจะสามารถอ่านได้เฉพาะ Client กับเครื่อง Server เท่านั้น นิยมใช้กับเว็บไซต์ที่ต้องการความปลอดภัยสูง เช่น เว็บไซต์ของธนาคาร ร้านค้าออนไลน์ เป็นต้น [14]

ข้อมูลที่ส่งด้วย HTTPS จะได้รับการรักษาความปลอดภัยผ่านโพรโทคอล Transport Layer Security (TLS) ซึ่งให้การปกป้องหลัก 3 ชั้นดังนี้

- 1) การเข้ารหัส หมายถึง การเข้ารหัสข้อมูลที่แลกเปลี่ยนเพื่อรักษาความปลอดภัยจากผู้ลักลอบดูข้อมูล ซึ่งหมายความว่าขณะที่ผู้ใช้เรียกดูเว็บไซต์ จะไม่มีใครสามารถฟังบทสนทนาเพื่อติดตามกิจกรรมของผู้ใช้ได้ หรือขโมยข้อมูลของผู้ใช้ได้
- 2) ความถูกต้องสมบูรณ์ของข้อมูล หมายถึง จะไม่สามารถแก้ไขหรือทำให้ข้อมูลเสียหายในช่วงที่ถ่ายโอนข้อมูล ไม่ว่าจะเจตนาหรือไม่ก็ตาม โดยที่ไม่มีการตรวจพบ
- 3) การตรวจสอบสิทธิ์ หมายถึง การพิสูจน์ว่าผู้ใช้สื่อสารกับเว็บไซต์ถูกต้อง และมีสิทธิ์ในการเข้าถึงข้อมูล โดยจะป้องกันการโจมตีจากบุคคลอื่นที่ไม่มีสิทธิ์เข้าถึงข้อมูล [15]

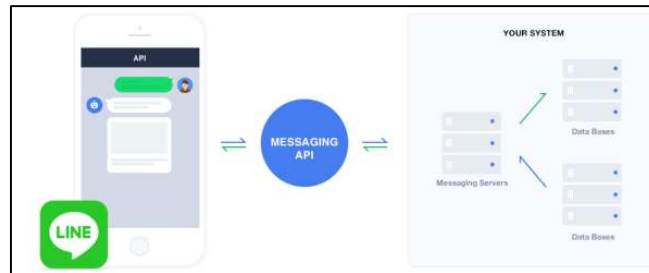


รูปที่ 3.13 ภาพความแตกต่างของ HTTP และ HTTPS

### 3.7 การทำงานของ Messaging API LINE

Messaging API จะเชื่อมต่อระหว่างผู้ใช้ ผ่านทาง LINE official account หรือ LINE@ account ซึ่ง Messaging API สามารถ Accept Friend รวมถึงส่ง Message หาผู้ใช้คนอื่นๆ ที่เป็นเพื่อนใน LINE ได้ โดยผ่านหน้า LINE@ Manager ที่ได้กำหนด การใช้งาน Messaging API ทำให้สามารถส่งข้อมูลระหว่าง Server ของเราไปยังผู้ใช้ LINE ผ่านทาง LINE Platform ซึ่ง Request ที่ใช้ส่งข้อมูลต้องอยู่ในรูป JSON format โดยตัว Server จะต้องเชื่อมต่อกับ LINE Platform และเมื่อมีผู้ใช้เพิ่ม Account LINE เป็นเพื่อน หรือ ส่งข้อความมาหาทาง LINE Platform จะทำการส่ง Request มายัง Server ที่

ได้ผูกไว้กับ LINE Account ที่จัดเตรียมไว้ทันที วิธีนี้เรียกว่า Webhook ซึ่งมันทำให้ผู้ใช้งานรู้สึกเหมือนกับว่าได้โต้ตอบกับคนจริงๆ [16] ดังรูปที่ 3.14



รูปที่ 3.14 ภาพอธิบายการรับส่งของบอทด้วย Messaging API

### 3.8 การรับและส่งข้อมูลจาก LINE Platform

การรับข้อมูลจาก LINE Platform จะรับจากข้อมูลจะถูกส่งมาที่ URL ของ Server เมื่อไรก็ตามที่ผู้ใช้ได้ทำการติดต่อกับ Account ที่กำหนดไว้ ซึ่งมีด้วยกัน 2 แบบ คือ

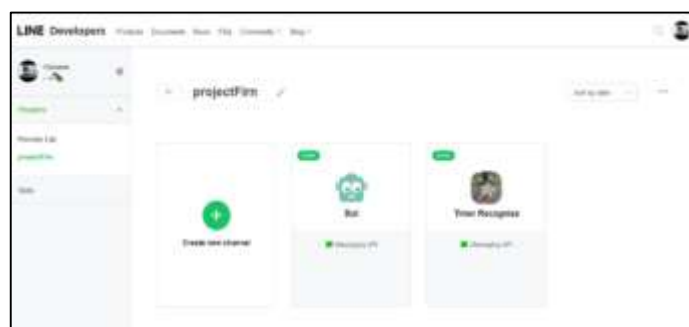
1. User ส่งข้อความ (message)
2. User มี action เช่น Add LINE ID ของเราเป็นเพื่อน (Operation)

ในการรับข้อมูลที่ส่งมาผ่านทาง HTTP ที่ได้ตั้ง URL ไว้กับ LINE โดยข้อมูลจะอยู่ในรูป JSON format และทุก ๆ Request จะมี Signature ใส่ไปในส่วน Header ซึ่ง Server ที่กำหนดไว้จะต้องทำการตรวจสอบ Signature ว่าถูกส่งมาจาก LINE Platform จริงๆ ไม่ใช่มีผู้ปลอมแปลงส่งมา

การส่งข้อมูลไปหา LINE Platform สามารถใช้ APIs ที่ทาง LINE Platform ให้ไว้ใช้สำหรับการส่งข้อมูลจาก Server เราไปหาผู้ใช้ได้ ซึ่งความสามารถที่ทำได้คือ

1. ส่งข้อความไปยัง User หรือใครก็ตามที่เพิ่มเราเป็นเพื่อนไว้
2. ดึงข้อมูลชื่อ Display ของ User

ซึ่ง server เราสามารถเรียก APIs เพื่อส่งข้อความ ไปหา User ได้ตลอดเวลา โดยจะต้องทำการตั้งค่า Channel access token (Channel ID), Channel secret และ Channel MID ซึ่ง APIs ทั้งหมดนี้ต้องใช้ผ่าน HTTPS ก่อน [17]



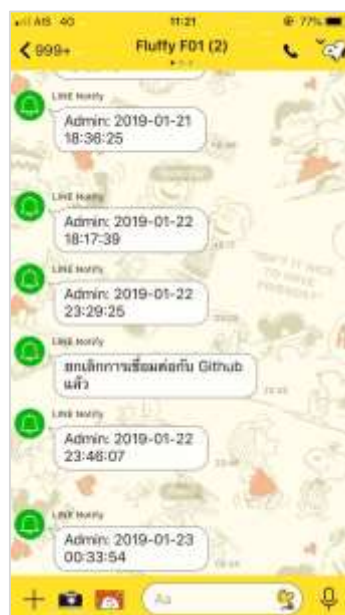
รูปที่ 3.15 ภาพการสร้างบอท

### 3.9 LINE Notify

เป็นการแจ้งเตือนผ่าน LINE [18] การใช้งานนั้นต้องมี LINE Account ก่อน แล้วจะสามารถสมัครใช้ LINE Notify ได้ การเรียกใช้งานเรียกโดย Add LINE Notify เป็นเพื่อนก่อน แล้วจะสามารถเชิญ LINE Notify ที่สร้างไว้เข้ากลุ่มที่กำหนดได้ จากนั้นออก Access Token (สำหรับผู้พัฒนา) โดยใส่ชื่อ Token เข้าไป ชื่ออะไรก็ได้ ส่วนห้องแชต นั้นเลือกได้ทั้งแบบโต้ตอบส่วนตัวกับเรา หรือ จะเลือกเป็นกลุ่ม LINE ก็ได้ นอกจากนี้หากเราไม่พอใจสามารถลบและสร้างใหม่ได้เรื่อย ๆ เรียกว่าเอาที่สบายใจเลยละกัน แต่ Access Token จะแสดงให้เห็นหลังจากออก Token เท่านั้น ดังนั้นจะต้อง Copy ไว้เลยเพราะมันจะไม่สามารถกลับมาเปิดที่หลังได้ [19]

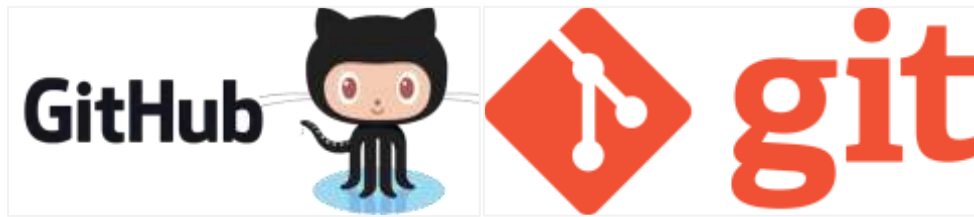


รูปที่ 3.16 ภาพกลุ่มของ LINE Notify



รูปที่ 3.17 ภาพการแจ้งเตือนมายังกลุ่ม Fluffy F01 ในLINE

### 3.10 ความรู้เบื้องต้นเกี่ยวกับ GitHub

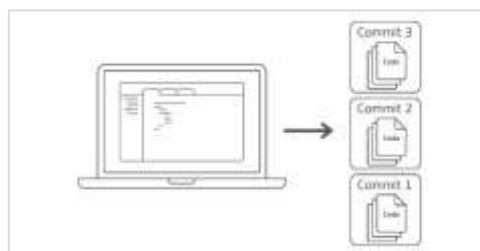


รูปที่ 3.18 ภาพ GitHub

GitHub คือ website Git (version control repository) ที่อยู่บน internet มีการทำงานแบบเดียวกับ Git เลย [20] ซึ่ง Developer จำนวนมากร่วมกันเขียน และปรับแต่งโค้ดร่วมกันได้โดย Git จะช่วยให้ code ที่ร่วมกันเขียนขึ้นมา ไม่ทับกันมั่วแน่นอน [21] การเข้าถึงข้อมูลและจัดการข้อมูลผ่าน web นั้นมีทั้งแบบเสียเงิน ไม่ต้องเสียเงิน และแค่ code project ทั้งหมดจะถูกแจกจ่ายให้คนอื่น ๆ สามารถเห็นได้ด้วย (public) ซึ่ง GitHub ก็มีการเสนอ plan แบบส่วนตัว (private) ให้ถ้าอยากให้ code ไม่ถูกแจกจ่ายออกไปโดยจะมีค่าใช้จ่ายตรงนี้ ปัจจุบันมีมากกว่า 20 ล้าน user รวมกันกว่า 60 ล้าน repository บนระบบแล้ว [22]

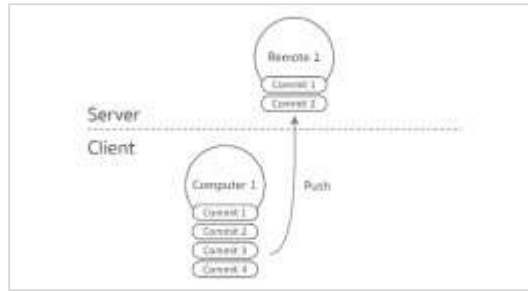
#### คำต่างๆที่ควรรู้เกี่ยวกับการใช้งาน Git

- 1) Repository คือ Folder ที่ใช้เก็บข้อมูลนั่นเอง อยากจะเก็บอะไรไว้ในนั้นก็กดเข้าไปได้เลย (ในความเป็นจริง 1 Repository สามารถเก็บ Project เท่าไรก็ได้ตามที่ต้องการ แต่ส่วนใหญ่นิยมเก็บ Project 1 ตัวต่อ 1 Repository)
- 2) Clone คือการ Copy Repository จาก Remote มาลงเครื่อง
- 3) Commit เวลาที่มีข้อมูลที่แก้ไขเสร็จแล้ว (โค้ดที่เขียนคำสั่งบางอย่างเสร็จแล้ว) แล้วอยากจะทำ Backup เก็บไว้ใน VCS จะเรียกกันว่า Commit ซึ่งการ Commit จะสามารถเลือกได้ว่าเอาไฟล์ไหนบ้าง (ไม่จำเป็นต้องเลือกทุกไฟล์)



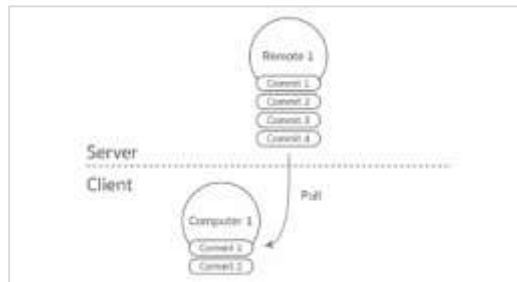
รูปที่ 3.19 ภาพอธิบายการ Commit ของ GitHub

- 4) Unstaged เวลาเราแก้ไขโค้ดหรือแก้ไขข้อมูล ไฟล์ที่ถูกแก้ไขจะอยู่ในสถานะ Unstaged
- 5) Staged ไฟล์ที่ถูกแก้ไขเสร็จเรียบร้อยแล้ว อยู่ในสถานะ Staged ก่อนถึงจะทำการ Commit ได้
- 6) Push เวลาที่มี Commit อยู่ในเครื่องและต้องการจะ Sync ขึ้นไปเก็บไว้ใน Remote จะเรียกขั้นตอนนี้ว่า Push



รูปที่ 3.20 ภาพอธิบายการ Push ของ GitHub

- 7) Pull เป็นการดึงข้อมูล Commit ใหม่ ๆ ที่อยู่ใน Repository ลงมาเก็บไว้ในเครื่อง



รูปที่ 3.21 ภาพการ Pull ของ GitHub

- 8) Fetch ในบางครั้งเราอาจจะไม่ต้องการ Pull ข้อมูลลงมาเก็บไว้ในเครื่องทันที แค่อยากเช็คสถานะของ Remote เฉย ๆ ว่ามีใคร Push ข้อมูลใหม่ขึ้นไป Remote
- 9) Conflict ในขณะที่กำลังจะ Merge อยู่ นั้น จะเห็นว่าโค้ดของนาย A และนาย B มันชนกัน ไปแก้โค้ดที่เดียวกัน ดังนั้น Git จะแจ้งว่าเกิด Conflict หรือก็คือโค้ดที่ทับซ้อนกันนั่นเอง
- 10) Merge Commit สมมติว่านาย A กับนาย B เขียนโค้ดด้วยกันอยู่ และทั้งคู่ก็เขียนโค้ดที่อยู่ในไฟล์เดียวกัน แล้วใครคนใดคนหนึ่ง push ขึ้น Remote ก่อน อีกคนก็จะไม่สามารถ push ได้ ดังนั้นก่อนที่จะ Push ของตัวเองขึ้นไปได้ ต้อง Pull จาก Remote ลงมาใหม่ก่อนเพื่ออัปเดต Commit แล้ว Push ขึ้นไป ซึ่งเราเรียกขั้นตอนนี้ว่า Merge Commit [23]





## บทที่ 4 ขั้นตอนและแผนการดำเนินงาน

ในบทนี้จะแสดงถึงขั้นตอนการดำเนินงาน แสดงรายละเอียดในแต่ละขั้นตอน รวมไปถึงการแสดงผลแผนการดำเนินงานทั้งหมด และเวลาที่ใช้ในแต่ละขั้นตอนการดำเนินงาน

### 4.1 รวบรวมข้อมูลและกำหนดขอบเขต

รวบรวมข้อมูล ผลงานที่เกี่ยวข้อง ทฤษฎีที่เกี่ยวข้อง เพื่อนำมาวิเคราะห์และกำหนดขอบเขตของโครงการ

### 4.2 ศึกษาทฤษฎีและความรู้ที่เกี่ยวข้อง

ศึกษาทฤษฎีและความรู้ที่เกี่ยวข้อง เพื่อนำความรู้ที่ได้มาประยุกต์ใช้กับโครงการ

### 4.3 ออกแบบและพัฒนาหน้า Web Application

ออกแบบและพัฒนาหน้า Web Application ในส่วนที่ติดต่อกับผู้ใช้ (User Interface) รวมทั้งออกแบบฐานข้อมูลสำหรับจัดเก็บข้อมูลสมาชิกของเจ้าหน้าที่ภาควิชาคอมพิวเตอร์

### 4.4 พัฒนาระบบ สร้างฐานข้อมูลในการจัดเก็บข้อมูลสมาชิก สร้าง Web Application

พัฒนาระบบ และสร้างฐานข้อมูลที่จัดเก็บข้อมูลสมาชิกของเจ้าหน้าที่ภาควิชาคอมพิวเตอร์

### 4.5 ทดสอบระบบ

ทดสอบระบบการทำงานของอุปกรณ์ว่าสามารถทำงานเป็นไปตามเป้าหมายหรือไม่ และปรับปรุงแก้ไขให้เป็นไปตามเป้าหมาย

### 4.6 จัดทำเอกสารและคู่มือประกอบโครงการ

จัดทำเอกสารประกอบโครงการ ตั้งแต่เริ่มต้นจนถึงขั้นตอนสุดท้ายของโครงการ และจัดทำคู่มือการใช้งานของระบบ

#### 4.7 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	เดือน									
	ส.ค. 2561	ก.ย. 2561	ต.ค. 2561	พ.ย. 2561	ธ.ค. 2561	ม.ค. 2562	ก.พ. 2562	มี.ค. 2562	เม.ย. 2562	พ.ค. 2562
1. รวบรวมข้อมูลและกำหนดขอบเขต	↕		↗							
2. ศึกษาทฤษฎีและความรู้ที่เกี่ยวข้อง	↘				↗					
3. ออกแบบและพัฒนาหน้า Web Application			↘			↗				
4. พัฒนาระบบ สร้างฐานข้อมูลจัดเก็บข้อมูลสมาชิก					↘		↗			
5. ทดสอบระบบ									↕	
6. จัดทำเอกสารและคู่มือประกอบโครงการ		↘								↗

รูปที่ 4.1 ภาพแผนดำเนินงาน

## บทที่ 5 การวิเคราะห์และการออกแบบ

### 5.1 โครงสร้างหลักของระบบงาน (Process Hierarchy)

ระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพแบ่งออกเป็น 2 ส่วนดังนี้

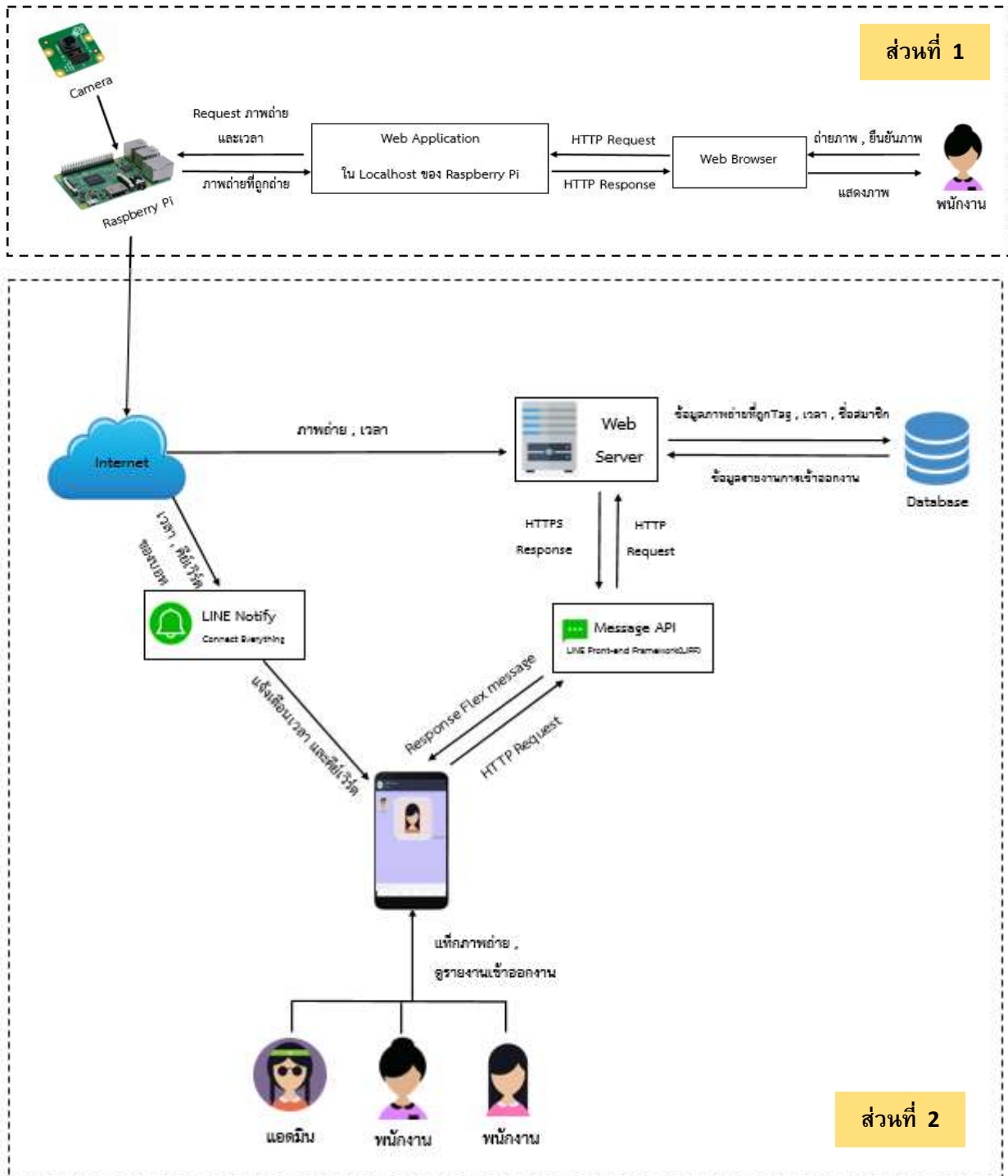
#### 5.1.1 การทำงานของผู้ใช้งานกับตัวอุปกรณ์

เป็นส่วนที่ผู้ใช้งานติดต่อกับตัวอุปกรณ์เพื่อทำการสั่งให้ตัวอุปกรณ์ถ่ายภาพของผ่านหน้าเว็บแอปพลิเคชันโดย

- 1) ก่อนที่จะสั่งให้ตัวอุปกรณ์ถ่ายภาพ ผู้ใช้จะต้องเลือกช่วงเช้างานหรือออกงานก่อนโดยการกดปุ่มที่เว็บแอปพลิเคชัน
- 2) การถ่ายภาพสามารถถ่ายใหม่ได้หลายครั้ง
- 3) หลังจากถ่ายภาพแล้วจะแสดงภาพถ่ายที่ถูกถ่ายก่อนทำการส่ง
- 4) เมื่อผู้ใช้นั้นการส่งภาพถ่ายแล้ว ภาพถ่ายจะถูกบันทึกในฐานข้อมูลสำรองและถูกส่งไปแจ้งเตือนลงในไลน์กลุ่มของผู้ใช้งาน
- 5) เมื่อส่งรูปภาพเรียบร้อยแล้ว จะมีข้อความแสดงทางหน้าจอของตัวอุปกรณ์ว่าได้ทำการส่งสำเร็จแล้ว ก่อนกลับไปยังหน้าหลัก

#### 5.1.2 การทำงานของบอทและระบบการจัดการแท็กภาพถ่ายก่อนจัดเก็บลงฐานข้อมูล

- 1) ในไลน์กลุ่มจะมีบอทที่คอยรับคำสั่งงานจากผู้ใช้งาน โดยผู้ใช้งานจะต้องพิมพ์คีย์เวิร์ดลงในไลน์กลุ่มเพื่อคุยกับบอท จากนั้นบอทจะตอบกลับมาเป็น Flex Message ซึ่งคีย์เวิร์ดที่คุยกับบอทมีดังนี้
  - คีย์เวิร์ดสำหรับแท็กเจ้าของภาพถ่าย (แท็ก, แท็กภาพถ่าย, T, t, Tag, tag)
  - คีย์เวิร์ดสำหรับดูประวัติการทำงาน (รายงาน, เช็คเวลาเข้าออกงาน, R, r, Report, report)
  - คีย์เวิร์ดตัวช่วย (ตัวช่วย, All, all, Help, help)
- 2) เมื่อผู้ใช้ทำการกดปุ่มแท็กภาพถ่ายที่บอทตอบกลับมาแล้ว จะมีหน้าเว็บแอปพลิเคชันแสดงขึ้นมาภายในไลน์
- 3) ผู้ใช้จะต้องทำการล็อกอินเข้าสู่ระบบก่อนที่จะทำกระบวนการแท็กเจ้าของภาพถ่าย และกระบวนการเรียกดูประวัติการทำงาน



รูปที่ 5.1 ภาพสถาปัตยกรรมของระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ

จากรูปที่ 5.1 เป็นสถาปัตยกรรมของระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ โดยจากภาพจะเห็นเป็น 2 ส่วน โดยอธิบายหลักการทำงานดังนี้

1. ในส่วนแรกเป็นการทำงานของตัวอุปกรณ์เมื่อได้รับคำสั่งจากผู้ใช้งาน (เจ้าหน้าที่ภาควิชาคอมพิวเตอร์) โดยจะใช้ Raspberry Pi ในการประมวลผลคำสั่ง ซึ่งจะเชื่อมต่อกับกล้อง Raspberry Pi Camera เพื่อใช้ในการถ่ายภาพของผู้ใช้ และใช้ Raspberry Pi 3 7" Touchscreen Display ซึ่งเป็นหน้าจอที่ใช้สำหรับรับคำสั่งและแสดงผลให้กับผู้ใช้ โดยจะมีกระบวนการทำงานดังนี้



รูปที่ 5.2 อุปกรณ์ Raspberry Pi ต่อกับกล้อง



รูปที่ 5.3 ภาพจอ Touch Screen

- ผู้ใช้งานได้สั่งให้ตัวอุปกรณ์ถ่ายภาพ เพื่อบันทึกเวลาทำงานและออกงานในแต่ละวันโดยผ่านทาง Web Application ที่ถูกพัฒนาขึ้นแบบ Localhost ใน Raspberry Pi ทางหน้าจอ Touchscreen ผ่าน Web Browser



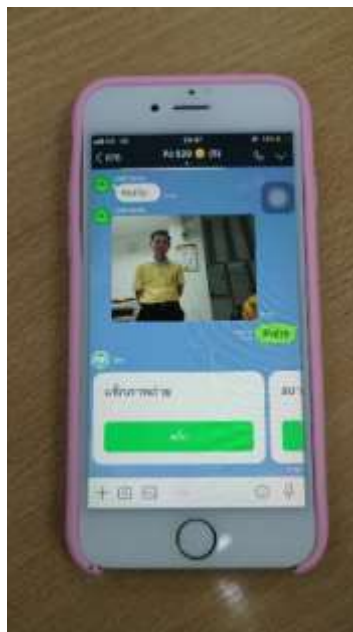
รูปที่ 5.4 ภาพหน้าเว็บการเข้าออกงานบนหน้าจอ Touch Screen ที่ต่อกับ Raspberry Pi3

- เมื่อผู้ใช้ถ่ายภาพแล้ว ได้ทำการกดยืนยันถ่ายภาพ ภาพถ่ายและเวลาจะถูกส่งไปแจ้งเตือนในไลน์กลุ่ม และถูกบันทึกลงในฐานข้อมูลสำรอง

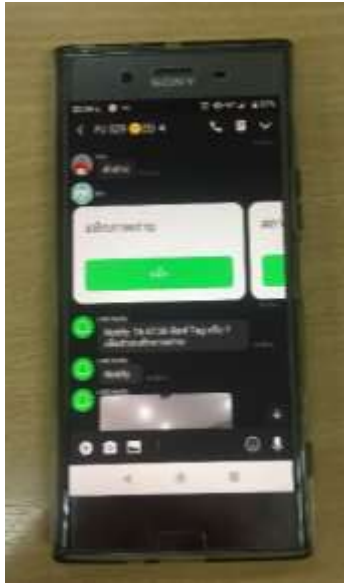


รูปที่ 5.5 ภาพการแจ้งเตือนภาพถ่ายและเวลา มาแสดงในกลุ่มไลน์

2. ในส่วนที่สอง เป็นกระบวนการต่อจากส่วนแรกโดยตัวอุปกรณ์ Raspberry Pi ได้เชื่อมต่อกับเครือข่าย Internet ส่งภาพถ่ายและเวลาไปเก็บไว้ที่ฐานข้อมูลสำรอง และใช้ Message API ของ LINE ในการแจ้งเตือนยังผู้ใช้งาน และใช้บอทเป็นตัวกลางในการสื่อสารกับผู้ใช้ โดยกระบวนการทำงานของบอทที่ได้ทำการสร้างไว้มีดังนี้
  - บอทจะคอยรับคีย์เวิร์ดจากผู้ใช้งานและทำการตอบกลับกับผู้ใช้



รูปที่ 5.6 ภาพการคุ้ยโต้ตอบของผู้ใช้กับบอทบนมือถือ iPhone



รูปที่ 5.7 ภาพการคุยโต้ตอบของผู้ใช้กับบอทบนมือถือ Sony

- บอทเป็นตัวที่เชื่อมโยงไปยังหน้าเว็บแอปพลิเคชันที่ได้พัฒนาบน Server ผ่าน LINE LIFF แสดงหน้าเว็บแอปพลิเคชันในไลน์



รูปที่ 5.8 เป็นภาพหน้าเว็บที่มีภาพถ่ายที่ยังไม่ได้แท็กเจ้าของภาพจากตัวอุปกรณ์



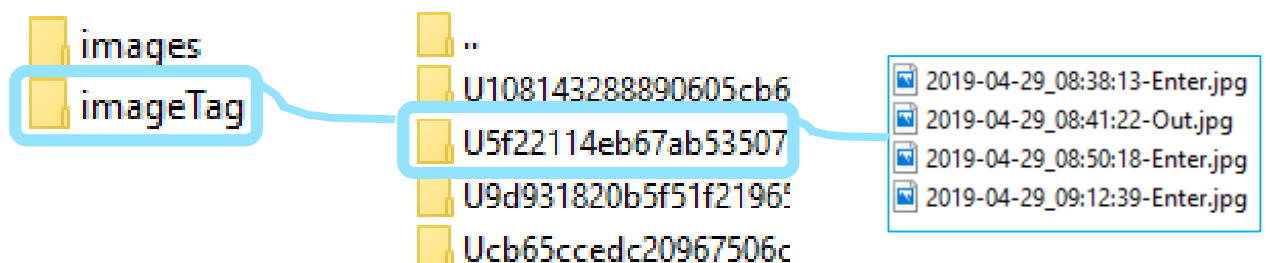


รูปที่ 5.9 เป็นภาพหน้าเว็บที่มีชื่อเล่นของผู้ใช้งานสำหรับแท็กเจ้าของภาพถ่าย



รูปที่ 5.10 เป็นภาพหน้าเว็บให้ผู้ใช้งานยืนยันเพื่อแท็กเจ้าของภาพถ่าย

3. เมื่อผู้ใช้ทำการแท็กเจ้าของภาพถ่ายแล้ว ภาพถ่าย เวลาที่ถ่ายภาพ ไอดีผู้ใช้งานไลน์ของเจ้าของภาพ และคนแท็กภาพถ่าย จะถูกบันทึกลงฐานข้อมูลหลัก



รูปที่ 5.11 ภาพการเก็บภาพถ่ายลงโฟลเดอร์ของภาพถ่าย

image	userID_Owner	dateEnter	timeEnter	who
2019-01-30_12_44_21-Enter.jpg	U5f22114eb67a [REDACTED]	2019-01-30	12:44:21	U108143288890 [REDACTED]
2019-03-20_08_28_33-Enter.jpg	U108143288890 [REDACTED]	2019-03-20	08:28:33	U5f22114eb67a [REDACTED]

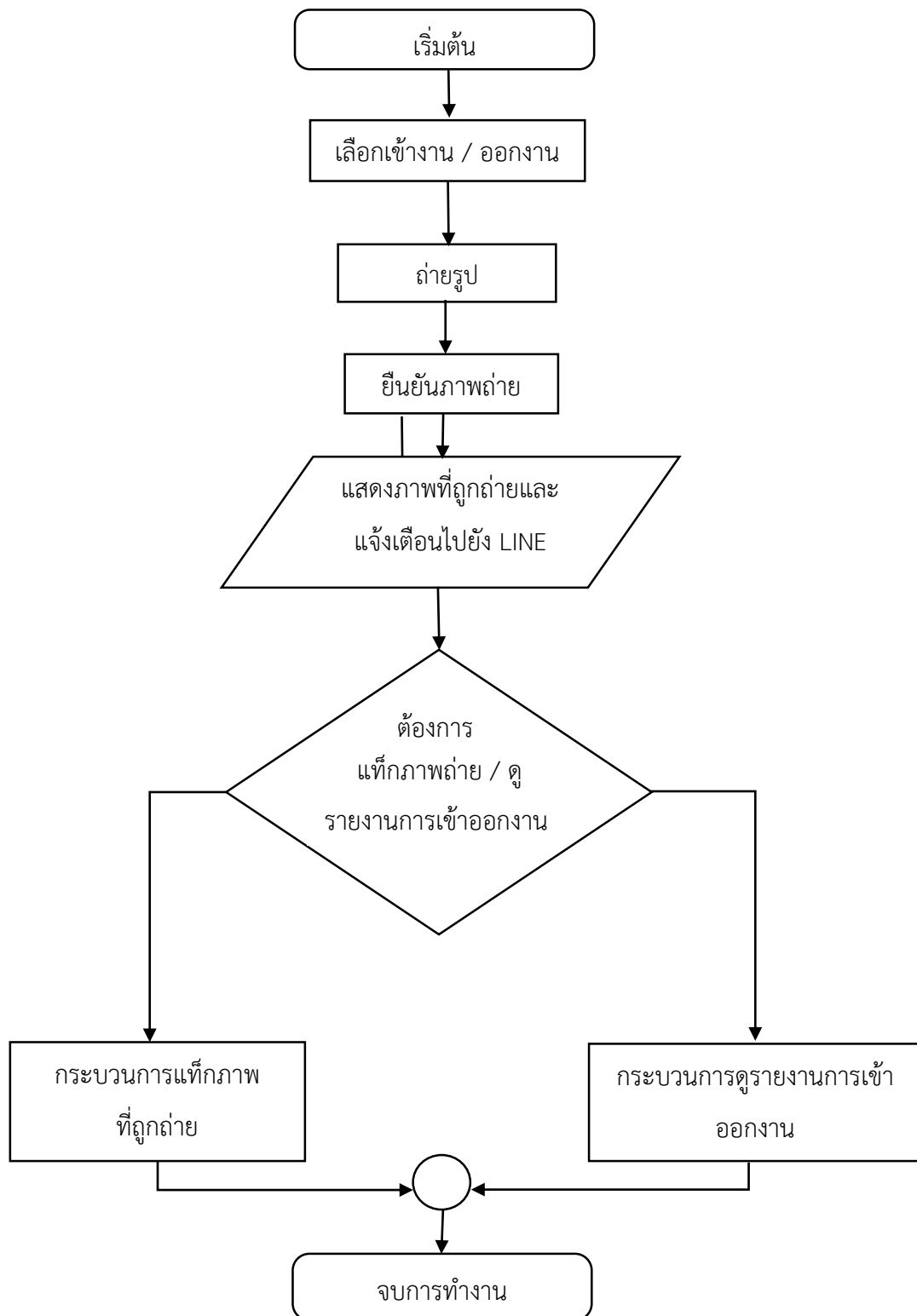
รูปที่ 5.12 ภาพการบันทึกข้อมูลการแท็กภาพถ่ายขาเข้าลงฐานข้อมูล

image	userID_Owner	dateOut	timeOut	who
2019-03-20_16_47_36-Out.jpg	U36a63feb7e031 [REDACTED]	2019-03-20	16:47:36	U5f22114eb67a [REDACTED]
2019-03-20_17_10_08-Out.jpg	Ub43a06e8a9d84 [REDACTED]	2019-03-20	17:10:08	U5f22114eb67a [REDACTED]

รูปที่ 5.13 ภาพการบันทึกข้อมูลการแท็กภาพถ่ายขาออกลงฐานข้อมูล

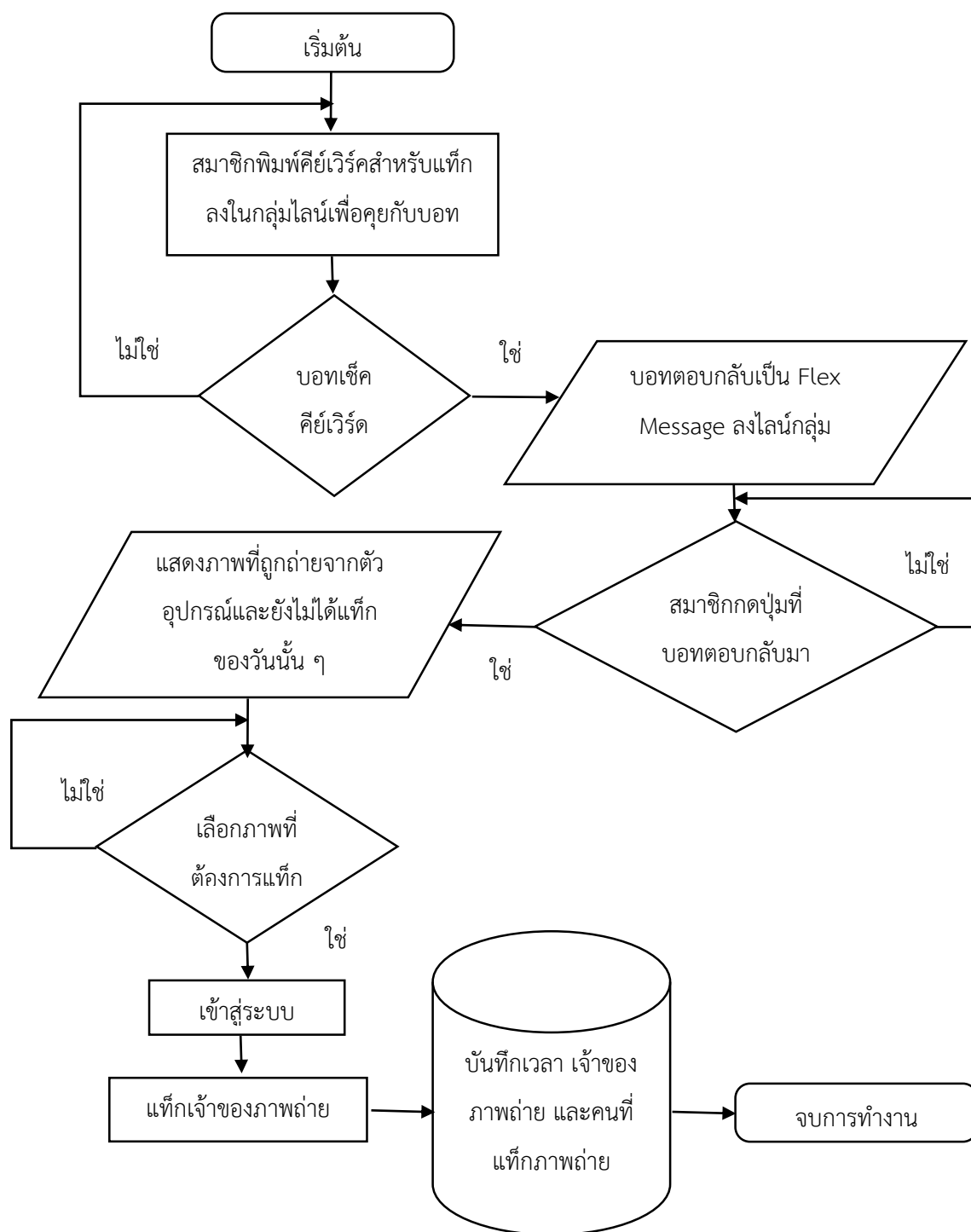
## 5.2 ผังงาน (Flow Chart)

### 5.2.1 ผังการทำงานโดยรวมของระบบ



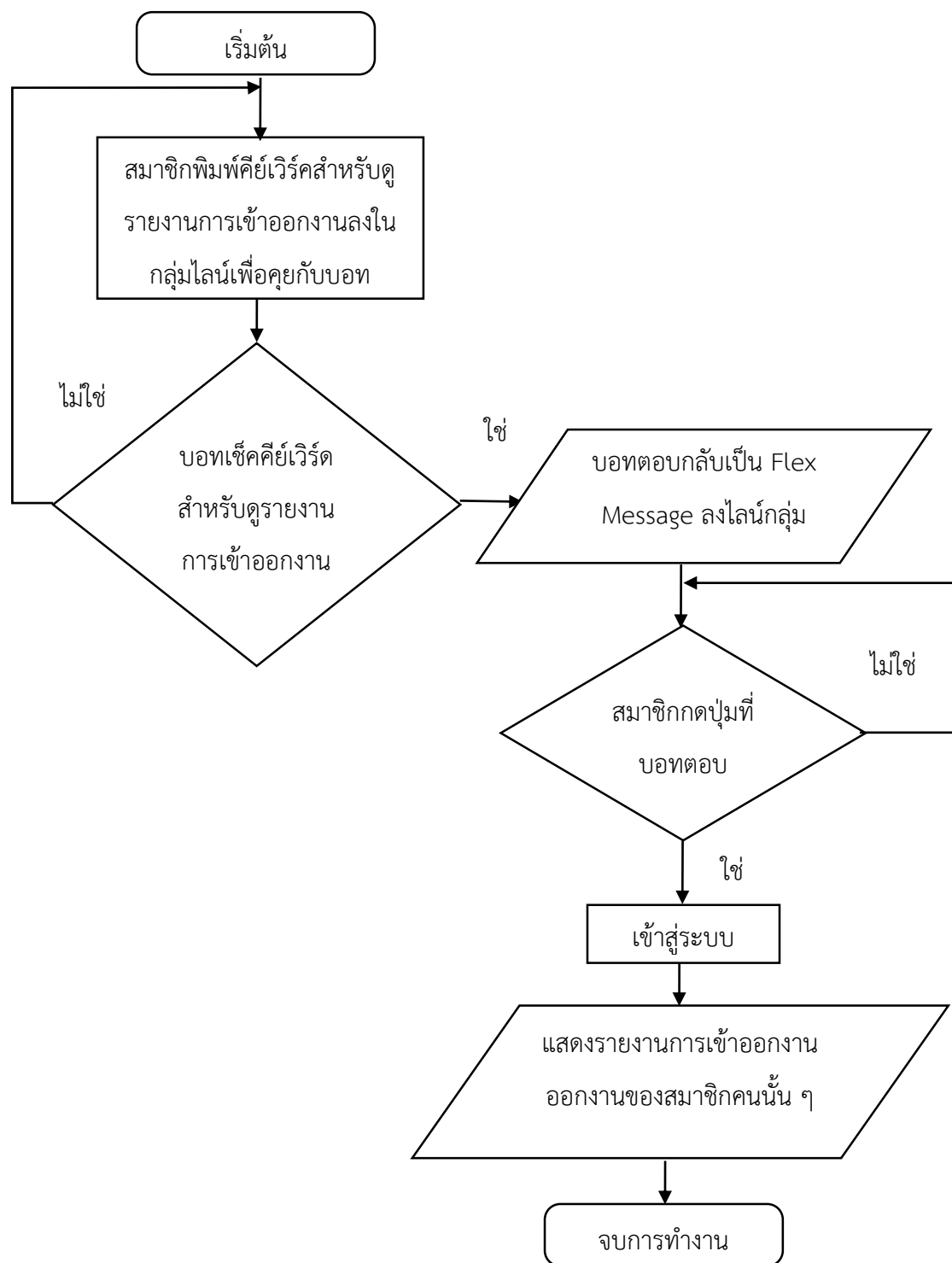
รูปที่ 5.14 ภาพ Flow Chart ภาพรวมของระบบ

## 5.2.2 ฟังก์ชันการทำงานของผู้ใช้กับบอทในการจัดการกระบวนการแท็กภาพที่ถูกถ่าย



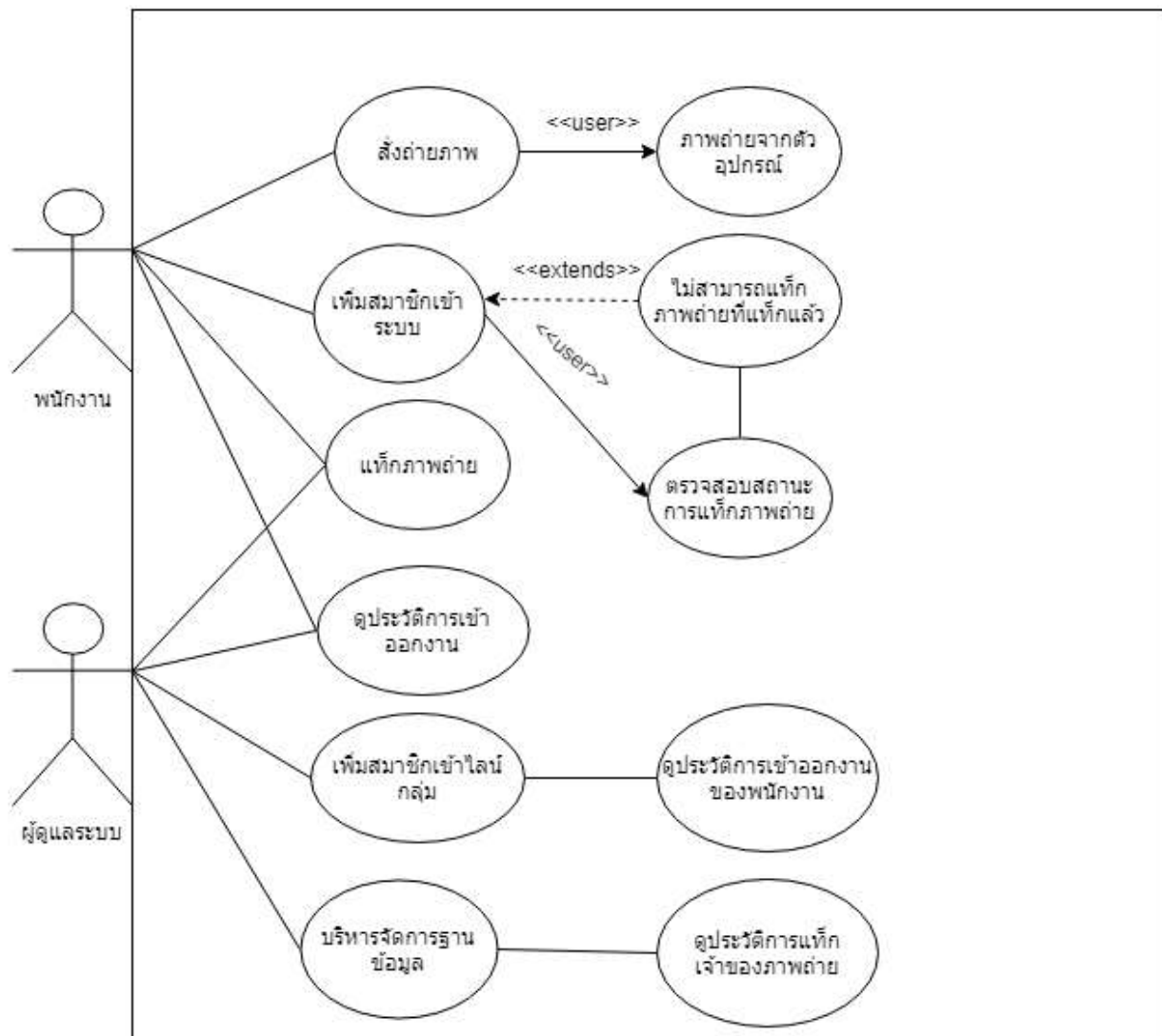
รูปที่ 5.15 ภาพ Flow Chart กระบวนการแท็กภาพถ่ายของผู้ใช้กับบอท

### 5.2.3 ฟังก์ชันการทำงานเมื่อผู้ใช้ต้องการดูรายงานการเข้างานออกงาน



รูปที่ 5.16 ภาพ Flow Chart กระบวนการดูรายงานการเข้างานออกงานของผู้ใช้กับบอท

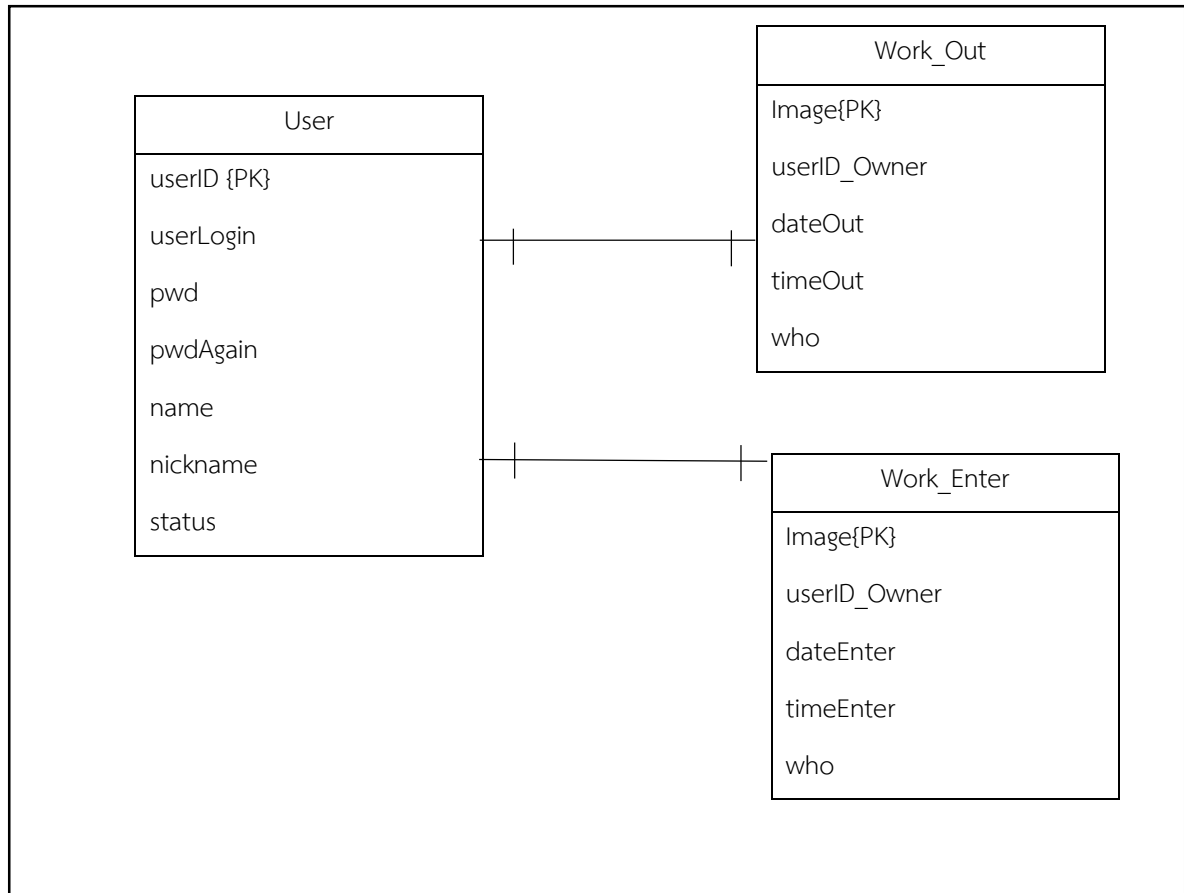
### 5.3 Use case Diagram



รูปที่ 5.17 ภาพ Use case Diagram ของระบบ

## 5.4 การออกแบบฐานข้อมูลและ Data Dictionary

### 5.4.1 ไตอะแกรมแสดงความสัมพันธ์ระหว่างผู้ใช้งานกับช่วงเวลาขาเข้าและขาออกงาน



รูปที่ 5.18 ภาพความสัมพันธ์ของผู้ใช้งานกับช่วงเวลาขาเข้างานและขาออกงาน

## 5.4.2 Data Dictionary

### 1) User

ตารางที่ 5.1 ผู้ใช้งาน

Field Name	Field Type	Description
userID {PK}	char(33)	ไอดีผู้ใช้งานLINE
userLogin	varchar(50)	ชื่อผู้ใช้งาน
pwd	varchar(15)	รหัสผ่าน
pwdAgain	varchar(15)	รหัสผ่าน
name	varchar(50)	ชื่อและนามสกุล
nickname	varchar(20)	ชื่อเล่น
status	varchar(5)	สถานะผู้ใช้งาน

### 2) Work\_Enter

ตารางที่ 5.2 บันทึกเวลาทำงานขาเข้า

Field Name	Field Type	Description
Image{PK}	varchar(30)	ชื่อภาพถ่าย
userID_Owner	char(33)	ไอดีผู้ใช้งานLINEของเจ้าของภาพถ่าย
dateEnter	date	วันที่
timeEnter	time	เวลา
who	char(33)	ไอดีผู้ใช้งานLINEของคนแท็กภาพถ่าย

### 3) Work\_Out

ตารางที่ 5.3 บันทึกเวลาทำงานขาออก

Field Name	Field Type	Description
Image{PK}	varchar(30)	ชื่อภาพถ่าย
userID_Owner	char(33)	ไอดีผู้ใช้งานLINEของเจ้าของภาพถ่าย
dateOut	date	วันที่
timeOut	time	เวลา
who	char(33)	ไอดีผู้ใช้งานLINEของคนแท็กภาพถ่าย





## บทที่ 6 ผลการดำเนินงาน ข้อจำกัด และข้อเสนอแนะ

### 6.1 ผลการดำเนินงาน

#### 6.1.1 การทำงานของเว็บแอปพลิเคชันของตัวอุปกรณ์

- อุปกรณ์สามารถรับคำสั่งถ่ายภาพผ่านหน้าเว็บแอปพลิเคชันจากผู้ใช้งานได้อย่างถูกต้อง
- อุปกรณ์สามารถบันทึกเวลาขณะที่ผู้ใช้งานได้ถ่ายภาพของตนเองได้อย่างถูกต้อง
- หน้าเว็บแอปพลิเคชันใช้งานง่าย ไม่มีตัวเลือกให้สับสน
- หน้าเว็บแอปพลิเคชันสามารถแสดงเวลา ณ ปัจจุบันได้อย่างถูกต้อง
- หน้าเว็บแอปพลิเคชันสามารถแสดงวิดีโอสตริมให้ผู้ใช้งานเห็นตนเองก่อนทำการกดถ่ายภาพได้
- เมื่อผู้ใช้กดถ่ายภาพจากหน้าเว็บแอปพลิเคชัน หน้าเว็บสามารถแสดงภาพถ่ายของผู้ใช้งานได้ถูกต้อง
- ผู้ใช้งานสามารถทำการถ่ายภาพใหม่อีกครั้งได้
- เมื่อผู้ใช้งานถ่ายภาพและกดยืนยันภาพถ่ายจากหน้าเว็บแอปพลิเคชัน เวลาขณะที่ผู้ใช้ถ่ายภาพและภาพถ่ายจะถูกบันทึกลงฐานข้อมูล และแจ้งเตือนไปยังไลน์กลุ่มของผู้ใช้งานได้

#### 6.1.2 การทำงานของเว็บแอปพลิเคชันบนไลน์กลุ่ม

- ผู้ใช้งานสามารถพิมพ์คีย์เวิร์ดเพื่อแท็กภาพถ่ายและดูรายงานประวัติการเข้าออกงานได้
- บอทสามารถตอบกลับตามคีย์เวิร์ดที่ผู้ใช้พิมพ์ลงในไลน์กลุ่ม
- บอทสามารถส่งค่า UserID LINE จาก Log LINE ไปยังหน้าเว็บแอปพลิเคชันได้
- หน้าเว็บแอปพลิเคชันง่ายต่อการใช้งาน ไม่ทำให้ผู้ใช้งานสับสน
- ผู้ใช้งานสามารถแท็กเจ้าของภาพถ่ายและสามารถเรียกดูประวัติการเข้างานออกงานได้

### 6.2 ข้อจำกัด

- ตัวอุปกรณ์ต้องมีไฟเลี้ยงตลอดเวลา
- ตัวอุปกรณ์จะต้องเชื่อมต่อกับอินเทอร์เน็ตตลอดเวลา
- ผู้ใช้จะต้องเพิ่มตัวเองเข้าสู่ระบบครั้งแรกเมื่อถูกเพิ่มเข้าไลน์กลุ่ม
- จะต้องมีการเข้าสู่ระบบก่อนผู้ใช้ทำการแท็กเจ้าของภาพถ่ายหรือดูรายงานประวัติการเข้าออกงาน

### 6.3 ข้อเสนอแนะ

- ระบบควรมีการตรวจสอบชื่อเล่นที่ซ้ำกัน หรือใช้ชื่อจริงแทนชื่อเล่น ทำให้แท็กเจ้าของได้ถูกต้อง
- พัฒนาต่อยอดให้ระบบสามารถตรวจจับหน้าคนได้ก่อนถ่ายภาพ เพื่อให้ผู้ใช้งานถ่ายภาพได้อย่างถูกต้องตามรูปแบบที่กำหนด
- พัฒนาให้ตัวอุปกรณ์รู้จำใบหน้าผู้ใช้งานได้ โดยใช้ข้อมูลภาพถ่ายที่เก็บอยู่ในฐานข้อมูล



## บรรณานุกรม

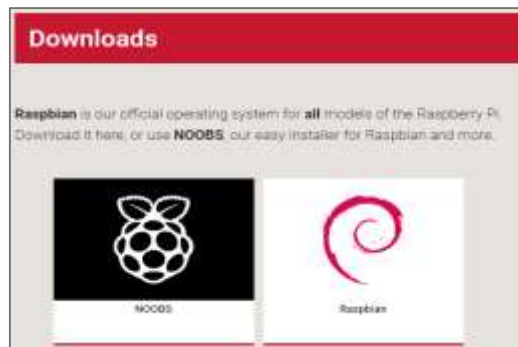
- [1] Kewin Rueangraklikhit, 2561, "การนับคนเข้าออกด้วย Raspberry Pi", [ออนไลน์], เข้าถึงได้: <https://medium.com/@kewinrueangraklikhit/การนับคน-เข้า-ออก-ประตู-ด้วย-raspberry-pi-7903c9478c50> (สืบค้นวันที่ 14 กันยายน 2561)
- [2] OneDee.io, 2561, "ระบบลงเวลาทำงานพนักงาน (OneDee.io)", [ออนไลน์], เข้าถึงได้: <https://www.onedee.io/onedee-โปรแกรมลงเวลาพนักงาน> (สืบค้นวันที่ 14 ตุลาคม 2561)
- [3] อินเทลส์ไปร์ จำกัด, 2561, "TimeMint App ลงเวลาพนักงานด้วยมือถือ", [ออนไลน์], เข้าถึงได้: <https://www.timehint.co/> (สืบค้นวันที่ 14 ตุลาคม 2561)
- [4] ดร. เกตุจำ, 2561, "แนวคิด Internet of Things", [ออนไลน์], เข้าถึงได้: [http://203.155.220.230/bmainfo/data\\_DDS/document/internet-of-things.pdf](http://203.155.220.230/bmainfo/data_DDS/document/internet-of-things.pdf) (สืบค้นวันที่ 15 กันยายน 2561)
- [5] blog SoGoodWeb, 2560, "กลุ่มของ Internet of Things ที่แบ่งตามตลาด", [ออนไลน์], เข้าถึงได้: <https://blog.sogoodweb.com/Article/Detail/59554> (สืบค้นวันที่ 15 ตุลาคม 2561)
- [6] อ. ไวรรรยา, 2560, "ความรู้เบื้องต้นเกี่ยวกับ API", นครปฐม, ภาควิชาคอมพิวเตอร์มหาวิทยาลัยศิลปากรวิทยาเขตพระราชวังสนามจันทร์ (สืบค้นวันที่ 14 ตุลาคม 2561)
- [7] mindphp, 2560, "API คืออะไร", [ออนไลน์], เข้าถึงได้: <http://mindphp.com/คู่มือ/73-คืออะไร/2038-api-คืออะไร.html> (สืบค้นวันที่ 14 ตุลาคม 2561)
- [8] LINE Developers, 2561, "มารู้จักกับ Line Front-end Framework (LIFF)", [ออนไลน์], เข้าถึงได้: <https://developers.line.me/en/docs/liff/overview/> (สืบค้นวันที่ 15 กันยายน 2561)
- [9] Warit Wanwithu, 2561, "มารู้จักกับ LIFF—สิ่งที่จะมาเติมเต็มในการทำ Chat Bot กัน", [ออนไลน์], เข้าถึงได้: <https://medium.com/lindevth/introduction-to-liff-7d708e2f42ec>. (สืบค้นวันที่ 15 กันยายน 2561)
- [10] ArduinosPro, 2561, "Raspberry Pi คืออะไร", [ออนไลน์], เข้าถึงได้: <https://www.arduinopro.com/article/51/raspberry-pi-tutorial-ep1-raspberry-pi-คืออะไร> (สืบค้นวันที่ 15 ตุลาคม 2561)
- [11] PoundXI, 2560, "Raspberry Pi ทำอะไรได้บ้าง", [ออนไลน์], เข้าถึงได้: <https://poundxi.com/raspberry-pi-คืออะไร> (สืบค้นวันที่ 15 ตุลาคม 2561).
- [12] RaspberryPi, 2561, "ระบบปฏิบัติการของ Raspberry Pi", [ออนไลน์], เข้าถึงได้: <https://www.raspberrypi.org/downloads/> (สืบค้นวันที่ 16 ตุลาคม 2561).
- [13] W3School, 2561, "มารู้จัก W3School", [ออนไลน์], เข้าถึงได้: <https://www.w3schools.com/about/> (สืบค้นวันที่ 15 ตุลาคม 2561).
- [14] mindphp, 2560, "HTTP HTTPS คืออะไร โพรโทคอล แลกเปลี่ยนข้อมูล", [ออนไลน์], เข้าถึงได้: <http://mindphp.com/คู่มือ/73-คืออะไร/2046-http-https-คืออะไร.html> (สืบค้นวันที่ 16 ตุลาคม 2561).
- [15] 2561, "รักษาเว็บไซต์ของคุณให้ปลอดภัยด้วย HTTPS", [ออนไลน์], เข้าถึงได้: <https://support.google.com/webmasters/answer/6073543?hl=th> (สืบค้นวันที่ 16 ตุลาคม 2561).

- [16] Saixiii, 2561, "การทำงานของ Messaging API (LINE API)", [ออนไลน์], เข้าถึงได้: <https://saixiii.com/chapter2-line-api-official/> (สืบค้นวันที่ 15 ตุลาคม 2561).
- [17] Saixiii, 2561, "การรับและส่งข้อมูลจาก LINE Platform", [ออนไลน์], เข้าถึงได้: <https://saixiii.com/chapter2-line-api-official/> (สืบค้นวันที่ 15 ตุลาคม 2561).
- [18] LINE Notify, 2562, "LINE Notify", [ออนไลน์], เข้าถึงได้: <https://notify-bot.line.me/doc/en/> (สืบค้นวันที่ 02 มกราคม 2562).
- [19] Piyanat Nimkhuntod, 2560, "ทำ Line Notify กันใหม่", [ออนไลน์], เข้าถึงได้: <https://medium.com/dev2pro/ทำ-line-notify-กันใหม่-4a2a7d62ac05> (สืบค้นวันที่ 20 มกราคม 2562).
- [20] Saixiii, 2562, "Github คืออะไร เป็น Git host ที่เก็บข้อมูลการเปลี่ยนแปลงของ file", [ออนไลน์], เข้าถึงได้: <https://saixiii.com/what-is-github/> (สืบค้นวันที่ 20 มกราคม 2562).
- [21] Saixiii, 2562, "Git คืออะไร Git คือ ศูนย์จัดเก็บข้อมูลการเปลี่ยนแปลงของ file", [ออนไลน์], เข้าถึงได้: <https://saixiii.com/what-is-git/> (สืบค้นวันที่ 20 มกราคม 2562).
- [22] Thanatcha Kromsang, 2562, "เรียนรู้ Git และ Github ฉบับเด็กมหาวิทยาลัย", [ออนไลน์], เข้าถึงได้: <https://medium.com/@thanatchakromsang/เรียนรู้-git-และ-github-ฉบับเด็กมหาวิทยาลัย-7311034c6527> (สืบค้นวันที่ 20 มกราคม 2562).
- [23] NEXTZY, 2562, "มาเรียนรู้ Git แบบง่ายๆกันเถอะ", [ออนไลน์], เข้าถึงได้: <https://blog.nextzy.me/มาเรียนรู้-git-แบบง่ายๆกันเถอะ-427398e62f82> (สืบค้นวันที่ 20 มกราคม 2562).
- [24] RaspberryPi, 2561, "Noobs", [ออนไลน์], เข้าถึงได้: <https://www.raspberrypi.org/downloads/noobs/> (สืบค้นวันที่ 28 ตุลาคม 2561).
- [25] PoundXI, 2560, "การควบคุมอุปกรณ์อิเล็กทรอนิกส์ของ Raspberry Pi", 2561, [ออนไลน์], เข้าถึงได้: <https://poundxi.com/raspberry-pi-คืออะไร> (สืบค้นวันที่ 15 ตุลาคม 2561).

## ภาคผนวก ก คู่มือการติดตั้งระบบ

### ก.1 การติดตั้งระบบปฏิบัติการลงบน Raspberry Pi3 Model B

1. เตรียม Micro SD Card ขนาดความจุแนะนำมากกว่า 8 GB เพื่อเก็บระบบปฏิบัติการ Raspbian และสิ่งที่จำเป็นกับระบบงาน จากนั้นดาวน์โหลด Raspbian ลงไว้บนคอมพิวเตอร์จาก <https://www.raspberrypi.org/downloads/>





รูปผนวกที่ ก.1 การดาวน์โหลด Raspbian

2. ดาวน์โหลดโปรแกรม Etcher สำหรับแฟลช Raspbian ลง Micro SD Card ที่ได้เตรียมไว้เพื่อใช้กับ Raspberry Pi จาก <https://etcher.en.softonic.com/>




รูปผนวกที่ ก.2 ภาพโปรแกรม Etcher สำหรับแฟลชไฟล์

	2018-11-13-raspbian-stretch-lite	12/20/2018 11:16 ...	File folder	
	2018-11-13-raspbian-stretch-lite.zip	12/9/2018 11:13 PM	WinRAR ZIP archive	359,794 KB

รูปผนวกที่ ก.3 ภาพโฟลเดอร์ที่เก็บ Raspbian ที่ดาวน์โหลดมา

3. ขั้นตอนต่อไปทำการแฟลชลง Micro SD Card โดยเลือกระบบปฏิบัติการ Raspbian ที่ทำการแตกไฟล์

	2018-11-13-raspbian-stretch-lite.img	11/13/2018 9:02 PM	Disc Image File	1,822,720 KB
---	--------------------------------------	--------------------	-----------------	--------------

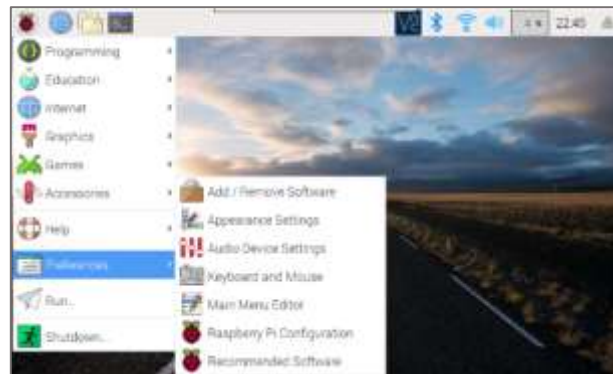
รูปผนวกที่ ก.4 Image File ของ Raspbian ที่ดาวน์โหลดมา



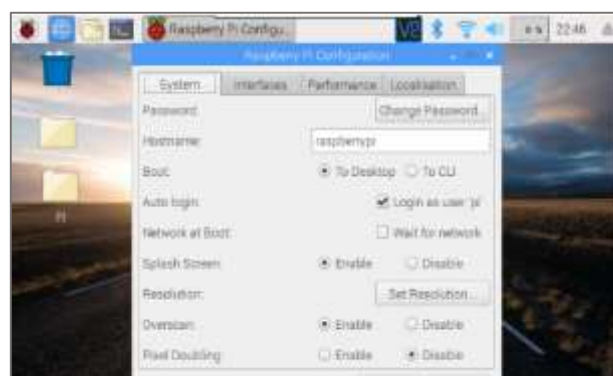
รูปผนวกที่ ก.5 ภาพแสดงการแฟลช Image File ลง SD Card

## ก.2 การตั้งค่าเบื้องต้นให้กับ Raspberry Pi3 Model B

1. เปิดเครื่อง Raspberry Pi รอจนติดตั้ง Raspbian OS เสร็จเรียบร้อยแล้ว
2. ไปที่ตั้งค่าเลือก Raspberry Pi Configuration ให้ตั้งค่า Interface และ Localisation ดังรูปผนวกที่ ก.6 – ก.9



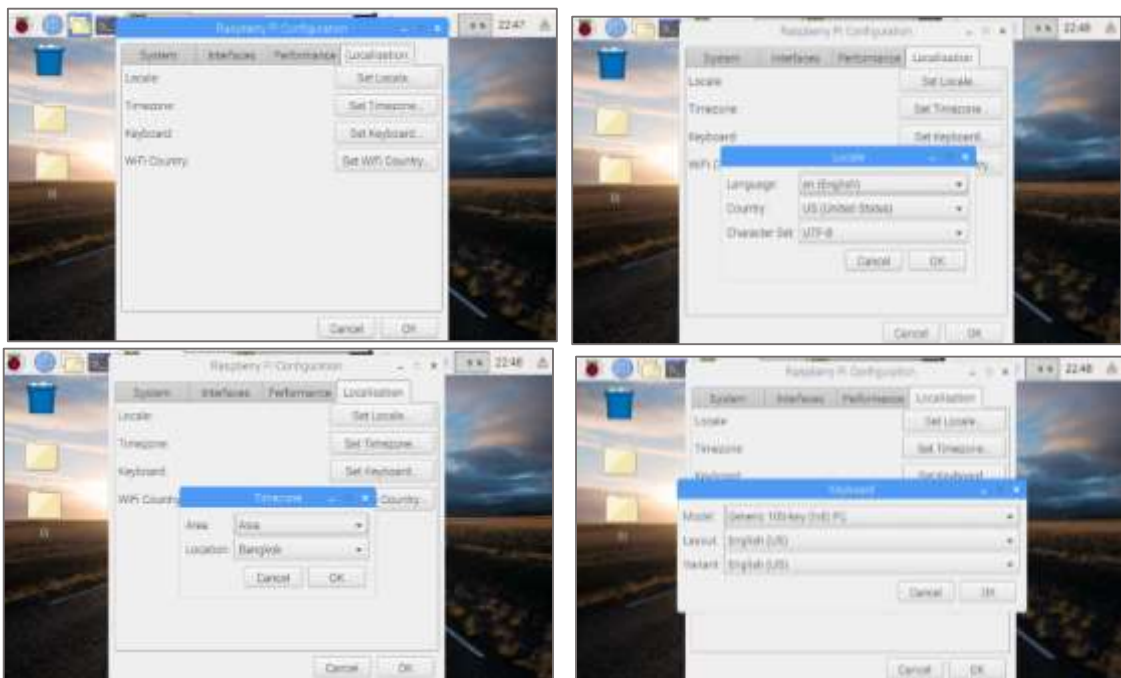
รูปผนวกที่ ก.6 ภาพเลือก Raspberry Pi Configuration



รูปผนวกที่ ก.7 เลือกที่ System และทำการตั้งชื่อให้กับHostname ของ Raspberry Pi



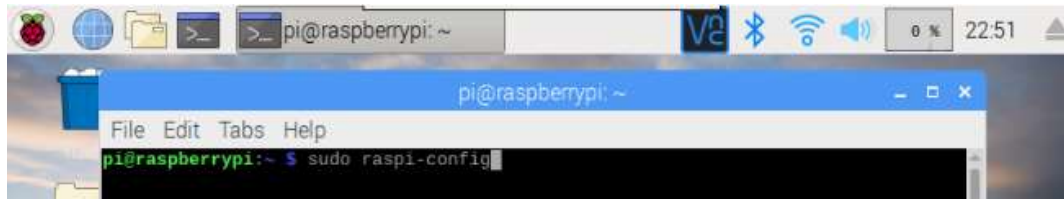
รูปผนวกที่ ก.8 เลือกที่ Interfaces ทำการตั้งค่า Enable และ Disable



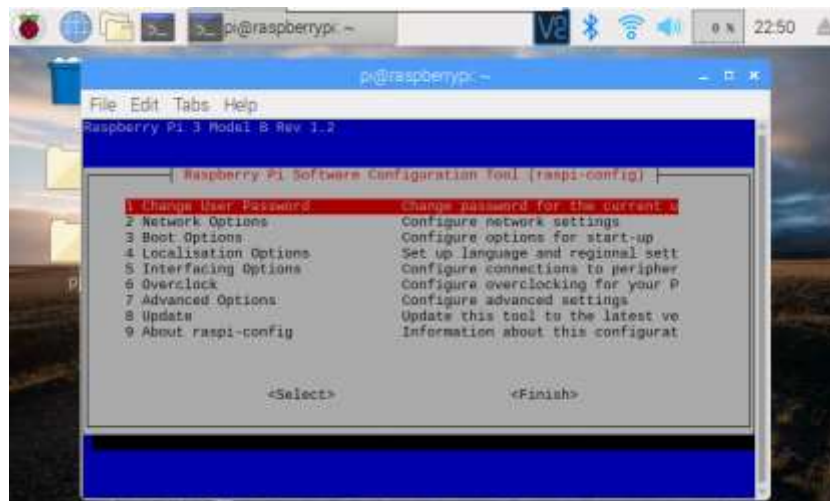
รูปผนวกที่ ก.9 เลือกที่ Localisation ทำการตั้งค่า Locale , Time Zone และ Keyboard

3. เปลี่ยนรหัสผ่านในการเข้าใช้งานเครื่องRaspberry Pi
  - พิมพ์คำสั่ง `sudo raspi-config`
  - เลือก Chang User Password
  - เลือก OK แล้วใส่ Password ที่ต้องการเปลี่ยน
4. ทำการรีบูทโดยพิมพ์ `sudo reboot`





รูปผนวกที่ ก.10 ภาพการสั่งตั้งค่า Raspberry Pi



รูปผนวกที่ ก.11 ภาพการเปลี่ยนรหัส Raspberry Pi ใหม่

### ก.3 การติดตั้ง PiCamera และ OpenCV

PiCamera เป็นไลบรารีที่ลงเพื่อจะเรียกใช้การถ่ายภาพจากโมดูลกล้อง Raspberry Pi Camera ส่วน OpenCV เป็นไลบรารีข้ามแพลตฟอร์ม สนับสนุนการเขียนโปรแกรมที่มุ่งเป้าไปการแสดงผลแบบเรียลไทม์ และยังสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึกอีกด้วย โดยจะใช้ CV2 ในการจัดการเกี่ยวกับภาพถ่าย ก่อนที่เราจะใช้ virtualenv and virtualenvwrappe จะต้องติดตั้งตามขั้นตอนต่อไปนี้

1. เปิด Command Prompt ของ Raspberry Pi
2. ทำการ Update และ Upgrade โดยใช้ apt-get โดย
  - sudo apt-get update
  - sudo apt-get upgrade
  - sudo rpi-update
  - sudo reboot
3. ติดตั้ง Developer tool
  - sudo apt-get install build-essential git cmake pkg-config
4. ติดตั้ง สิ่งที่สำคัญในการรับส่งรูปภาพ (image I/O packages) และโหลดไฟล์รูปเช่น JPEG, PNG, TIFF เป็นต้น
  - sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev

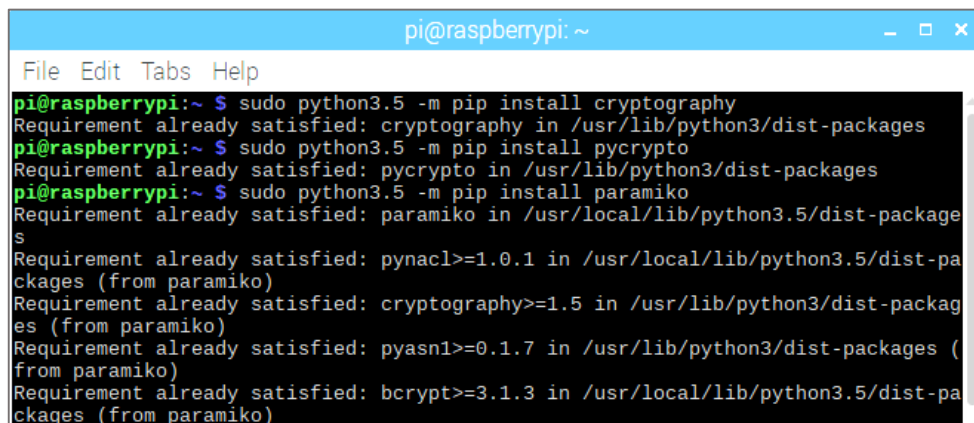
5. ติดตั้ง video I/O packages
  - `sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev`
  - `sudo apt-get install libxvidcore-dev libx264-dev`
6. ติดตั้งไลบรารี GTK ใช้ในการสร้าง GUIs และ เป็นตัวที่จะเรียก OpenCV library ด้วย (highgui) ทำให้สามารถเรียกดูภาพออกมาทางหน้าจอได้
  - `sudo apt-get install libgtk2.0-dev`
7. ติดตั้งไลบรารี เพื่อประสิทธิภาพให้กับ OpenCV library
  - `sudo apt-get install libatlas-base-dev gfortran`
8. ติดตั้ง Python 2.7 and Python 3 เพื่อใช้ compile OpenCV และ python
  - `sudo apt-get install python2.7-dev python3-dev`
9. Setup python สำหรับ OpenCV โดย install pip
  - `wget https://bootstrap.pypa.io/get-pip.py`
  - `sudo python get-pip.py`
10. ติดตั้ง virtualenv and virtualenvwrapper ด้วย pip
  - `sudo pip install virtualenv virtualenvwrapper`
  - `sudo rm -rf ~/.cache/pip`
11. Update ~/.profile file โดยเพิ่ม 3 บรรทัดด้านล่าง ที่ไฟล์ profile ใช้คำสั่ง
  - `nano ~/.profile`
  - เพิ่ม 3 บรรทัดด้านล่างลงท้ายไฟล์
  - `# virtualenv and virtualenvwrapper`  
`Export WORKON_HOME=$HOME/.virtualenvs`  
`source /usr/local/bin/virtualenvwrapper.sh`
12. ให้ทำการ Reload profile อีกครั้ง
  - `source ~/.profile`
13. สร้าง vision virtual environment โดยให้เลือกว่าจะใช้ python2 หรือ python3
  - สำหรับ Python2 พิมพ์คำสั่ง `mkvirtualenv cv`
  - สำหรับ Python3 พิมพ์คำสั่ง `mkvirtualenv cv -p python3`
14. ทำการ reboot ระบบโดยการปิดและเปิด terminal ใหม่จากนั้น Reload Profile อีกครั้งแล้วเข้า virtual environment
  - `source ~/.profile`
  - `workon cv`
15. ติดตั้ง numpy ซึ่งเป็นตัวสำคัญที่ใช้ในการ compiling ไฟล์ Python ที่ใช้ OpenCV
  - `pip install numpy`
16. ดาวน์โหลด OpenCV
  - `workon cv`

- git clone https://github.com/opencv/opencv.git
  - cd opencv
  - mkdir mybin
  - cd mybin
  - cmake
    - DCMMAKE\_BUILD\_TYPE=RELEASE
    - DCMMAKE\_INSTALL\_PREFIX=/usr/local
    - DENABLE\_PRECOMPILED\_HEADERS=OFF
    - DWITH\_FFMPEG=OFF ..
17. setup และ build
- make
18. สุดท้ายติดตั้ง OpenCV
- sudo make install
  - sudo ldconfig
19. นำ OpenCV ไปใช้ต้องเชื่อมโยงไลบรารี OpenCV กับ site-packages directory ทั้งหมดก่อนถึงจะสามารถใช้ OpenCV library ได้
- cd ~/.virtualenvs/cv/lib/python2.7/site-packages/
  - ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so
  - ln -s /usr/local/lib/python2.7/site-packages/cv.py cv.py
20. ทดสอบเรียกใช้ OpenCV library โดยทำการเรียกดูเวอร์ชันด้วย python ให้เปิด terminal ขึ้นมาแล้วให้พิมพ์คำสั่งต่อไปนี้
- workon cv
  - python
  - >>> import cv2
  - >>> cv2.\_\_version\_\_
- จะเห็นได้ว่ามีเวอร์ชันแสดงออกมา '2.4.10'
21. ถ้าง่วงแล้วเกิดข้อผิดพลาดให้ลงดังต่อไปนี้
- Python3 -m pip install numpy
  - Python3 -m pip install opencv-python
  - sudo apt install libgtk4
  - sudo apt install libgtk4-test

## ก.4 การติดตั้งไลบรารีที่จำเป็นกับระบบงาน

เป็นไลบรารีที่จำเป็นต้องใช้ของตัวอุปกรณ์โดยมีขั้นตอนการติดตั้งไลบรารีดังต่อไปนี้

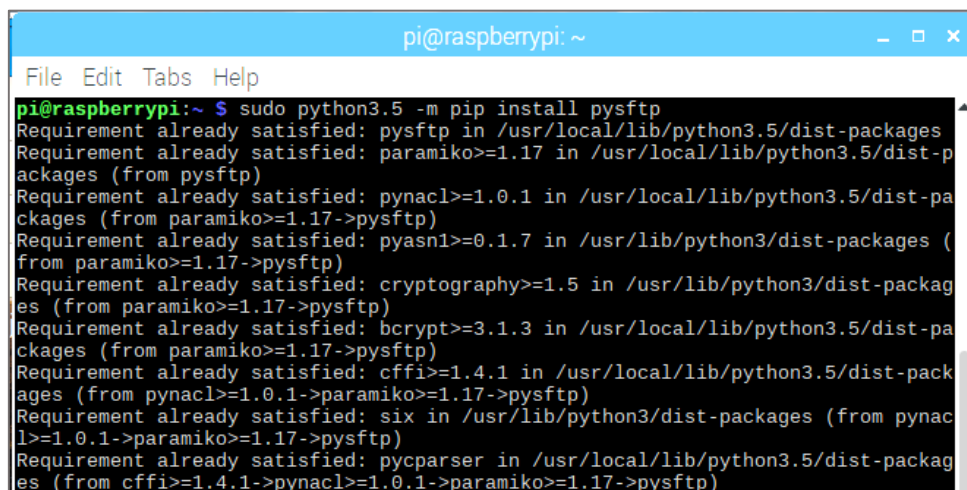
1. ติดตั้ง pysftp ลงบน Raspberry Pi3
  - sudo python3.5 -m pip install cryptography
  - sudo python3.5 -m pip install pycrypto
  - sudo python3.5 -m pip install paramiko



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo python3.5 -m pip install cryptography  
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages  
pi@raspberrypi:~ $ sudo python3.5 -m pip install pycrypto  
Requirement already satisfied: pycrypto in /usr/lib/python3/dist-packages  
pi@raspberrypi:~ $ sudo python3.5 -m pip install paramiko  
Requirement already satisfied: paramiko in /usr/local/lib/python3.5/dist-packages  
Requirement already satisfied: pynacl>=1.0.1 in /usr/local/lib/python3.5/dist-packages (from paramiko)  
Requirement already satisfied: cryptography>=1.5 in /usr/lib/python3/dist-packages (from paramiko)  
Requirement already satisfied: pyasn1>=0.1.7 in /usr/lib/python3/dist-packages (from paramiko)  
Requirement already satisfied: bcrypt>=3.1.3 in /usr/local/lib/python3.5/dist-packages (from paramiko)
```

รูปผนวกที่ ก.12 ภาพติดตั้ง cryptography , pycrypto และ paramiko

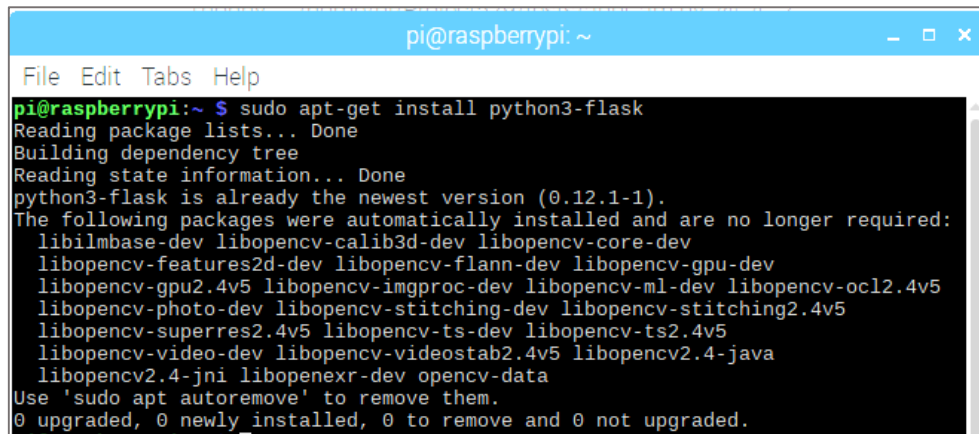
- sudo python3.5 -m pip install pysftp



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo python3.5 -m pip install pysftp  
Requirement already satisfied: pysftp in /usr/local/lib/python3.5/dist-packages  
Requirement already satisfied: paramiko>=1.17 in /usr/local/lib/python3.5/dist-packages (from pysftp)  
Requirement already satisfied: pynacl>=1.0.1 in /usr/local/lib/python3.5/dist-packages (from paramiko>=1.17->pysftp)  
Requirement already satisfied: pyasn1>=0.1.7 in /usr/lib/python3/dist-packages (from paramiko>=1.17->pysftp)  
Requirement already satisfied: cryptography>=1.5 in /usr/lib/python3/dist-packages (from paramiko>=1.17->pysftp)  
Requirement already satisfied: bcrypt>=3.1.3 in /usr/local/lib/python3.5/dist-packages (from paramiko>=1.17->pysftp)  
Requirement already satisfied: cffi>=1.4.1 in /usr/local/lib/python3.5/dist-packages (from pynacl>=1.0.1->paramiko>=1.17->pysftp)  
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from pynacl>=1.0.1->paramiko>=1.17->pysftp)  
Requirement already satisfied: pycparser in /usr/local/lib/python3.5/dist-packages (from cffi>=1.4.1->pynacl>=1.0.1->paramiko>=1.17->pysftp)
```

รูปผนวกที่ ก.13 ภาพติดตั้ง pysftp

2. ติดตั้ง Flask ลงบน Raspberry Pi3
  - sudo apt-get install python3-flask



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo apt-get install python3-flask  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python3-flask is already the newest version (0.12.1-1).  
The following packages were automatically installed and are no longer required:  
  libilmbase-dev libopencv-calib3d-dev libopencv-core-dev  
  libopencv-features2d-dev libopencv-flann-dev libopencv-gpu-dev  
  libopencv-gpu2.4v5 libopencv-imgproc-dev libopencv-ml-dev libopencv-ocl2.4v5  
  libopencv-photo-dev libopencv-stitching-dev libopencv-stitching2.4v5  
  libopencv-superres2.4v5 libopencv-ts-dev libopencv-ts2.4v5  
  libopencv-video-dev libopencv-videostab2.4v5 libopencv2.4-java  
  libopencv2.4-jni libopenexr-dev opencv-data  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

รูปผนวกที่ ก.14 ภาพติดตั้ง Flask

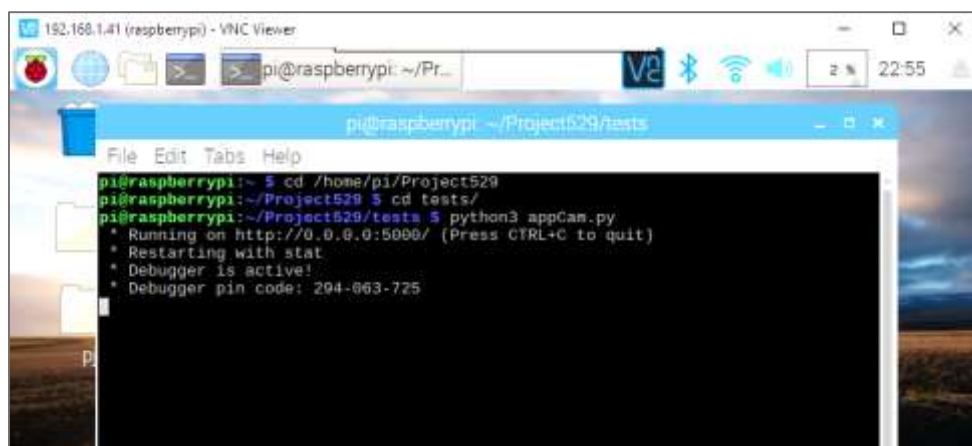
- ทดลองรัน Flask
  - สร้างไฟล์ appCam.py และเขียนไฟล์ดังรูปผนวกที่ ก.4-4



```
from flask import Flask  
app = Flask(__name__)  
@app.route('/')  
def index():  
    return 'Hello world'  
if __name__ == '__main__':  
    app.run(debug=True, port=80, host='0.0.0.0')
```

รูปผนวกที่ ก.15 ภาพโค้ดสำหรับรันและกำหนด Port ของ Flask

- ทดสอบรัน appCam.py โดยพิมพ์ python3 appCam.py



```
192.168.1.41 (raspberrypi) - VNC Viewer  
pi@raspberrypi: ~/Project529  
pi@raspberrypi:~$ cd /home/pi/Project529  
pi@raspberrypi:~/Project529$ cd tests/  
pi@raspberrypi:~/Project529/tests$ python3 appCam.py  
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)  
* Restarting with stat  
* Debugger is active!  
* Debugger pin code: 294-0663-725
```

รูปผนวกที่ ก.16 ภาพการรัน Flask บน Raspberry Pi

- จะมี Hello World แสดงหน้าเว็บ



รูปผนวกที่ ก.17 ภาพการเปิดเว็บแอปพลิเคชันที่ใช้ Flask บน Raspberry Pi

## ก.5 การสมัครและใช้งานบอทไลน์

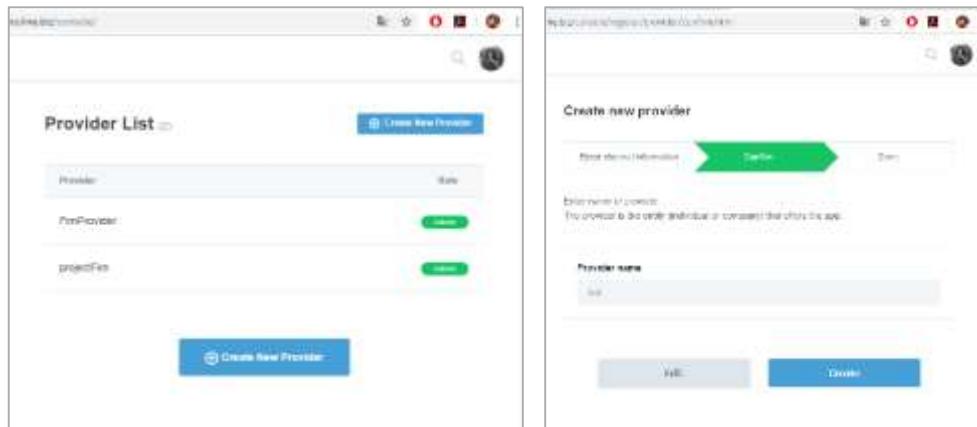
เราจะใช้บอทไลน์มาช่วยให้ทำงานง่ายขึ้น ไม่ว่าจะส่ง Notify ต่างๆ หรือสั่งให้บอทดึงค่าจาก Database จากอุปกรณ์ มาแสดงเป็นรูปภาพ หรือข้อความกลับมาให้เราที่ไลน์ได้

1. สมัครใช้งานไลน์จาก <https://developers.line.biz/en/> เข้าสู่ระบบบัญชีไลน์



รูปผนวกที่ ก.18 ภาพการเข้าสู่ระบบเพื่อใช้บริการไลน์ในการสร้างบอท

2. สร้าง Provider สำหรับเก็บไลน์บอทที่เราสร้าง ให้เลือก Create New Provider



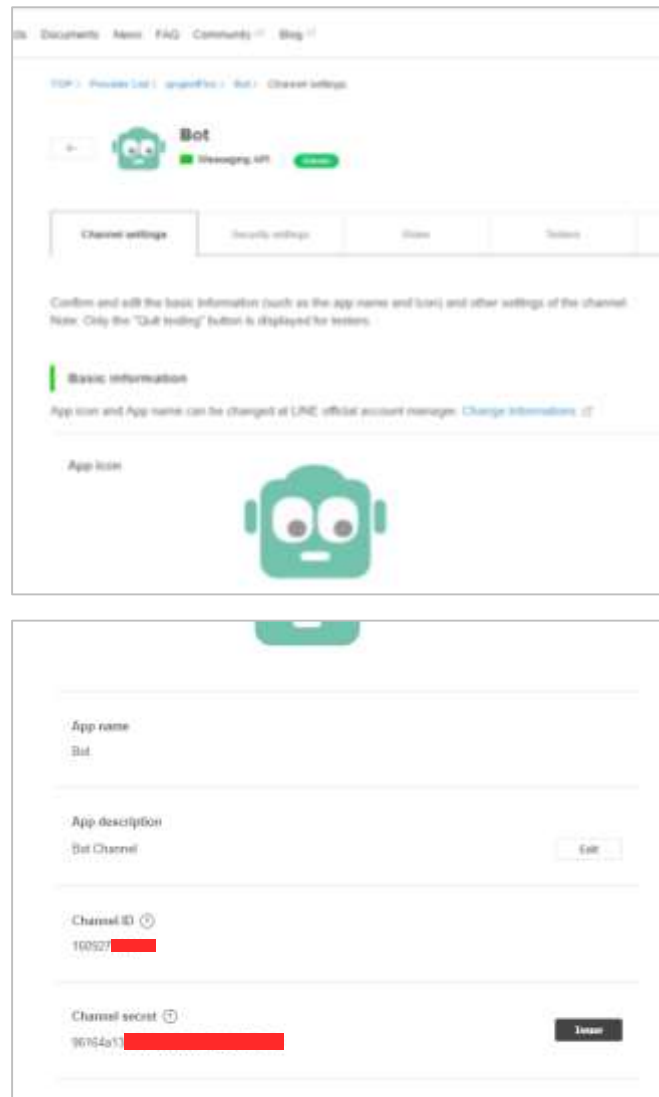
รูปผนวกที่ ก.19 ภาพการสร้าง Provider สำหรับเก็บบอทไลน์

3. เริ่มสร้างบอทไลน์โดยเลือก Messaging API



รูปผนวกที่ ก.20 ภาพการเริ่มสร้างบอทไลน์โดยเลือก Messaging API

4. เลือกที่ Channel Settings แล้วเพิ่มไอคอนที่ App icon แก้ไขชื่อบอทที่ App name และเพิ่มคำอธิบายเกี่ยวกับบอทที่ App description



รูปผนวกที่ ก.21 ภาพการสร้างบอทไลน์โดยกำหนดค่าที่ Channel Setting

5. สร้างไฟล์ bot\_test.php สำหรับกำหนด Action ต่าง ๆ ให้กับไลน์บอท เช่น เมื่อผู้ใช้พิมพ์คุยกับบอทในไลน์ โดยพิมพ์ hi หรือ hello ไลน์บอทจะตอบกลับไปเป็นคำทักทายที่ผู้ใช้พิมพ์



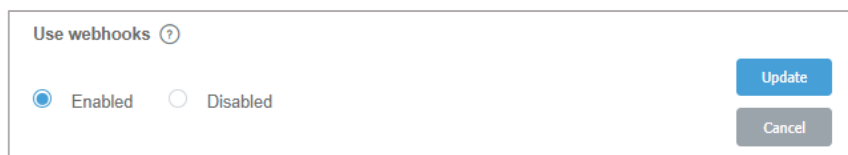
```

1 <?php
2 $API_URL = 'https://api.line.me/v2/bot/message'; // URL API LINE
3 $ACCESS_TOKEN = '5Drfo4t/S4oZbsMzAbenhjM70kuXfWe6Xp6S1LnmVbNwTrYa';
4 $CHANNEL_SECRET = '96164a13e36916';
5 $POST_HEADER = array('Content-Type: application/json', 'Authorization: Bearer ' . $ACCESS_TOKEN); // Set HEADER
6 $request = file_get_contents('php://input'); // Get request content
7 $request_array = json_decode($request, true); // Decode JSON to Array
8 //เป็นการ Get ข้อมูลที่ได้จากการที่ User ที่มีการกระทำใน Channel
9 if (sizeof($request_array['events']) > 0) {
10     // $json_encode = json_encode($request_array['events']);
11     foreach ($request_array['events'] as $event) {
12         $json_encode = json_encode($request_array);
13         $reply_token = $event['replyToken']; // Build message to reply back
14         if ($event['type'] == 'message') {
15             if ($event['message']['type'] == 'text') {
16                 $userID = $event['source']['userId'];
17                 $groupID = $event['source']['groupId'];
18                 $text = $event['message']['text'];
19                 if ($text == 'hi' || $text == 'hello') {
20                     $text = 'สวัสดีจ้า';
21                     $data = ['replyToken' => $reply_token,
22                             'messages' => [
23                                 // ['type' => 'text', 'text' => $json_encode],
24                                 ['type' => 'text', 'text' => $text],
25                             ]
26                     ];
27                     $post_body = json_encode($data);
28                     $send_result = send_reply_message($API_URL.'/reply', $POST_HEADER, $post_body);
29                 }
30             }
31         }
32     }
33 }
34 >>

```

รูปผนวกที่ ก.22 ภาพเกี่ยวกับโค้ดที่ทักทายของบอทไลน์

- เปิดใช้งาน Webhook โดยไปที่ Use Webhook เลือก Edit เปลี่ยนโหมดเป็น Enabled



Use webhooks ?

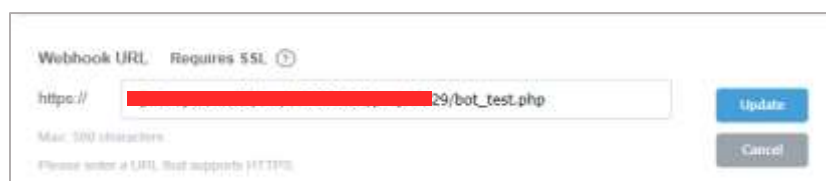
☒ Enabled ☐ Disabled

Update

Cancel

รูปผนวกที่ ก.23 ภาพการกำหนดค่าให้ Use Webhooks โดยเลือก Enabled

- ใส่ URL ไลน์บอทที่เราเตรียมไว้ ไปที่ Webhook URL เลือก Edit แล้วใส่ URL ลงไป โดยจะต้องเป็นแบบ HTTPS เท่านั้น จากนั้นให้เลือก Verify เพื่อเชื่อมกับบอทที่ได้เขียนไว้ เมื่อมีเครื่องหมาย ✓ ด้านใดแสดงว่าสามารถใช้บอทได้



Webhook URL Requires SSL ?

https:// [redacted] /bot\_test.php

Max. 500 characters

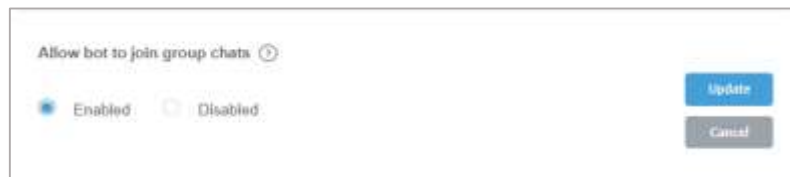
Please enter a URL that supports HTTPS

Update

Cancel

รูปผนวกที่ ก.24 ภาพการใส่ URL ของบอทไลน์ที่ Webhook URL

8. อนุญาตให้เพิ่มไลน์บอทเข้าไลน์กลุ่มได้ ให้ไปที่ Allow bot to join group chats เลือกโหมดให้เป็น Enabled



รูปผนวกที่ ก.25 ภาพการกำหนดค่า Enabled ให้กับ Allow bot to join group chart เมื่อต้องการใช้งานบอทในกลุ่มไลน์

9. ทดลองใช้ไลน์บอทโดยสแกน QR Code ที่ให้มา และพิมพ์เวิร์ดที่กำหนดไว้ลงในไลน์



รูปผนวกที่ ก.26 ภาพ QR-Code ของบอทไลน์



รูปผนวกที่ ก.27 ภาพการคุยกับบอทของผู้ใช้งาน

## ก.6 การทดลองสร้างบอทด้วยการใช้ ngrok

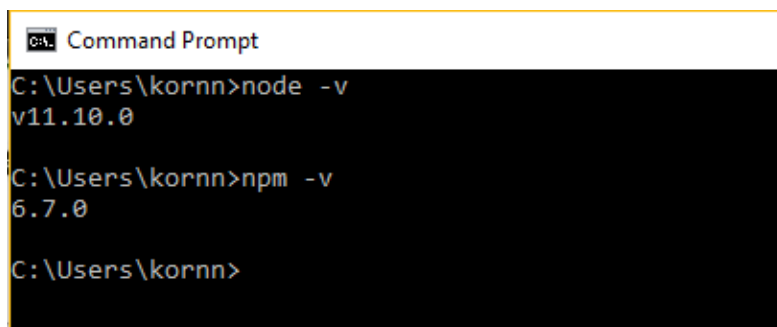
เราจะใช้ ngrok ในการสร้างบอทไลน์โดยมีวิธีการติดตั้งและเรียกใช้งาน ngrok ดังต่อไปนี้

1. ติดตั้ง ngrok ลงในคอมพิวเตอร์
2. ติดตั้ง Node.js โดยปกติแล้ว npm จะถูกติดตั้งมาพร้อมกับ node.js ทำหน้าที่ในการจัดการ package เสริมต่าง ๆ ไม่ว่าจะเป็น application หรือ module ต่าง ๆ เพื่อให้เราสามารถเรียกใช้งานได้
  - โดยเข้าไปที่เว็บไซต์ <https://nodejs.org/en/>



รูปผนวกที่ ก.28 ภาพการดาวน์โหลด Node.js

- ดาวน์โหลดเสร็จแล้ว จะได้ตัว Install node.js มา ให้ทำการ Double click เพื่อติดตั้ง
3. เมื่อลงเสร็จเรียบร้อยแล้วให้เข้า Start -> cmd -> node -v และ npm -v ก็จะแสดงเวอร์ชันของ node.js และ npm



รูปผนวกที่ ก.29 ภาพการเช็คเวอร์ชันของ Node.js และ npm

4. clone Folder
  - git clone <https://github.com/kamnan43/line-bot-nodejs-starter>
  - cd line-bot-nodejs-starter
5. ทำการแก้ไขที่ไฟล์ config.json โดยเพิ่ม ChannelAccessToken และ ChannelSecret LINE ของเราลงไป

```
"port" : "3000",  
"channelAccessToken": "5Drfo4t/S4oZbs[REDACTED]  
"channelSecret": "96164a13e3691[REDACTED]"
```

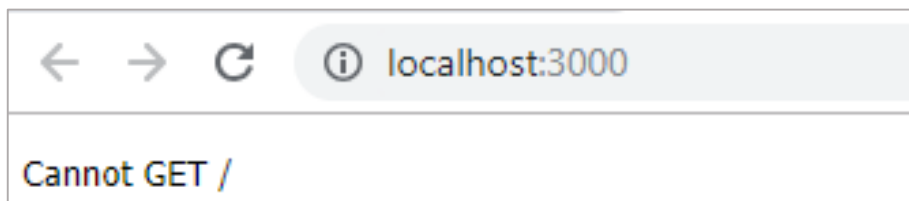
รูปผนวกที่ ก.30 ภาพการแก้ไข Channel Access Token และ Channel Secret ของบอทไลน์

- ทดสอบการใช้ npm โดย เปิด cmd แล้วเข้าไปยังไดเรกทอรีไฟล์ที่เรา Clone มา จากนั้นพิมพ์ npm start เพื่อทำการ start bot ดังรูปผนวกที่ ก.31

```
C:\Users\kornn>cd Desktop\line-bot-nodejs-starter  
C:\Users\kornn\Desktop\line-bot-nodejs-starter>npm start  
  
> line-bot-nodejs-starter@1.0.0 start C:\Users\kornn\Desktop\line-bot-nodejs-starter  
> node index.js  
  
listening on 3000
```

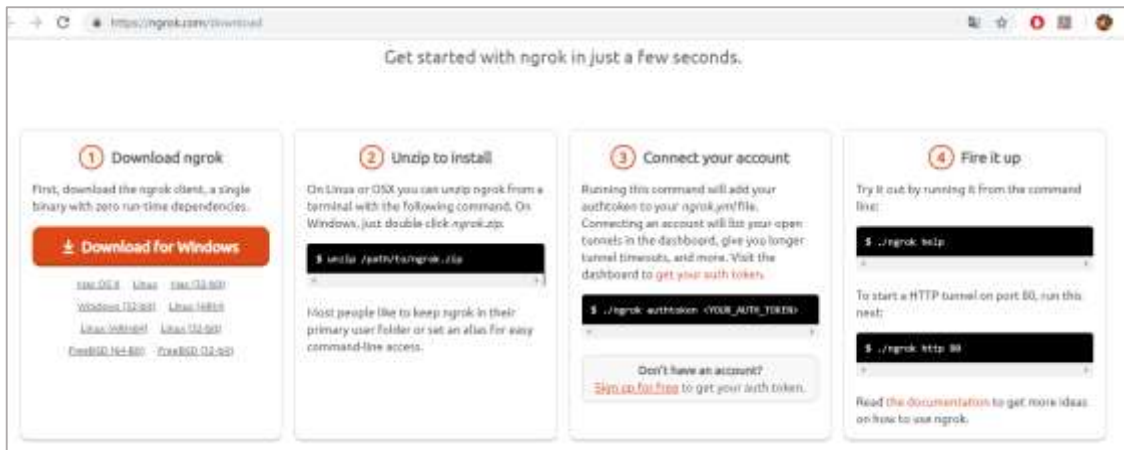
รูปผนวกที่ ก.31 ภาพการเริ่มใช้งานบอทไลน์ที่ดาวน์โหลดมา

- ตอนนี้ บอท พร้อมทำงานแล้วเข้าไปที่ <http://localhost:3000>



รูปผนวกที่ ก.32 ภาพการทำงานของบอทบนหน้าเว็บ

- เปิดใช้งาน ngrok แต่บอทจะยังไม่สามารถเชื่อมต่อกับ LINE ได้ในตอนนี้เนื่องจาก LINE ต้องการ public url และต้องเป็น https ดังนั้นเราจะใช้ ngrok เพื่อ expose localhost ของเรา ออก public เพื่อให้สามารถเชื่อมต่อกับ LINE ได้
- เข้าไปที่ <https://ngrok.com/download> และทำตามข้อ 1 - 3 ดังรูปผนวกที่ ก.33



รูปผนวกที่ ก.33 ภาพการติดตั้งและตั้งค่าเพื่อใช้ ngrok บน PC

8. รัน ngrok เพื่อ expose port ที่ต้องการ เช่น ถ้าเราต้องการรัน Bot ของเราที่ port 3000 เข้าไปที่ไดเรกทอรีของ ngrok และใช้คำสั่ง ดังนี้

- พิมพ์ ngrok http 3000 เพื่อเชื่อมกับ LINE BoT

```

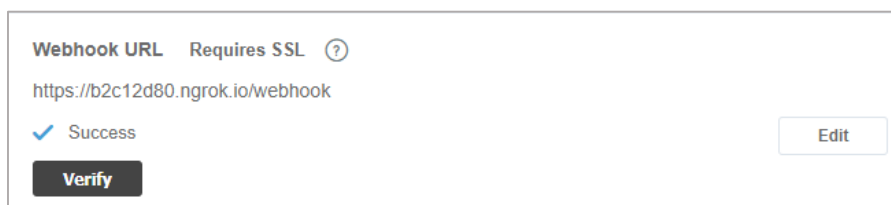
C:\> Command Prompt
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\korinn>cd Desktop\ngrok-stable-windows-amd64 (1)

C:\Users\korinn\Desktop\ngrok-stable-windows-amd64 (1)>ngrok http 3000
  
```

รูปผนวกที่ ก.34 ภาพการรัน ngrok เพื่อเชื่อมกับบอลไลน์

9. นำ url ที่เป็น https ต่อท้ายด้วย /webhook ไปใส่ใน Webhook URL ของ LINE และกด verify ให้เป็นดังรูปผนวกที่ ก.35 (ต้องเชื่อมบอทกับ ngrok จะต้องรัน npm ไว้ด้วย)



รูปผนวกที่ ก.35 ภาพการตรวจสอบ Webhook URL ของบอท

```
Command Prompt - ngrok http 3000
ngrok by @inconshreveable

Session Status      online
Account             Kornnalin Chawengsub (Plan: Free)
Version             2.2.8
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://b2c12d80.ngrok.io -> localhost:3000
                   https://b2c12d80.ngrok.io -> localhost:3000

Connections         ttl    opn    rt1    rt5    p50    p90
                   1      0      0.00   0.00   5.75   5.75

HTTP Requests
-----
POST /webhook       200 OK
```

รูปผนวกที่ ก.36 ภาพการรัน ngrok บน PC

10. ทดลองส่งข้อความคุยกับบอทในกลุ่ม จะเห็นว่าบอทตอบกลับมาเป็นข้อความที่ส่งไป ส่วนที่ cmd จะแสดงสถานะ 200 OK



รูปผนวกที่ ก.37 ภาพการตอบกลับของบอทกับผู้ใช้งาน

```

C:\> Select Command Prompt - ngrok http 3000

ngrok by @inconshreveable

Session Status      online
Account             Kornnalin Chawengsub (Plan: Free)
Version             2.2.8
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://b2c12d80.ngrok.io -> localhost:3000
                    https://b2c12d80.ngrok.io -> localhost:3000

Connections         ttl    opn    rt1    rt5    p50    p90
                    4      0      0.00   0.00   2.07   5.75

HTTP Requests
-----
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK
POST /webhook        200 OK

```

รูปผนวกที่ ก.38 ภาพสถานะการรับส่งค่าผ่าน Message API LINE

11. ส่วนใน npm ที่รันไว้จะแสดงเป็นlog ต่างๆในไลน์กลุ่มดังรูปผนวกที่ ก.39

```

C:\Users\kornn\Desktop\line-bot-nodejs-starter>npm start

> line-bot-nodejs-starter@1.0.0 start C:\Users\kornn\Desktop\line-bot-nodejs-starter
> node index.js

listening on 3000
event { replyToken: '00000000000000000000000000000000',
  type: 'message',
  timestamp: 1551358388553,
  source:
    { type: 'user', userId: 'Udeadbeefdeadbee[REDACTED]' },
  message: { id: '100001', type: 'text', text: 'Hello, world' } }
event { replyToken: 'ffffffffffffffffffffffffffffffff',
  type: 'message',
  timestamp: 1551358388553,
  source:
    { type: 'user', userId: 'Udeadbeefdeadbee[REDACTED]' },
  message:
    { id: '100002', type: 'sticker', packageId: '1', stickerId: '1' } }
event { type: 'message',
  replyToken: '2d6c6ba05e0e4ef[REDACTED]',
  source:
    { userId: 'U5f22114eb67ab53507dbc659a03ff468', type: 'user' },
  timestamp: 1551358881159,
  message: { type: 'text', id: '9433948105413', text: 'Kornnalin' } }
event { type: 'message',
  replyToken: '41d71a20e39149a[REDACTED]',
  source:
    { userId: 'U5f22114eb67ab53507dbc659a03ff468', type: 'user' },
  timestamp: 1551358892208,
  message: { type: 'text', id: '9433949197874', text: 'Hi' } }
event { type: 'message',
  replyToken: 'e1c9a8ce51aa4136bf4fa73582f368da',

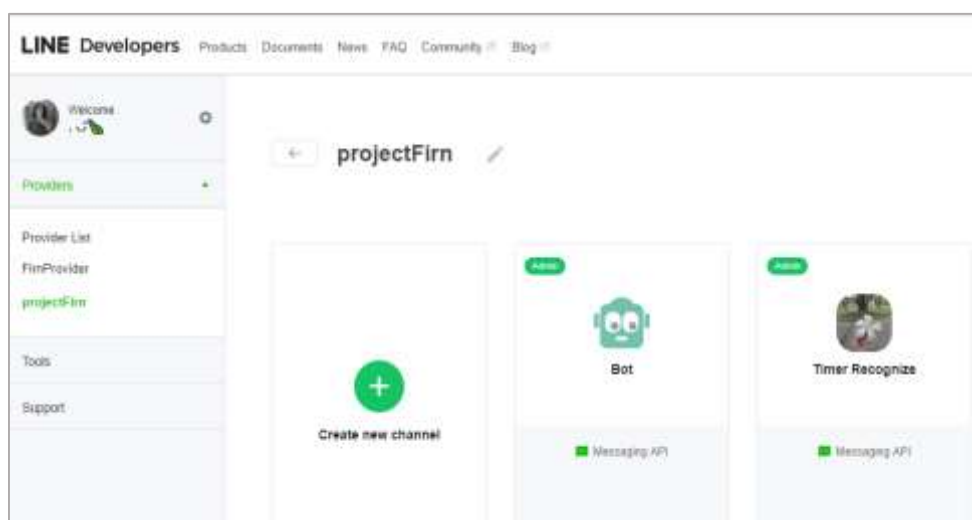
```

รูปผนวกที่ ก.39 ภาพ Log ของบอทไลน์

## ก.7 การใช้ LINE LIFF

LIFF เป็นบริการที่ LINE จัดเตรียมไว้ให้โดยโดยในปฏิญานิพนธ์นี้ ต้องการเปิดหน้าเว็บแอปพลิเคชันที่ได้พัฒนาขึ้น เพื่อใช้สำหรับการแท็กภาพถ่ายและดูรายงานการเข้าออกงานของผู้ใช้ในไลน์กลุ่มบนมือถือ ดังนั้นหากต้องการใช้บริการ LINE LIFF เปิดหน้าเว็บแอปพลิเคชันที่พัฒนาไว้จะต้องมี LIFF ID เพื่อเข้าถึงหน้าเว็บต่างๆ ในแชทส่วนตัว หรือกลุ่มแชทของไลน์ได้เลย โดยไม่ต้องแจ้งเตือนออกไปเปิดหน้าเว็บแอปพลิเคชันที่เบราว์เซอร์อื่นๆ เช่น Safari Chrome เป็นต้น และวิธีกำหนดค่าเพื่อหา LIFF ID มีต่อไปนี้

1. เข้าไปที่ LINE Developer เว็บไซต์ <https://developers.line.biz/en/> ไปยัง Provider ที่เก็บบอทที่เราสร้าง



รูปผนวกที่ ก.40 ภาพบอทไลน์ที่ถูกสร้างขึ้น

2. เข้าไปยังบอทที่เราได้ทำการสร้างไว้ เลือก LIFF และเลือก ADD



รูปผนวกที่ ก.41 ภาพการใช้บริการ LIFF LINE

3. กำหนดค่าเพื่อเพิ่ม LIFF โดยกำหนดชื่อ (Name) เลือกขนาด (Size) แสดงผลบนจอโทรศัพท์ โดย Full จะแสดงเต็มหน้าจอ 100%, Tall จะแสดง 75% ของหน้าจอโทรศัพท์ และ Compact 50% ของหน้าจอโทรศัพท์



รูปผนวกที่ ก.42 ภาพสร้าง LIFF ใหม่

4. จะได้ LIFF ที่เราสร้าง พร้อมกับ LIFF URL ดังรูปผนวกที่ ก.43

รูปผนวกที่ ก.43 ภาพ LIFF URL ของ Add Member LIFF ที่สร้างขึ้น

5. ทดลองใช้ LIFF แสดงหน้าเว็บแอปพลิเคชันที่เราสร้างโดยใช้เพิ่ม โค้ดในส่วนตอบกลับของบอทดังต่อไปนี้
  - สร้างคีย์เวิร์ดคำที่ผู้ใช้พิมพ์คุยกับบอทเกี่ยวกับการเพิ่มสมาชิกใหม่

```
$keyword_addMember = array('M', 'm', 'Member', 'MERBER', 'member', 'สมัครสมาชิก', 'เพิ่มสมาชิกใหม่', 'สมาชิกใหม่');
```

รูปผนวกที่ ก.44 ภาพการกำหนดคีย์เวิร์ดเกี่ยวกับการเพิ่มสมาชิกใหม่ให้กับบอทไลน์

- สร้างฟังก์ชันในตอบกลับโดยเพิ่มโค้ดดังรูปผนวกที่ ก.45

```
function getBubble( $title, $btn_url ) {
    $bubble = array(
        "type"=> "bubble",
        "header"=> array(
            "type"=> "box",
            "layout"=> "vertical",
            "contents"=>
                [
                    array(
                        "type"=> "text",
                        "text"=> $title,
                        "size"=> "x1"
                    )
                ]
        ),
        "body"=> array(
            "type"=> "box",
            "layout"=> "vertical",
            "contents"=>
                [
                    array(
                        "type"=> "button",
                        "style"=> "primary",
                        "action"=> array(
                            "type"=> "uri",
                            "label"=> "คลิก",
                            "uri"=> $btn_url
                        )
                    )
                ]
        )
    );
    return $bubble;
};
```

รูปผนวกที่ ก.45 ภาพโค้ดฟังก์ชันที่เขียนขึ้น เพื่อให้บอทตอบกลับผู้ใช้ในรูปแบบ Flex Message

- นำ LIFF URL ที่ได้มาใส่ในโค้ดบอทเพื่อที่จะให้บอทตอบกลับเป็นหน้าเว็บแอปพลิเคชันที่ต้องการบนมือถือในกลุ่มไลน์

```

120 //เป็นการ Get ข้อมูลที่ได้จากการที่ User ที่มีการกระทำใน Channel
121 if (sizeof($request_array['events']) > 0) {
122     // $json_encode = json_encode($request_array['events']);
123     foreach ($request_array['events'] as $sevent) {
124         $json_encode= json_encode($request_array);
125         $reply_token = $sevent['replyToken']; // Build message to reply back
126         if ($sevent['type'] == 'message') {
127             if($sevent['message']['type'] == 'text'){
128                 $userID = $sevent['source']['userId'];
129                 $groupID = $sevent['source']['groupId'];
130                 $text = $sevent['message']['text'];
131                 //Add memberJoined
132                 foreach ($keyword_addMember as $key => $member) {
133                     if ($text == $member) {
134                         $contents = array(
135                             "type"=> "carousel",
136                             "contents"=> [
137                                 $contents = getBubble( "สมัครสมาชิกใหม่", "line://app/160927[REDACTED]?user=".$userID),
138                             ]
139                         );
140                         $messages = ['type'=>'flex', 'altText'=>'Report', 'contents'=> $contents];
141
142                         $data = [ 'replyToken' => $reply_token, 'messages' => [$messages], ];
143                         $post_body = json_encode($data);
144                         $send_result = send_reply_message($API_URL.'/reply', $POST_HEADER, $post_body);
145                     }
146                 }
147             }
148         }
149     }
150 }
151 echo "bot test";
152 ?>

```

รูปผนวกที่ ก.46 ภาพโค้ดบอทไลน์

- ทดลองพิมพ์คีย์เวิร์ดคุยกับบอท โดยพิมพ์ “Member” จะเห็นว่าบอทตอบกลับมาเป็น Bubble Message



รูปผนวกที่ ก.47 ภาพการตอบกลับของบอทในรูปแบบ Flex Message

- เมื่อผู้ใช้ในไลน์กดปุ่ม บอทจะแสดงหน้าเว็บแอปพลิเคชันดังรูปผนวกที่ ก.48

รูปผนวกที่ ก.48 ภาพหน้าเว็บแอปพลิเคชันแสดงการสมัครสมาชิกใหม่

## ก.8 การสมัครใช้งานไลน์ Notify

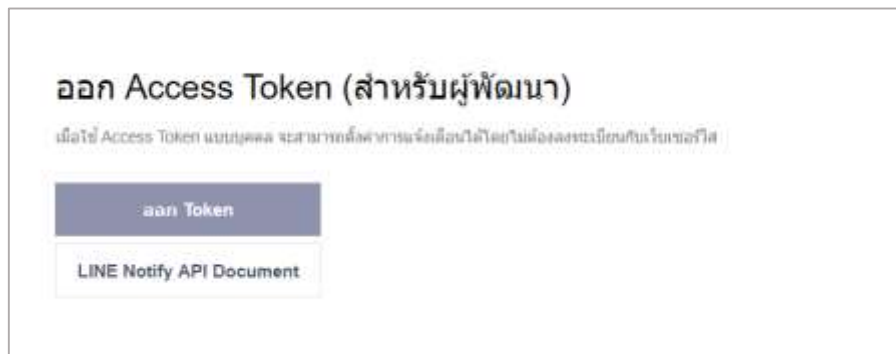
เป็นบริการของ LINE ที่สามารถส่งข้อความการแจ้งเตือนต่าง ๆ ไปยังกลุ่มไลน์ต่าง ๆ ผ่านทาง API ที่ LINE เตรียมไว้ โดยจะประยุกต์ใช้กับอุปกรณ์ ซึ่งในปริญญานิพนธ์นี้จะส่งการแจ้งเตือนไปยังไลน์กลุ่มจากตัวอุปกรณ์ Raspberry Pi การแจ้งเตือนไปยังกลุ่มไลน์นั้นจะต้องมี Token ในการเข้าถึงเพื่อระบุและส่งการแจ้งเตือนได้ถูกแชท หรือกลุ่มแชท เป็นต้น

### ขั้นตอนการใช้ LINE Notify

1. ไปลงชื่อเข้าใช้งานที่ <https://notify-bot.line.me/th>

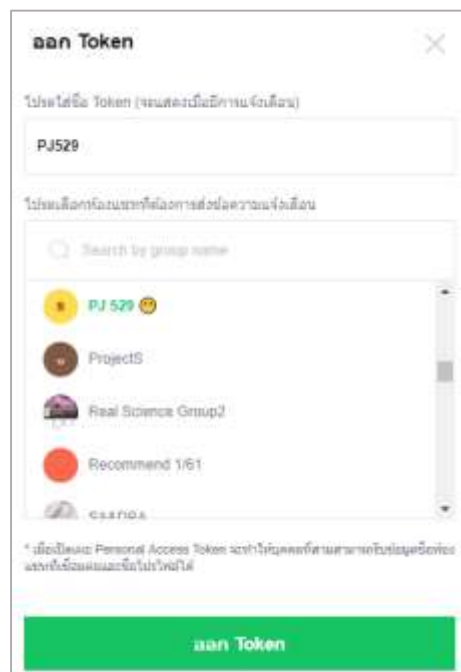
รูปผนวกที่ ก.49 ภาพเข้าสู่ระบบ LINE Notify

- เลือก ออก Token เพื่อออก Token ใช้ระบุกลุ่มที่จะแจ้งเตือน



รูปผนวกที่ ก.50 ภาพการออก Access Token เพื่อใช้บริการของ LINE Notify

- ใส่ชื่อ Token เลือกกลุ่มที่ต้องการใช้ Notify ในการแจ้งเตือน และเลือก ออก Token



รูปผนวกที่ ก.51 ภาพการตั้งชื่อ Token และเลือกกลุ่มที่ต้องการแจ้งเตือน

- เมื่อเลือกออก Token แล้วจะมี Token แสดงขึ้นมาให้นำไปใช้และจะแสดง Token เพียงครั้งเดียวเท่านั้น เลือกคัดลอกเพื่อนำ Token ไปใช้



รูปผนวกที่ ก.52 ภาพ Token ที่ออกให้ของ LINE Notify

5. เมื่อออก Token เรียบร้อยแล้วจะแสดงบริการที่เชื่อมต่อกับไลน์ Notify และมีไลน์Notifyแจ้งเตือนเข้ามายังไลน์ให้เพิ่มเข้ากลุ่มไลน์ที่กำหนดไว้ตอนแรก



รูปผนวกที่ ก.53 ภาพการแจ้งเตือนครั้งแรกของ LINE Notify โดยให้เชิญบัญชี LINE Notify เข้าร่วมในกลุ่มไลน์ที่กำหนดไว้ตอนออก Token

6. จากนั้นทดสอบส่งการแจ้งเตือนจากตัวอุปกรณ์

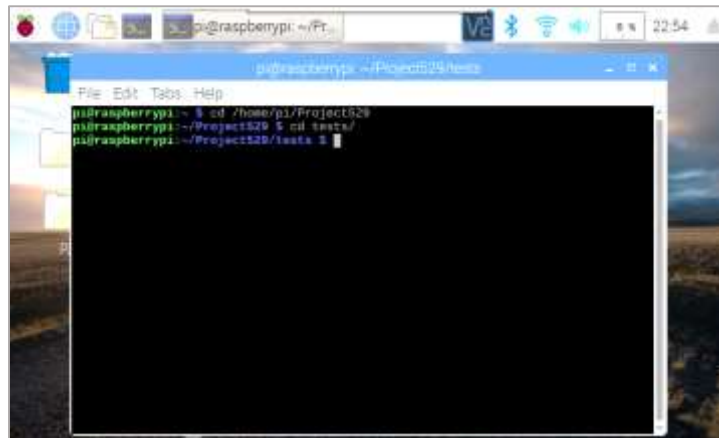


รูปผนวกที่ ก.54 ภาพการแจ้งเตือนเวลาและคิวเวิร์ดจากตัวอุปกรณ์ Raspberry Pi หลังจากผู้ทำกระบวนการถ่ายภาพเพื่อเข้าออกงานเรียบร้อยแล้ว

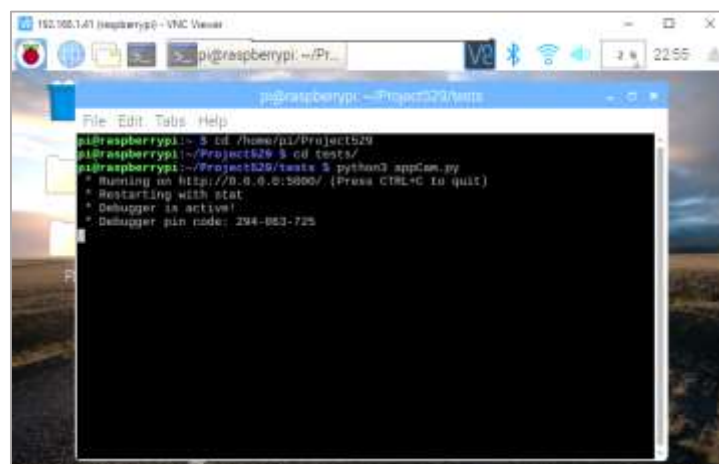
## ภาคผนวก ข คู่มือการใช้งาน

### ข.1 ส่วนของเว็บแอปพลิเคชันบนตัวอุปกรณ์ Raspberry Pi3

ก่อนที่ผู้ใช้จะเข้าใช้เว็บแอปพลิเคชันที่ถูกพัฒนาบนตัวอุปกรณ์ จะต้องทำการรัน Flask ทิ้งไว้ก่อนโดยให้เข้าไปยังโพลเดอร์ที่เก็บไฟล์ python เขียนโดยใช้ Flask Framework จากนั้นรันไฟล์โดยใช้คำสั่ง  
python3 appCam.py

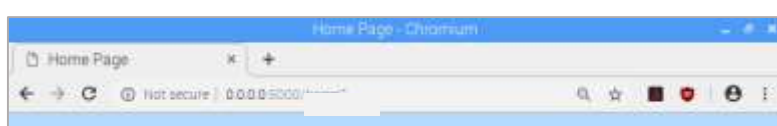


รูปผนวกที่ ข.1 ภาพหน้าจอ Raspberry Pi แสดงการเข้าไปยังที่อยู่ของไฟล์ appCam.py



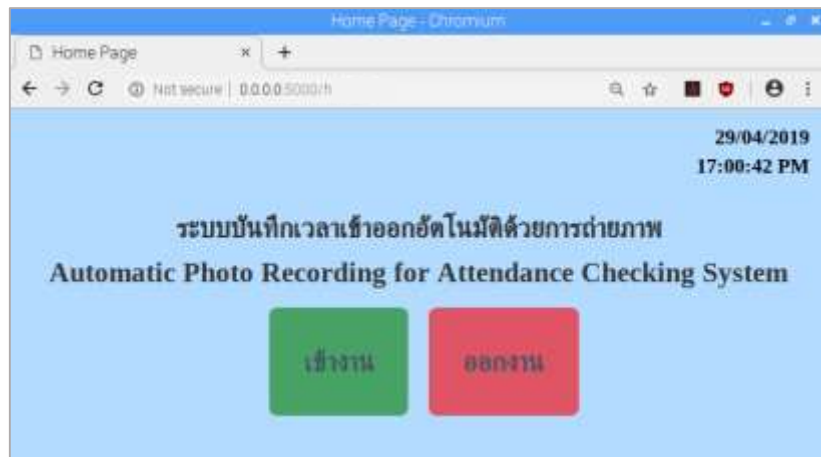
รูปผนวกที่ ข.2 ภาพการรันไฟล์ appCam.py เพื่อใช้ Flask Framework

จากนั้นเปิด Chrome ขึ้นมาเข้าไปที่ 0.0.0.0:5000 จากนั้นจะมีหน้าเว็บแอปพลิเคชันที่พัฒนาไว้แสดงขึ้นมาเป็นหน้าที่ติดต่อกับผู้ใช้งานโดยเป็นหน้าหลักของระบบบันทึกเวลาเข้าออกอัตโนมัติด้วยการถ่ายภาพ ดังรูปผนวกที่ ข.3



รูปผนวกที่ ข.3 ภาพ URL ที่สามารถเข้าถึงหน้าเว็บแอปพลิเคชันที่รัน Flask Framework โดยเข้าถึงได้ที่ 0.0.0.0:5000

เมื่อผู้ใช้งานต้องการบันทึกเวลาเพื่อบอกว่ามาทำงานหรือเลิกงานของวันนั้น ๆ ผู้ใช้ต้องเลือกกดปุ่มเพื่อทำรายการซึ่งมีอยู่ 2 ปุ่ม คือเข้างาน และออกงาน



รูปผนวกที่ ข.4 หน้าหลักของเว็บแอปพลิเคชันบนตัวอุปกรณ์ Raspberry Pi

เมื่อผู้ใช้เลือกปุ่มเข้างานหรือออกงานแล้ว จะแสดงหน้าที่ให้ผู้ใช้ถ่ายภาพ โดยจะมีวิดีโอสดตามภาพตอนนั้น และมีปุ่มอยู่ 2 ปุ่มคือ 1) ปุ่มถ่ายรูป จะถ่ายภาพของผู้ใช้จากตัวอุปกรณ์เมื่อผู้ใช้กดปุ่ม 2) ปุ่มกลับหน้าหลัก คือปุ่มที่เมื่อกดแล้วจะกลับไปหน้าแรก

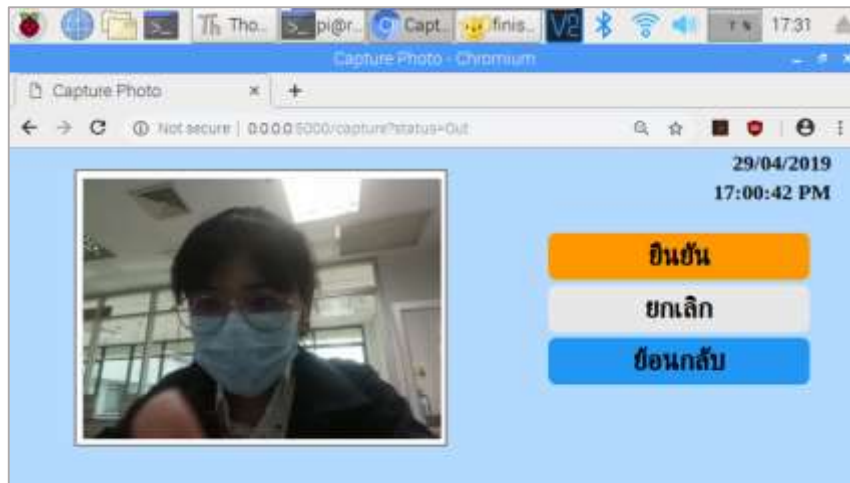


รูปผนวกที่ ข.5 หน้าวิดีโอสดบนเว็บแอปพลิเคชันเพื่อให้ผู้ใช้ทำการถ่ายภาพตนเองบน Raspberry Pi

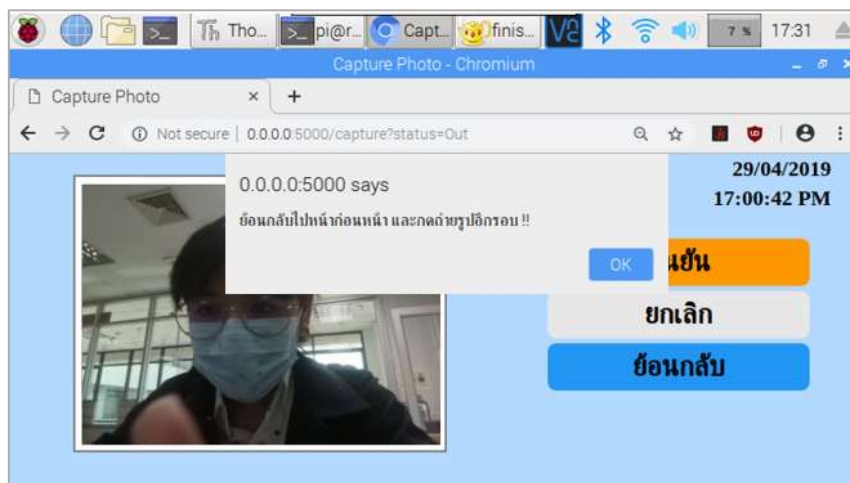
เมื่อผู้ใช้ได้ทำการถ่ายภาพตนเองโดยกดปุ่ม “ถ่ายรูป” แล้วจะแสดงภาพถ่ายที่ถูกถ่ายจากตัวอุปกรณ์ ทางหน้าจอพร้อมปุ่มให้ผู้ใช้เลือกกด 3 ปุ่ม ดังนี้

- 1) ปุ่มยืนยัน เป็นการยืนยันภาพถ่ายเมื่อผู้ใช้กดปุ่ม ภาพถ่ายจะถูกบันทึกในโฟลเดอร์ภาพที่รอการแท็กจากผู้ใช้งาน และแจ้งเตือนเวลาและศักราชไปที่ไลน์กลุ่มที่กำหนด
- 2) ปุ่มยกเลิก เมื่อผู้ใช้กดปุ่มยกเลิก จะแสดง Alert ขึ้นมาบนหน้าจอ
- 3) ย้อนกลับ เมื่อกดปุ่มย้อนกลับจะกลับไปหน้าก่อนหน้า ซึ่งเป็นหน้าให้ผู้ใช้ถ่ายรูปตัวเองใหม่อีกครั้ง



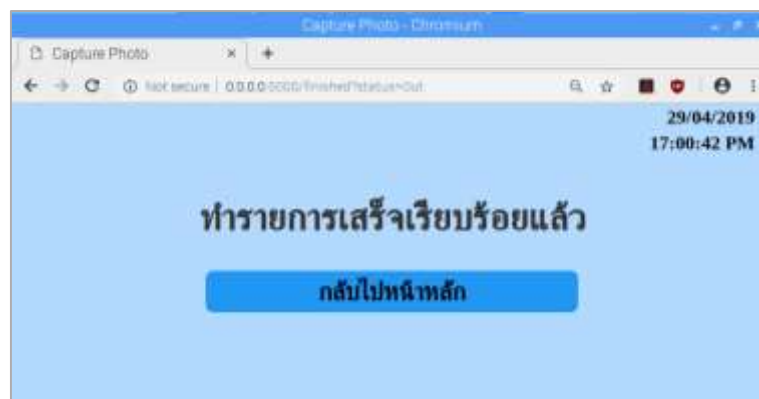


รูปผนวกที่ ข.6 หน้ายืนยันภาพถ่ายที่ถูกถ่ายจากตัวอุปกรณ์ Raspberry Pi



รูปผนวกที่ ข.7 ภาพแสดงการ Alert เมื่อผู้ใช้กดปุ่ม ยกเลิก บนเว็บแอปพลิเคชันที่แสดงบนจอ Raspberry Pi

หน้าทำการรายการเสร็จเรียบร้อยแล้ว และปุ่มกลับไปหน้าหลัก จะแสดงเมื่อผู้ใช้ได้ทำการถ่ายภาพและยืนยันภาพถ่ายตัวเองเสร็จเรียบร้อยแล้ว



รูปผนวกที่ ข.8 หน้าบอกสถานะของกระบวนการถ่ายภาพเพื่อบันทึกเวลาเข้าออกงานเสร็จเรียบร้อยแล้ว

## ข.2 ส่วนของเว็บแอปพลิเคชันบนมือถือที่แสดงบนไลน์

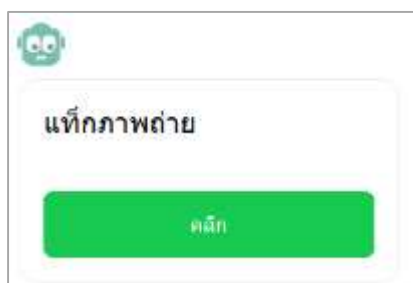
1. ส่วนการแจ้งเตือน เมื่อผู้ใช้ทำการถ่ายภาพและยืนยันภาพถ่ายตัวเองเสร็จเรียบร้อยแล้ว จะใช้ LINE Notify แจ้งเตือนเวลาและคีย์เวิร์ดจากตัวอุปกรณ์มายังไลน์กลุ่ม เพื่อให้ผู้ใช้พิมพ์คุยกับบอทได้อย่างถูกต้อง



รูปผนวกที่ ข.9 ภาพแสดงการแจ้งเตือน เวลา และคีย์เวิร์ดที่ถูกส่งมาจากตัวอุปกรณ์ Raspberry Pi

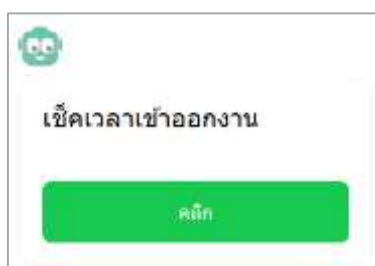
2. ส่วนคีย์เวิร์ดพิมพ์คุยกับบอทในไลน์กลุ่ม เป็นคีย์เวิร์ดที่ผู้ใช้ต้องพิมพ์คุยกับบอทในไลน์กลุ่ม เพื่อทำการแท็กภาพถ่าย หรือดูรายงานการเข้าออกงาน โดยคีย์เวิร์ดที่ผู้ใช้สามารถพิมพ์คุยกับบอทไลน์มีดังนี้

- คีย์เวิร์ดของบอทสำหรับการแท็กภาพถ่าย คือ T, t, TAG, Tag, แท็ก, แท๊ก, แท็กภาพถ่าย, สถานะการแท็ก เป็นต้น



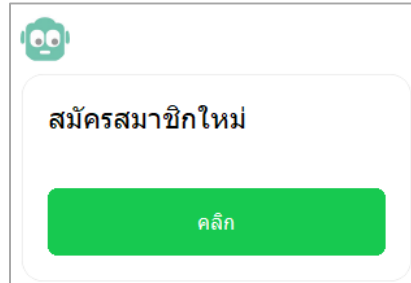
รูปผนวกที่ ข.10 การตอบกลับของบอทไลน์เพื่อแท็กภาพถ่าย

- คีย์เวิร์ดของบอทสำหรับดูรายงานการเข้าออกงาน คือ R, r, Report, Reports, report, reports, รายงาน, เช็คเวลาเข้าออกงาน, รายงานเวลาเข้าออกงาน เป็นต้น



รูปผนวกที่ ข.11 การตอบกลับของบอทไลน์เพื่อเช็คเวลาเข้าออกงาน

- คีย์เวิร์ดของบอทสำหรับสมัครสมาชิกใหม่ จะสมัครเพียงครั้งแรกเท่านั้นที่เข้ากลุ่มไลน์เท่านั้นและไม่สามารถสมัครซ้ำได้อีกเพราะจะใช้ UserID LINE ของแต่ละคนในการระบุตัวตน โดยคีย์เวิร์ดมีดังนี้ M, m, Member, MERBER, member, สมัครสมาชิก, เพิ่มสมาชิกใหม่, สมาชิกใหม่ เป็นต้น



รูปผนวกที่ ข.12 การตอบกลับของบอทไลน์เพื่อสมัครสมาชิกใหม่

- คีย์เวิร์ดของบอทสำหรับเป็นตัวช่วย คือ help, All, Help, HELP, all, ตัวช่วย เป็นต้น ซึ่งจะช่วยแสดงรายการทั้งแท็กภาพถ่าย และดูรายงานการเข้าออกงาน



รูปผนวกที่ ข.13 การตอบกลับของบอทไลน์เพื่อแท็กภาพถ่ายและเช็คเวลาเข้าออกงาน

**3. ส่วนการแท็กภาพถ่าย** เมื่อผู้พิมพ์คีย์เวิร์ดสำหรับแท็กภาพถ่ายลงในไลน์กลุ่ม บอทจะตอบกลับมาเป็น Flex Message เมื่อต้องการทำการแท็กภาพถ่ายให้ผู้พิมพ์คีย์เวิร์ดที่ตอบกลับมาจากบอทดังรูปผนวกที่ ข.14



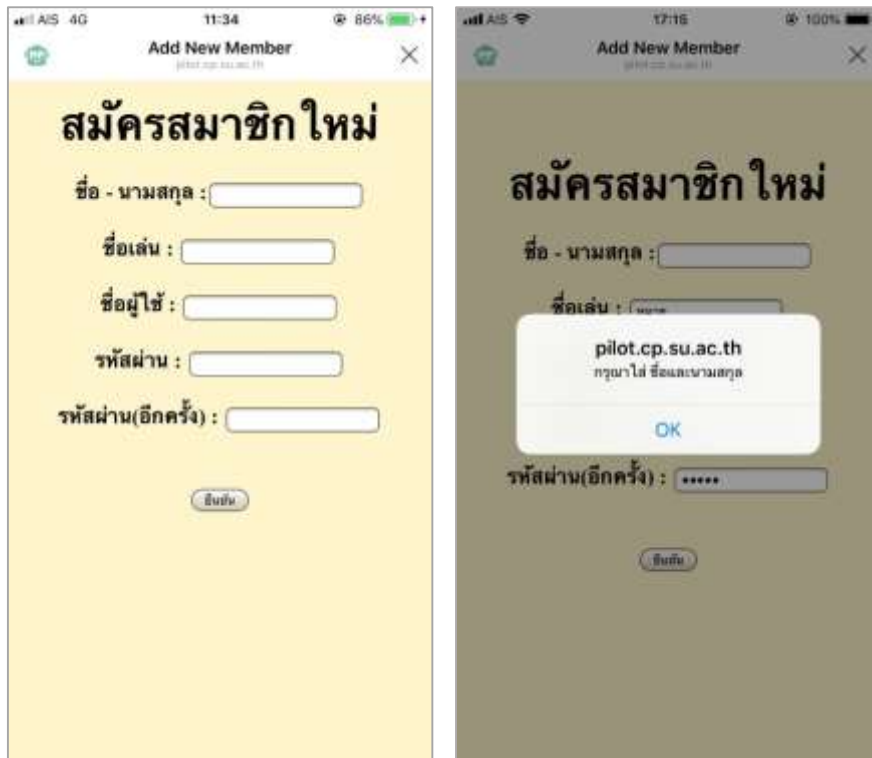
รูปผนวกที่ ข.14 การตอบกลับของบอทไลน์เมื่อผู้ใช้พิมพ์คีย์เวิร์ดเพื่อแท็กภาพถ่าย

จากภาพเป็นหน้าเว็บแอปพลิเคชันที่แสดงบนกลุ่มไลน์แสดงบนมือถือโดยใช้บริการของไลน์ (LINE LIFF) แสดงภาพถ่ายที่ถูกถ่ายจากตัวอุปกรณ์ Raspberry Pi3 และยังไม่ได้แท็กเจ้าของภาพถ่าย เมื่อผู้ใช้ต้องการแท็กเจ้าของภาพถ่าย ให้กดไปที่ภาพนั้น ๆ ดังรูปผนวกที่ ข.15



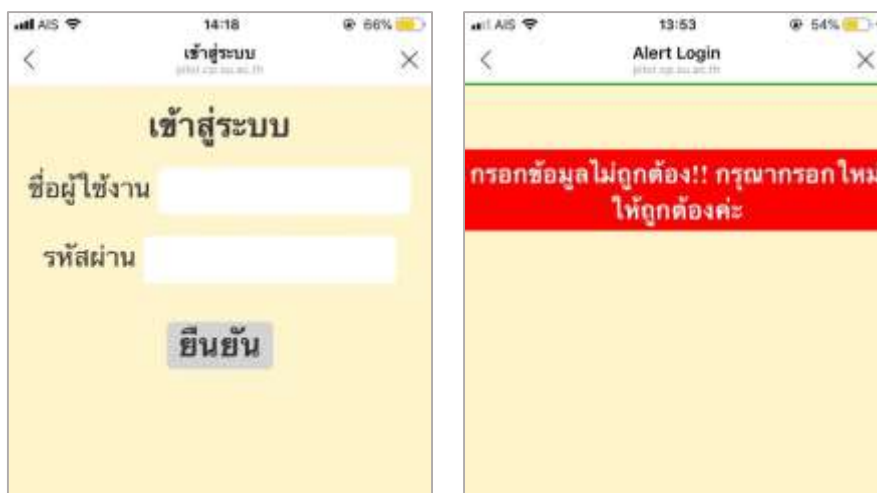
รูปผนวกที่ ข.15 หน้าเว็บแอปพลิเคชันภาพที่ยังไม่ถูกแท็กเจ้าของภาพ

จากนั้นจะแสดงหน้าเข้าสู่ระบบให้ผู้ใช้งานทำการเข้าสู่ระบบก่อนทำการแท็กเจ้าของภาพนั้น ๆ โดยหากผู้ใช้อย่างไม่เป็นสมาชิกในระบบ จะต้องสมัครสมาชิกก่อนและต้องกรอกข้อมูลทุกช่องให้ครบ หากกรอกไม่ครบจะเตือนให้ผู้ใช้งานกรอกให้ครบดังรูปผนวกที่ ข.16 และเมื่อสมัครสมาชิกเสร็จเรียบร้อยแล้วจะกลับไปยังหน้าเข้าสู่ระบบเพื่อทำการแท็กภาพถ่ายต่อนั่นเอง



รูปผนวกที่ ข.16 หน้าเว็บแอปพลิเคชันสมัครสมาชิกใหม่เข้าสู่ระบบและการ Alert เมื่อผู้ใช้กรอกข้อมูลไม่ครบ

เป็นหน้าเข้าสู่ระบบเพื่อทำการแท็กภาพถ่าย โดยผู้ใช้เป็นสมาชิกในระบบแล้ว และเลือกภาพถ่าย ต้องการแท็กเจ้าของภาพถ่ายภาพแล้ว ผู้ใช้จะต้องทำการเข้าสู่ระบบให้ถูกต้อง หากกรอกชื่อผู้ใช้งาน หรือกรอกรหัสผ่านผิด ก็จะมีการ Alert บอกว่าผู้ใช้ได้กรอกข้อมูลไม่ถูกต้องให้กรอกให้ถูกต้อง



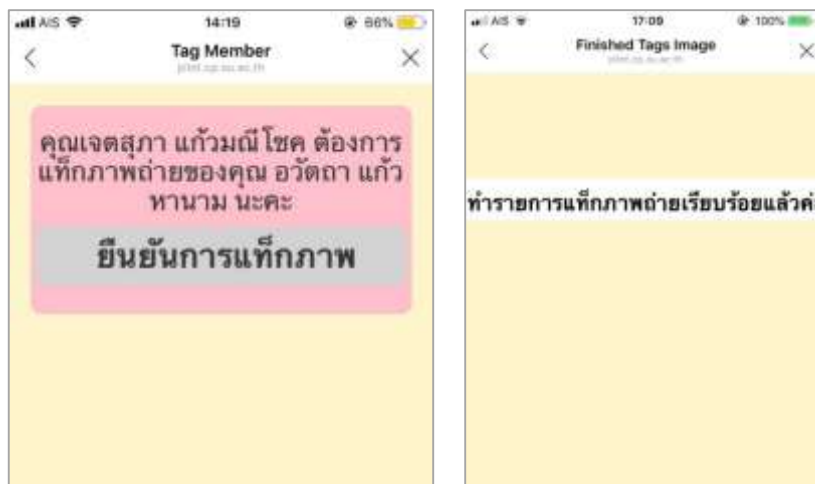
รูปผนวกที่ ข.17 หน้าเว็บแอปพลิเคชันหน้าเข้าสู่ระบบและการ Alert เมื่อผู้ใช้กรอกข้อมูลไม่ถูกต้อง

เมื่อผู้ใช้เข้าสู่ระบบเรียบร้อยแล้ว จะแสดงหน้าแท็กเจ้าของภาพถ่าย จะมีภาพที่ผู้ใช้เลือกและมีปุ่มที่เป็นชื่อเล่นของผู้ใช้ทั้งหมด ให้ผู้ใช้เลือกกดเพื่อทำการแท็กเจ้าของภาพดังรูปผนวกที่ ข.18



รูปผนวกที่ ข.18 หน้าเว็บแอปพลิเคชันหน้าแท็กเจ้าของภาพถ่าย

เป็นหน้ายืนยันการแท็กภาพ เป็นการยืนยันก่อนจะแท็กภาพถ่าย โดยจะมีชื่อคนที่แท็กและชื่อเจ้าของภาพที่ถูกแท็ก หากต้องการยืนยันผู้ใช้ต้องกดปุ่มยืนยันการแท็กภาพ และจะแสดงหน้าจอผู้ใช้ว่ารายการแท็กสำเร็จแล้ว



รูปผนวกที่ ข.19 หน้าเว็บแอปพลิเคชันหน้ายืนยันการแท็กเจ้าของภาพถ่าย

เมื่อผู้ใช้ทำการแท็กภาพถ่ายเรียบร้อยแล้ว จะกลับไปยังหน้าภาพที่ยังไม่ได้แท็กเจ้าของภาพ



รูปผนวกที่ ข.20 หน้าเว็บแอปพลิเคชันหน้าภาพถ่ายที่ยังไม่ถูกแท็กเจ้าของภาพ

#### 4. ส่วนดูรายงานการเข้าออกงาน

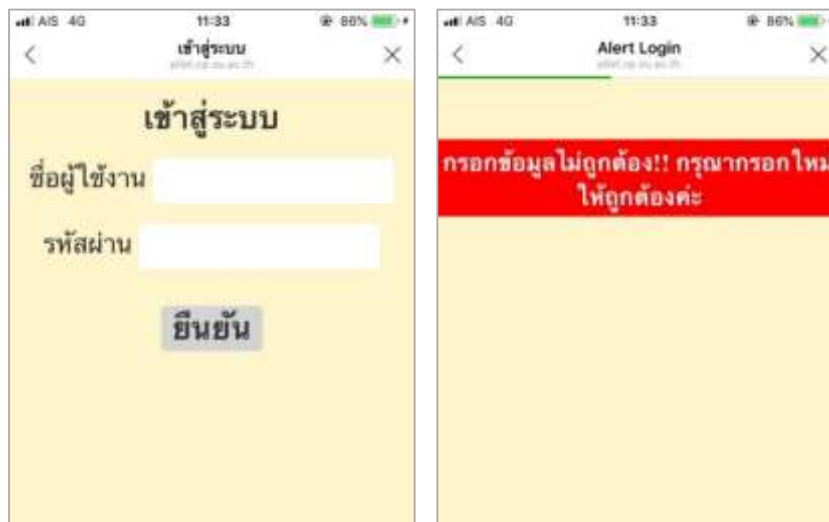
เมื่อผู้ใช้พิมพ์คีย์เวิร์ดสำหรับดูรายงานการเข้าออกงานลงในไลน์กลุ่ม บอทจะตอบกลับมาเป็น Flex Message เมื่อต้องการทำรายงานให้ผู้ใช้กดปุ่มที่ตอบกลับมาจากบอท



รูปผนวกที่ ข.21 การตอบกลับของบอทไลน์เมื่อผู้ใช้พิมพ์คีย์เวิร์ดเพื่อเช็คเวลาเข้าออกงาน

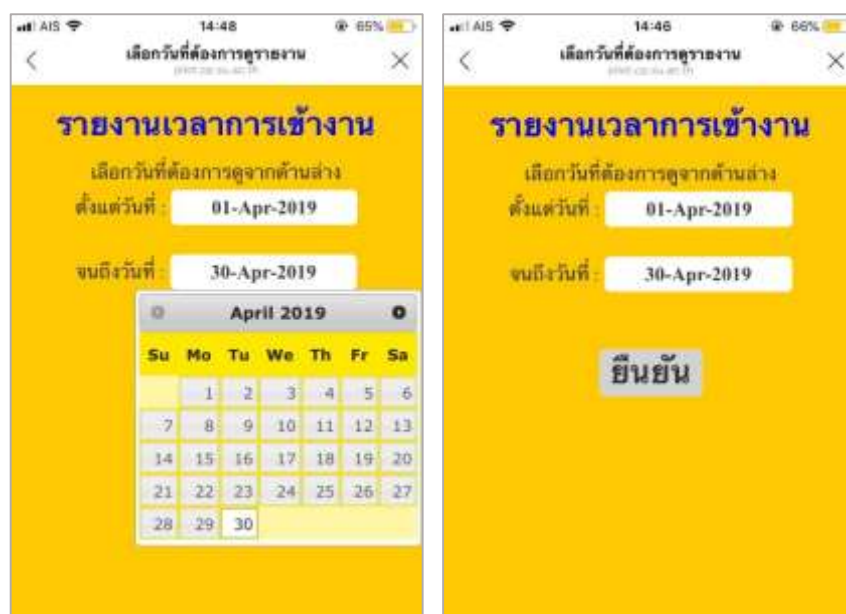


ก่อนที่ผู้ใช้ดูรายงานการเข้าออกงานจะต้องเข้าสู่ระบบก่อน หากกรอกชื่อผู้เข้าใช้งาน หรือกรอกรหัสผ่าน ผิด ก็จะมีการ Alert บอกว่าผู้ใช้ได้กรอกข้อมูลไม่ถูกต้องให้กรอกใหม่ให้ถูกต้องดังรูปผนวกที่ ข.22



รูปผนวกที่ ข.22 หน้าเว็บแอปพลิเคชันหน้าเข้าสู่ระบบและการ Alert เมื่อผู้ใช้กรอกข้อมูลไม่ถูกต้อง

เมื่อเข้าสู่ระบบเพื่อดูรายงานการเข้างานออกงาน หากระบบเช็คว่าเป็นผู้ใช้งาน(User) เข้าสู่ระบบจะแสดงหน้าให้เลือกช่วงวันที่ที่ต้องการดูรายงานแบบรายเดือน โดยผู้ใช้งานจะต้องเลือกวันที่ให้ถูกต้อง ระบบจะมีการตรวจสอบความผิดพลาดเมื่อเลือกวันที่ช่วงท้ายน้อยกว่าหรือเลือกเกินวันที่ปัจจุบัน ก็จะไม่สามารถทำรายการได้ และจะอยู่หน้าเดิมให้ผู้เลือกใหม่อีกครั้งดังรูปผนวกที่ ข.23



รูปผนวกที่ ข.23 หน้าเว็บแอปพลิเคชันหน้าเลือกวันที่ต้องการดูรายงานการเข้าออกงาน

หากรบบเช็คว่าเป็นผู้ดูแลระบบ (Admin) เข้าสู่ระบบเพื่อดูรายงานการเข้างานออกงาน จะแสดงหน้าจอให้เลือกเลือกรายชื่อผู้ใช้งานในระบบที่ต้องการดูรายงานการเข้างานแบบรายเดือนเพื่อตรวจสอบดูการเข้าออกงานของผู้ใช้งานในระบบเบื้องต้นก่อน ดังรูปผนวกที่ ข.24



รูปผนวกที่ ข.24 หน้าเว็บแอปพลิเคชันหน้ารายชื่อผู้ใช้งานเพื่อดูรายงานเวลาเข้างานออกงาน

เมื่อผู้ใช้เลือกช่วงวันที่ที่ต้องการดูรายงานการเข้าออกงาน หรือผู้ดูแลระบบเลือกคนที่ต้องการดูรายงานการเข้าออกงานแล้ว จะแสดงหน้ารายงานเวลาการเข้างานดังรูปผนวกที่ ข.25 โดยจะแสดงชื่อและนามสกุล และแสดงเป็นช่วงวันที่ที่เข้าออกงานโดย จะแสดงวันที่ เวลาเข้างาน เวลาออกงาน และภาพถ่าย

The screenshot shows a mobile app interface titled 'Work Reports' with a subtitle 'pilot.ccp.ac.th'. The main heading is 'รายงานเวลาการเข้างาน' (Work Time Report). Below it is the instruction 'อวดตา แก้วหานาม' (Show to Kaewhanam). There is a table with the following data:

วันที่	เวลาเข้า	เวลาออก	ภาพถ่าย
2019-04-26	15:59:49	16:30:20	คลิก
2019-04-29	08:36:42	17:09:21	คลิก

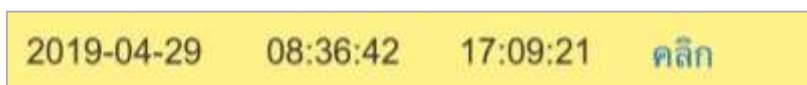
รูปผนวกที่ ข.25 หน้าเว็บแอปพลิเคชันหน้ารายงาน ช่วงเวลาในการเข้าออกงานของผู้ใช้งาน

หากผู้ใช้งานลืมบันทึกเวลาช่วงเช้าหรือออกงาน ช่วงเวลานั้น ๆ จะแสดง ( - ) ดังรูปผนวกที่ ข.26



รูปผนวกที่ ข.26 หน้าเว็บแอปพลิเคชันหน้าช่วงเวลาเข้าออกงานของผู้ใช้งานเป็นเครื่องหมาย ( - ) เมื่อผู้ใช้ไม่ได้บันทึกเวลาช่วงเข้าออกงานในช่วงนั้น ๆ

เมื่อผู้ใช้งานต้องการดูภาพถ่าย ให้กดที่ “คลิก” ดังรูปผนวกที่ ข.27 จากนั้นจะแสดงภาพถ่ายและเวลาที่ถ่ายภาพดังรูปผนวกที่ ข.28



รูปผนวกที่ ข.27 ภาพวันที่ ช่วงเวลาเข้า ช่วงเวลาออก และคลิกเพื่อดูภาพถ่าย



รูปผนวกที่ ข.28 หน้าเว็บแอปพลิเคชันหน้าเวลาและภาพถ่ายของผู้ใช้งาน