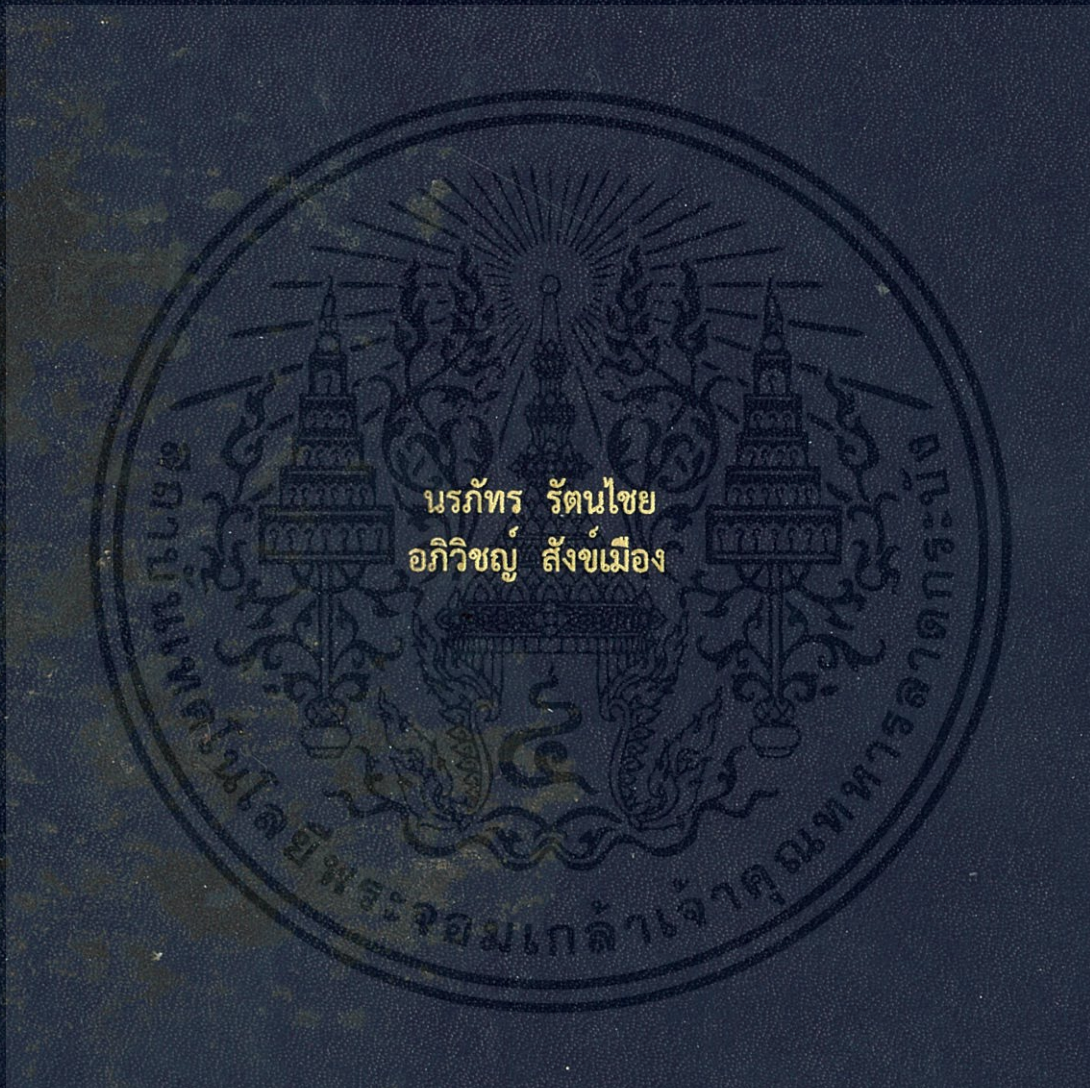


ระบบแจ้งเตือนทางแอปพลิเคชันไลน์  
สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์

ALERT SYSTEM VIA LINE APPLICATION  
FOR DEVICES IN THE SERVER ROOM



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)  
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบแจ้งเตือนทางแอปพลิเคชันไลน์  
สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์

ALERT SYSTEM VIA LINE APPLICATION  
FOR DEVICES IN THE SERVER ROOM



นรภัทร รัตนไชย  
อภิวิชญ์ สังข์เมือง

เลขหมู่.....  
เลขทะเบียน..... 148953  
วัน,เดือน,ปี..... ๗.๘ S.ค. 2560

b. 1๒๘๖๙๗๒1  
i. ....

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)  
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALERT SYSTEM VIA LINE APPLICATION  
FOR DEVICES IN THE SERVER ROOM



NORAPHAT RATTANACHAI  
APIWICH SUNGMUANG

A COOPERATIVE EDUCATION SUBMITTED IN  
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)  
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADEMIC YEAR 2015



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา ระบบแจ้งเตือนทางแอปพลิเคชันไลน์สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์  
Alert System via Line Application for Devices in the  
Server Room

ชื่อนักศึกษา นายนรภัทร รัตน์ไชย รหัสนักศึกษา 55050339  
นายอภิวิชญ์ สังข์เมือง รหัสนักศึกษา 55050530

ปริญญา วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)  
ภาควิชา วิทยาการคอมพิวเตอร์  
คณะ วิทยาศาสตร์  
ปีการศึกษา 2558  
อาจารย์ที่ปรึกษา ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง(สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร. วรางคณา กิมปาน ประธานกรรมการ	
ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	ระบบแจ้งเตือนทางแอปพลิเคชันสำหรับอุปกรณ์ในท้องเซิร์ฟเวอร์
ชื่อนักศึกษา	นายนรภัทร รัตนไชย รหัสนักศึกษา 55050339 นายอภิวิชญ์ สังข์เมือง รหัสนักศึกษา 55050530
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์

### บทคัดย่อ

การที่แอปพลิเคชันไลน์มีผู้ใช้งานเป็นจำนวนมาก โดยมีการใช้งานอย่างแพร่หลายทั้งในเรื่องส่วนตัวและในการกระจายข่าวสารประชาสัมพันธ์ รวมทั้งการติดต่อสื่อสารระหว่างบุคคลในองค์กร และระบบการแจ้งเตือนปัญหาอุปกรณ์ในท้องเซิร์ฟเวอร์ในปัจจุบันใช้การส่งข้อความผ่านทางเอสเอ็มเอสหรืออีเมลไปที่ผู้ดูแลระบบและเจ้าหน้าที่ฝ่ายเทคนิค ซึ่งผู้ส่งไม่สามารถทราบได้ว่าผู้รับอ่านข้อความแล้วหรือไม่ โครงการสหกิจศึกษานี้จึงมีวัตถุประสงค์ในการใช้แอปพลิเคชันไลน์ในการแจ้งเตือนปัญหาในท้องเซิร์ฟเวอร์เพื่อให้ระบบสามารถทราบได้ว่าผู้รับอ่านข้อความแล้วหรือไม่ เมื่อโปรแกรมตรวจสอบระบบเครือข่ายที่ชื่อว่า PRTG Network Monitor ในท้องเซิร์ฟเวอร์ตรวจจับปัญหาได้ จะทำการส่งข้อมูลผ่านโปรโตคอลเอชทีทีพี (http) ไปที่ระบบซึ่งติดตั้งบนราสเบอร์รี่พายที่เชื่อมต่อกับระบบอินเทอร์เน็ตผ่านทางพอร์ตยูเอสบี ระบบจะทำการอ่านชื่อผู้ที่เกี่ยวข้องจากไฟล์ที่ทำการบันทึกข้อมูลของผู้เกี่ยวข้องตามลำดับชั้นของตำแหน่งงานไว้และส่งข้อมูลไปที่แอปพลิเคชันไลน์โดยใช้เอพีไอ LineSharp การส่งข้อมูลไปที่เจ้าหน้าที่จะส่งตามลำดับรายชื่อในไฟล์ ซึ่งถ้าไม่มีการอ่านข้อมูลในระยะเวลาที่กำหนดระบบจะทำการส่งข้อมูลไปยังหัวหน้างานในลำดับถัดไป ระบบจะหยุดส่งข้อความเมื่อมีผู้อ่านข้อความนั้น ผู้จัดทำคาดว่าระบบนี้จะให้ประโยชน์ในการลดค่าใช้จ่ายในการส่งข้อความเตือน เนื่องจากราสเบอร์รี่พายมีราคาถูก มีขนาดเล็ก และง่ายในการติดตั้ง

คำสำคัญ : การแจ้งเตือน โซเชียลเน็ตเวิร์ก โปรโตคอลเอชทีทีพี พอร์ตยูเอสบี ระบบปฏิบัติการลินุกซ์ สมาร์ทโฟน หน่วยความจำ อินเทอร์เน็ต แอปพลิเคชันไลน์ ฮาร์ดแวร์

<b>Title</b>	Alert System via Line Application for Devices in the Server Room		
<b>Students</b>	Mr. Noraphat Rattanachai	Student ID 55050339	
	Mr. Apiwich Sungmuang	Student ID 55050530	
<b>Degree</b>	Bachelor of Science (Computer Science)		
<b>Department</b>	Computer Science		
<b>Faculty</b>	Science		
<b>University</b>	King Mongkut's Institute of Technology Ladkrabang (KMITL)		
<b>Academic Year</b>	2558		
<b>Advisor</b>	Dr. Rungrat Wiangsripanawan		

### Abstract

Line application is used by a number of users and widely used both in personnel and in business as a new channel to distribute news and to communicate among users. Also, the current notification system for problems in the server room uses SMS and email to notify problems to the system administrators or the technicians. This does not inform the sender whether the message is read or not. Therefore, this cooperative project aims at using the Line application to notify problems in the server room so that the system will know whether the recipient has read the message yet. When the PRTG Network Monitoring in the server room detects the problem, it will send the http message to the alert system which is installed on the Raspberry Pi which is connected to the Internet via USB port. Then, the system retrieves its administrators and system engineers' Line contact from the configuration file and uses LineSharp API to send the alert message to Line application. The contact list is arranged according to the order of command in the organization. If there is no response within the timeout interval, the system will send the message to his superintendent until the end of the list. It is expected that this system will help to reduce costs in sending alert messages. In addition, Raspberries Pi is cheap, small and easy to plugin to the system.

**Keywords :** Alert, Social Network, HTTP Protocol, USB Port, Linux Operation System, Internet, Smartphone, Memory, Line application, Hardware

## กิตติกรรมประกาศ

ในการจัดทำสหกิจศึกษาเรื่องระบบแจ้งเตือนทางแอปพลิเคชันสำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์ (Alert System via Line Application for Devices in the Server Room) สำเร็จลุล่วงได้ เนื่องจากได้รับความอนุเคราะห์จากบุคคลหลายท่าน

ขอขอบพระคุณ ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ อาจารย์ที่ปรึกษาสหกิจ ที่ได้ให้คำปรึกษา คำแนะนำ รวมถึงการแก้ไข และตรวจสอบข้อบกพร่องต่างๆ ด้วยความเอาใจใส่เป็นอย่างดี เพื่อให้สหกิจนี้สมบูรณ์มากที่สุด

ขอขอบคุณอาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ทุกๆ ท่านที่ได้ประสิทธิ์ประสาทความรู้ในทั้งภาคทฤษฎีและภาคปฏิบัติแก่ผู้จัดทำมาตลอดเวลา 4 ปี จนกระทั่งสหกิจศึกษาเสร็จสมบูรณ์ได้ด้วยดี

ขอขอบพระคุณ คุณธีรยุทธ พึ่งเกียรติไพบูลย์ และคุณฉัตรชัย ตั้งวัฒนา ที่คอยดูแล ให้คำปรึกษาและชี้แนวทางการศึกษาตลอดช่วงการทำสหกิจศึกษาที่บริษัท ซีดีจี ซิสเต็มส์ จำกัด

สุดท้ายนี้ ขอขอบพระคุณบิดา มารดา ที่คอยให้กำลังใจ ให้คำปรึกษา และให้การสนับสนุนมาโดยตลอด และขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกท่านที่ให้คำปรึกษา คอยให้กำลังใจ คอยเคียงข้าง และคอยช่วยเหลือในการทำสหกิจศึกษาเสมอมา

นรภัทร รัตนไชย  
อภิวินัย สังข์เมือง

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ก
บทคัดย่อภาษาอังกฤษ .....	ข
กิตติกรรมประกาศ .....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการดำเนินงาน.....	1
1.3 ขอบเขตของการดำเนินงาน .....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
1.6 อุปกรณ์ที่ใช้ในการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง .....	2
2.1 เทคโนโลยีไร้สาย (Wireless Technology).....	2
2.2 โมบายเทคโนโลยี (Mobile Technology) .....	2
2.3 เทคโนโลยีการติดต่อสื่อสารบนโทรศัพท์มือถือ.....	4
2.3.1 เทคโนโลยี 3G.....	4
2.3.2 เทคโนโลยี 4G.....	5
2.4 โซเชียลแอปพลิเคชันไลน์.....	5
2.5 LineSharp.....	6
2.6 คอมพิวเตอร์ขนาดเล็ก Raspberry Pi.....	6
2.7 PRTG Network Monitor.....	9
บทที่ 3 วิธีดำเนินการ.....	9
3.1 สถาปัตยกรรมระบบ (Architecture).....	9
3.2 Use Case Diagram.....	12

## สารบัญ (ต่อ)

	หน้า
3.3 Activity Diagram .....	13
3.3.1 การตั้งค่าระบบแจ้งเตือนฯ.....	14
3.3.2 การเริ่มต้นระบบแจ้งเตือนฯ.....	15
3.3.3 การรับคำร้องจาก PRTG Network Monitor.....	16
3.3.4 การส่งข้อความผ่านแอปพลิเคชันไลน์ .....	17
3.3.5 การตรวจสอบการอ่านข้อความ.....	18
3.3.6 การส่งข้อความไปยังบุคคลถัดไป.....	19
3.3.7 การตรวจสอบล็อกไฟล์สถานะการทำงาน .....	20
3.4 Sequence Diagram .....	21
3.4.1 การตั้งค่าข้อมูลผู้ติดต่อสำหรับระบบไลน์และเปิดการเชื่อมต่ออินเทอร์เน็ต .....	22
3.4.2 การเริ่มต้นใช้งานระบบแจ้งเตือนฯ.....	23
3.4.3 การแจ้งเตือน (กรณีผู้รับอ่านข้อความ).....	24
3.4.4 การแจ้งเตือน (กรณีผู้รับไม่ได้อ่านข้อความ).....	25
3.5 Flowchart Diagram .....	27
3.5.1 การทำงานของโปรแกรมจากสถานะเริ่มต้นจนถึงสถานะเตรียมพร้อม.....	28
3.5.2 การทำงานของเซรตสำหรับดักจับอีเว้นท์ที่เกิดขึ้นบนระบบ.....	30
3.5.3 การทำงานของเซรตส่งข้อความ .....	31
3.6 การออกแบบระบบ .....	33
3.6.1 การตั้งค่าไฟล์ Configuration .....	33
3.6.2 ฟังก์ชันการเพิ่มระดับความสำคัญของบุคคลในการแจ้งเตือน .....	35
3.6.3 โปรแกรมส่งข้อความโดยใช้ LineSharp .....	36
3.6.4 กระบวนการทำงาน .....	38
บทที่ 4 ผลลัพธ์ของระบบ.....	41
4.1 ขั้นตอนการทดสอบระบบ .....	41
4.1.1 เปิดการใช้งานระบบแจ้งเตือน Social Alert.....	41
4.1.2 ทดสอบปิดการเชื่อมต่อเครือข่ายเน็ตเวิร์ค.....	43
4.1.3 ตรวจสอบผลลัพธ์ที่เกิดขึ้น .....	46

## สารบัญ (ต่อ)

	หน้า
4.2 ข้อเปรียบเทียบระหว่างการแจ้งเตือนผ่านข้อความ SMS กับ LINE .....	49
บทที่ 5 สรุปผลการดำเนินการ.....	50
5.1 ข้อสรุปผลการดำเนินงาน.....	50
5.2 ข้อเสนอแนะ.....	51
5.3 ข้อจำกัด.....	51
เอกสารอ้างอิง .....	52
ภาคผนวก .....	52
ภาคผนวก ก. การติดตั้งโปรแกรมระบบแจ้งเตือนๆ สำหรับใช้งานบนอุปกรณ์ Raspberry Pi.....	54
ภาคผนวก ข. การติดตั้ง Android Debug Bridge (adb).....	57
ภาคผนวก ค. การเปิดฟังก์ชันการใช้งานอินเทอร์เน็ต.....	63
ภาคผนวก ง. การตั้งค่าโปรแกรม PRTG Network Monitor.....	70
ภาคผนวก จ. การทดสอบปิดการเชื่อมต่อเครือข่าย .....	76
ภาคผนวก ฉ. การใช้งานระบบแจ้งเตือนปัญหาอุปกรณ์ผ่านแอปพลิเคชันไลน์.....	83

# สารบัญตาราง

ตารางที่	หน้า
2.1 รายละเอียดคุณสมบัติทางเทคนิคของบอร์ด Raspberry Pi .....	7
4.1 ข้อเปรียบเทียบระหว่างการแจ้งเตือนผ่านข้อความ SMS กับ LINE.....	49



# สารบัญรูป

รูปที่	หน้า
2.1 ระบบปฏิบัติการของ Raspberry Pi .....	6
2.2 ส่วนประกอบของบอร์ด Raspberry Pi 2 .....	7
2.3 โปรแกรม PRTG Network Monitor .....	10
3.1 สถาปัตยกรรมของระบบแจ้งเตือนทางแอปพลิเคชันไลน์ .....	11
3.2 Use Case Diagram ระบบแจ้งเตือนทางแอปพลิเคชันไลน์ .....	12
3.3 Activity Diagram การตั้งค่าระบบแจ้งเตือนฯ.....	14
3.4 Activity Diagram การเริ่มต้นระบบแจ้งเตือนฯ.....	15
3.5 Activity Diagram การรับคำร้องจาก PRTG Network Monitor .....	16
3.6 Activity Diagram การส่งข้อความผ่านแอปพลิเคชันไลน์ .....	17
3.7 Activity Diagram การตรวจสอบการอ่านข้อความ .....	18
3.8 Activity Diagram การส่งข้อความไปยังบุคคลถัดไป.....	19
3.9 Activity Diagram การตรวจสอบล็อกไฟล์สถานะการทำงาน .....	20
3.10 Sequence Diagram ของการตั้งค่าข้อมูลผู้ติดต่อสำหรับระบบไลน์.....	22
3.11 Sequence Diagram การเริ่มต้นใช้งานระบบแจ้งเตือนฯ.....	23
3.12 Sequence Diagram การแจ้งเตือน (กรณีผู้รับอ่านข้อความ).....	24
3.13 Sequence Diagram การแจ้งเตือน (กรณีผู้รับไม่ได้อ่านข้อความ).....	25
3.14 Flowchart Diagram การทำงานของโปรแกรมจาก .....	28
3.15 Flowchart Diagram การทำงานของเซิร์ฟเวอร์ที่เก็บข้อมูลที่เกิดขึ้นบนระบบ .....	30
3.16 Flowchart Diagram การทำงานของเซิร์ฟเวอร์ส่งข้อความ.....	31
3.17 โครงสร้างไฟล์ Configuration ที่ใช้เป็นข้อมูลสำหรับส่งข้อความ .....	33
3.18 แสดงการจำลองฟังก์ชันเพิ่มระดับความสำคัญของบุคคลในการแจ้งเตือน .....	35
3.19 โปรแกรมส่งข้อความและรับข้อความเบื้องต้น .....	36
3.20 ผลลัพธ์ของโปรแกรม .....	37
3.21 การรีโมตสำหรับใช้งาน Raspberry PI.....	38
3.22 ได้รับคำร้องขอจากเซิร์ฟเวอร์ PRTG.....	39
3.23 ส่งข้อความผ่าน API Line .....	40
3.24 ข้อความทางไลน์ของผู้รับ.....	40
4.1 คำสั่งที่ใช้งานโปรแกรม.....	41

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.2 ตรวจสอบอุปกรณ์ที่เชื่อมต่อ .....	42
4.3 ผลลัพธ์เมื่อไม่พบการเชื่อมต่ออุปกรณ์ .....	42
4.4 การจัดเตรียมข้อมูลสำหรับการส่งข้อความผ่านไลน์เรียบบร้อย .....	43
4.5 เซิร์ฟเวอร์สำหรับทดสอบ (Ubuntu-SocialAlert).....	43
4.6 ปิดการทำงาน Network adapter .....	44
4.7 สถานะการทดสอบอินเทอร์เน็ตเป็น Up .....	44
4.8 สถานะการทดสอบอินเทอร์เน็ตเป็น Warning .....	45
4.9 หน้าต่างย่อยแสดงรายการข้อผิดพลาดที่เกิดขึ้น .....	45
4.10 สถานะการทดสอบอินเทอร์เน็ตเป็น Down .....	46
4.11 พารามิเตอร์ที่ได้รับจาก URL .....	46
4.12 กระบวนการส่งข้อความผ่านไลน์.....	47
4.13 ข้อความที่ถูกส่งไปยังไลน์ผู้รับ .....	47
4.14 ระบบไลน์อยู่ในสถานะพร้อมใช้งาน .....	48
4.15 สถานะการทำงาน success.....	48
4.16 สถานะการทำงาน fail .....	48
4.17 ข้อมูลที่บันทึกบนคล็อดไฟล์ .....	49
ก.1 หน้าแรกโปรแกรม FileZilla .....	54
ก.2 กรอกรายละเอียดสำหรับเชื่อมต่อ Raspberry Pi.....	54
ก.3 แสดงหน้าต่างเชื่อมต่อสำเร็จ .....	55
ก.4 การอัปโหลดไฟล์โปรแกรมระบบแจ้งเตือนฯ .....	55
ก.5 หน้าต่างแสดงสถานะการอัปโหลด .....	56
ข.1 หน้าแรกโปรแกรม FileZilla.....	57
ข.2 กรอกรายละเอียดสำหรับเชื่อมต่อ Raspberry Pi.....	57
ข.3 แสดงหน้าต่างเชื่อมต่อสำเร็จ .....	58
ข.4 การอัปโหลดไฟล์โปรแกรม adb .....	58
ข.5 หน้าต่างแสดงสถานะการอัปโหลด .....	59
ข.6 หน้าแรกโปรแกรม PuTTY .....	59
ข.7 หน้าแรก SSH Secure Shell บน Raspberry Pi.....	60

## สารบัญรูป (ต่อ)

รูปที่	หน้า
ข.8 หน้าแสดงผลจากคำสั่ง ls -l.....	60
ข.9 หน้าแสดงผลจากคำสั่ง chmod 755 adb และ ls -l .....	61
ข.10 คำสั่งสำหรับคัดลอกไฟล์ adb.....	61
ข.11 หน้าแสดงการติดตั้ง Android debug bridge เรียบร้อย .....	62
ข.12 หน้าแสดงผลการเชื่อมต่อ Smartphone เข้ากับ Raspberry Pi .....	62
ค.1 หน้าต่างโฮมบนสมาร์ตโฟน.....	63
ค.2 หน้าต่างการตั้งค่าบนสมาร์ตโฟน.....	63
ค.3 หน้าต่าง Wireless & networks บนสมาร์ตโฟน .....	64
ค.4 หน้าต่างเมนู Tethering & portable hotspot บนสมาร์ตโฟน .....	64
ค.5 ข้อความยืนยันการ Tethering & portable hotspot บนสมาร์ตโฟน.....	65
ค.6 หน้าต่างแสดงการเปิดฟังก์ชัน USB tethering บนสมาร์ตโฟนเสร็จสมบูรณ์ .....	65
ค.7 ไอคอนโปรแกรม PuTTY.....	66
ค.8 หน้าต่างโปรแกรม PuTTY .....	66
ค.9 หน้าต่างโปรแกรม PuTTY เมื่อเชื่อมต่อ Raspberry PI แล้ว.....	67
ค.10 หน้าต่างแสดง network Interface จากคำสั่ง ifconfig -a.....	67
ค.11 หน้าต่างแสดง IP Routing table จากคำสั่ง netstat -r.....	68
ค.12 คำสั่งลบเส้นทางที่เป็น default gateway.....	68
ค.13 หน้าต่างแสดง IP Routing table หลังจากลบเส้นทางที่เป็น default.....	68
ค.14 หน้าต่างแสดงผลเส้นทางเครือข่ายเน็ตเวิร์คที่ออก .....	69
ง.1 หน้าเว็บสำหรับดาวน์โหลดโปรแกรม PRTG Network Monitor.....	70
ง.2 ไอคอนโปรแกรม PRTG Enterprise Console.....	70
ง.3 หน้าแรกโปรแกรม PRTG Enterprise Console .....	71
ง.4 หน้าต่างแถบ Setup.....	71
ง.5 หน้าต่างเมนู Notification Delivery .....	72
ง.6 การตั้งค่า Configuration Mode .....	72
ง.7 การตั้งค่า Custom URL .....	73
ง.8 ตั้งค่า Encoding for SMS.....	73
ง.9 ปุ่มบันทึก.....	73

## สารบัญรูป (ต่อ)

รูปที่	หน้า
จ.10 หน้าต่างเมนู Notification Contacts.....	74
จ.11 ปุ่ม Add SMS Contact.....	74
จ.12 การสร้าง Notification Contact .....	74
จ.13 รายการ Notification Contact ที่สร้างขึ้น.....	75
จ.1 เริ่มต้นโปรแกรม VMware vSphere Client.....	76
จ.2 ยืนยันการรับรองความปลอดภัยทางอิเล็กทรอนิกส์.....	76
จ.3 หน้าต่างรายละเอียดของ VMware vSphere Client.....	77
จ.4 หน้าต่างการตั้งค่าของ VMware vSphere Client.....	77
จ.5 หน้าต่างการตั้งค่าของ VMware vSphere Client (หลังเปลี่ยนแปลง).....	78
จ.6 แถบสถานะการทำงานหลังการเปลี่ยนแปลง.....	78
จ.7 เริ่มต้นโปรแกรม Remote Desktop Connection .....	79
จ.8 หน้าต่างยืนยันข้อมูลผู้ใช้งาน .....	79
จ.9 หน้าต่างการตรวจสอบใบรองรับการเข้าถึง .....	80
จ.10 หน้าต่างรอการเชื่อมต่อ .....	80
จ.11 หน้าต่างโปรแกรม PRTG สถานะการทำงานปกติ (Up) .....	81
จ.12 หน้าต่างโปรแกรม PRTG สถานะการทำงานเริ่มมีปัญหา (Warning) .....	81
จ.13 หน้าต่างโปรแกรม PRTG แสดงสถานะการทำงานปัจจุบัน.....	82
จ.14 หน้าต่างโปรแกรม PRTG สถานะการทำงานมีข้อผิดพลาด (Down).....	82
ฉ.1 ไดรกทอรีไฟล์ alertConf.ini.....	83
ฉ.2 โครงสร้างไฟล์ alertConf.ini.....	83
ฉ.3 SSH Secure Shell บน Raspberry Pi.....	84
ฉ.4 คำสั่งเริ่มต้นโปรแกรมระบบแจ้งเตือนปัญหาอุปกรณ์ผ่านทางแอปพลิเคชันไลน์ .....	84
ฉ.5 ไฟล์ล็อกสถานะการทำงาน .....	85

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากซอฟต์แวร์ในการติดตามและแจ้งเตือนปัญหาทรัพยากรของอุปกรณ์ในห้องเซิร์ฟเวอร์ เช่น หน่วยความจำหลักหรือฮาร์ดดิสก์ไม่เพียงพอ หน่วยประมวลผลกลางทำงานเกินขนาด หรือระบบต่างๆ หยุดการทำงานลง ใช้การแจ้งเตือนในรูปแบบเก่าที่ไม่สามารถตรวจสอบได้ว่าข้อความนั้นไปถึงผู้รับหรือไม่ เช่น การส่งเอสเอ็มเอสหรืออีเมล จึงได้ออกแบบระบบที่สามารถส่งข้อความแจ้งเตือนผ่านทางโซเชียลมีเดีย ซึ่งได้แก่ แอปพลิเคชันไลน์ ที่เป็นที่นิยมและเข้าถึงการใช้งานของผู้รับข้อความ เพื่อให้มั่นใจได้ว่าข้อความต่างๆ จะส่งถึงผู้รับ จึงได้เกิดผลงานระบบแจ้งเตือนทางแอปพลิเคชันไลน์สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์นี้ขึ้นมา

### 1.2 วัตถุประสงค์ของการดำเนินงาน

- 1) เพื่อลดปัญหาการส่งข้อความแจ้งเตือนที่อาจจะส่งไม่ถึงมือผู้รับ
- 2) เพื่อให้ระบบสามารถส่งข้อความแจ้งเตือนผ่านแอปพลิเคชันไลน์ได้
- 3) เพื่อประหยัดงบประมาณการส่งข้อความแจ้งเตือน ซึ่งการส่งผ่านแอปพลิเคชันไลน์จะเสียค่าใช้จ่ายเฉพาะค่าบริการอินเทอร์เน็ต ซึ่งไม่เหมือนกับ SMS ที่จะต้องเสียค่าใช้จ่ายในการส่งแต่ละครั้ง
- 4) เพื่อให้ระบบสามารถทำงานร่วมกับโปรแกรม PRTG Network Monitoring ได้

### 1.3 ขอบเขตของการดำเนินงาน

- 1) ระบบสามารถตั้งค่าข้อมูลผู้ติดต่อสำหรับการส่งข้อความแจ้งเตือนได้
- 2) ระบบสามารถทำงานได้บนระบบปฏิบัติการ Linux และ Windows
- 3) ระบบสามารถยืนยันการได้รับทราบการแจ้งเตือนที่ได้ส่งไปผ่านแอปพลิเคชันไลน์
- 4) ระบบสามารถแจ้งเตือนไปยังบุคคลที่มีลำดับสำคัญกว่ากรณีที่คุณคลิกก่อนหน้าไม่มีการอ่านข้อความ

#### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ระบบสามารถจัดการปัญหาการแจ้งเตือน กรณีเกิดปัญหากับระบบต่างๆ ที่เราติดตาม ให้สามารถส่งข้อความแจ้งเตือนมายังฝ่ายเทคนิคที่รับผิดชอบได้
- 2) ระบบสามารถแจ้งเตือนและเข้าถึงฝ่ายเทคนิคมากขึ้นเนื่องจากปัจจุบันการใช้แอปพลิเคชันไลน์กำลังเป็นที่นิยมในชีวิตประจำวัน
- 3) ระบบสามารถส่งข้อความแจ้งเตือน โดยการร้องขอผ่านโปรโตคอล HTTP ได้
- 4) ระบบสามารถทำงานร่วมกับโปรแกรมที่ทำหน้าที่มอนิเตอร์ระบบเครือข่ายได้ เช่น โปรแกรม PRTG Network Monitor เป็นต้น

#### 1.5 ขั้นตอนการดำเนินงาน

- 1) วิเคราะห์ปัญหา
- 2) วางแผนการดำเนินงาน
- 3) ศึกษาวิธีการทำงานของ Raspberry Pi ที่มีการต่อเข้ากับอุปกรณ์ระบบปฏิบัติการแอนดรอยด์ ผ่าน USB Port
- 4) ศึกษาวิธีการใช้งานโปรแกรม PRTG Network Monitor เบื้องต้น
- 5) วิเคราะห์ความต้องการของระบบ
- 6) ออกแบบแอปพลิเคชัน
- 7) พัฒนาแอปพลิเคชัน
- 8) ทดสอบการใช้งานแอปพลิเคชันให้ตรงกับความต้องการของระบบ
- 9) แก้ไข และปรับปรุงแอปพลิเคชัน

#### 1.6 อุปกรณ์ที่ใช้ในการดำเนินงาน

- 1) สมาร์ทโฟน หรือแท็บเล็ตสำหรับแชร์อินเทอร์เน็ต
- 2) บอร์ด Raspberry Pi
- 3) เครื่องคอมพิวเตอร์โน้ตบุ๊ก

## บทที่ 2

# ทฤษฎีที่เกี่ยวข้อง

### 2.1 เทคโนโลยีไร้สาย (Wireless Technology)

เป็นเทคโนโลยีการคมนาคมการสื่อสารรูปแบบใหม่ที่พัฒนาขึ้นจากโครงสร้างการคมนาคมการสื่อสารแบบเก่าที่ใช้สายเคเบิลในการส่งผ่านข้อมูลเสียงไปบนเครือข่าย เทคโนโลยีไร้สายถูกพัฒนาขึ้นมาให้ผู้ใช้งานสามารถเชื่อมต่อการสื่อสารถึงกันได้ทุกสถานที่ทุกเวลาโดยการแปลงสัญญาณเสียงหรือชุดข้อความให้เป็นคลื่นความถี่วิทยุและทำการส่งไปถึงปลายทางผู้รับโดยตรงผ่านทางสายเคเบิล ซึ่งก่อนที่สัญญาณถูกส่งตรงมาถึงผู้รับนั้นสัญญาณที่เป็นคลื่นความถี่จะต้องเดินทางผ่านปัจจัยหลายอย่าง เช่น ผ่านอากาศ, สะท้อนกับอาคาร, กระแทกต้นไม้, ผ่านคลื่นวิทยุ เป็นต้น ซึ่งปัจจัยที่กล่าวมานั้นเป็นผลกระทบที่ทำให้ระบบการสื่อสารไร้สายนั้นเกิดปัญหาด้านความต่อเนื่องในการสื่อสาร เช่น สัญญาณขาดหาย, สัญญาณถูกส่งไปผิดปลายทาง, อับสัญญาณ เป็นต้น

ที่กล่าวมานั้น เป็นปัญหาที่พบขึ้นของการสื่อสารไร้สาย แต่ในปัจจุบันได้มีการพัฒนาเทคโนโลยีให้มีประสิทธิภาพมากขึ้น และด้วยข้อดีของการสื่อสารไร้สาย เช่น ความคล่องตัวในการใช้งานสูง, ลดขั้นตอนการติดตั้งและลดค่าใช้จ่าย เป็นต้น ทำให้เทคโนโลยีไร้สายเข้ามามีบทบาทในการใช้ชีวิตประจำวันมากขึ้น

### 2.2 โบายเทคโนโลยี (Mobile Technology)

คืออุปกรณ์ติดต่อสื่อสารแบบไร้สายที่มีการใช้งานและพัฒนาอย่างต่อเนื่องในปัจจุบัน ที่สามารถนำติดตัวไว้ในสถานที่ต่าง ๆ ได้อย่างสะดวกสบาย เช่น เครื่องคอมพิวเตอร์พกพา, โทรศัพท์มือถือ, พีดีเอ (PDA), อุปกรณ์บอกตำแหน่งพื้นผิวโลก (Global Position System: GPS) เป็นต้น ซึ่งได้มีการพัฒนาอย่างต่อเนื่องทำให้อุปกรณ์ติดต่อสื่อสารไร้สายเป็นได้มากกว่าอุปกรณ์รับส่งข้อความในยุค 1G, 2G โดยมีแอปพลิเคชันที่สนับสนุนการใช้ชีวิตประจำวันเช่น กล้องถ่ายภาพ, สมุดจดบันทึก, แผนที่, เข็มทิศ เป็นต้น อีกทั้งยังสามารถเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย ทำให้การรับส่งข้อมูลเป็นไปได้อย่างอิสระทำให้ผู้บริโภคเลือกที่จะใช้งานอุปกรณ์ติดต่อสื่อสารแบบไร้สายมากขึ้น และที่สำคัญคือการพัฒนาของอุปกรณ์ติดต่อสื่อสารแบบไร้สายทั้งซอฟต์แวร์ที่ใช้ควบคุม, ลักษณะอุปกรณ์ มีลักษณะการพัฒนาไปในทิศทางเดียวกันกับเทคโนโลยีที่เข้ามาใหม่เรื่อยๆ ในอนาคต ทำให้

ผู้บริโภคไม่จำเป็นต้องเสียค่าใช้จ่ายในการเปลี่ยนอุปกรณ์ติดต่อสื่อสารแบบไร้สายให้รองรับกับเทคโนโลยีใหม่ๆ

## 2.3 เทคโนโลยีการติดต่อสื่อสารบนโทรศัพท์มือถือ

เทคโนโลยีการติดต่อสื่อสารบนโทรศัพท์มือถือเกิดขึ้นมาและมีการพัฒนาตามลำดับจากยุค 1G, 2G, 3G, 4G ซึ่งเป็นตัวช่วยในการเชื่อมต่อเครือข่ายการติดต่อสื่อสารให้เป็นไปอย่างอิสระ แอปพลิเคชันหรือซอฟต์แวร์ที่พัฒนาขึ้นบนโทรศัพท์มือถือบางประเภทนั้นจำเป็นต้องมีเครือข่ายการติดต่อสื่อสารในการรับส่งข้อมูลถึงกัน แต่ผู้บริโภคไม่จำเป็นต้องเปลี่ยนโทรศัพท์มือถือให้รองรับกับระบบใหม่ที่เกิดขึ้นจึงเป็นข้อดีอย่างหนึ่งที่ทำให้ผู้บริโภคเข้ามาใช้บริการในจำนวนมากขึ้น

### 2.3.1 เทคโนโลยี 3G

3G [2,3] คือ เทคโนโลยีการสื่อสารในยุค 3G ซึ่งได้มีการเปลี่ยนแปลงจากการรับส่งข้อมูลแบบอะนาล็อกเป็นดิจิทัลแทนทำให้การรับส่งข้อมูลมีปริมาณสูงขึ้น ความเร็วในการรับส่งข้อมูลมีกำลังความเร็วสูงขึ้นแตกต่างโดยสิ้นเชิงกับยุค 2G โทรศัพท์เคลื่อนที่สามารถรับ-ส่งข้อมูลในความเร็วสูงโดยจะเน้นการเชื่อมต่อแบบ Wireless ด้วยความเร็วสูง ทำให้การบริการทางด้านมัลติมีเดียหรือโซเชี่ยลแอปพลิเคชันสามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น มาตรฐานโทรศัพท์มือถือยุคที่ 3 ได้รับการพัฒนาขึ้นมาจากพื้นฐานของมาตรฐาน IMT-2000 ภายใต้กลุ่มของสหภาพโทรคมนาคมระหว่างประเทศ โดยมีกรอบนโยบายมุ่งหวังให้เป็นเครือข่ายโทรคมนาคมที่สามารถหลอมรวมการให้บริการได้หลากหลายทั้งการติดต่อสื่อสารทางเสียง, ข้อมูล, อินเทอร์เน็ตและมัลติมีเดีย โดยผู้ใช้งานมีลักษณะการใช้งานทั้งประจำที่หรือเคลื่อนที่ โดยมาตรฐาน IMT-2000 โดยสรุปมีดังนี้

- 1) มีโครงสร้างพื้นฐานที่รองรับการใช้งานบริการต่างๆ เช่น บริการรับส่งข้อมูลประจำที่, บริการรับส่งข้อมูลเคลื่อนที่, การสื่อสารด้วยเสียง, การรับส่งข้อมูลที่เป็นอิสระ, สามารถเข้าถึงอินเทอร์เน็ตได้, การให้บริการสื่อมัลติมีเดียทุกประเภท จะต้องเป็นไปในแนวทางเดียวกัน คือสามารถโอนถ่ายข้อมูลไปยังอุปกรณ์โทรคมนาคมอื่นที่สามารถรับส่งข้อมูลได้
- 2) โครงข่ายข้ามแดน (Global Roaming) สามารถใช้งานอุปกรณ์เดียวกันในพื้นที่ใดก็ได้ทั่วโลก
- 3) ความต่อเนื่องในการสื่อสาร (Seamless Delivery Service) สามารถใช้งานการติดต่อสื่อสารได้อย่างต่อเนื่องถึงแม้ว่าจะมีการเคลื่อนที่จนทำให้เกิดการเปลี่ยนแปลงของสถานีรับส่งสัญญาณ
- 4) อัตราความเร็วการรับส่งข้อมูล (Transmission Rate)
  - ขณะอยู่ประจำที่ไม่ได้เคลื่อนที่ ความเร็วในการรับส่งข้อมูลอย่างน้อย 2 Mbps
  - ขณะเคลื่อนที่ด้วยความเร็วระดับยานพาหนะ ความเร็วในการรับส่งข้อมูลอย่างน้อย 384 Kbps
  - ทุกสภาพการใช้งาน มีความเร็วในการรับส่งข้อมูลอย่างน้อย 14.4 Mbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2 เทคโนโลยี 4G

4G [2,3] คือเทคโนโลยีถูกพัฒนาขึ้นจากโดยมีมาตรฐานมาจากมาตรฐานเครือข่ายโทรศัพท์เคลื่อนที่ยุค 3G โดยจะเน้นเพิ่มความเร็วในการรับ-ส่งข้อมูลให้สูงขึ้น ทำให้การใช้งานแอปพลิเคชันต่างๆ มีความเร็วในการส่งข้อมูลเร็วขึ้นตามไปด้วย เช่น การท่องเว็บไซต์, เกมออนไลน์, ชมภาพยนตร์หรือโอบายที่วีความละเอียดคมชัดสูง, วีดีโอคอนเฟอเรนซ์, การถ่ายทอดสด Live Broadcast แบบ Real-time, Cloud Service สื่อการเรียนการสอนผ่านทาง e-Learning, การรักษาผ่านทางไกล, ทำให้ยุคปัจจุบันกลายเป็นยุคสำหรับการทำธุรกิจด้านมัลติมีเดียและวีดีโอออนไลน์ในอนาคต ซึ่งอุปกรณ์โทรศัพท์มือถือในปัจจุบันนั้นได้รับการพัฒนาบนมาตรฐานเดียวกันกับโทรศัพท์ในยุค 3G ทำให้ไม่จำเป็นต้องเปลี่ยนตัวอุปกรณ์อีกทั้งยังสามารถเข้าใช้งานการติดต่อสื่อสารบนโทรศัพท์มือถือบน 4G ได้ทันที โดยในประเทศไทยโทรศัพท์ยุค 2G นั้นไม่รองรับต่อเทคโนโลยี 3G, 4G แล้ว

ปัจจุบัน 4G ที่ใช้งานในเชิงพาณิชย์อยู่ 2 ระบบ คือ มาตรฐาน Mobile WiMAX และมาตรฐาน Long Term Evolution (LTE) ซึ่งได้มีการพัฒนาให้สอดคล้องกับมาตรฐาน IMT-Advance ของ ITU โดยมาตรฐานของ IMT-Advance มีคุณสมบัติที่สำคัญดังต่อไปนี้

- 1) อยู่บนมาตรฐานการสื่อสารข้อมูลบนมาตรฐานไอพี (All-IP packet switched network)
- 2) ความต่อเนื่องในการสื่อสาร (Seamless Delivery Service) สามารถใช้งานการติดต่อสื่อสารได้อย่างต่อเนื่องถึงแม้ว่าจะมีการเคลื่อนที่จนทำให้เกิดการเปลี่ยนแปลงของสถานีรับส่งสัญญาณต่างเครือข่าย
- 3) สามารถให้บริการมัลติมีเดียคุณภาพสูงรองรับมัลติมีเดียในยุคต่อไปได้
- 4) อัตราความเร็วการรับส่งข้อมูล (Transmission Rate)
  - ขณะอยู่ประจำที่ไม่ได้เคลื่อนที่ ความเร็วในการรับส่งข้อมูลอย่างน้อย 1 Gbps
  - ขณะเคลื่อนที่ด้วยความเร็วระดับยานพาหนะ ความเร็วในการรับส่งข้อมูลอย่างน้อย 100 Mbps

### 2.4 โซเชียลแอปพลิเคชันไลน์

Line [1] เป็นโซเชียลแอปพลิเคชันชนิดหนึ่งบนโทรศัพท์มือถือ Android, iPhone, Window Phone, Pc ที่นิยมใช้อย่างแพร่หลายในสังคมปัจจุบัน โดยฟังก์ชันการทำงานมีใช้งานมากมายเช่น ทั้งการใช้งานในการติดต่อสื่อสาร, โทรศัพท์ผ่านทางไกลโดยไม่เสียค่าใช้จ่ายในการใช้บริการ, วีดีโอติดต่อสื่อสารแบบเรียลไทม์, ส่งไฟล์ข้อมูล เป็นต้น ทำให้ผู้คนนำมาใช้ในทางการค้าและชีวิตประจำวันมากขึ้น โดยผู้ใช้งานสามารถติดตั้งตัวแอปพลิเคชันได้ผ่านทางช่องทางต่างๆ โดยตัวแอปพลิเคชันไลน์ จะทำการเชื่อมต่อโทรศัพท์มือถือด้วยเบอร์โทรศัพท์ประจำ SIM Card เมื่อทำการสมัครสมาชิกและยืนยันผ่านทางระบบเรียบร้อยแล้วผู้ใช้งานจะถือเป็นโคล์แอนท์เครื่องหนึ่งที่ติดต่อกับเซิร์ฟเวอร์ Line ผู้ใช้งานสามารถสร้างการเชื่อมต่อกับผู้ใช้งานอื่นได้โดยการใช้เบอร์โทรศัพท์หรือ ID Line ของผู้ใช้ที่

ต้องการเพื่อทำการค้นหาเส้นทางการเชื่อมต่อบนเซิร์ฟเวอร์ เมื่อพบเส้นทางการเชื่อมต่อเซิร์ฟเวอร์จะบันทึกเส้นทางการติดต่อระหว่างไคลแอนท์ทั้งสองไว้บนระบบ

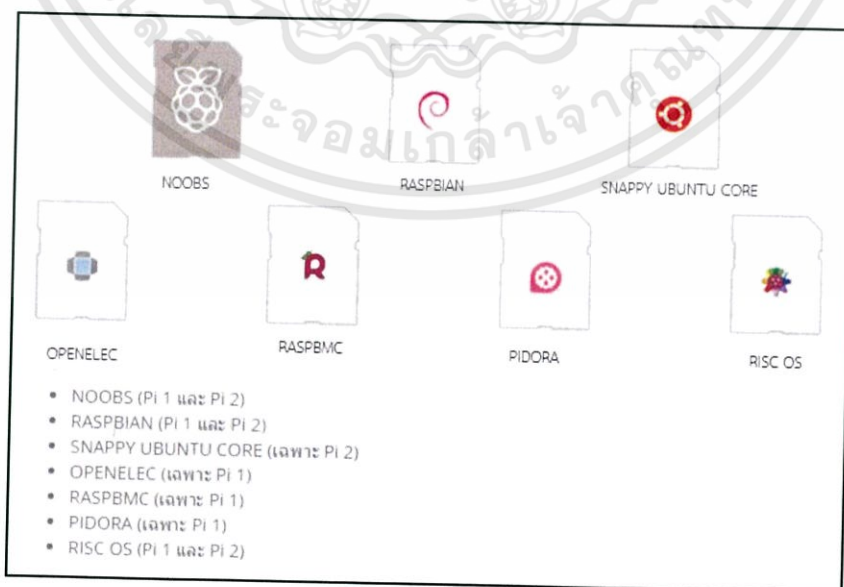
## 2.5 LineSharp

LineSharp [5] เป็น Open Source ที่ถูกเขียนขึ้นด้วยโครงสร้างภาษา C# ถูกพัฒนาโดยทีม Banandana ใช้ในการร้องขอเซอร์วิสการทำงานต่างๆของไลน์โปรโตคอล ซึ่งเซอร์วิสต่างๆ ถูกนำมาเขียนให้อยู่ในรูปของโครงสร้างภาษาการเขียนโปรแกรมที่ใกล้เคียงกับภาษามนุษย์มากที่สุดเพื่อให้ผู้พัฒนาโปรแกรมสามารถนำไปประยุกต์ใช้ในการพัฒนาซอฟต์แวร์

การทำงานของ LineSharp จะทำการร้องขอการทำงานบนโปรโทคอลของเซิร์ฟเวอร์ไลน์ ดังนั้นผู้ใช้งานจะต้องเชื่อมต่อเครือข่ายอินเทอร์เน็ตทุกครั้งในการใช้งาน การพัฒนาโปรแกรมโดยใช้ LineSharp ผู้พัฒนาจะต้องสร้างส่วนการเชื่อมต่อกับเครือข่ายไลน์และใช้รหัสยืนยันตัวบุคคลในการร้องขอเข้าใช้งานเหมือนกับการใช้งานแอปพลิเคชันไลน์บนโทรศัพท์มือถือ เมื่อทำการสร้างการเชื่อมต่อเสร็จสิ้นผู้พัฒนาสามารถร้องขอเรียกใช้คำสั่งการทำงานต่างๆบนไลน์ได้ เช่น การแสดงรายชื่อผู้ติดต่อ, การส่งข้อความถึงผู้รับที่ต้องการ, การสร้างรายชื่อผู้ติดต่อใหม่ เป็นต้น

## 2.6 คอมพิวเตอร์ขนาดเล็ก Raspberry Pi

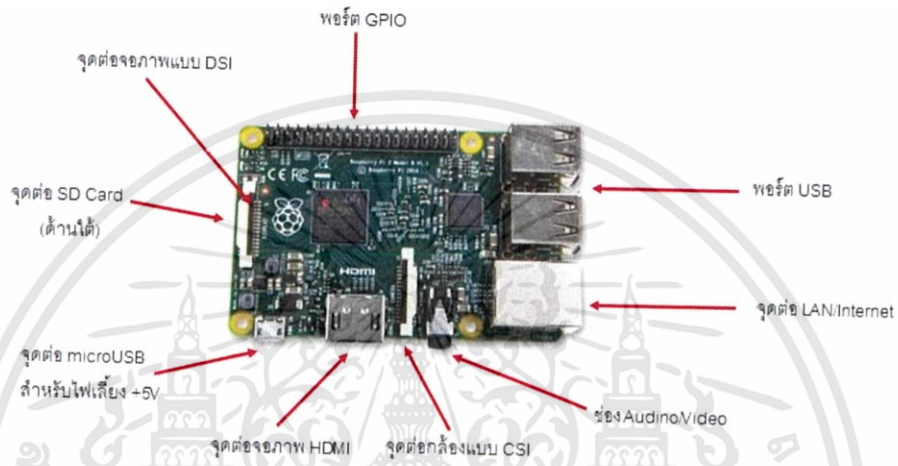
Raspberry Pi [4,6] เป็นคอมพิวเตอร์ชนิดหนึ่งที่ถูกสร้างบน Single Circuit Board ให้มีขนาดเล็กเทียบเท่าได้กับบัตรเครดิต พัฒนาโดยบริษัท Raspberry Pi Foundation ที่ประเทศสหรัฐอเมริกา บนตัวบอร์ดจะเชื่อมกับอุปกรณ์ต่างๆ เช่น memory (RAM), Microprocessor, Input/output (I/O) เพื่อทำให้การทำงานของบอร์ดเหมือนกับคอมพิวเตอร์ส่วนตัวทั่วไปและรองรับระบบปฏิบัติการลินุกซ์ (Linux Operation System) ได้หลายระบบ เช่น Raspbian (Debian), Pidora (Fedora), Arch Linux, Window 10 เป็นต้น



รูปที่ 2.1 ระบบปฏิบัติการของ Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยติดตั้งระบบปฏิบัติการลงบน SD Card ที่ทำหน้าที่เป็น Hard Disk บนเครื่องคอมพิวเตอร์ปกติ GPIO จะเป็นส่วนหนึ่งของบอร์ดที่แตกต่างจากคอมพิวเตอร์ทั่วไปไว้สำหรับการติดต่อการทำงานกับอุปกรณ์อิเล็กทรอนิกส์ภายนอก ทำให้ตัว Raspberry Pi เป็นคอมพิวเตอร์ที่สามารถควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้อย่างอิสระโดยการเขียนโปรแกรมควบคุมและสามารถนำไปประยุกต์ใช้ได้หลายแบบ โดยส่วนประกอบของ input/output บนตัวบอร์ดและรายละเอียดของแต่ละส่วนสามารถแจกแจงได้ดังนี้



รูปที่ 2.2 ส่วนประกอบของบอร์ด Raspberry Pi 2

ตารางที่ 2.1 รายละเอียดคุณสมบัติทางเทคนิคของบอร์ด Raspberry Pi

	โมเดล A	โมเดล B (Revision 2)
System on a chip (SoC)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM and Single USB Port)	
CPU	700MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)	
GPU	Broadcom Video Core IV @ 250 MHz OpenGL ES 2.0 (24 GFLOPS) MPEG-2 and VC-1, 1080p 30 h.264/MPEG-4 AVC high-profile decoder and encoder	
Memory (SDRAM)	256 MB (Shared with GPU)	512 MB (Shared with GPU)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

USB 2.0 Ports	1(direct form BCM2835)	2 (via the build in integrated 3-port USB hub)
Video Input	A CSI input connector allows for the connection of RPF designed camera module (ออกแบบมาให้เชื่อมต่อกับRaspberry Pi Camera Moduleโดยเฉพาะ)	
Video Outputs	Composite RCA (PAL and NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards. (มีทั้งสองแบบ คือ แบบ RCA และแบบ HDMI)	

ตารางที่ 2.1 (ต่อ) รายละเอียดคุณสมบัติทางเทคนิคของบอร์ด Raspberry Pi

Audio Outputs	3.5 mm jack, HDMI, and as of revision 2 boards, I2S audio (also potentially for audio input)	
Onboard storage	SD/ MMC/ SDIO card slot (3.3V card power support only)	
Onboard network	None	10/100 Ethernet (8P8C) USB adapter on the third port of the USB hub
Low-level peripherals	8 x GPIO, UART, I2C Bus, SPI Bus with two chip selects, I2S audio +3.3V, +5V, Ground	
Power ratings	300 mA (1.5 W)	700 mA (3.5 W)
Power source	5 Volt via Micro USB or GPIO header	
Size	85.60 mm x 53. Mm (3.370 inch x 2.125 inch)	
Weight	45 g. (1.6 oz.)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 PRTG Network Monitor

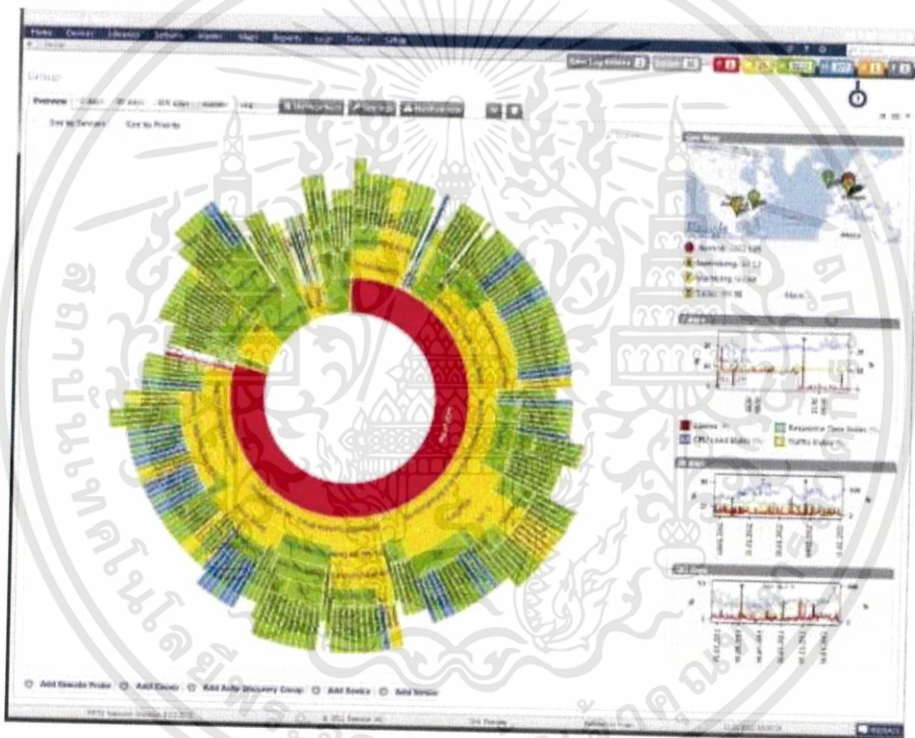
เป็นโปรแกรมที่เหมาะสมสำหรับบุคคลที่ต้องการจัดการเกี่ยวกับระบบเน็ตเวิร์ค ไม่ว่าจะเป็นแอดมินหรือผู้ที่ต้องการศึกษาก็ตาม สามารถใช้โปรแกรม PRTG Network Monitor [7] ช่วยในการจัดการได้ สามารถแจ้งรายละเอียดการทำงานในระบบเน็ตเวิร์คได้อย่างละเอียด ไม่ว่าจะเป็นรายละเอียดการใช้แบนด์วิดท์ (Bandwidth) ต่างๆ รวมไปถึง การส่งข้อมูลจากโปรแกรมหรือระบบภายใน การส่งไฟล์ การใช้งานไฟล์ การใช้อินเทอร์เน็ต และอีกมากมาย โดยมีรายละเอียดแยกตามสีและประเภทแบบชัดเจน ทั้งยังสามารถแสดงออกมาให้อยู่ในรูปแบบของกราฟได้ นอกจากนี้ยังสามารถวัดอุณหภูมิของระบบเน็ตเวิร์คและอุปกรณ์ที่ใช้งานบางชิ้น และยังครอบคลุมการแจก IP Address และการใช้งานไปยังเว็บไซต์ต่างๆ โดยสามารถดูได้ด้วยโปรแกรมเดียว ไม่ว่าจะเป็นเข้าจากเว็บเบราว์เซอร์ตัวใดก็ตาม โปรแกรมนี้ถูกพัฒนาโดยทีมผู้พัฒนาจากประเทศเยอรมัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คุณสมบัติและความสามารถของโปรแกรม PRTG Network Monitor เพิ่มเติม

- ใช้งานง่าย และครอบคลุมการใช้งานที่หลากหลายในการควบคุมปริมาณการใช้เน็ตเวิร์ค ทั้งแบบ LAN และแบบ WAN
- สามารถใช้งานได้ดีกับอุปกรณ์เน็ตเวิร์คหลายค่าย เช่น VMware, CISCO เป็นต้น
- เก็บรายละเอียดการทำงานในแต่ละเครื่อง แต่ละเครือข่ายได้อย่างละเอียด ขึ้นอยู่กับการตั้งค่าของผู้ใช้งาน
- ทดสอบการดาวน์โหลด อัปโหลดไฟล์ ได้ง่ายๆ และสามารถนำข้อมูลเหล่านั้นมาทำเป็น กราฟรูปแบบต่างๆ ได้
- รองรับการใช้งานทั้ง คอมพิวเตอร์ทั่วไป สมาร์ทโฟนและแท็บเล็ต



รูปที่ 2.3 โปรแกรม PRTG Network Monitor

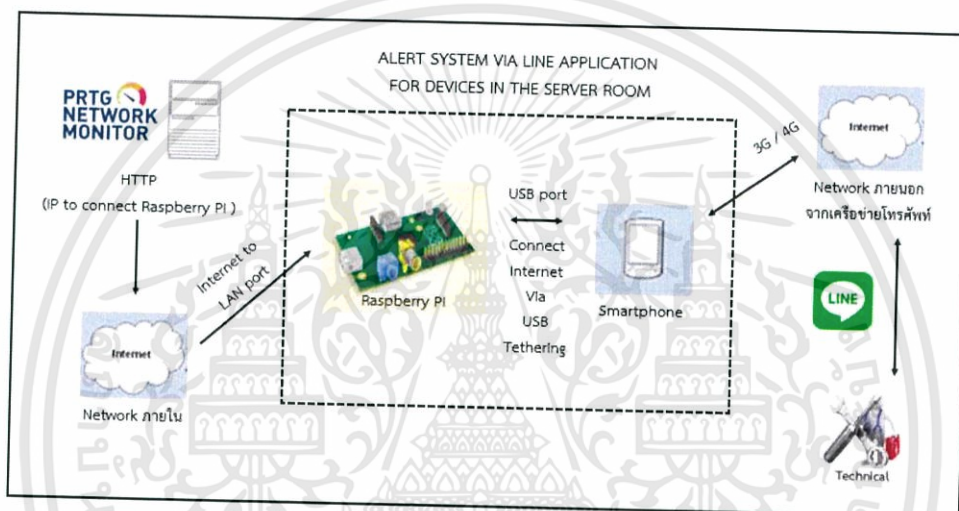
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3

## วิธีการดำเนินงาน

เนื้อหาในบทนี้ประกอบไปด้วยการอธิบายรายละเอียดของระบบการทำงาน ซึ่งประกอบด้วยสถาปัตยกรรมของระบบ Use Case Diagram Activity Diagram Sequence Diagram Flowchart Diagram การตั้งค่าไฟล์ configuration ฟังก์ชันต่างๆ ที่เกี่ยวข้องกับการทำงานของระบบ

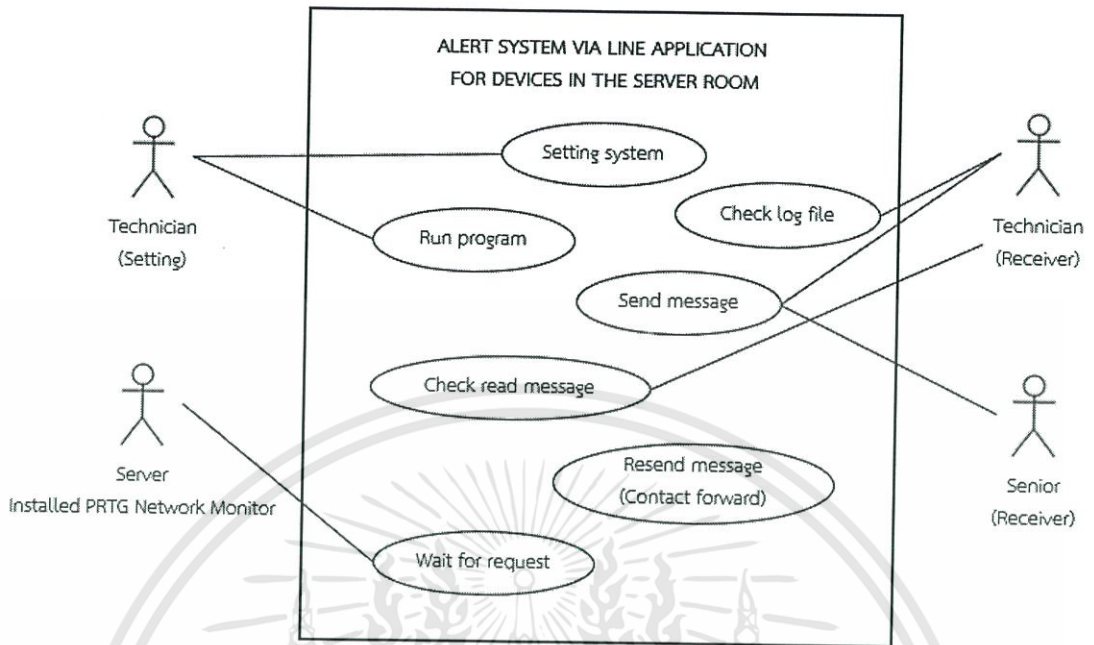
### 3.1 สถาปัตยกรรมระบบ (Architecture)



รูปที่ 3.1 สถาปัตยกรรมของระบบแจ้งเตือนทางแอปพลิเคชันไลน์สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์

ระบบแจ้งเตือนทางแอปพลิเคชันไลน์สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์สามารถทำงานร่วมกับโปรแกรม PRTG Network Monitor หรือโปรแกรม Monitor อื่นๆ ที่ติดตั้งอยู่บนเครื่องเซิร์ฟเวอร์ได้ โดยเมื่อเครื่องเซิร์ฟเวอร์ที่ทำการติดตามอยู่นั้นเกิดปัญหาขึ้นโปรแกรม Monitor จะทำการร้องขอผ่านโปรโตคอล HTTP ซึ่งมีการเชื่อมต่อกันแบบเครือข่ายภายในมายังระบบแจ้งเตือนฯ ที่พัฒนาขึ้นและติดตั้งไว้บน Raspberry PI โดยมีการส่งพารามิเตอร์ในส่วนของกลุ่มข้อมูลผู้รับและข้อความสำหรับแจ้งเตือน ระบบแจ้งเตือนฯ สามารถแจ้งเตือนผ่านช่องทางไลน์ที่มีการนำโค้ดบางส่วนที่เป็น Open Source มา implement ให้สามารถส่งข้อความผ่านไลน์ได้ ในการส่งแต่ละครั้งจะมีการนำพารามิเตอร์ของกลุ่มข้อมูลผู้รับไปค้นหาในไฟล์ alertConf.ini เพื่อดึงเอาข้อมูล ได้แก่ ชื่อไลน์ของผู้รับที่ต้องการส่งข้อความ โดยในการส่งไลน์นั้นจะใช้อินเทอร์เน็ต 3G หรือ 4G ซึ่งมีการเชื่อมต่อกันแบบเครือข่ายโทรศัพท์ทั่วไป เพื่อให้ Raspberry Pi สามารถใช้งานแอปพลิเคชันไลน์ได้นั้นจะต้องทำการเชื่อมต่ออินเทอร์เน็ตโดยต่อเข้ากับโทรศัพท์ผ่านโหมด USB Tethering ซึ่งเปิดการใช้งานไว้แล้ว

### 3.2 Use Case Diagram



รูปที่ 3.2 Use Case Diagram ระบบแจ้งเตือนทางแอปพลิเคชันไลน์ สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์

ในแต่ละไซต์งานจะมีเครื่องเซิร์ฟเวอร์ที่ติดตั้งโปรแกรม PRTG Network Monitor ในการดูแลระบบ เมื่อมีการใช้ทรัพยากรของระบบเกินขนาดหรือมีการทำงานผิดพลาดตามที่ได้กำหนดไว้ เครื่องเซิร์ฟเวอร์จะทำการส่งคำสั่งร้องขอมายังระบบแจ้งเตือนฯ ที่พัฒนาขึ้น ซึ่งมีการรอคำสั่งอยู่ เพื่อส่งข้อความแจ้งเตือนถึงสาเหตุของปัญหายังฝ่ายเทคนิคเพื่อให้ดำเนินการแก้ไข โดยระบบจะทำหน้าที่เป็นเซิร์ฟเวอร์ที่เปิดการเชื่อมต่อรอคำสั่งร้องขอจากเครื่องเซิร์ฟเวอร์ที่ไซต์โดยมีการระบุพารามิเตอร์ ซึ่งประกอบด้วยข้อมูลผู้รับปลายทางและข้อความสำหรับแจ้งเตือน มีการนำไลน์เอพีไอที่เป็นแอปพลิเคชันโซเซียลมีเดียมาประยุกต์ใช้ในการส่งข้อความไปยังผู้รับหรือฝ่ายเทคนิคตามพารามิเตอร์ดังกล่าว หากผู้รับไม่มีการยืนยันว่าได้รับข้อความ ระบบจะดำเนินการส่งข้อความแจ้งเตือนไปยังบุคคลหรือหน่วยงานที่มีระดับที่สำคัญกว่า (แสดงรูปที่ 3.2)

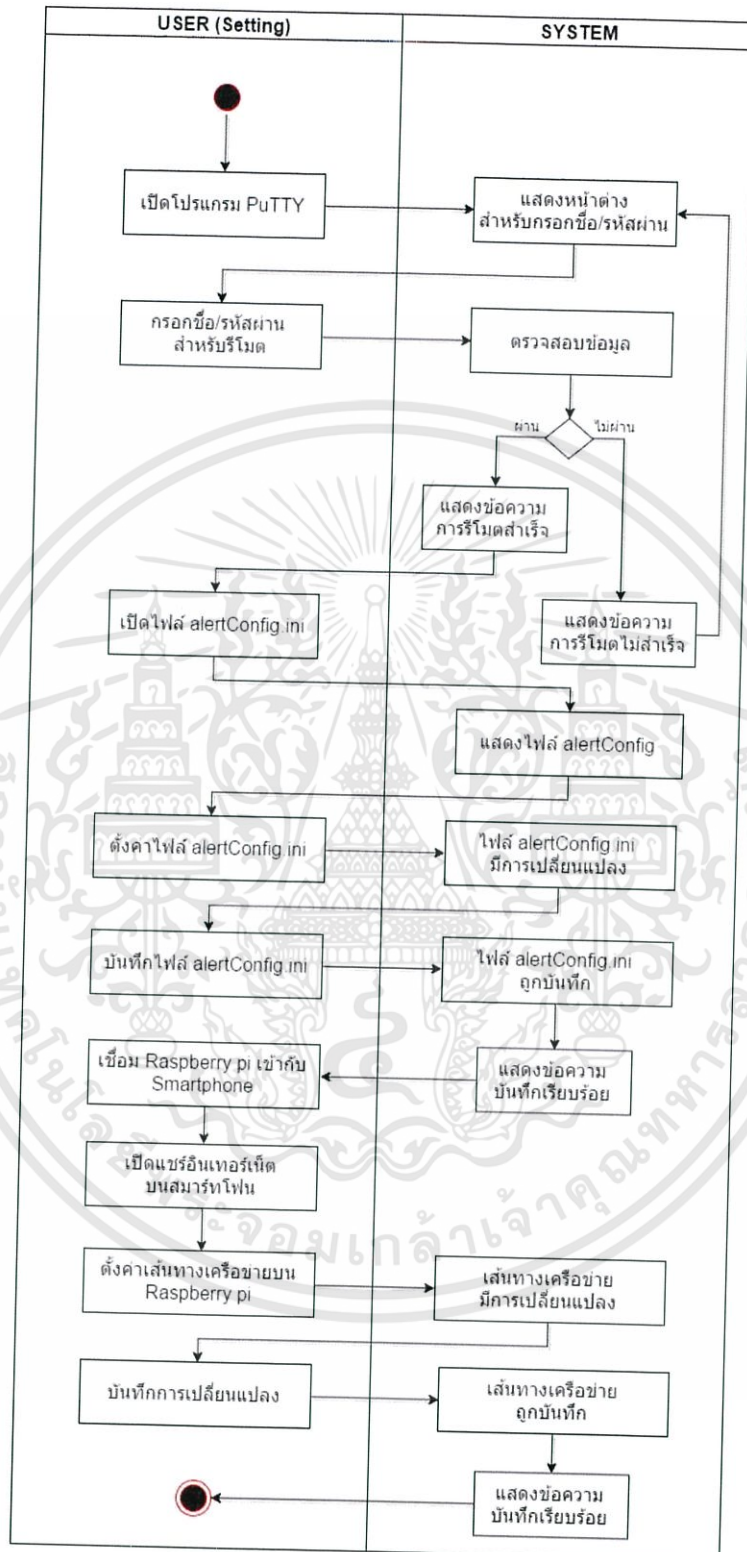
### 3.3 Activity Diagram

Activity Diagram ของระบบแบ่งออกเป็น 7 ส่วนย่อยคือ

- 1) Activity Diagram การตั้งค่าระบบแจ้งเตือนฯ
- 2) Activity Diagram การเริ่มต้นระบบแจ้งเตือนฯ
- 3) Activity Diagram การรับคำร้องจาก PRTG Network Monitor
- 4) Activity Diagram การส่งข้อความผ่านแอปพลิเคชันไลน์
- 5) Activity Diagram การตรวจสอบการอ่านข้อความ
- 6) Activity Diagram การส่งข้อความไปยังบุคคลถัดไป
- 7) Activity Diagram การตรวจสอบล็อกไฟล์สถานะการทำงาน



### 3.3.1 Activity Diagram การตั้งค่าระบบแจ้งเตือนฯ

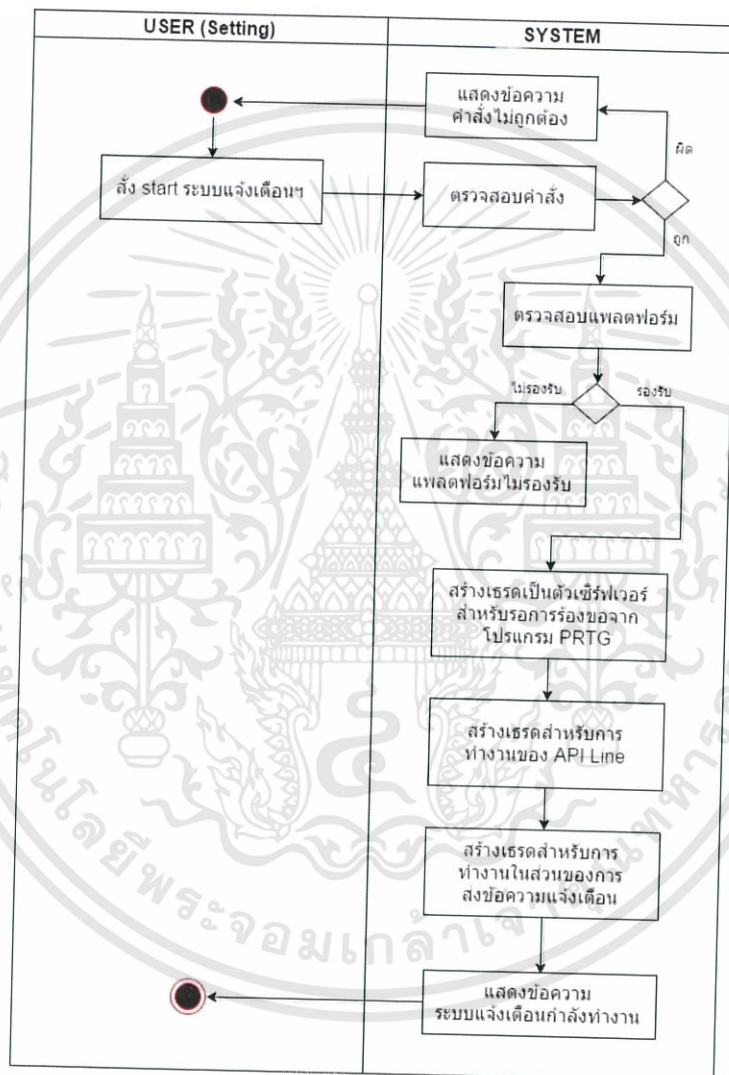


รูปที่ 3.3 Activity Diagram การตั้งค่าระบบแจ้งเตือนฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนแรกจะเป็นการตั้งค่าระบบแจ้งเตือนฯ ในส่วนของไฟล์ alertConf.ini ที่ใช้ในการระบุชื่อผู้รับสำหรับส่งข้อความและการเชื่อมต่ออินเทอร์เน็ตให้กับ Raspberry Pi โดยทำการรีโมตขึ้นไปใช้งาน SSH บน Raspberry Pi เพื่อทำการตั้งค่าไฟล์ alertConf.ini และตั้งค่าเส้นทางเครือข่าย (Route IP) ของอินเทอร์เน็ตที่ได้รับมาจากสมาร์ตโฟนที่เชื่อมต่อกันด้วยยูเอสบี ผ่านโปรแกรม PuTTY (แสดงรูปที่ 3.3)

### 3.3.2 Activity Diagram การเริ่มต้นระบบแจ้งเตือนฯ

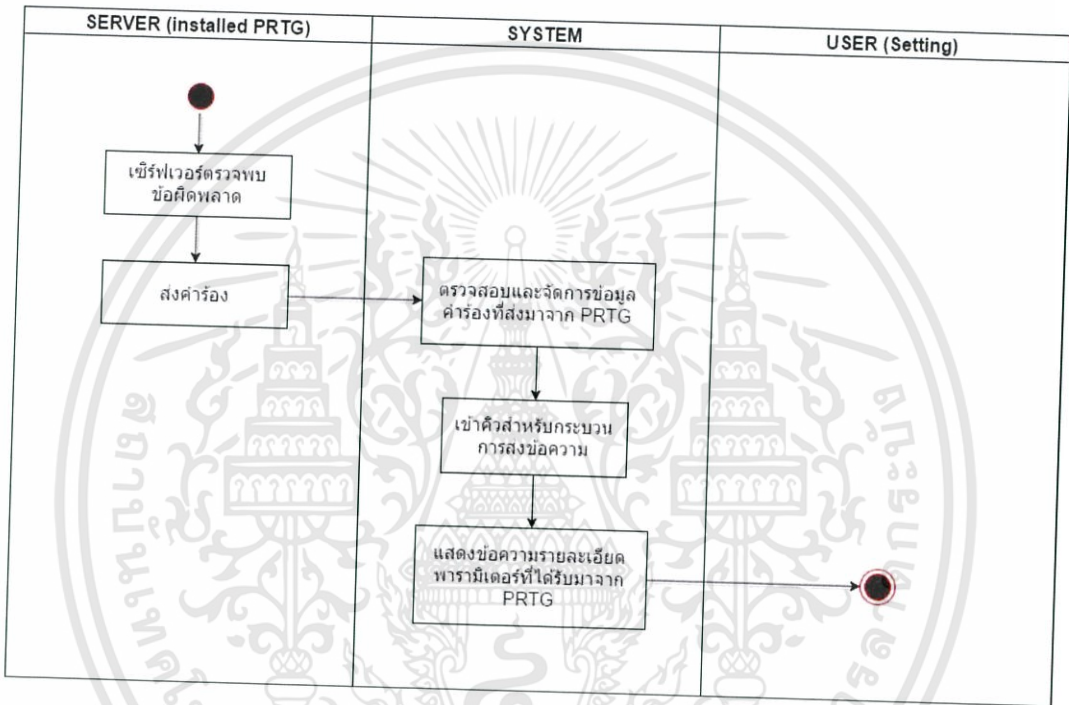


รูปที่ 3.4 Activity Diagram การเริ่มต้นระบบแจ้งเตือนฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนนี้เป็นการเริ่มต้นใช้งานระบบแจ้งเตือนฯ โดยจะต้องทำการรีโมตเข้าไปใช้งาน SSH บน Raspberry Pi ผ่านโปรแกรม PuTTY เมื่อยืนยันตัวตนเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการตรวจสอบแพลตฟอร์มที่เรียกใช้งาน ซึ่งแพลตฟอร์มที่ทางระบบสามารถรองรับ ได้แก่ Windows และ Linux เมื่อแพลตฟอร์มที่เรียกใช้งานโดยระบบสามารถรองรับได้ก็จะสร้างเซรตที่เป็นตัวเซิร์ฟเวอร์สำหรับรอคำร้องขอจากโปรแกรม PRTG ต่อมาสร้างเซรตสำหรับการทำงานของไลน์เอพีไอ และสร้างเซรตสำหรับการทำงานในส่วนของการส่งข้อความแจ้งเตือน (แสดงรูปที่ 3.4)

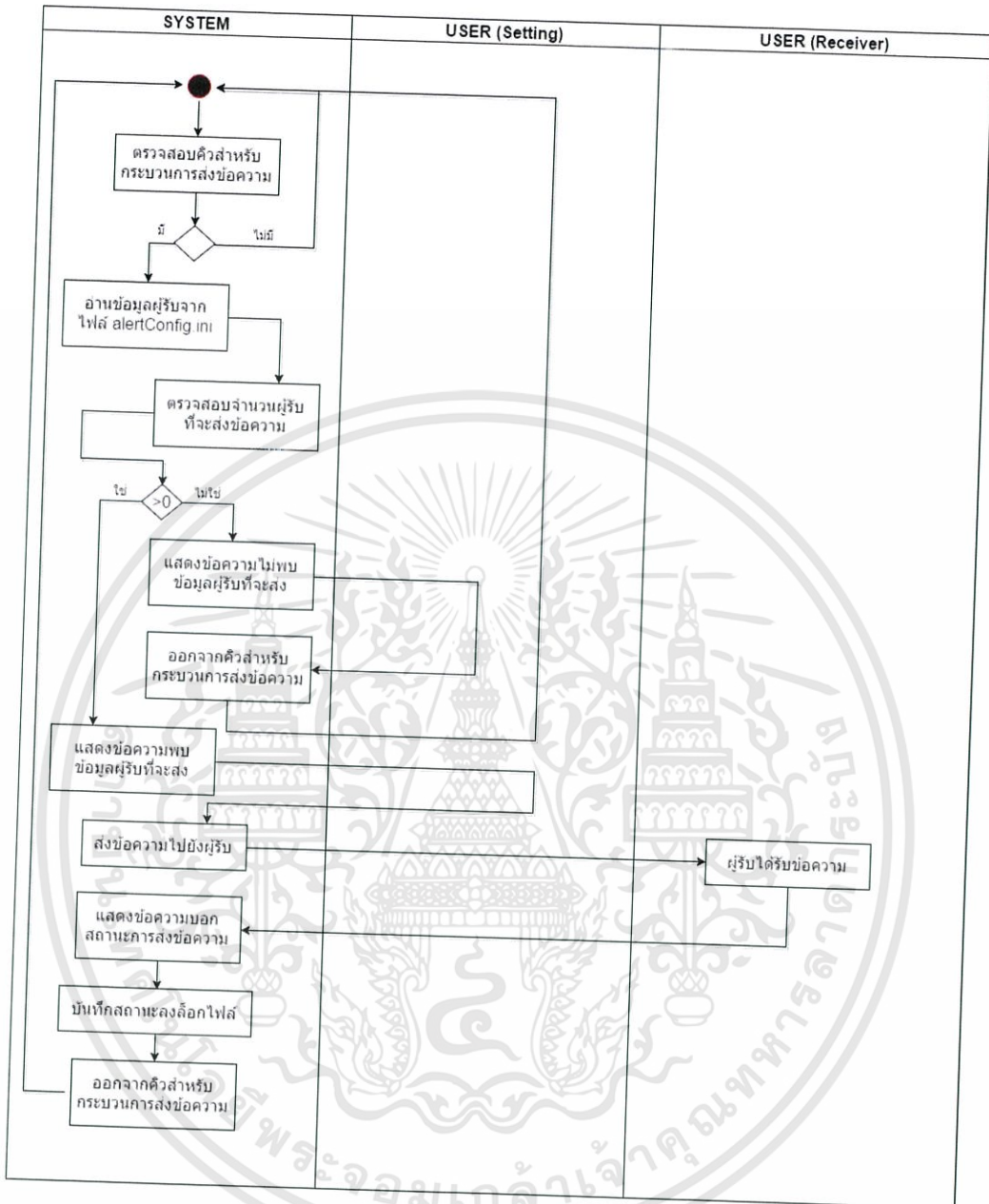
### 3.3.3 Activity Diagram การรับคำร้องจาก PRTG Network Monitor



รูปที่ 3.5 Activity Diagram การรับคำร้องจาก PRTG Network Monitor

ในขั้นตอนนี้จะสามารถทำงานได้หลังจากมีการสั่งเริ่มต้นระบบแจ้งเตือนฯ ไปแล้ว โดยเริ่มต้นเมื่อเซิร์ฟเวอร์ที่มอนิเตอร์นั้นมีข้อผิดพลาดเกิดขึ้นก็จะส่งคำร้องขอมาที่ระบบแจ้งเตือนฯ ที่มีการรอรับคำร้องอยู่ จากนั้นทำการจัดการข้อมูลโดยแยกพารามิเตอร์ที่ส่งมาพร้อมกับ url จากนั้นจะทำการเข้าคิวและนำพารามิเตอร์ดังกล่าวไปใช้ในกระบวนการส่งข้อความต่อไป (แสดงรูปที่ 3.5)

3.3.4 Activity Diagram การส่งข้อความผ่านแอปพลิเคชันไลน์

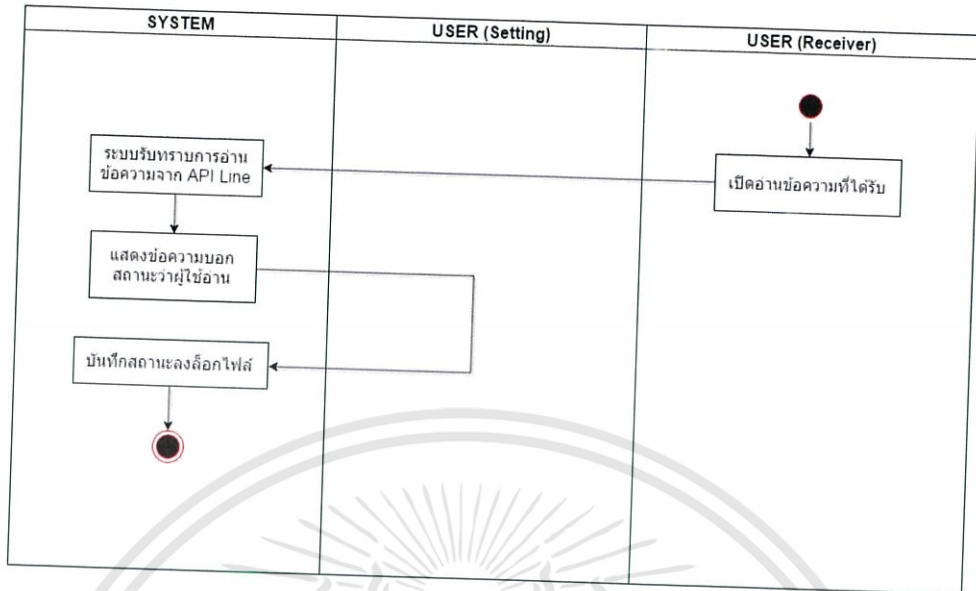


รูปที่ 3.6 Activity Diagram การส่งข้อความผ่านแอปพลิเคชันไลน์

ในขั้นตอนนี้จะเกิดขึ้นหลังจากที่ระบบแจ้งเตือนฯ ได้รับคำสั่งมาแล้วจะมีการเช็คว่ามีข้อมูลในคิวหรือไม่ หากมีข้อมูลในคิวก็ดึงข้อมูลผู้รับ จากไฟล์ alertConfig.ini จากนั้นก็จะทำการเช็คจำนวนผู้รับที่จะส่งข้อความ หากมีข้อมูลผู้รับก็จะดำเนินการส่งข้อความแจ้งเตือนไปยังผู้รับดังกล่าว โดยมีการแสดงข้อความบอกทุกครั้งที่มีการส่งข้อความ เพื่อสามารถยืนยันความถูกต้องระบบมีการบันทึกสถานะการทำงานลงล็อกไฟล์ทุกครั้งก่อนที่จะนำโพเซสการส่งข้อความปัจจุบันออกจากคิว (แสดงรูปที่ 3.6)

148953

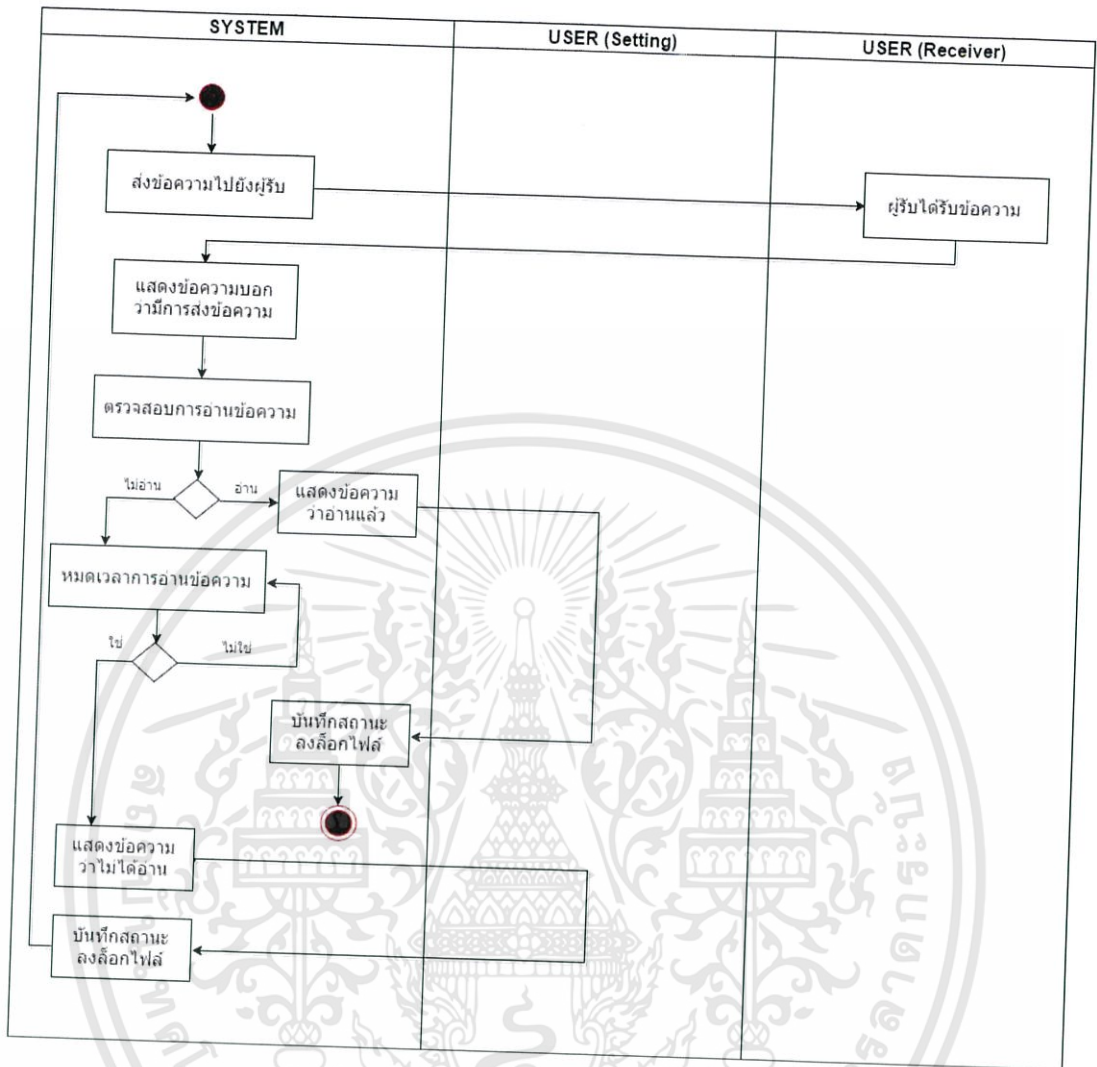
### 3.3.5 Activity Diagram การตรวจสอบการอ่านข้อความ



รูปที่ 3.7 Activity Diagram การตรวจสอบการอ่านข้อความ

ในขั้นตอนนี้จะเกิดขึ้นหลังจากที่ส่งข้อความไปยังผู้รับแล้ว หากผู้รับมีการเปิดอ่านข้อความไลน์เอพีไอจะทำหน้าบอกไปยังระบบแจ้งเตือนๆ ว่ามีการอ่านข้อความจากผู้รับเกิดขึ้น โดยมีการแสดงข้อความบอกทุกครั้งที่มีการอ่านข้อความ และบันทึกสถานะการทำงานทุกครั้งลงล็อกไฟล์ (แสดงรูปที่ 3.7)

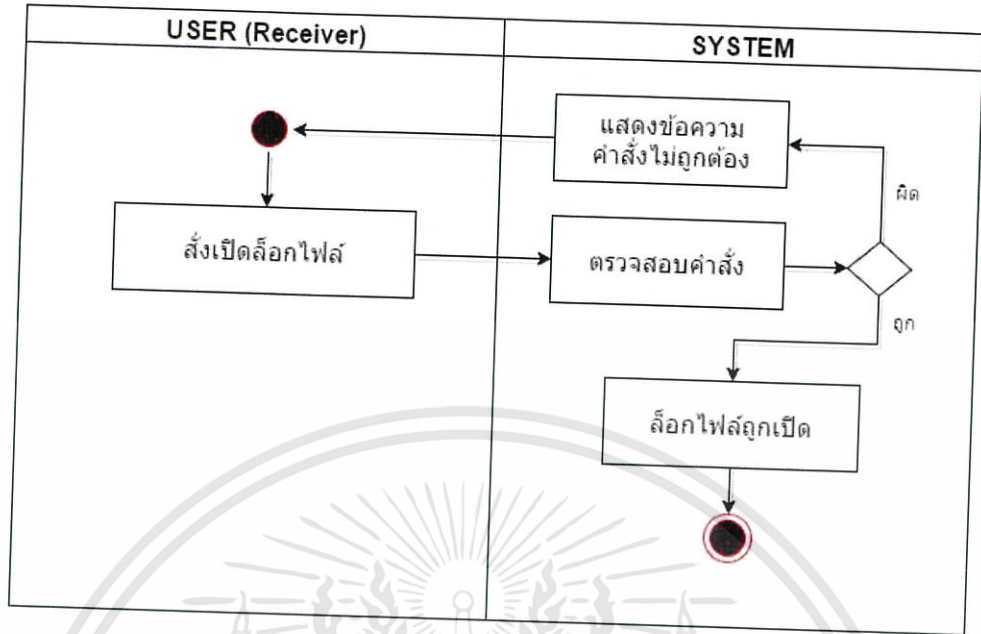
### 3.3.6 Activity Diagram การส่งข้อความไปยังบุคคลถัดไป



รูปที่ 3.8 Activity Diagram การส่งข้อความไปยังบุคคลถัดไป

ในขั้นตอนนี้จะเกิดขึ้นกรณีที่มีการระบุรายชื่อผู้รับที่ต้องการส่งข้อความแจ้งเตือนมากกว่าหนึ่งคน โดยหลังจากส่งข้อความไปยังผู้รับคนแรกแล้ว จะมีการตรวจสอบว่าผู้รับมีการอ่านข้อความภายในระยะเวลาที่กำหนดไว้หรือไม่ หากไม่ก็จะดำเนินการส่งข้อความไปยังบุคคลถัดไป โดยแสดงข้อความบอกและบันทึกข้อมูลลงล็อกไฟล์บอกสถานะการทำงานกรณีผู้รับไม่ได้อ่านข้อความ (แสดงรูปที่ 3.8)

### 3.3.7 Activity Diagram การตรวจสอบล็อกไฟล์สถานะการทำงาน



รูปที่ 3.9 Activity Diagram การตรวจสอบล็อกไฟล์สถานะการทำงาน

ในขั้นตอนนี้ผู้ใช้สามารถใช้คำสั่งเปิดล็อกไฟล์ที่ระบบแจ้งเตือนฯ ได้ทำการบันทึกไว้ได้ โดยผ่าน SSH บน Raspberry pi ไปยังไดเรกทอรีที่เก็บล็อกไฟล์ไว้ หากคำสั่งที่ใช้เปิดล็อกไฟล์อ้างไปยังไดเรกทอรีของล็อกไฟล์ถูกต้องก็จะสามารถเปิดล็อกไฟล์ขึ้นมาตรวจสอบได้ (แสดงรูปที่ 3.9)

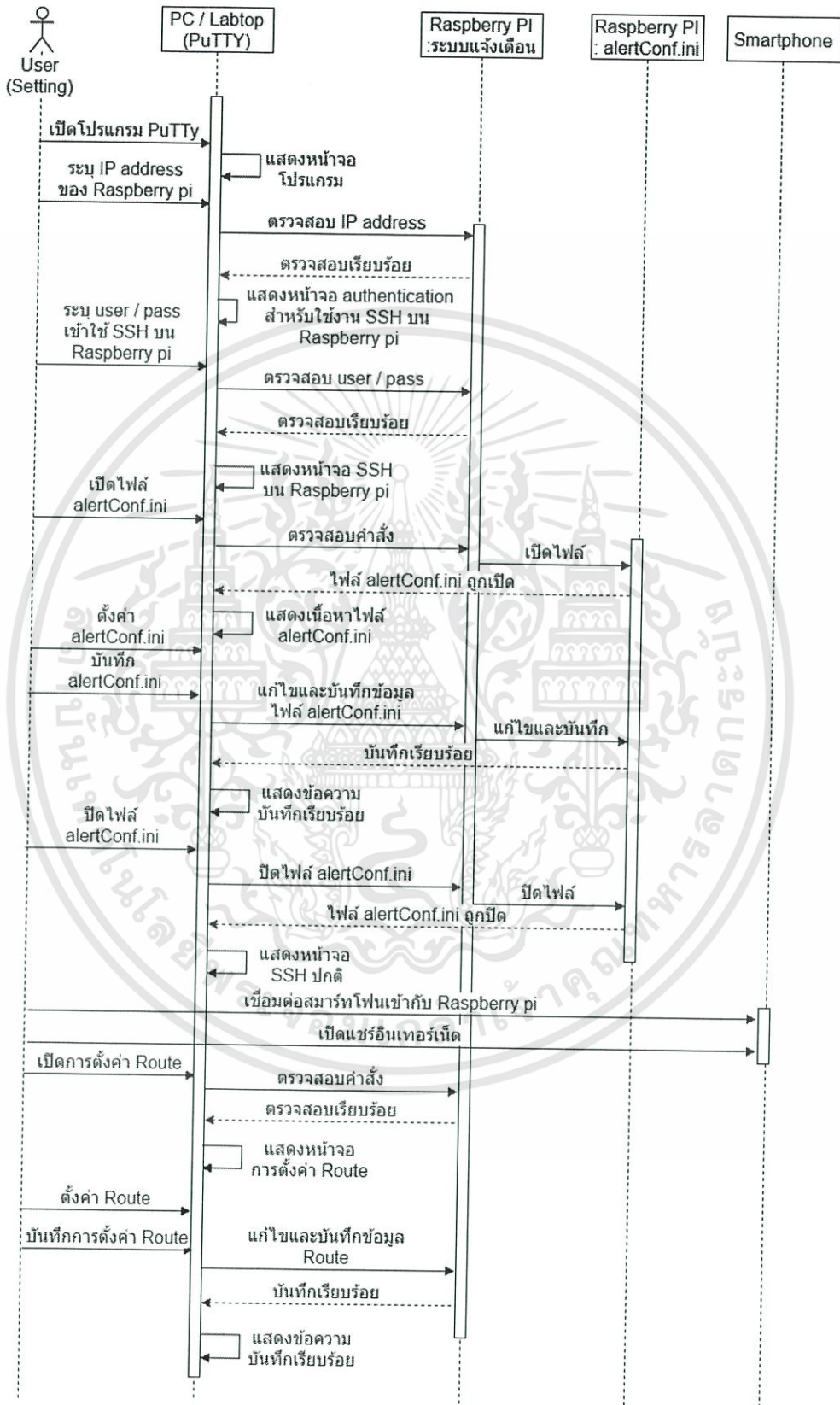
### 3.4 Sequence Diagram

Sequence Diagram เป็นการออกแบบขั้นตอนการทำงานของระบบแบ่งออกเป็น 4 ส่วนย่อยคือ

- 1) Sequence Diagram การตั้งค่าข้อมูลผู้ติดต่อสำหรับระบบไลน์และเปิดการเชื่อมต่ออินเทอร์เน็ต
- 2) Sequence Diagram การเริ่มต้นใช้งานระบบแจ้งเตือนฯ
- 3) Sequence Diagram การแจ้งเตือน (กรณีผู้รับอ่านข้อความ)
- 4) Sequence Diagram การแจ้งเตือน (กรณีผู้รับไม่ได้อ่านข้อความ)



### 3.4.1 Sequence Diagram การตั้งค่าข้อมูลผู้ติดต่อสำหรับระบบไลน์และเปิดการเชื่อมต่ออินเทอร์เน็ต

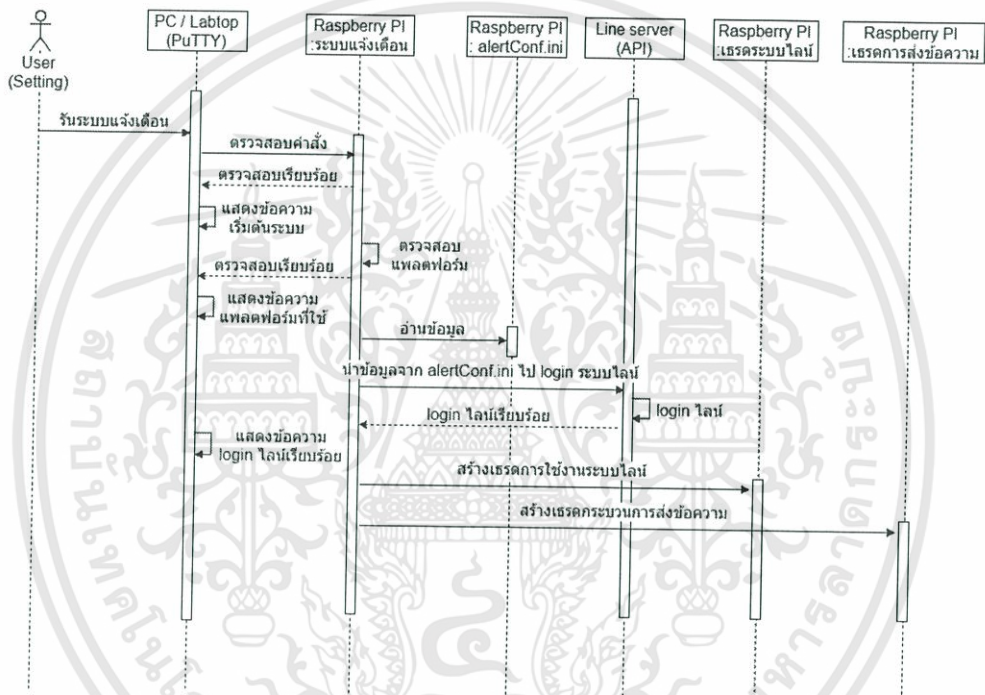


รูปที่ 3.10 Sequence Diagram ของการตั้งค่าข้อมูลผู้ติดต่อสำหรับระบบไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 อธิบายการทำงานได้ว่าผู้ใช้จะต้องทำการตั้งค่าข้อมูลผู้ติดต่อและเปิดการเชื่อมต่ออินเทอร์เน็ตก่อนเริ่มใช้ระบบแจ้งเตือนฯ โดยมีขั้นตอนคือผู้ใช้จะต้องรีโมตไปใช้งาน SSH บน Raspberry pi ผ่านโปรแกรม PuTTY ซึ่งจะต้องผ่านการตรวจสอบยืนยัน (authentication) โดยใช้ username และ password ของ Raspberry pi ที่ได้จากขั้นตอนการติดตั้ง OS หลังจากตรวจสอบยืนยันเข้าใช้งาน SSH ได้แล้ว ผู้ใช้สามารถเปิดไฟล์ alertConf.ini ขึ้นมาเพื่อตั้งค่าข้อมูลผู้รับสำหรับระบบไลน์ได้ เมื่อแก้ไขหรือตั้งค่าเสร็จเรียบร้อยให้ทำการบันทึกและปิดไฟล์ จากนั้นให้ผู้ใช้เชื่อมต่อ Raspberry pi เข้ากับสมาร์ตโฟนที่เปิดโหมดแชร์อินเทอร์เน็ตและทำการตั้งค่าเส้นทางเครือข่ายให้ส่งออกโดยผ่านเครือข่ายของโทรศัพท์เป็นอันเสร็จกระบวนการ

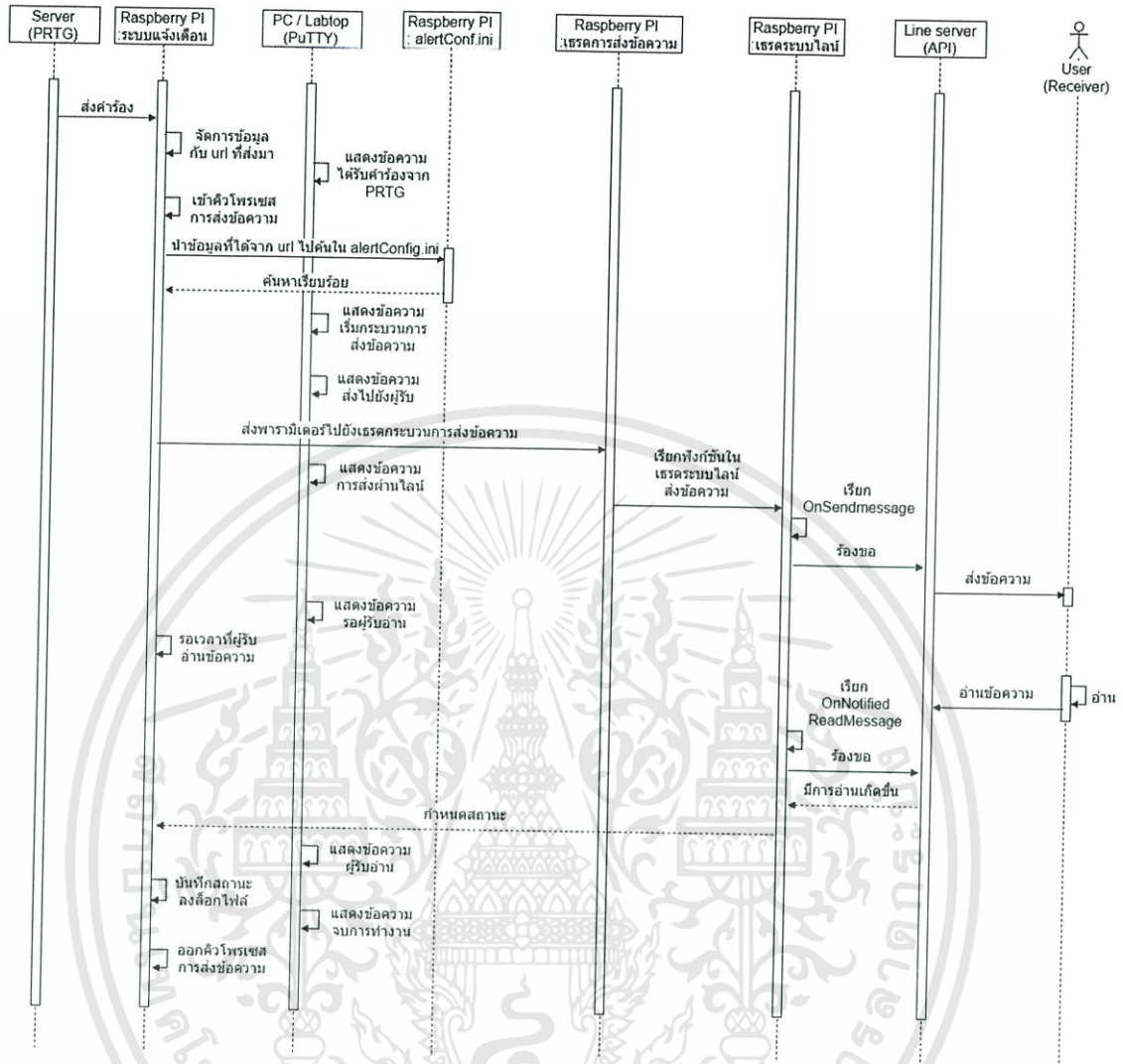
### 3.4.2 Sequence Diagram การเริ่มต้นใช้งานระบบแจ้งเตือนฯ



รูปที่ 3.11 Sequence Diagram การเริ่มต้นใช้งานระบบแจ้งเตือนฯ

รูปที่ 3.12 อธิบายการทำงานได้ว่าหลังจากที่ผู้ใช้รีโมตไปใช้งาน SSH บน Raspberry pi ผ่านโปรแกรม PuTTY และทำการตรวจสอบยืนยันเรียบร้อยแล้ว ให้ผู้ใช้ใส่คำสั่งเปิดโปรแกรมระบบแจ้งเตือนฯ ผ่าน SSH จากนั้นโปรแกรมระบบแจ้งเตือนฯ ก็จะเริ่มทำงานโดยอันดับแรกจะต้องตรวจสอบแพลตฟอร์มที่ใช้รันโปรแกรมก่อนจากนั้นจะอ่านข้อมูลจากไฟล์ alertConf.ini เพื่อนำมาใช้ในการล็อกอินระบบไลน์เพื่อจะใช้ในการส่งข้อความ หลังจากล็อกอินเสร็จแล้วจะเป็นการสร้างเซิร์ฟเวอร์การทำงานขึ้นมา 2 ตัว คือเซิร์ฟเวอร์ที่เป็นตัวคอยอัปเดตการทำงานของระบบไลน์และเซิร์ฟเวอร์ระบบการส่งข้อความ

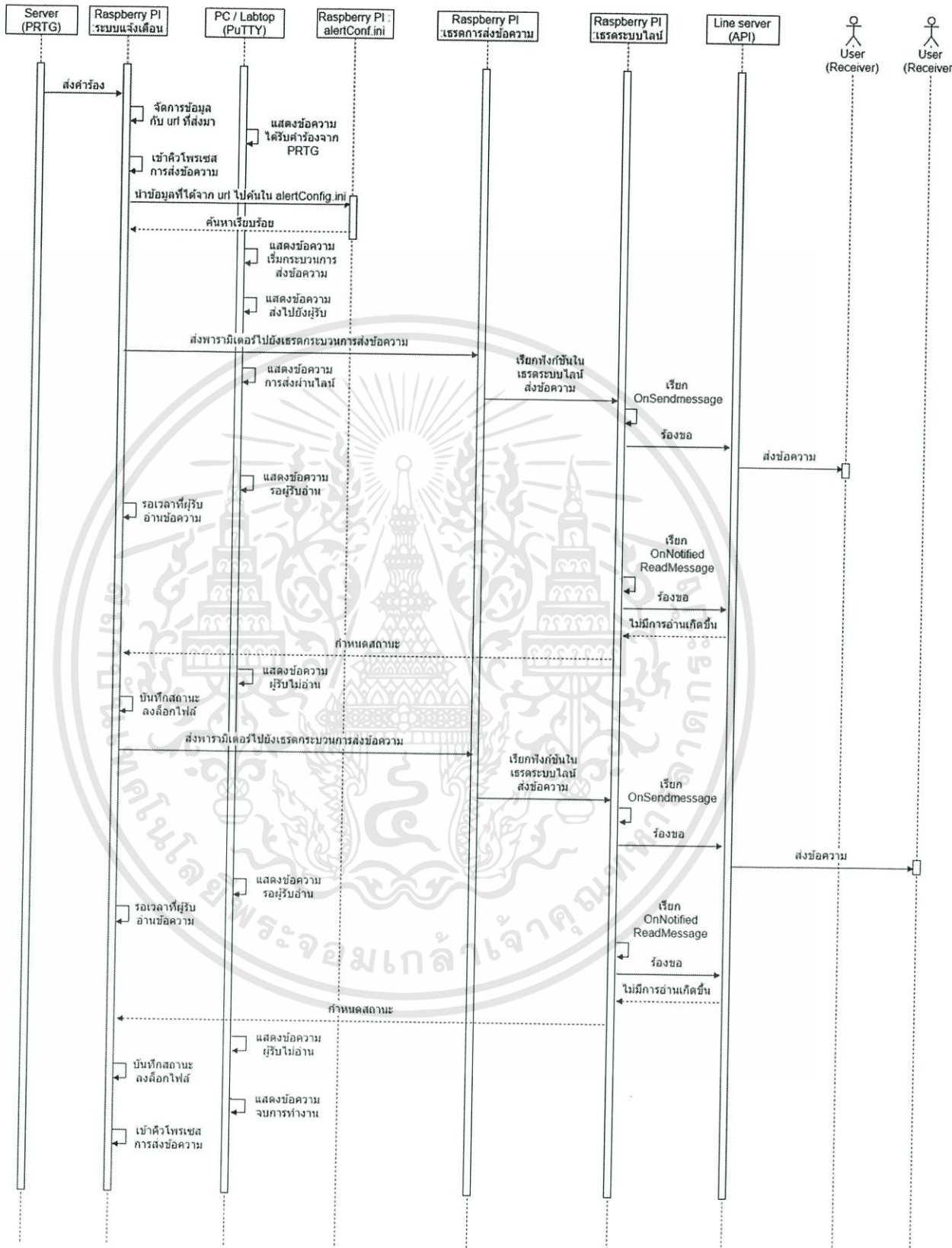
### 3.4.3 Sequence Diagram การแจ้งเตือน (กรณีผู้รับอ่านข้อความ)



รูปที่ 3.12 Sequence Diagram การแจ้งเตือน (กรณีผู้รับอ่านข้อความ)

รูปที่ 3.13 อธิบายการทำงานได้ว่าระบบแจ้งเตือนฯ นั้นทำงานร่วมกับโปรแกรมมอนิเตอร์ระบบ (PRTG Network Monitor) เมื่อมีข้อผิดพลาดเกิดขึ้นโปรแกรมมอนิเตอร์จะส่งคำร้องขอผ่านโปรโตคอล HTTP โดยส่งพารามิเตอร์ไปยังระบบแจ้งเตือนฯ ที่รันการทำงานไว้ ระบบจะรับพารามิเตอร์ที่ส่งมาเก็บในตัวแปรสำหรับนำไปค้นหารายชื่อผู้รับ จากนั้นจะเข้าคิวในกระบวนการส่งข้อความโดยเอาตัวแปรที่เก็บพารามิเตอร์ไปค้นหารายชื่อผู้รับจากไฟล์ alertConfig.ini แล้วเรดกระบวนการส่งข้อความจะเรียกฟังก์ชันส่งข้อความผ่านไลน์ในเซิร์ฟเวอร์ออนไลน์เพื่อส่งคำร้องไปยังเซิร์ฟเวอร์ไลน์แล้วทำการส่งข้อความแจ้งเตือนไปยังผู้รับ โดยเซิร์ฟเวอร์ไลน์จะเรียกฟังก์ชันอ่านข้อความมายังเซิร์ฟเวอร์เพื่อรอเวลาที่ผู้รับอ่านข้อความ หากผู้รับได้อ่านข้อความที่ส่งมาระบบจะกำหนดสถานะว่าอ่านแล้วและทำการบันทึกข้อมูลการส่งข้อความแจ้งเตือนลงล็อกไฟล์และแสดงสถานะจบการทำงานแล้วออกจากคิวเป็นอันจบกระบวนการส่งข้อความแจ้งเตือน

### 3.4.4 Sequence Diagram การแจ้งเตือน (กรณีผู้รับไม่ได้อ่านข้อความ)



รูปที่ 3.13 Sequence Diagram การแจ้งเตือน (กรณีผู้รับไม่ได้อ่านข้อความ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.14 มีกระบวนการทำงานคล้ายกับรูปที่ 3.13 เมื่อระบบแจ้งเตือนได้ทำการส่งข้อความแจ้งเตือนไปยังผู้รับแล้ว ระบบจะมีการตั้งเวลารอคอยผู้ใช้อ่านข้อความ หากผู้ใช้นั้นดูแล้วไม่ได้อ่านข้อความที่ส่งมาภายในระยะเวลาที่กำหนด ระบบจะกำหนดสถานะเป็นไม่ได้อ่าน แล้วทำการส่งข้อความไปยังผู้รับบุคคลถัดไปได้ระบุไว้ในส่วนของรายชื่อข้อมูลผู้รับ ระบบจะสิ้นสุดกระบวนการทำงานเมื่อมีการส่งข้อความไปยังผู้รับจนครบทุกคนแล้วและยังไม่ได้มีการอ่านข้อความหรือส่งข้อความไปยังผู้รับแล้วมีการอ่านเกิดขึ้น ระบบจะทำการบันทึกข้อมูลการส่งข้อความแจ้งเตือนลงล็อกไฟล์และแสดงสถานะจบการทำงานแล้วออกจากคิวเป็นอันจบกระบวนการส่งข้อความแจ้งเตือน



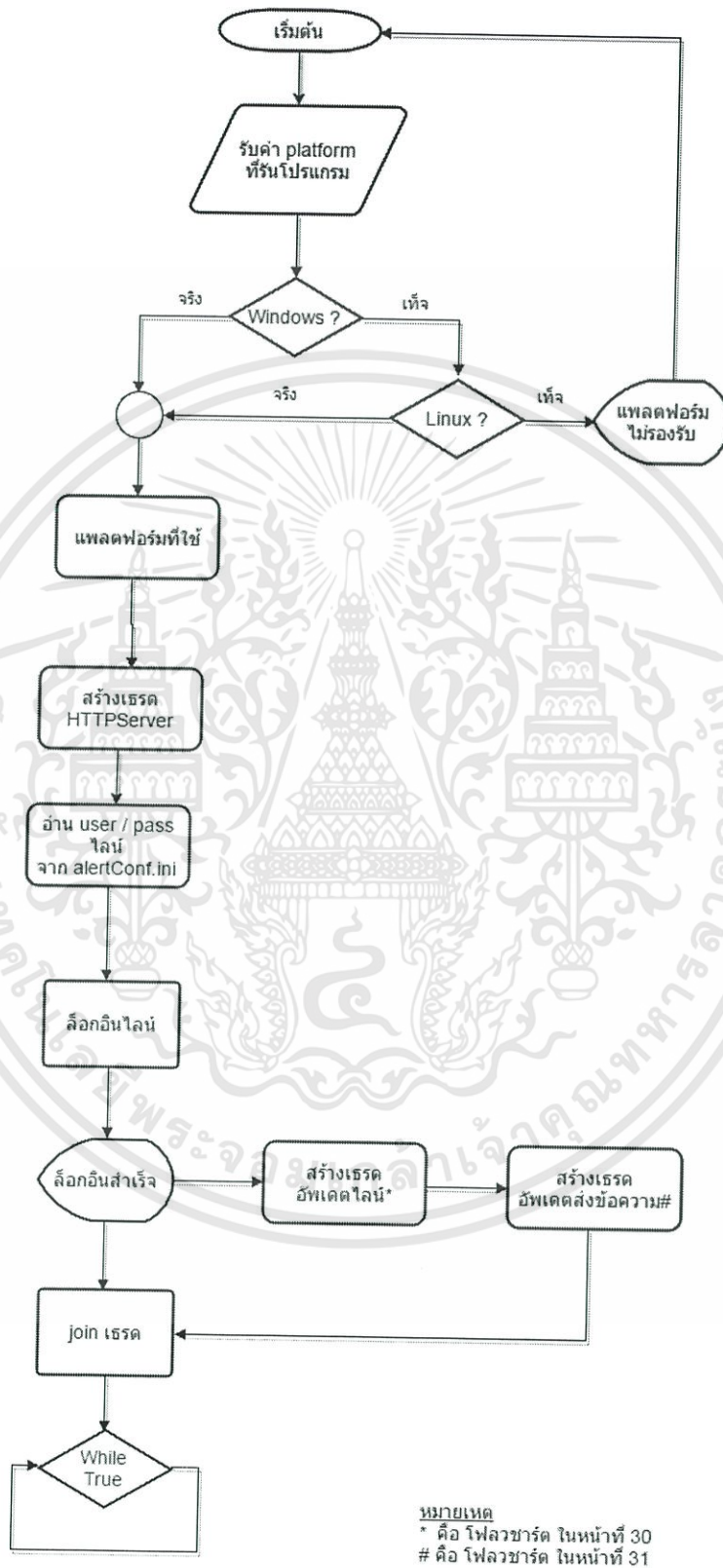
### 3.5 Flowchart Diagram

Flowchart Diagram เป็นการออกแบบขั้นตอนการทำงานของระบบแบ่งออกเป็น 3 ส่วนย่อยคือ

- 1) Flowchart Diagram การทำงานของโปรแกรมจากสถานะเริ่มต้นจนถึงสถานะเตรียมพร้อม
- 2) Flowchart Diagram การทำงานของเซตสำหรับดักจับอีเว้นท์ที่เกิดขึ้นบนระบบ
- 3) Flowchart Diagram การทำงานของเซตส่งข้อความ



### 3.5.1 Flowchart Diagram การทำงานของโปรแกรมจากสถานะเริ่มต้นจนถึงสถานะเตรียมพร้อม



รูปที่ 3.14 Flowchart Diagram การทำงานของโปรแกรมจากสถานะเริ่มต้นจนถึงสถานะเตรียมพร้อม

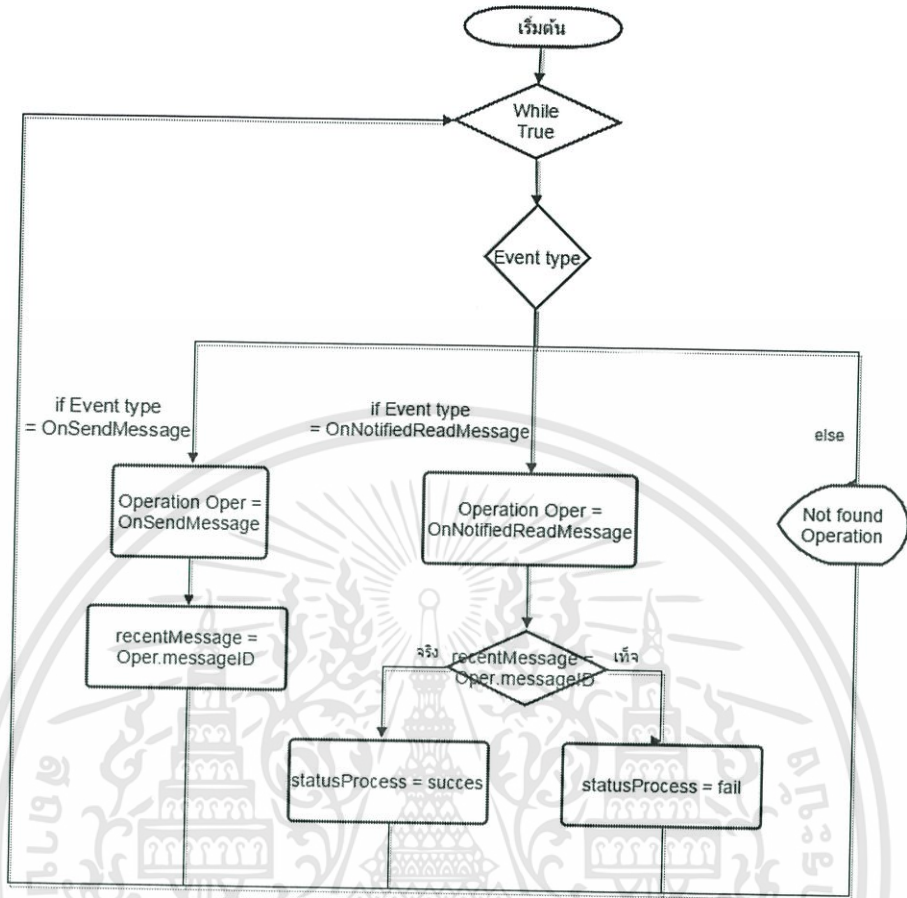
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.14 จะเป็นแผนภาพการทำงานของโปรแกรมในการเตรียมความพร้อมสำหรับส่งข้อความ โดยเริ่มจากการสั่งให้โปรแกรมทำงาน เริ่มจากการตรวจสอบระบบปฏิบัติการที่รันโปรแกรม ถัดจากนั้นจะสร้างเซิร์ฟเวอร์ HTTP Server ซึ่งทำหน้าที่รับการร้องขอจากเซิร์ฟเวอร์ที่ติดตั้งโปรแกรม PRTG ผ่านโปรโตคอล http เข้าสู่โปรแกรมและบันทึกข้อมูลเข้าสู่คิวการส่งข้อความ หลังจากนั้นจะทำการดึงข้อมูลรหัสผ่านจากไฟล์ตั้งค่าที่เก็บอยู่ที่เดียวกับโปรแกรมเพื่อล็อกอินเชื่อมต่อระบบไลน์ เมื่อทำการเชื่อมต่อเสร็จสิ้นจะทำการสร้างเซิร์ฟเวอร์ไลน์อัปเดตเพื่อรับการตอบสนองการแจ้งเตือนจากเซิร์ฟเวอร์ไลน์ และสร้างเซิร์ฟเวอร์การส่งข้อความผ่านไลน์ตามลำดับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.2 Flowchart Diagram การทำงานของเซรด์สำหรับดักจับอีเว้นท์ที่เกิดขึ้นบนระบบ



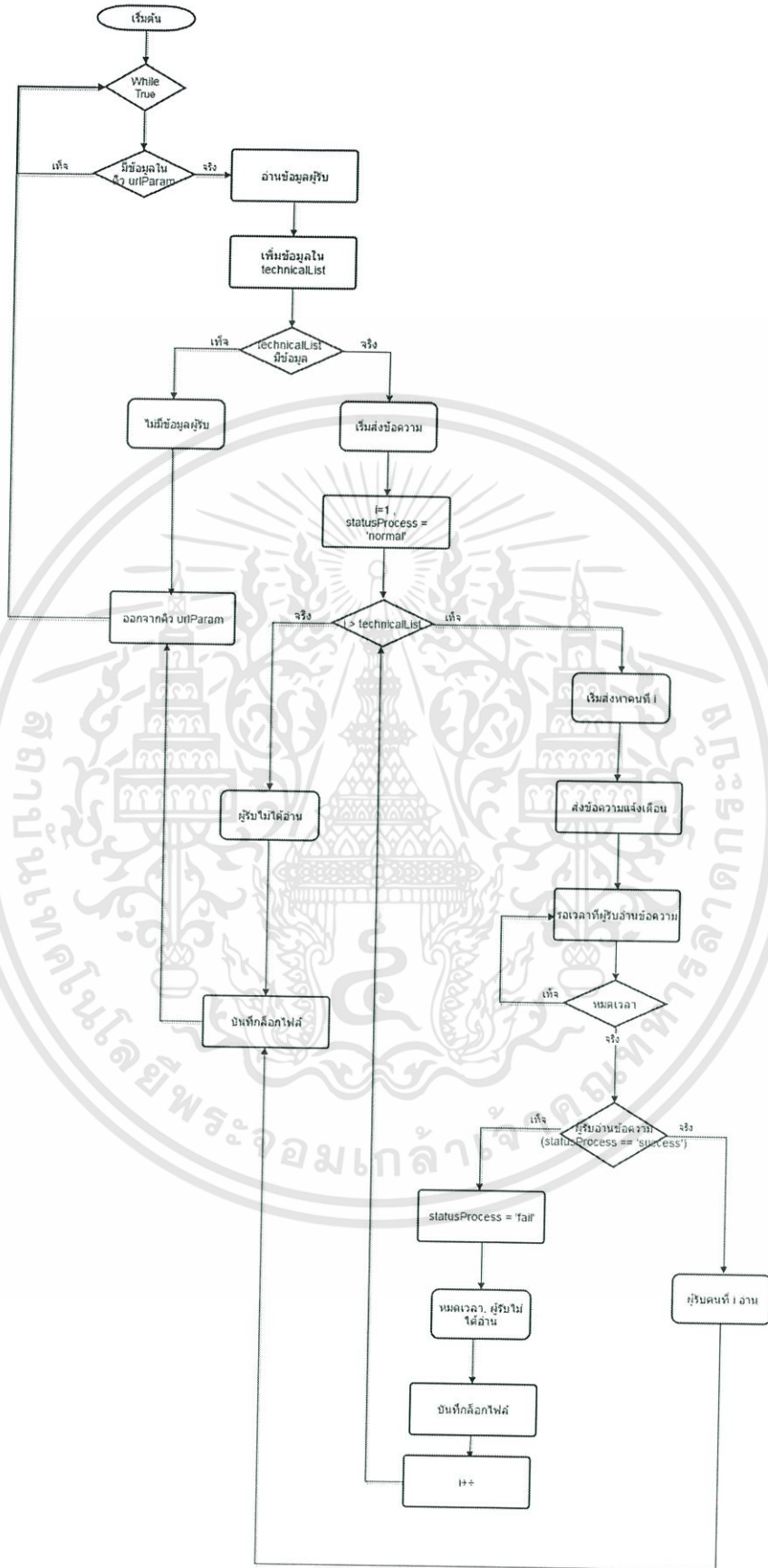
รูปที่ 3.15 Flowchart Diagram การทำงานของเซรด์ดักจับอีเว้นท์ที่เกิดขึ้นบนระบบ

รูปที่ 3.15 เป็นส่วนการทำงานของเซรด์ไลน์อัปเดตที่จะคอยทำหน้าที่ในการรับการตอบสนองจากไลน์ โดยจะมีการทำงานในลักษณะที่เป็นอีเว้นท์ทริกเกอร์เมื่อเกิดเหตุการณ์ขึ้นจะทำให้ทำการดักจับพารามิเตอร์ที่ถูกส่งมาทางไลน์เข้าสู่โปรแกรม โดยโปรแกรมที่พัฒนานั้นจะมีการใช้อีเว้นท์ทริกเกอร์อยู่ 2 ส่วนดังนี้

1) อีเว้นท์ OnSendMessage การทำงานของฟังก์ชันนี้จะเกิดขึ้นเมื่อมีการส่งข้อความออกไปหาผู้รับโดยผู้พัฒนาได้เพิ่มเติมส่วนของการทำงานลงไปในส่วนนี้โดยให้ทำการเก็บไอดีของข้อความปัจจุบันที่ส่งออกไปเพื่อเอาไว้ใช้ในการยืนยันว่าข้อความที่ถูกอ่านเป็นข้อความเดียวกันกับข้อความที่ฟังก์ชันส่งออกไปจากโปรแกรม

2) อีเว้นท์ OnNotifiedReadMessage การทำงานของฟังก์ชันนี้จะเกิดขึ้นเมื่อผู้รับได้อ่านข้อความทางไลน์และจะทำการส่งข้อมูลเป็นไอดีของข้อความที่ถูกอ่านและไอดีของผู้อ่านข้อความกลับมาโดยผู้พัฒนาได้เพิ่มเติมส่วนของการทำงานนี้ลงไปโดยเมื่อเกิดเหตุการณ์นี้ให้เช็คข้อความที่ถูกอ่านด้วยไอดีว่าตรงกับไอดีของข้อความที่ส่งไปหรือไม่ ถ้าตรงกันถือว่าการส่งข้อความนั้นประสบความสำเร็จ

3.5.3 Flowchart Diagram การทำงานของเซตส่งข้อความ



รูปที่ 3.16 Flowchart Diagram การทำงานของเซตส่งข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.16 เป็นส่วนของการทำงานบนเรดส่งข้อความที่จะทำหน้าที่ในการค้นหารายชื่อผู้รับที่ต้องการส่งข้อความ การส่งข้อความและรอการตอบรับจากผู้รับในช่วงระยะเวลาหนึ่งการจะอัปเดตผลของการส่งข้อความครั้งนั้น โดยจะเริ่มจากนำข้อมูลที่ได้รับมาไปค้นหารายชื่อเพื่อสร้างคิวลำดับการส่งข้อความ โดยมีเงื่อนไขว่าถ้าค้นหารายชื่อไม่พบก็จะไม่ทำการส่งข้อความและเริ่มคิวถัดไป แต่ถ้าหากพบก็จะเข้าสู่การเช็คเงื่อนไขลำดับการส่งปัจจุบันยังมีค่าไม่เท่ากับจำนวนรายชื่อของผู้ส่งข้อความ ถ้าจริงก็ให้ทำการดึงข้อมูลลำดับปัจจุบันและดำเนินการส่งข้อความ จากนั้นทำการรอการตอบรับจากการอ่านข้อความ หากเกินเวลาที่กำหนดจริงสถานะการส่งข้อความนั้นจะถือว่าล้มเหลวและดึงข้อมูลของรายชื่อคนถัดไปมาทำการส่งข้อความต่อไปจนครบรายชื่อผู้รับข้อความ ถ้าข้อความที่ถูกส่งมีการตอบรับการอ่านโดยผู้รับที่ถูกส่งไปถึง จะถือว่าการส่งครั้งนั้นประสบความสำเร็จและทำการดึงข้อมูลจากคิวการส่งข้อความถัดไปมาดำเนินการต่อ



### 3.6 การออกแบบระบบ

#### 3.6.1 การตั้งค่าไฟล์ Configuration

เนื่องจากระบบแจ้งเตือนทางแอปพลิเคชันไลน์สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์ ออกแบบมาให้สามารถปรับแต่งข้อมูลที่ระบุไปยังผู้รับข้อความปลายทางที่ใช้สำหรับการส่งข้อความ ได้แก่ ข้อมูล user, password ในการเข้าใช้งานระบบไลน์ รายชื่อของกลุ่มผู้ที่ต้องการส่งข้อความ การนัดหมายเวลาสำหรับส่งข้อความแต่ละครั้ง โดยสามารถแก้ไขผ่านทางไฟล์ Configuration เพื่ออำนวยความสะดวกให้แก่ผู้ใช้ โดยที่ผู้ใช้ไม่จำเป็นต้องดำเนินการแก้ไขในส่วนของโปรแกรมซึ่งอาจจะทำให้เกิดข้อผิดพลาดขึ้นได้ ซึ่งการปรับแต่งไฟล์ Configuration สามารถเขียนได้ด้วยโปรแกรม text editor ทั่วไปและทำการบันทึกไฟล์เป็นสกุล .ini โดยมีโครงสร้างพื้นฐานมีดังนี้

```

1  [System]
2  username = social_alert@kmitl.ac.th
3  password = P@ssw0rd
4  time_next = 5
5  time_out = 30
6
7  [database_dep]
8  receiver1 = {John}
9  receiver2 = {Bob}
10 receiver3 = {Cartor}
11
12 [network_dep]
13 receiver1 = {Jerry}
14 receiver2 = {Tony}
15 receiver3 = {Susie}
16 receiver4 = {Deb}
17
18

```

รูปที่ 3.17 โครงสร้างไฟล์ Configuration ที่ใช้เป็นข้อมูลสำหรับส่งข้อความ

ในการปรับแต่งไฟล์ Configuration สำหรับการทำงานของโปรแกรม จำเป็นต้องมีโครงสร้างพื้นฐานดังรูปที่ 3.17 หากมีรูปแบบโครงสร้างไม่ตรงโปรแกรมจะไม่สามารถดำเนินการต่อไปได้ ซึ่งแต่ละส่วนจะมีความสำคัญแบ่งเป็น 2 ส่วนดังนี้

ส่วนที่ 1 [System] ประกอบด้วย 4 ส่วนคือ

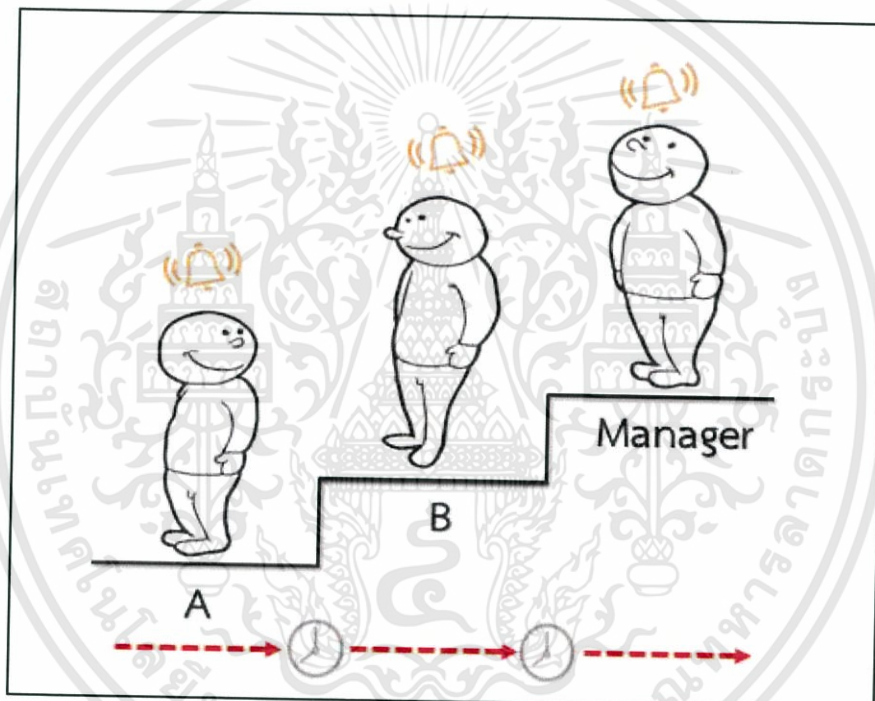
- 1) username คือ ใอดีในการใช้งาน Line
- 2) password คือ รหัสผ่านในการใช้งาน Line
- 3) time\_next คือ ระยะเวลาในการส่งข้อความ เมื่อเกิดปัญหาในการส่งข้อความไม่สำเร็จ เพื่อลดระยะเวลาในการรอส่งข้อความครั้งต่อไป
- 4) time\_out คือ ระยะเวลาในการส่งข้อความกรณีส่งข้อความได้ปกติ หากไม่มีการตอบกลับจากผู้รับจนกระทั่งหมดเวลาดังกล่าว ระบบจะดำเนินการส่งไปยังบุคคลถัดไป

ส่วนที่ 2 รายชื่อข้อมูลกลุ่มบุคคลที่ใช้ในการส่งข้อความ เช่น [database\_dep], [network\_dep], ...

ในส่วนนี้เป็นการระบุกลุ่มของบุคคลที่ใช้ในการส่งข้อความจากต้นทางไปยังไปยังทุกคนในรายชื่อ หากข้อมูลในบรรทัดใดมีข้อผิดพลาดหรือรูปแบบโครงสร้างไม่ถูกต้องข้อมูลจะไปถูกนำเข้าไปประมวลผล และจะแสดงผลข้อความเตือน เพื่อให้ทราบว่าข้อมูลที่ระบุนั้นไม่ถูกต้อง โดยแต่ละชุดข้อมูล (receiver) ประกอบด้วย ชื่อของบุคคลที่แสดงบนไลน์ โดยข้อมูลกลุ่มรายชื่อสามารถเพิ่มเติมได้ในภายหลัง

### 3.6.2 ฟังก์ชันการเพิ่มระดับความสำคัญของบุคคลในการแจ้งเตือน (Contact forward)

ระบบแจ้งเตือนทางแอปพลิเคชันสำหรับอุปกรณ์ในท้องเซิร์ฟเวอร์สามารถส่งข้อความแจ้งเตือนไปยังผู้รับตามลำดับความสำคัญที่ระบุข้อมูลผู้รับที่จะส่งไว้ในไฟล์ Configuration ได้ ยกตัวอย่างเช่น ชุดข้อมูลผู้รับที่จะส่งนี้ประกอบด้วย นาย A นาย B และ Manager เมื่อมีข้อผิดพลาดเกิดขึ้น ระบบจะเริ่มส่งข้อความแจ้งเตือนโดยเริ่มส่งไปหานาย A ก่อนลำดับแรก หากนาย A ไม่ได้ตอบรับข้อความภายในระยะเวลาที่กำหนด ระบบจะทำการส่งข้อความต่อไปยังนาย B หากนาย B มีการเปิดอ่านข้อความระบบจะยุติการส่งข้อความในการแจ้งเตือนครั้งนั้น หรือถ้าหากนาย B ไม่ได้มีการเปิดอ่านข้อความภายในระยะเวลาที่กำหนดเช่นเดียวกับนาย A ระบบก็จะทำการส่งข้อความไปยังบุคคลถัดไป ดังรูปที่ 3.18 จะเป็น Manager ในฝ่ายนั้น เพื่อที่จะได้รับทราบและติดต่อนาย A หรือนาย B เพื่อเร่งดำเนินการแก้ไขข้อผิดพลาดที่เกิดขึ้น



รูปที่ 3.18 แสดงการจำลองฟังก์ชันเพิ่มระดับความสำคัญของบุคคลในการแจ้งเตือน

### 3.6.3 โปรแกรมส่งข้อความโดยใช้ LineSharp

โปรแกรมรับชื่อผู้ติดต่อที่ต้องการส่งข้อความ “Hi” ผ่านทางเป็นพิมพ์จากผู้ใช้งาน และทำการส่งข้อความไปยังผู้ติดต่อ ตัวอย่างแสดงดังรูปที่ 3.19

```

1 static void Main(string[] args)
2 {
3     LineClient line = new LineClient();
4     List<Contact> ct = line.GetContacts(line.GetContactIDs());
5     Console.WriteLine("Enter the name of a user you would like to send a 'Hi' to:");
6     string name = Console.ReadLine();
7     for (int i = 0; i < ct.Count; i++)
8     {
9         if (ct[i].Name.Contains(name))
10        {
11            line.SendMessage(ct[i].ID, "Hi!");
12        }
13    }
14    line.OnReceiveMessage += (o, eventArgs) => Console.WriteLine(eventArgs.Message.Text);
15    line.Login(username, password);
16    while (true) line.Update();
17 }

```

รูปที่ 3.19 โปรแกรมส่งข้อความและรับข้อความเบื้องต้น

#### คำอธิบายโปรแกรม

- บรรทัดที่ 3 เป็นการประกาศตัวแปร line เป็นคลาส LineClient เพื่อใช้งานฟังก์ชันต่างๆ ที่ได้รับรูไว้  
ไว้ในคลาส
- บรรทัดที่ 4 เป็นการประกาศตัวแปร ct เป็นลิสของข้อมูลประเภท Contract เพื่อรับรายชื่อของผู้ติดต่อ  
ติดต่อไลน์ในระบบ ผ่านฟังก์ชัน GetContacts( Parameter ) ซึ่งจะต้องส่ง Parameter ที่ได้รับมาจากฟังก์ชัน  
GetContractIDs ( )
- บรรทัดที่ 5 เป็นคำสั่งแสดงข้อความโต้ตอบกับผู้ใช้โปรแกรมให้พิมพ์ข้อมูลชื่อผู้ติดต่อบน Line  
สู่ระบบ
- บรรทัดที่ 6 เป็นการประกาศตัวแปร name เพื่อรับค่าจากข้อมูลผู้ใช้โปรแกรมส่งกลับมา
- บรรทัดที่ 7 เป็นคำสั่ง for loop โดยมีการตรวจสอบเงื่อนไข ถ้าเงื่อนไขไม่เป็นเท็จจะทำคำสั่ง  
ภายใน loop จนถึงบรรทัดสุดท้าย ก่อนจะเพิ่มจำนวน counter คือ i++ และทำการ  
เช็คเงื่อนไขอีกครั้ง จนกระทั่งเงื่อนไขเป็นเท็จจะถือเป็นการจบ loop การทำงาน
- บรรทัดที่ 9 เป็นคำสั่งเช็คเงื่อนไข ถ้าข้อมูลชื่อในลิสของผู้ติดต่อตรงกับชื่อที่ได้รับค่ามาจาก  
คีย์บอร์ดถ้าใช่ให้ทำเงื่อนไขบรรทัดที่ 8
- บรรทัดที่ 11 เป็นคำสั่งส่งข้อความไปยังผู้ติดต่อโดยใช้ฟังก์ชัน SendMessage( parameter1,  
parameter2 ) โดยที่ Parameter1 คือ ข้อมูล ID ของผู้ติดต่อที่ได้จากการวน loop  
ค้นหา ส่วนParameter2 คือ ข้อความประเภท String ที่ต้องการส่งหาผู้ติดต่อ

- บรรทัดที่ 14 เป็นการประกาศสร้าง Event OnReceiveMessage เป็นอีเว้นที่จะกระทำโดยอัตโนมัติ เมื่อเกิดเหตุการณ์ที่ได้รับข้อความส่งมา เมื่อเกิดเหตุการณ์ดังกล่าวจะทำคำสั่งแสดงผลข้อความที่ได้รับโดยฟังก์ชัน Console.WriteLine(eventArgs.Message.Text) โดยที่ eventArgs.Message.Text คือพารามิเตอร์ที่มีโครงสร้างเป็น Object รับข้อความที่ส่งเข้ามาแสดงผลออกทางหน้าจอ
- บรรทัดที่ 15 เป็นคำสั่งที่ใช้ในการสั่ง Login เข้าสู่ระบบ Line โดยใช้ฟังก์ชัน Login( username, password ) โดยที่ username คือพารามิเตอร์ประเภท String รับข้อมูลบัญชีผู้ใช้ที่เข้าสู่ระบบ password คือพารามิเตอร์ประเภท String รับข้อมูลรหัสผ่านสำหรับบัญชีผู้ใช้ในการยืนยันความถูกต้อง
- บรรทัดที่ 16 เป็นฟังก์ชันที่ใช้ในการอ่านข้อมูล event ที่เกิดขึ้นบนระบบ Line เช่น การส่งข้อความ , การได้รับข้อความ, การอ่านข้อความ เป็นต้น โดยมีการทำงานแบบ While loop แบบไม่ได้กำหนดเงื่อนไขสิ้นสุดการทำงานเอาไว้

### ผลลัพธ์ของโปรแกรม

จากชุดคำสั่งข้างต้น เมื่อนำมาประมวลผลจะได้ผลลัพธ์ดังรูปที่ 3.20

```

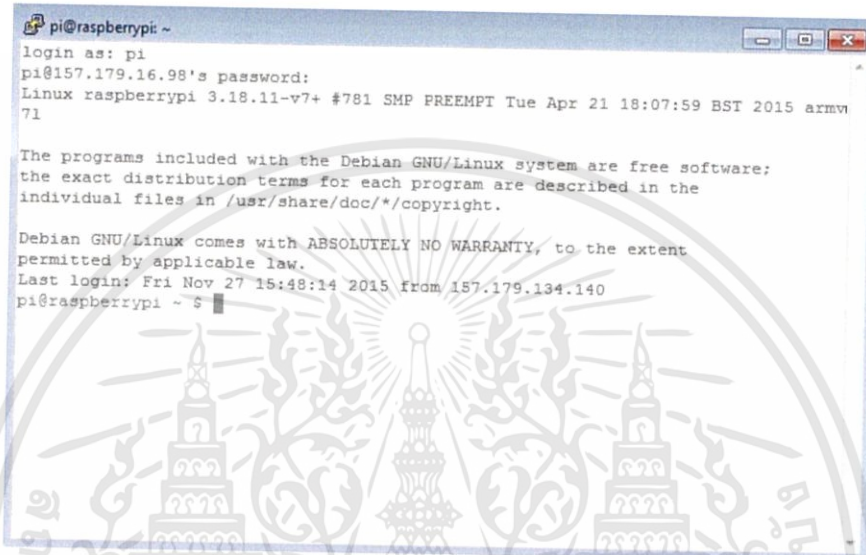
Operation recieved.
>>Type: RECEIVED_MESSAGE
>>Oprev: 5129
>>Param1: 0
>>Param2:
>>Param3:
>>Message:
>>>>Text: 11/26/2015 11:18:30 AM -- helloworld
>>>>CreatedTime: 1448511520307
>>>>DeliveredTime: 0
>>>>ID: 3519289957356
>>>>To:
>>>>From: ua848a9d4ca09b97a58cc0fe9ee8315c5
>>>>Type: NONE
  
```

รูปที่ 3.20 ผลลัพธ์ของโปรแกรม

### 3.6.4 กระบวนการทำงาน

#### 1) การรีโมตสำหรับใช้งาน Raspberry Pi

เนื่องจาก Raspberry Pi เป็นอุปกรณ์คอมพิวเตอร์พกพาขนาดเล็กที่ไม่ได้มีโมดูลของอุปกรณ์แสดงผล ในการเข้าไปใช้งาน terminal เพื่อรันโปรแกรมระบบแจ้งเตือนนั้นจะต้องทำการรีโมตขึ้นไปใช้งานโดยผ่านโปรแกรม PuTTY เมื่อทำการตรวจสอบยืนยันตัวตนเรียบร้อยแล้ว จะปรากฏหน้าต่างของโปรแกรมดังรูปที่ 3.21



```

pi@raspberrypi: ~
login as: pi
pi@157.179.16.98's password:
Linux raspberrypi 3.18.11-v7+ #781 SMP PREEMPT Tue Apr 21 18:07:59 BST 2015 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 27 15:48:14 2015 from 157.179.134.140
pi@raspberrypi ~ $
  
```

รูปที่ 3.21 การรีโมตสำหรับใช้งาน Raspberry Pi

## 2) การได้รับคำร้องขอจากเซิร์ฟเวอร์ที่ติดตั้งโปรแกรม PRTG Network Monitor

ในส่วนของการได้รับคำร้องขอจากเซิร์ฟเวอร์ที่ทำหน้าที่สังเกตการณ์ เมื่อมีข้อผิดพลาดเกิดขึ้นโปรแกรม PRTG Network Monitor จะส่งคำร้องขอมายังโปรแกรมระบบแจ้งเตือนที่มีการรอคำร้องอยู่ผ่านโปรโตคอล http โดยส่งพารามิเตอร์มา 3 ตัว คือ หมายเลขเครื่องที่เกิดข้อผิดพลาด (From) ชื่อกลุ่มผู้รับปลายทางสำหรับส่งข้อความ (To) และข้อความแจ้งเตือนข้อผิดพลาดที่เกิดขึ้น (text) จะปรากฏหน้าต่างของโปรแกรมดังรูปที่ 3.22



```

pi@raspberrypi: ~
starting: GET /notification.aspx?from=13013&to=network_dep&text=(PRTG+Network
+Monitor+(PRTG))+Test+Ping+(Ping)+Down+(Request+timed+out+(ICMP+error+%23+11010)) HTTP/1.1
readHeaders()
header: Host: 127.0.0.1
header: Connection:keep-alive
got headers
URL = /notification.aspx?from=13013&to=network_dep&text=(PRTG+Network+Monitor
+(PRTG))+Test+Ping+(Ping)+Down+(Request+timed+out+(ICMP+error+%23+11010))
myURI Query = ?from=13013&to=network_dep&text=(PRTG+Network+Monitor+(PRTG))
+Test+Ping+(Ping)+Down+(Request+timed+out+(ICMP+error+%23+11010))
From = 13013
To = network_dep
Text To Send = (PRTG Network Monitor (PRTG)) Test Ping (Ping) Down (Request timed out (ICMP error # 11010))
  
```

รูปที่ 3.22 ได้รับคำร้องขอจากเซิร์ฟเวอร์ PRTG

## 3) การส่งข้อความผ่าน API Line

หลังจากได้รับคำร้องขอจากเซิร์ฟเวอร์แล้ว ระบบจะนำพารามิเตอร์ในส่วนของผู้รับปลายทางสำหรับส่งข้อความ (To) ไปค้นในไฟล์ Configuration แล้วเรียกฟังก์ชัน SendMessage โดยนำข้อความแจ้งเตือนข้อผิดพลาดที่เกิดขึ้นส่งไปยังผู้รับตามรายชื่อในกลุ่มนั้น จะปรากฏหน้าต่างของโปรแกรมดังรูปที่ 3.21 จะเห็นว่าการส่งพารามิเตอร์สำหรับระบบไลน์ที่ใช้ในการส่งข้อความประกอบด้วย

Message.Text: ข้อความสำหรับแจ้งเตือน ซึ่งประกอบด้วย วันที่ เวลา ข้อมูลที่แจ้งเตือน

Message.CreatedTime: เวลาที่ส่งข้อความ

Message.DeliveredTime: เวลาที่ได้รับข้อความ

Message.ID: หมายเลขข้อความที่ส่ง

Message.To: ไลน์ผู้รับ

Message.From: ไลน์ผู้ส่ง

Message.Type: ประเภทการทำงานของ Line API

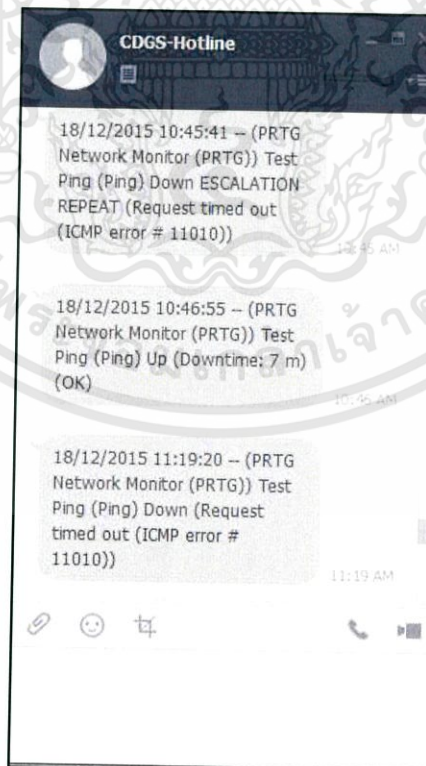
```

pi@raspberrypi: ~
Send Line Message...
[LineTransport] Attempting to flush the LineTransport.
[LineTransport] Attempting to flush with byte length of 172, accesskey of D2brbXAWDjGyK9twgp15.GelP1mOjUxPyZ
a+P7Om/Xq.nieowhmdjJOjVQmmERANc/BwVwUPW19a32s5KaVtw=, X-LS of 0 and URL of https://gd2u.line.naver.
jp.443/api/v4/TalkService.do
[LineTransport] Flushing and attempting to get a response.
[LineTransport] Got a response of size 81, reading from stream...
[LineTransport] Stream read complete, returning to the read buffer!
[PushClient] Acquired response from request, data length is 270
[PushClient] Reading response bytestream...
[PushClient] Finished reading, closing stream.
[OpHandler] Retrieved operation list of 2
[OpHandler] Dispatching operations with oprev=6171
[PushClient] Calling an API function using a byte array of 25 and a URL https://gd2u.line.naver.jp:443/P4
Operation received.
>>Type: SEND_MESSAGE
>>Oprev: 6171
>>Param1: 0
>>Param2:
>>Param3:
>>Message:
>>>Text: 18/12/2015 11:19:42 -- (PRTG Network Monitor (PRTG)) Test Ping (Ping) Down (Request timed out (ICMP
error # 11010))
>>>>CreatedTime: 1450412382313
>>>>DeliveredTime: 0
>>>>ID: 3621476645407
>>>>To: ue87a9b52fad6a8cfa4b4a905598a4ceb
>>>>From: ua848a9d4ca09b97a58cc0fe9ee8315c5
>>>>Type: NONE

```

### รูปที่ 3.23 ส่งข้อความผ่าน API Line

หลังจากโปรแกรมระบบแจ้งเตือนประมวลผลเสร็จเรียบร้อยแล้ว ข้อความดังกล่าวจะถูกส่งไปยังไลน์ของผู้รับ ดังรูปที่ 3.24



### รูปที่ 3.24 ข้อความทางไลน์ของผู้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลลัพธ์ของระบบ

ระบบแจ้งเตือนทางแอปพลิเคชันไลน์สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์ เป็นโปรแกรมที่ทำหน้าที่ส่งข้อความแจ้งเตือนถึงสาเหตุของปัญหาที่เกิดขึ้นกับทรัพยากรระบบ โดยการส่งข้อความผ่านแอปพลิเคชันไลน์ ในส่วนของการแจ้งเตือนนั้นระบบสามารถทำงานคู่กับโปรแกรมที่ทำหน้าที่ควบคุมดูแลทรัพยากรการทำงานต่างๆ ของเครื่องผ่านระบบเครือข่ายเน็ตเวิร์คได้ เช่น โปรแกรม PRTG Network Monitor เมื่อทรัพยากรการทำงานของเครื่องเซิร์ฟเวอร์มีข้อผิดพลาดเกิดขึ้น โปรแกรม PRTG Network Monitor จะส่งคำร้องขอผ่านโปรโตคอลเอชทีทีพีในรูปแบบของ URL ที่ระบุพารามิเตอร์ประกอบด้วยหมายเลขเครื่องเซิร์ฟเวอร์ กลุ่มรายชื่อที่ต้องการส่งข้อความ ข้อความสาเหตุของข้อผิดพลาดที่เกิดขึ้น มายังระบบแจ้งเตือนฯ ที่มีการรันโปรแกรมสำหรับรอคำสั่งร้องขอไว้ จากนั้นระบบแจ้งเตือนฯ ก็จะทำการประมวลผลและส่งข้อความไปยังผู้รับตามข้อมูลกลุ่มรายชื่อที่ต้องการส่ง หากผู้รับไม่ได้อ่านข้อความที่ได้ภายในระยะเวลาที่กำหนด ระบบก็จะส่งข้อความไปยังบุคคลถัดไปที่อยู่ในกลุ่มรายชื่อ ระบบแจ้งเตือนฯ จะสิ้นสุดการทำงานในหนึ่งกระบวนการก็ต่อเมื่อมีผู้รับหนึ่งคนอ่านข้อความหรือมีการส่งข้อความไปยังบุคคลสุดท้ายแล้วไม่ได้มีการอ่านข้อความ โดยหลังจากเสร็จกระบวนการส่งข้อความในแต่ละครั้งแล้วระบบจะบันทึกสถานะการทำงานลงล็อกไฟล์

#### 4.1 ขั้นตอนการทดสอบระบบ

##### 4.1.1 เปิดการใช้งานระบบแจ้งเตือน Social Alert

ระบบมีการทำงานบน Raspberry Pi จะต้องทำการโอนแฟ้มข้อมูลระบบขึ้นไปติดตั้งผ่านโปรโตคอล FTP ด้วยโปรแกรม Filezilla ดังนั้นการที่จะเปิดโปรแกรมจะต้องทำการ Remote เพื่อเข้าไปใช้งาน Terminal บน Raspberry Pi (ระบบปฏิบัติการลินุกซ์) และสั่งรันโปรแกรมให้ทำงานโดยใช้คำสั่ง `sudo mono` ตามด้วยชื่อไฟล์ที่มีนามสกุลเป็น `.exe` (แสดงรูปที่ 4.1)

```
pi@raspberrypi ~ $ sudo mono SMSServercmd.exe
```

รูปที่ 4.1 คำสั่งที่ใช้รันโปรแกรม

โปรแกรมจะเริ่มประมวลผลตามลำดับโดยมีลำดับการทำงานดังนี้

ขั้นตอนที่ 1 โปรแกรมจะทำการค้นหาอุปกรณ์โทรศัพท์ที่เชื่อมต่อกับบอร์ด Raspberry Pi ผ่านทาง USB เพื่อแชร์อินเทอร์เน็ตจากอุปกรณ์โทรศัพท์ ผลการเชื่อมต่ออุปกรณ์ (แสดงรูปที่ 4.2)

```
Run from linux platform
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
CB5120K6ZG      device
devices list was returned
```

รูปที่ 4.2 ตรวจสอบอุปกรณ์ที่เชื่อมต่อ

หมายเหตุ พบอุปกรณ์ที่เชื่อมต่อ Serial no. CB5120K6ZG

หากไม่ได้เชื่อมต่ออุปกรณ์โทรศัพท์มือถือเข้ากับ Raspberry Pi โปรแกรมจะทำการแสดงข้อความว่าไม่พบการเชื่อมต่ออุปกรณ์ (แสดงรูปที่ 4.3)

```
Run from linux platform
Error - No Devices Connected
Please 'Enter' if you connected devices.
```

รูปที่ 4.3 ผลลัพธ์เมื่อไม่พบการเชื่อมต่ออุปกรณ์

ขั้นตอนที่ 2 ระบบจะทำการเชื่อมต่อกับระบบไลน์เอพีไอ อันดับแรกจะล็อกอินเข้าสู่ระบบการใช้งานแบบทั่วไป เมื่อทำการล็อกอินเรียบร้อยแล้ว ต่อไปจะเป็นการเตรียมข้อมูลรายชื่อผู้ติดต่อบนระบบไลน์ โดยจะทำการดึงข้อมูลมาเก็บไว้บนตัวแปรในระบบเพื่อใช้เป็นข้อมูลในการส่งข้อความแจ้งเตือน ซึ่งข้อมูลผู้ติดต่อนี้คือรายชื่อในกลุ่มของผู้ที่ต้องการส่งข้อความ เมื่อทำการจัดเตรียมข้อมูลเรียบร้อยแล้วจะแสดงผลการทำงาน (แสดงรูปที่ 4.4)

```

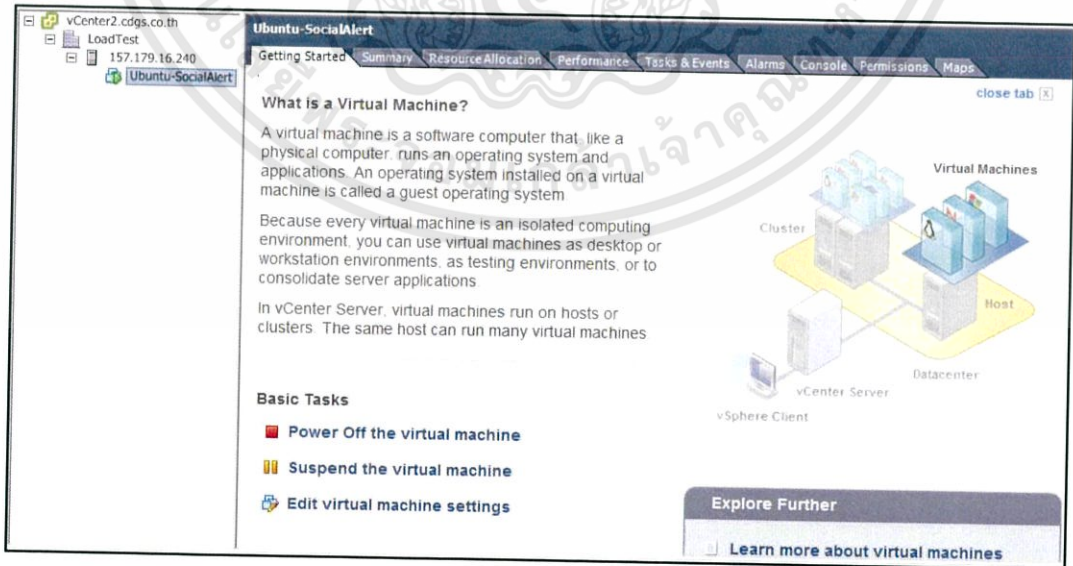
[LineTransport] Refreshing writestream!
[LineTransport] Attempting to flush the LineTransport.
[LineTransport] Attempting to flush with byte length of 386, accesskey of , X-LS
of 0 and URL of https://gd2u.line.naver.jp:443/api/v4/TalkService.do
[LineTransport] Flushing and attempting to get a response.
[LineTransport] Got a response of size 148, reading from stream...
[LineTransport] Stream read complete, returning to the read buffer!
[OpHandler] Starting the listening thread.
[LineTransport] Attempting to flush the LineTransport.
[LineTransport] Attempting to flush with byte length of 21, accesskey of DY9ZfIP
17mq6saHQZt45.QeIPImOjUxPyZa+P7OmvXq.osydXy+hTsPcCOVtMCAE+hr7TQ2XjB872A23PU48k/U
, X-LS of 0 and URL of https://gd2u.line.naver.jp:443/api/v4/TalkService.do
[LineTransport] Flushing and attempting to get a response.
[PushClient] Acquired response from request, data length is 26
[PushClient] Reading response bytestream...
[PushClient] Finished reading, closing stream.
[OpHandler] Beginning listener loop.
[OpHandler] Fetching operations with oprev=5493
[PushClient] Calling an API function using a byte array of 25 and a URL of https
://gd2u.line.naver.jp:443/P4
[PushClient] Sending request...
[LineTransport] Got a response of size 93, reading from stream...
[LineTransport] Stream read complete, returning to the read buffer!
[LineTransport] Attempting to flush the LineTransport.
[LineTransport] Attempting to flush with byte length of 84, accesskey of DY9ZfIP
17mq6saHQZt45.QeIPImOjUxPyZa+P7OmvXq.osydXy+hTsPcCOVtMCAE+hr7TQ2XjB872A23PU48k/U
, X-LS of 0 and URL of https://gd2u.line.naver.jp:443/api/v4/TalkService.do
[LineTransport] Flushing and attempting to get a response.
[LineTransport] Got a response of size 4837, reading from stream...
[LineTransport] Stream read complete, returning to the read buffer!
Line Login Processing... [Complete]
Thread Alert Process... [Start]

```

รูปที่ 4.4 การจัดเตรียมข้อมูลสำหรับการส่งข้อความผ่านไลน์เรียบร้อย

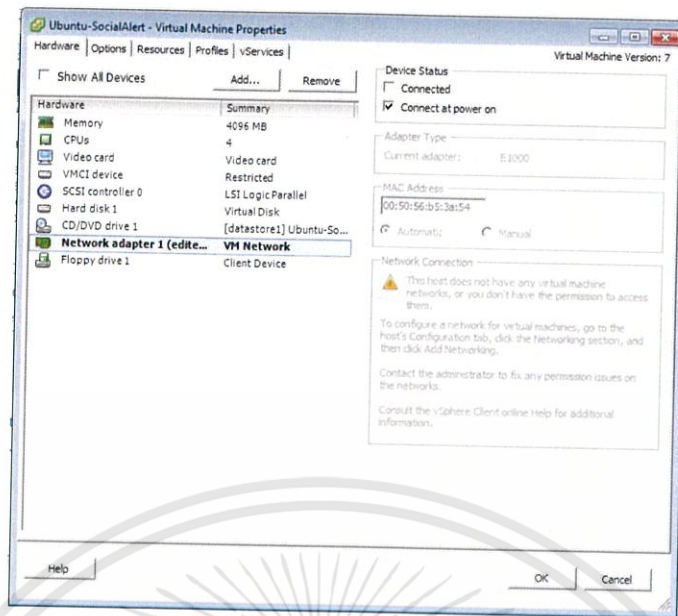
### 4.1.2 ทดสอบปิดการเชื่อมต่อเครือข่ายเน็ตเวิร์ค

ขั้นตอนที่ 1 ทำการทดสอบเซิร์ฟเวอร์ (Ubuntu-SocialAlert) (แสดงรูปที่ 4.5) โดย การปิดการเชื่อมต่อเครือข่ายเน็ตเวิร์ค (แสดงรูปที่ 4.6) จะสังเกตได้ว่าภายในประมาณ 1 นาที โปรแกรม PRTG Network Monitor จะเปลี่ยนสถานะจาก Up (แสดงรูปที่ 4.7) ไปเป็น Warning (แสดงรูปที่ 4.8) เนื่องจากไม่สามารถส่งการร้องขอในการตรวจสอบการเชื่อมต่อได้

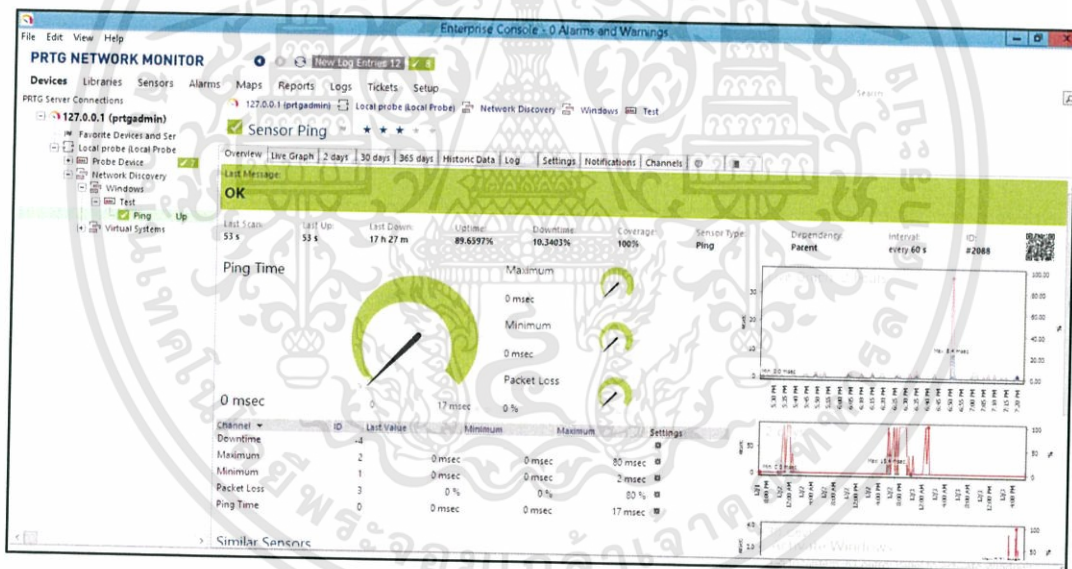


รูปที่ 4.5 เซิร์ฟเวอร์สำหรับทดสอบ (Ubuntu-SocialAlert)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

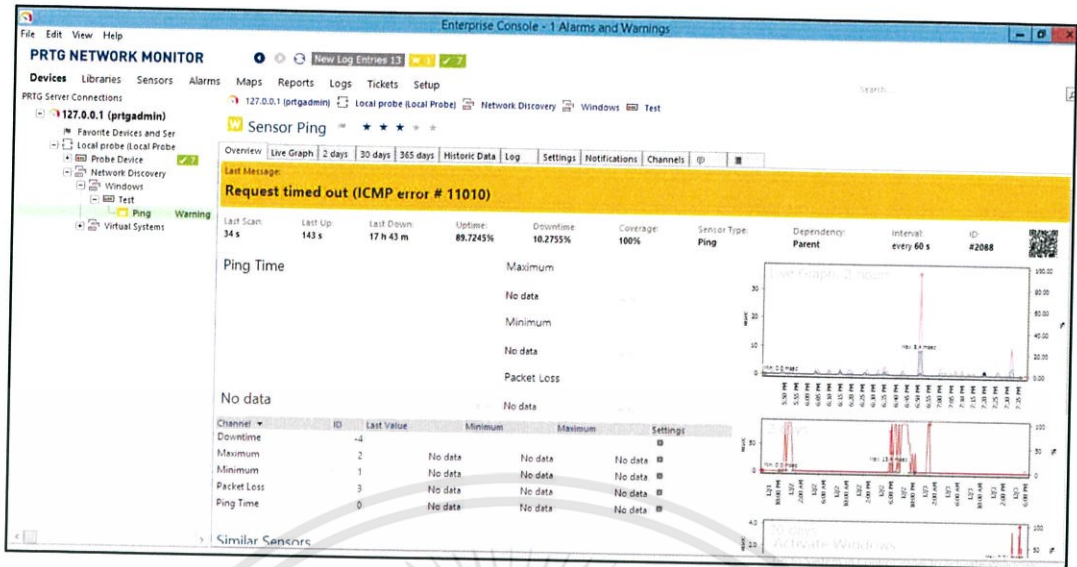


รูปที่ 4.6 ปิดการทำงานของ Network adapter



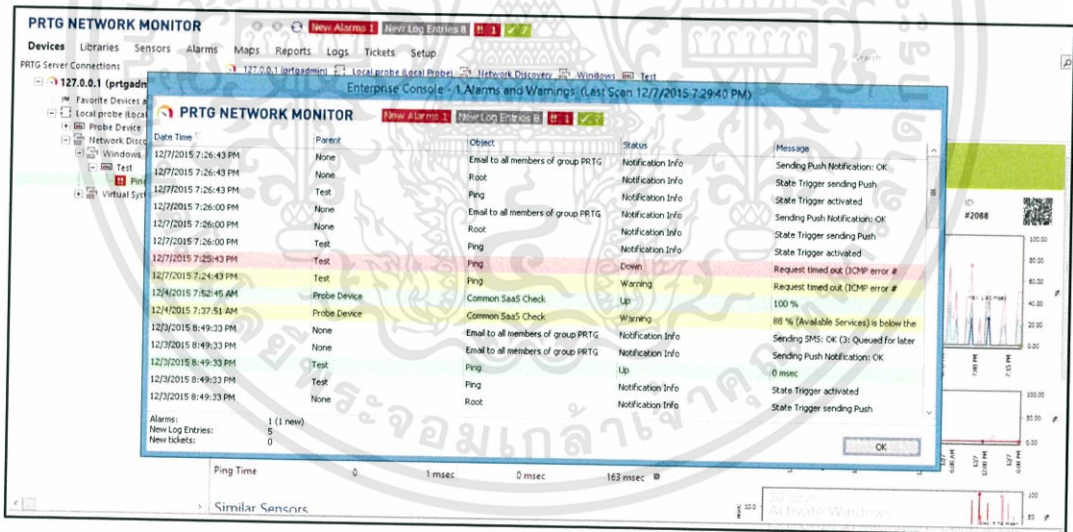
รูปที่ 4.7 สถานะการทดสอบอินเทอร์เน็ตเป็น Up

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



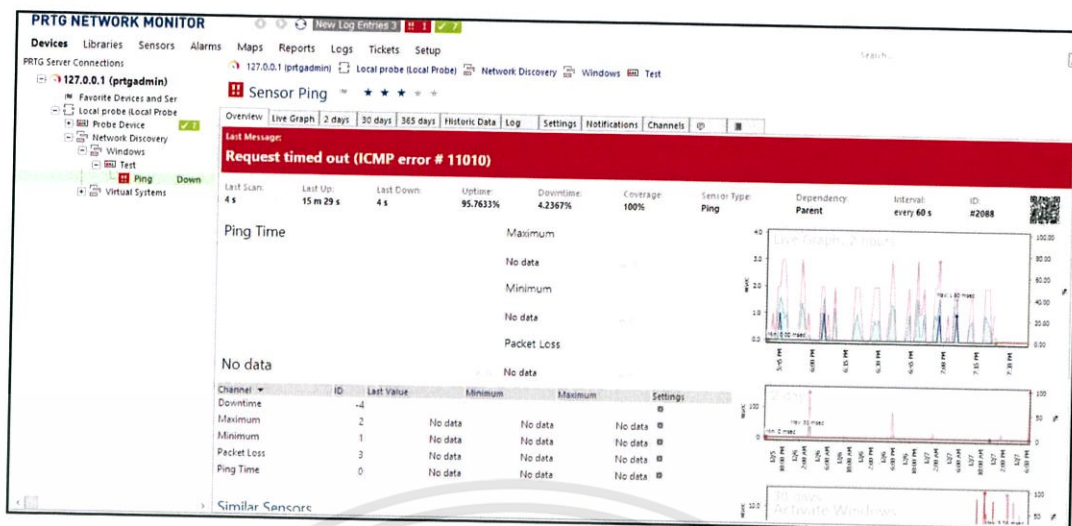
รูปที่ 4.8 สถานะการทดสอบอินเทอร์เน็ตเป็น Warning

จากนั้นอีกประมาณ 1 นาทีโปรแกรมจะแสดงหน้าต่างป๊อปอัปรายการข้อผิดพลาดที่เกิดขึ้น (แสดงรูปที่ 4.9) มีการสร้างตัวสำหรับแจ้งเตือน (New Alarm) ที่จะทำการร้องขอไปยังระบบแจ้งเตือน เพื่อใช้สำหรับส่งข้อความแจ้งเตือนไปยังผู้รับต่อไป จากนั้นโปรแกรมจะแสดงสถานะเป็น Down (แสดงรูปที่ 4.10)



รูปที่ 4.9 หน้าต่างย่อยแสดงรายการข้อผิดพลาดที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 สถานะการทดสอบอินเทอร์เน็ตเป็น Down

หมายเหตุ รายละเอียดขั้นตอนวิธีการทดสอบอยู่ภาคผนวก จ

#### 4.1.3 ตรวจสอบผลลัพธ์ที่เกิดขึ้น

ขั้นตอนที่ 1 เป็นการได้รับคำร้องขอจาก URL ที่ได้มีการดักจับไว้ ซึ่งถูกส่งคำร้องมาจากโปรแกรม PRTG Network Monitor ซึ่งจะทำให้การส่งข้อมูลที่สำคัญต่อกระบวนการการทำงานผ่านพารามิเตอร์มาทาง URL (แสดงรูปที่ 4.11)

```

pi@raspberrypi: ~
starting: GET /notification.aspx?from=13013&to=network_dep&text=(PRTG+Network+Monitor+(PRTG))+Test+Ping+(Ping)+Down+(Request+timed+out+(ICMP+error+%23+11010)) HTTP/1.1
readHeaders()
header: Host 127.0.0.1
header: Connection keep-alive
got headers
URL = /notification.aspx?from=13013&to=network_dep&text=(PRTG+Network+Monitor+(PRTG))+Test+Ping+(Ping)+Down+(Request+timed+out+(ICMP+error+%23+11010))
myURI Query = ?from=13013&to=network_dep&text=(PRTG+Network+Monitor+(PRTG))+Test+Ping+(Ping)+Down+(Request+timed+out+(ICMP+error+%23+11010))
From = 13013
To = network_dep
Text To Send = (PRTG Network Monitor (PRTG)) Test Ping (Ping) Down (Request timed out (ICMP error # 11010))
  
```

รูปที่ 4.11 พารามิเตอร์ที่ได้รับจาก URL

หมายเหตุ

from: ชื่อหรือหน่วยงานผู้ส่ง

to: ชื่อหรือหน่วยงานผู้รับ

text: ข้อความสำหรับการแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

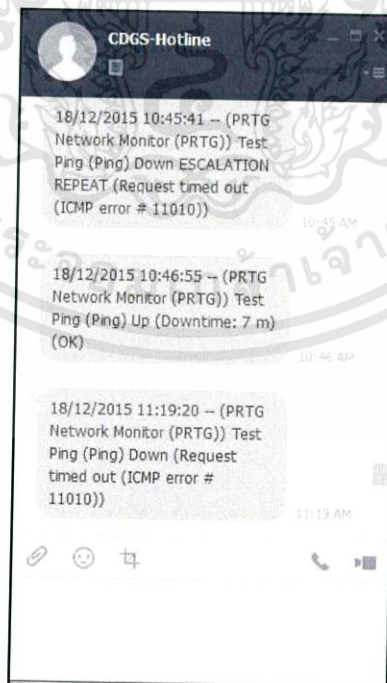
ขั้นตอนที่ 2 เมื่อมีการร้องขอให้ส่งข้อความมาแล้ว เซรตหลักจะทำการเพิ่มข้อมูลลงคิว และเซรต Alert Process ที่ทำงานอยู่ร่วมกันจะรับหน้าที่ต่อ โดยการนำข้อมูลในคิวไปทำการส่งข้อความ เมื่อเสร็จสิ้นกระบวนการจะทำการลบข้อมูลออกจากคิว ซึ่งโพรเซสการทำงานนั้นคือการส่งข้อความเป็นหลัก (แสดงรูปที่ 4.12) โดยจะทำการส่งข้อความไปยังผู้รับผ่านไลน์ (แสดงรูปที่ 4.13) ที่สามารถรับทราบได้ว่าผู้รับมีการอ่านข้อความหรือไม่ และใช้ระบบตอบกลับจากไลน์ในการยืนยันการได้รับข้อความ ซึ่งโพรเซสการส่งข้อความมีดังนี้

```

...Send Line Message
[LineTransport] Attempting to flush the LineTransport.
[LineTransport] Attempting to flush with byte length of 172, acceskey of DZbrbXAWDjGYK91wgp15.QeIP1mOjUxPyZa+P7OmvXq.niecwht
of 0 and URL of https://gd2u.line.naver.jp:443/api/v4/TalkService.do
[LineTransport] Flushing and attempting to get a response.
[LineTransport] Got a response of size 81, reading from stream...
[LineTransport] Stream read complete, returning to the read buffer!
[PushClient] Acquired response from request, data length is 270
[PushClient] Reading response byrestream...
[PushClient] Finished reading, closing stream.
[OpHandler] Retrieved operation list of 2
[OpHandler] Dispatching operations.
[OpHandler] Fetching operations with oprev=6171
[PushClient] Calling an API function using a byte array of 25 and a URL of https://gd2u.line.naver.jp:443/P4
[PushClient] Sending request...
Wait FallBack Processing
Operation recieved.
>>Type: SEND MESSAGE
>>Oprev: 6171
>>Param1: 0
>>Param2:
>>Param3:
>>Message:
>>>Text: 18/12/2015 11:19:42 -- (PRTG Network Monitor (PRTG)) Test Ping (Ping) Down (Request timed out (ICMP error # 11010))
>>>CreatedTime: 1450412382313
>>>DeliveredTime: 0
>>>ID: 3621476645407
>>>To: ue87a9b52fad6a8cfa4b4a905598a4ceb
>>>From: ua848a9d4ca09b97a58cc0fe9ee8315c5
>>>Type: NONE

```

รูปที่ 4.12 กระบวนการส่งข้อความผ่านไลน์



รูปที่ 4.13 ข้อความที่ถูกส่งไปยังไลน์ผู้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเปิดการใช้งานพื้นฐานไลน์เอพีโอแล้ว หากไม่มีการส่งข้อความเป็นเวลานาน ระบบจะใช้ฟังก์ชันเรียกตัวเองใหม่อีกครั้ง เพื่อยืนยันว่าระบบไลน์อยู่ในสถานะพร้อมใช้งาน (แสดงรูปที่ 4.14)

```
[PushClient] A long polling request returned status code 410, resending request...
[PushClient] Calling an API function using a byte array of 25 and a URL of https://gd2u.line.naver.jp:443/P4
[PushClient] Sending request...
[PushClient] A long polling request returned status code 410, resending request...
[PushClient] Calling an API function using a byte array of 25 and a URL of https://gd2u.line.naver.jp:443/P4
[PushClient] Sending request...
[PushClient] A long polling request returned status code 410, resending request...
[PushClient] Calling an API function using a byte array of 25 and a URL of https://gd2u.line.naver.jp:443/P4
[PushClient] Sending request...
[PushClient] A long polling request returned status code 410, resending request...
[PushClient] Calling an API function using a byte array of 25 and a URL of https://gd2u.line.naver.jp:443/P4
```

รูปที่ 4.14 ระบบไลน์อยู่ในสถานะพร้อมใช้งาน

ขั้นตอนที่ 3 หลังจากผู้รับได้รับข้อความเรียบร้อยแล้ว โปรแกรมจะแสดงข้อความบอกสถานะการทำงานปัจจุบัน ยกตัวอย่างเช่น การส่งข้อความไปหาผู้รับคนแรก แล้วมีการอ่านข้อความภายในเวลาที่กำหนด (แสดงรูปที่ 4.15)

```
Fallback Thread finished. at escalation : 1
Status Fallback Thread : success
0 async Fallback processes executing
Received Response From Technical Fallback!!
Status Process : success
End Finite State Machine
Queue is empty
```

รูปที่ 4.15 สถานะการทำงาน success

และกรณีไม่มีการอ่านข้อความจากผู้รับ โปรแกรมจะแสดงข้อความบอกสถานะการทำงานปัจจุบัน (แสดงรูปที่ 4.16)

```
Process 0 Fallback timeout
0 async Fallback processes executing
No Response From Technical Fallback!!
Status Process : fail
End Finite State Machine
Queue is empty
```

รูปที่ 4.16 สถานะการทำงาน fail

ขั้นตอนสุดท้ายก่อนที่จะเริ่มกระบวนการส่งข้อความชุดต่อไปโปรแกรมจะทำการบันทึกสถานะการส่งข้อความทั้งหมดที่ได้ดำเนินการส่งไปเพื่อเก็บไว้เป็นข้อมูลในการอ้างอิง ซึ่งข้อมูลดังกล่าวจะถูกสร้างเป็นไฟล์ใช้แบบวันต่อวันจุดประสงค์เพื่อที่จะจำแนกวันเวลาที่ดำเนินการส่งให้เป็นระเบียบ

Send to Group : [network_dep]			
Message : (PRIG Network Monitor (PRIG)) Test Ping (Ping) Down (Request timed out (ICMP error # 11010))			
Date/Time	To		Status
-----	---		-----
18/12/2015 10:46:23	[LINE] :Jerry		fail
18/12/2015 10:46:23	[LINE] :Tony		fail
-----			
Send to Group : [network_dep]			
Message : (PRIG Network Monitor (PRIG)) Test Ping (Ping) Down (Request timed out (ICMP error # 11010))			
Date/Time	To		Status
-----	---		-----
18/12/2015 10:46:58	[LINE] :Jerry		success

รูปที่ 4.17 ข้อมูลที่บันทึกบนล็อกไฟล์

## 4.2 ข้อเปรียบเทียบระหว่างการแจ้งเตือนผ่านข้อความ SMS กับ LINE

ตารางที่ 4.1 ข้อเปรียบเทียบระหว่างการแจ้งเตือนผ่านข้อความ SMS กับ LINE

SMS	LINE
<ul style="list-style-type: none"> <li>มีค่าใช้จ่ายที่สูงกว่าเมื่อเทียบกับการส่งข้อความแจ้งเตือนในแต่ละครั้ง</li> </ul>	<ul style="list-style-type: none"> <li>มีค่าใช้จ่ายที่ต่ำกว่าเนื่องจากชื่อเป็นแพ็คเกจอินเทอร์เน็ตแบบเหมาจ่าย</li> </ul>
<ul style="list-style-type: none"> <li>ทิศทางการสื่อสารจะมีลักษณะเป็นทางเดียว</li> </ul>	<ul style="list-style-type: none"> <li>มีลักษณะเป็นสองทิศทางเนื่องจากแอปพลิเคชันไลน์มีฟังก์ชันตรวจสอบการอ่านข้อความ</li> </ul>
<ul style="list-style-type: none"> <li>ไม่ค่อยเป็นที่นิยม มีการใช้งานเป็นบางครั้ง</li> </ul>	<ul style="list-style-type: none"> <li>เข้าถึงผู้ใช้ได้มากกว่าเนื่องจากเป็นแอปพลิเคชันที่เป็นที่นิยม</li> </ul>

## บทที่ 5

# สรุปผลการดำเนินงาน

### 5.1 ข้อสรุปผลการดำเนินงาน

ในการพัฒนาระบบส่งข้อความแจ้งเตือนผ่านทางโซเชียลมีเดียบนแอปพลิเคชันไลน์ที่ทำงานบนบอร์ด Raspberry Pi ที่มีการเชื่อมต่ออินเทอร์เน็ตโดยการแชร์ผ่านโทรศัพท์มือถือ ซึ่งทำให้สามารถส่งข้อความจากเซิร์ฟเวอร์เมื่อเกิดปัญหาแจ้งเตือนไปยังผู้รับผิดชอบเพื่อให้ดำเนินการแก้ไขได้อย่างเข้าถึงและรวดเร็ว ช่วยลดปัญหาการไม่ได้รับข้อความแจ้งเตือนเมื่อเกิดปัญหาขึ้น ซึ่งโดยส่วนใหญ่การแจ้งเตือนผ่านทางช่องทางเดิม เช่น e-mail หรือ sms นั้นมักจะมีปัญหา เช่น ไม่ได้รับข้อความที่ส่งมาเนื่องจากปัญหาทางผู้ให้บริการบ้าง หรือเวลาที่ผู้รับอ่านจะไม่สามารถยืนยันได้ว่าผู้รับได้ทำการอ่านข้อความแจ้งเตือนนั้นแล้ว ทางผู้พัฒนาจึงจัดทำระบบให้เพิ่มความสามารถในการส่งข้อความสำหรับแจ้งเตือนให้มีหลายช่องทางมากขึ้น และเพื่อจุดประสงค์หลักในการให้ผู้อ่านได้รับทราบถึงข้อความที่ส่งมา จึงได้นำเอาโซเชียลมีเดียแอปพลิเคชันที่เป็นที่นิยมใช้และมีการใช้งานที่เข้าถึงกว่า e-mail หรือ sms มาประยุกต์ใช้ โดยระบบจะทำการแจ้งเตือนไปหาผู้รับโดยการส่งข้อความผ่านทางไลน์ และมีการกำหนดช่วงเวลาในการอ่านของผู้รับ เมื่อมีการอ่านข้อความเกิดขึ้น แสดงว่าผู้รับได้ทราบถึงข้อความแจ้งเตือนนั้นแล้ว ซึ่งตรงตามจุดประสงค์ของการส่งข้อความก็จะสิ้นสุดการส่งข้อความถึงรายชื่อบุคคลหรือหน่วยงานถัดไป เมื่อมีการอ่านข้อความสถานะการทำงานของ การส่งข้อความนั้นจะแสดงผลเป็นสำเร็จ หรือถ้าหากไม่มีการอ่านข้อความสถานะการทำงานจะแสดงผลเป็นล้มเหลว ซึ่งสามารถเห็นข้อความได้จากหน้าต่างคอนโซล หรือสามารถเปิดดูล็อกไฟล์เพื่อตรวจสอบสถานะของการส่งข้อความได้ และผู้ใช้งานสามารถปรับเปลี่ยนข้อมูลรายชื่อกลุ่มผู้ที่ต้องการส่งข้อความผ่านไฟล์ Configuration ได้โดยไม่ต้องดำเนินการแก้ไขที่ตัวโปรแกรม

## 5.2 ข้อเสนอแนะ

ทางผู้พัฒนาเห็นว่าบางส่วนของโปรแกรมควรจะมีการพัฒนา หรือปรับเปลี่ยนการทำงานให้เหมาะสม และมีช่องทางการส่งข้อความแจ้งเตือนมากขึ้น ดังข้อเสนอแนะต่อไปนี้

- 1) การเพิ่มแพลตฟอร์มให้สามารถรันการทำงานบนระบบปฏิบัติการอื่นได้เพิ่มมากขึ้น ซึ่งปัจจุบันสามารถทำงานได้เพียง 2 ระบบปฏิบัติการคือ Windows, Linux
- 2) การเพิ่มฟังก์ชันการส่งข้อความไปยังช่องทางอื่นๆ เช่น Twitter, Facebook, Hi5 ฯลฯ
- 3) การเปลี่ยนอุปกรณ์ให้มีต้นทุนต่ำแต่สามารถใช้งานได้ปกติ เพื่อลดค่าใช้จ่ายในเรื่องของอุปกรณ์
- 4) การพัฒนาส่วนที่แสดงผลให้ชัดเจนและเข้าใจง่ายมากขึ้น เช่น การทำส่วนที่แสดงผลให้บ่งบอกสัดส่วนการทำงานของแต่ละส่วนให้ชัดเจนออกไป
- 5) การทำงานบนระบบปฏิบัติการ Linux มักจะพบปัญหาที่ไม่ทราบถึงสาเหตุได้บ่อยกว่าการทำงานบนระบบปฏิบัติการ Windows จึงต้องทำการพัฒนาและหาสาเหตุของปัญหาดังกล่าว

## 5.3 ข้อจำกัด

- 1) ระบบแจ้งเตือนฯ ไม่สามารถทำการส่งข้อความหลายๆ ข้อความในระยะเวลาอันสั้นได้
- 2) ชื่อผู้ติดต่อสำหรับส่งผ่านไลน์ต้องเป็นภาษาอังกฤษเท่านั้น หากชื่อเป็นภาษาไทย มีอักขระพิเศษ หรือรูปภาพ จะไม่สามารถใช้เป็นข้อมูลเพื่อทำการส่งข้อความได้
- 3) ระบบแจ้งเตือนฯ ยังไม่สามารถดำเนินการส่งรูปภาพ ไฟล์ หรือวีดีโอได้
- 4) ระบบแจ้งเตือนฯ ยังไม่มีฟังก์ชันเริ่มต้นการทำงานแบบอัตโนมัติ
- 5) ระบบแจ้งเตือนฯ ยังไม่สามารถตรวจสอบการอ่านข้อความที่ส่งไปยังผู้รับพร้อมกันหลายคนได้

## เอกสารอ้างอิง

- [1] ศักรินทร์ ต้นสุพงษ์. 2557. ปัจจัยที่ส่งผลต่อการยอมรับแอปพลิเคชันไลน์. วิทยาศาสตร์  
มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการจัดการ มหาวิทยาลัยกรุงเทพ
- [2] 3G 4G 5G. [ออนไลน์]. สืบค้นเมื่อ 16 เมษายน 2559. แหล่งที่มา :  
[www.ictforall.org/Articles/3G\\_%204G\\_5G.pdf](http://www.ictforall.org/Articles/3G_%204G_5G.pdf)
- [3] 1G 2G 3G 4G คืออะไร. [ออนไลน์]. สืบค้นเมื่อ 2 พฤศจิกายน 2558. แหล่งที่มา :  
<http://knightnew.exteen.com/20100728/1g-2g-3g-4g>.
- [4] เริ่มต้นกับ Raspberry Pi [ออนไลน์]. สืบค้นเมื่อ 2 พฤศจิกายน 2558. แหล่งที่มา :  
<http://www.sathittham.com/raspberry-pi>.
- [5] LineSharp. [ออนไลน์]. สืบค้นเมื่อ 2 พฤศจิกายน 2558. แหล่งที่มา :  
<http://www.kannel.org/overview.shtml>.
- [6] Raspberry Pi. [ออนไลน์]. สืบค้นเมื่อ 2 พฤศจิกายน 2558. แหล่งที่มา :  
[https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi).
- [7] PRTG Network Monitor. [ออนไลน์]. สืบค้นเมื่อ 29 พฤษภาคม 2559. แหล่งที่มา :  
<http://software.thaiware.com/7949-PRTG-Network-Monitor.html>



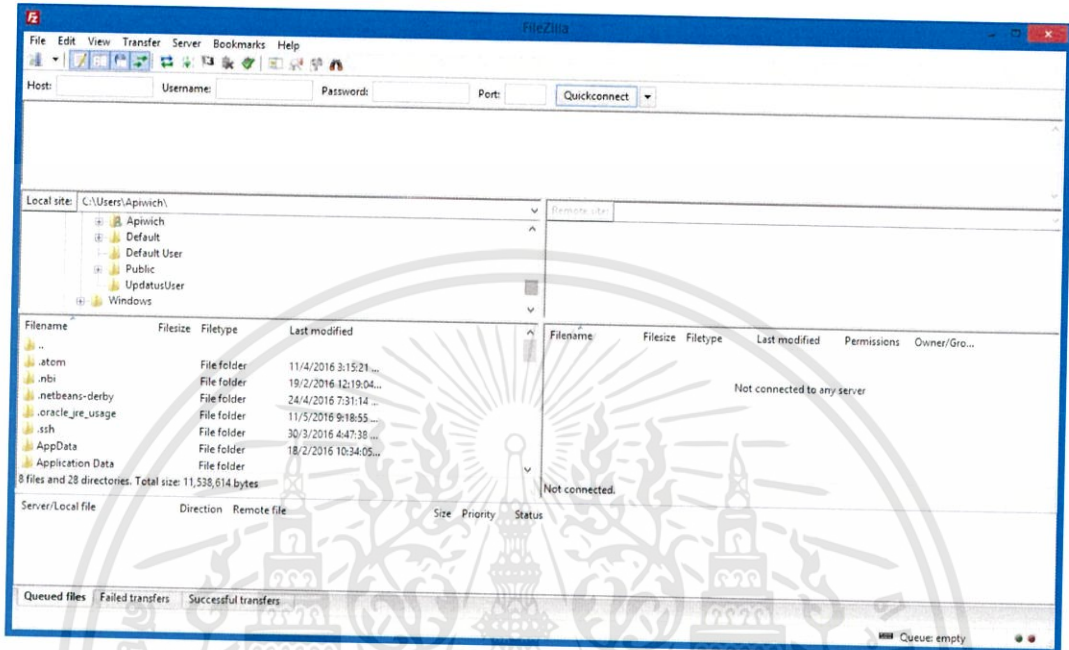
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก.

## การติดตั้งโปรแกรมระบบแจ้งเตือนๆ สำหรับใช้งานบนอุปกรณ์ Raspberry Pi

## 1. เปิดโปรแกรม FileZilla



รูปที่ ก.1 หน้าแรกโปรแกรม FileZilla

## 2. ทำการเชื่อมต่อไปยัง Raspberry Pi โดยระบุ Host Username และ Password แล้วกด Quickconnect

Host : หมายเลข IP ที่อยู่ของ Raspberry Pi

Username : ชื่อผู้ใช้สำหรับเข้าใช้งาน Raspberry Pi

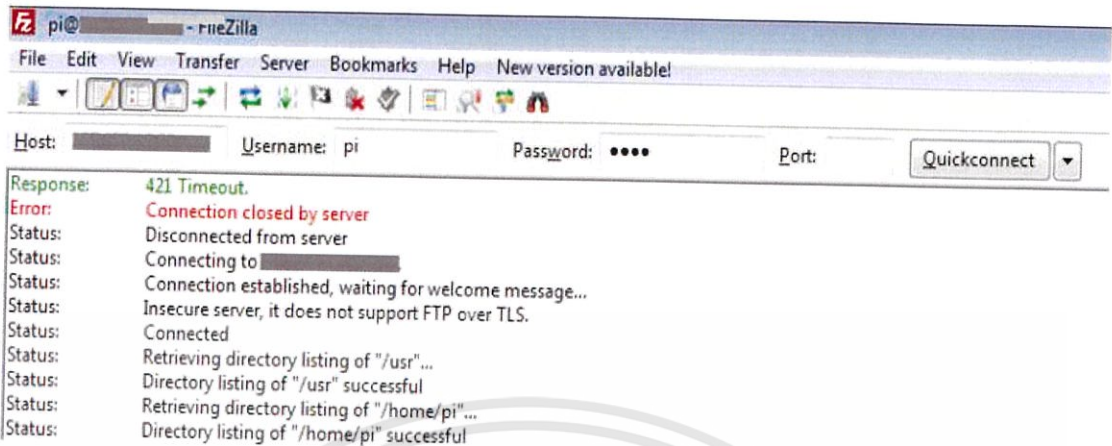
Password : รหัสสำหรับเข้าใช้งาน Raspberry Pi



รูปที่ ก.2 กรอกรายละเอียดสำหรับเชื่อมต่อ Raspberry Pi

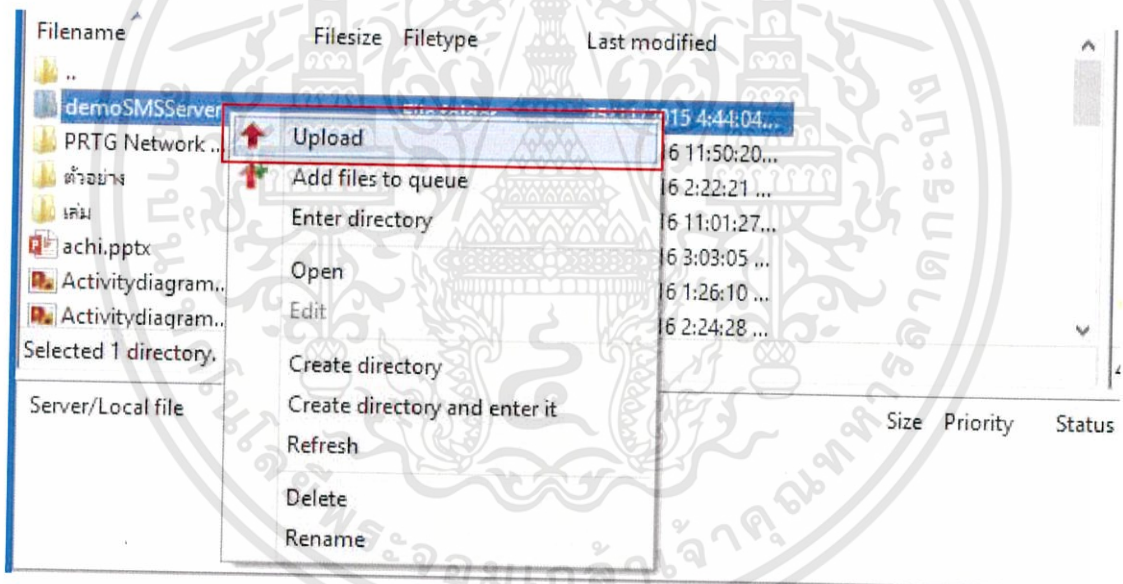
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หากเชื่อมต่อกับ Raspberry Pi สำเร็จ จะแสดงดังรูป ก.3



รูปที่ ก.3 แสดงหน้าต่างเชื่อมต่อสำเร็จ

4. ไปที่ไดเรกทอรีของโปรแกรมระบบแจ้งเตือนฯ คลิกขวาเลือก Upload



รูปที่ ก.4 การอัปโหลดไฟล์โปรแกรมระบบแจ้งเตือนฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เมื่ออัปโหลดไฟล์โปรแกรมสำเร็จ จะแสดงหน้าต่างดังรูปที่ ก.5

```

Host: ██████████ Username: pi Password: ██████████
Status: Directory listing of "/home/pi" successful
Response: 421 Timeout.
Error: Connection closed by server
Status: Connecting to ██████████...
Status: Connection established, waiting for welcome message...
Status: Insecure server, it does not support FTP over TLS.
Status: Connected
Status: Starting upload of D:\CDG\demoSMSServer
Status: File transfer successful, transferred 4,131,993 bytes in 1 second
Status: Retrieving directory listing of "/home/pi" ...
Status: Directory listing of "/home/pi" successful
  
```

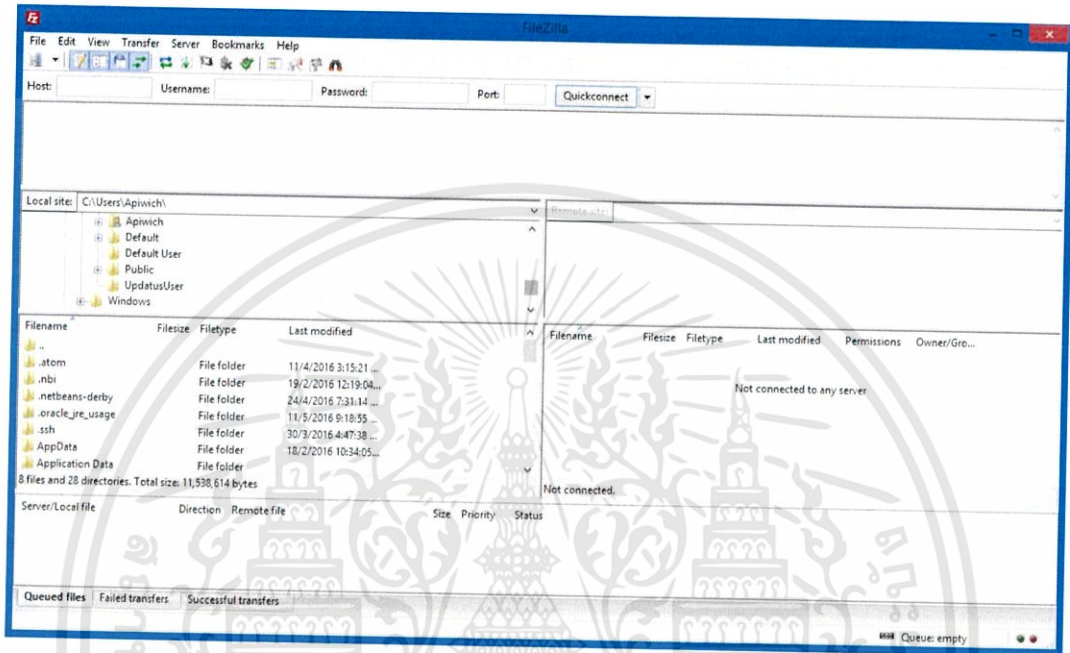
รูปที่ ก.5 หน้าต่างแสดงสถานะการอัปโหลด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.  
 การติดตั้ง Android Debug Bridge (adb)  
 บน Raspberry Pi สำหรับเชื่อมต่อกับอุปกรณ์สมาร์ทโฟน

1. เปิดโปรแกรม FileZilla



รูปที่ ข.1 หน้าแรกโปรแกรม FileZilla

2. ทำการเชื่อมต่อไปยัง Raspberry Pi โดยระบุ Host Username และ Password แล้วกด Quickconnect

Host : หมายเลข IP ที่อยู่ของ Raspberry Pi

Username : ชื่อผู้ใช้สำหรับเข้าใช้งาน Raspberry Pi

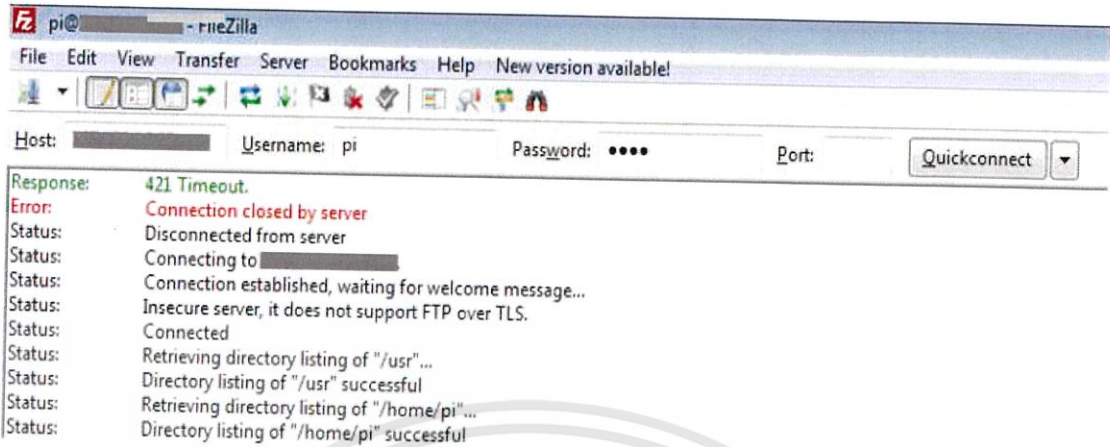
Password : รหัสสำหรับเข้าใช้งาน Raspberry Pi



รูปที่ ข.2 กรอกรายละเอียดสำหรับเชื่อมต่อ Raspberry Pi

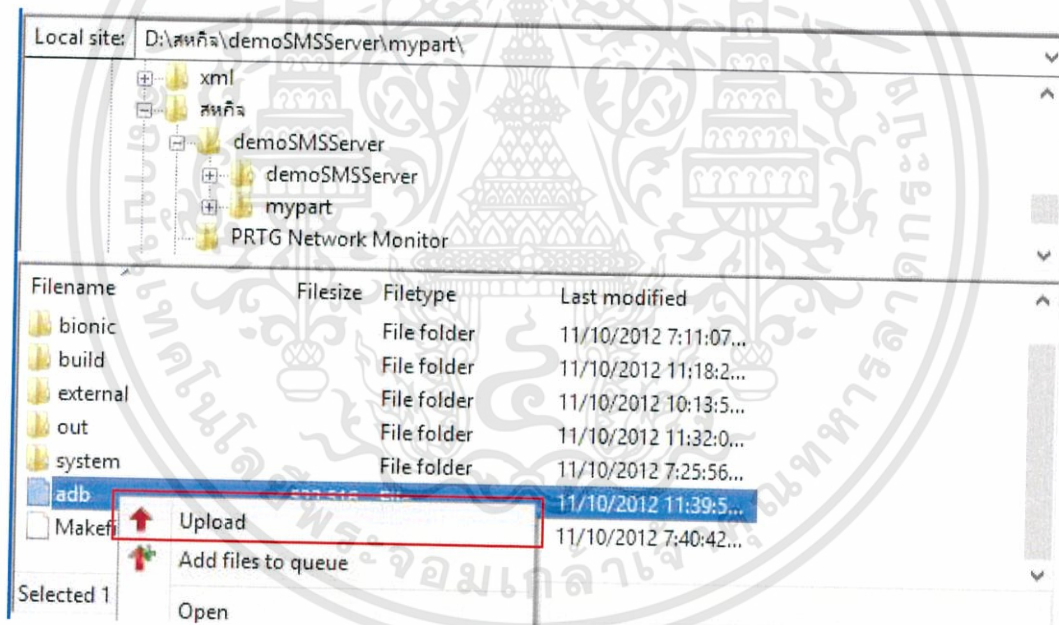
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หากเชื่อมต่อกับ Raspberry Pi สำเร็จ จะแสดงดังรูปที่ ข.3



รูปที่ ข.3 แสดงหน้าต่างเชื่อมต่อสำเร็จ

4. ไปที่ไดเรกทอรีของโปรแกรม adb คลิกขวาเลือก Upload



รูปที่ ข.4 การอัปโหลดไฟล์โปรแกรม adb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เมื่ออัปโหลดไฟล์โปรแกรมสำเร็จ จะแสดงหน้าต่างดังรูป ข.5

```

Host: [redacted] Username: pi Password: [redacted]
Status: Directory listing of "/home/pi" successful
Response: 421 Timeout.
Error: Connection closed by server
Status: Connecting to [redacted]...
Status: Connection established, waiting for welcome message...
Status: Insecure server, it does not support FTP over TLS.
Status: Connected
Status: Starting upload of D:\CDG\demoSMSServer
Status: File transfer successful, transferred 4,131,993 bytes in 1 second
Status: Retrieving directory listing of "/home/pi"...
Status: Directory listing of "/home/pi" successful

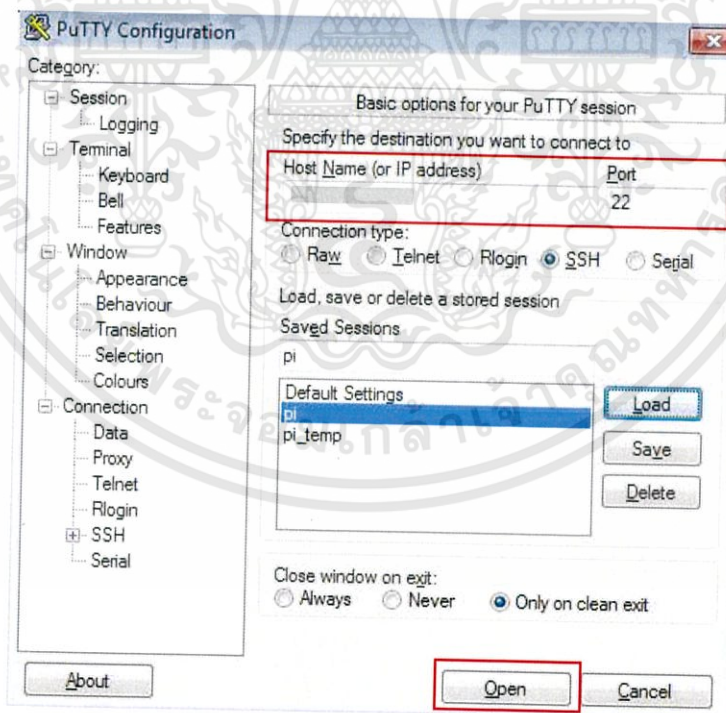
```

รูปที่ ข.5 หน้าต่างแสดงสถานการณ์อัปโหลด

6. เปิดโปรแกรม PuTTY เพื่อเรียกใช้ SSH Secure Shell บน Raspberry Pi สำหรับติดตั้ง adb โดยระบุ Host Name และ Port จากนั้นกด Open

Host Name (or IP address) : หมายเลข IP ที่อยู่ของ Raspberry Pi

Port : 22



รูปที่ ข.6 หน้าแรกโปรแกรม PuTTY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ล็อกอินเข้าใช้งาน Raspberry Pi โดยระบุ login as และ Password จากนั้นกด Enter

```

pi@raspberrypi: ~
login as: pi
pi@157.179.16.98's password:
Linux raspberrypi 3.18.11-v7+ #781 SMP PREEMPT Tue Apr 21 18:07:59 BST 2015 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov  2 13:43:34 2015 from 157.179.135.50
pi@raspberrypi ~ $

```

รูปที่ ข.7 หน้าแรก SSH Secure Shell บน Raspberry Pi

8. ใช้คำสั่ง `ls -l` เพื่อตรวจสอบว่าไฟล์ adb ได้ถูกอัปเดตขึ้นมาแล้ว (แสดงรูปที่ ข.8)

```

pi@raspberrypi ~ $ ls -l
total 6896
-rw----- 1 pi pi 4131993 Nov  2 13:48 adb
-rw----- 1 pi pi 2270720 Nov  2 11:33 AndroidLib.dll
-rwxr-xr-x 1 pi pi  5431 May  7 06:41 a.out
-rwxr-xr-x 1 pi pi  6401 Oct  5 15:40 blink
-rw-r--r-- 1 pi pi  727 Oct  5 15:39 blink.c
-rw----- 1 pi pi  198 Oct  5 16:30 blink.py
drwxr-xr-x 5 pi pi  4096 Oct 13 15:09 demoHttpRequest
drwxr-xr-x 2 pi pi  4096 May  7 06:32 Desktop
-rw----- 1 pi pi  962 Oct  5 16:19 GPIO-intro-script-2.py
-rwxr-xr-x 1 pi pi  5431 May  7 06:42 helloworld
-rw-r--r-- 1 pi pi  84 May  7 06:40 helloworld.c
-rw-r--r-- 1 pi pi  34 May  7 06:38 helloworld.py
drwxr-xr-x 4 pi pi  4096 Oct 13 13:49 lib
drwxr-xr-x 3 pi pi  4096 Oct 13 16:09 MonitorAgentTest
drwxrwxr-x 2 pi pi  4096 Jan 27 2015 python_games
-rw----- 1 pi pi  15360 Nov  2 11:33 SMSServercmd.exe
drwxr-xr-x 10 pi pi  4096 Oct  5 11:22 wiringPi
-rw-r--r-- 1 pi pi 567046 Oct  5 10:56 wiringPi.zip
pi@raspberrypi ~ $

```

รูปที่ ข.8 หน้าแสดงผลจากคำสั่ง `ls -l`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ทำการเปลี่ยนโหมดไฟล์ adb เพื่อให้สามารถทำการ Execute ได้ ด้วยคำสั่ง `chmod 755 adb` จากนั้นตรวจสอบไฟล์ด้วยคำสั่ง `ls -l` (แสดงรูปที่ ข.9)

```

pi@raspberrypi ~ $ chmod 755 adb
pi@raspberrypi ~ $ ls -l
total 6896
-rwxr-xr-x  1 pi pi 4131993 Nov  2 13:48 adb
-rw-----  1 pi pi 2270720 Nov  2 11:33 AndroidLib.dll
-rwxr-xr-x  1 pi pi   5431 May  7 06:41 a.out
-rwxr-xr-x  1 pi pi   6401 Oct  5 15:40 blink
-rw-r--r--  1 pi pi    727 Oct  5 15:39 blink.c
-rw-----  1 pi pi    198 Oct  5 16:30 blink.py
drwxr-xr-x  5 pi pi   4096 Oct 13 15:09 demoHttpRequest
drwxr-xr-x  2 pi pi   4096 May  7 06:32 Desktop
-rw-----  1 pi pi    962 Oct  5 16:19 GPIO-intro-script-2.py
-rwxr-xr-x  1 pi pi   5431 May  7 06:42 helloworld
-rw-r--r--  1 pi pi     84 May  7 06:40 helloworld.c
-rw-r--r--  1 pi pi     34 May  7 06:38 helloworld.py
drwxr-xr-x  4 pi pi   4096 Oct 13 13:49 lib
drwxr-xr-x  3 pi pi   4096 Oct 13 16:09 MonitorAgentTest
drwxrwxr-x  2 pi pi   4096 Jan 27  2015 python_games
-rw-----  1 pi pi  15360 Nov  2 11:33 SMSServercmd.exe
drwxr-xr-x 10 pi pi   4096 Oct  5 11:22 wiringPi
-rw-r--r--  1 pi pi  567046 Oct  5 10:56 wiringPi.zip
pi@raspberrypi ~ $

```

รูปที่ ข.9 หน้าแสดงผลจากคำสั่ง `chmod 755 adb` และ `ls -l`

10. ทำการคัดลอกไฟล์ไปเก็บไว้ยัง path `/usr/bin` เพื่อให้สามารถรันโปรแกรม adb แบบ global ได้ ด้วยคำสั่ง `sudo cp adb /usr/bin/`

```

pi@raspberrypi ~ $ sudo cp adb /usr/bin/
pi@raspberrypi ~ $

```

รูปที่ ข.10 คำสั่งสำหรับคัดลอกไฟล์ adb

11. ทดสอบรันโปรแกรม ด้วยคำสั่ง adb หากปรากฏข้อความดังภาพ (แสดงรูปที่ ข.11) แสดงว่าการติดตั้ง adb เสร็จสมบูรณ์

```

pi@raspberrypi ~ $ adb
Android Debug Bridge version 1.0.31

-d                - directs command to the only connected USB device
e                - returns an error if more than one USB device is
present.
-e              - directs command to the only running emulator.
returns an error if more than one emulator is r
unning.
-s <specific device> - directs command to the device or emulator with
the given
SERIAL
environment variable.
-p <product name or path> - simple product name like 'sooner', or
a relative/absolute path to a product
out directory like 'out/target/product/sooner'.
If -p is not specified, the ANDROID_PRODUCT_OUT
environment variable is used, which must
be an absolute path.
devices [-l]      - list all connected devices
('l' will also list device qualifiers)
connect <host>[:<port>] - connect to a device via TCP/IP

```

รูปที่ ข.11 หน้าแสดงการติดตั้ง Android debug bridge เสร็จเรียบร้อย

12. ทดสอบเชื่อมต่อ Smartphone เข้ากับ Raspberry Pi ผ่านสาย USB ด้วยคำสั่ง adb devices หากเชื่อมต่อสำเร็จจะแสดงดังรูปที่ ข.12

```

pi@raspberrypi ~ $ adb devices
List of devices attached
CB5120K6ZG    device
pi@raspberrypi ~ $

```

รูปที่ ข.12 หน้าแสดงผลการเชื่อมต่อ Smartphone เข้ากับ Raspberry Pi

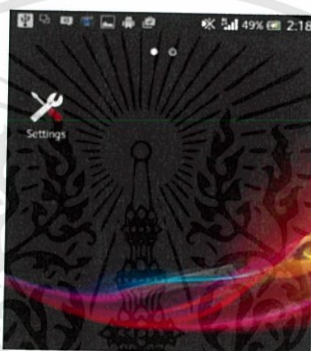
## ภาคผนวก ค.

### การเปิดฟังก์ชันการใช้งานอินเทอร์เน็ต

ระบบแจ้งเตือนฯ ที่ทีมผู้พัฒนาจัดทำขึ้น จำเป็นต้องใช้งานอินเทอร์เน็ตผ่านเครือข่ายบนสมาร์ตโฟน เพื่อใช้ส่งคำร้องขอสำหรับส่งข้อมูลการแจ้งเตือนไปยังผู้รับ โดยมีวิธีการตั้งค่าการเชื่อมต่ออินเทอร์เน็ตผ่านสายยูเอสบีซีดังนี้

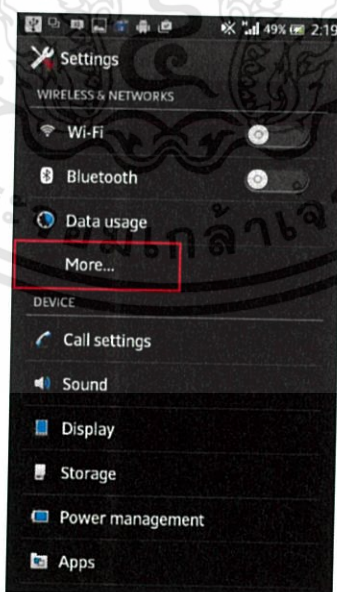
#### 1. ขั้นตอนการเปิดฟังก์ชันการใช้งานอินเทอร์เน็ตบนสมาร์ตโฟนผ่านสายยูเอสบีซี (Tethering & portable hotspot)

##### 1.1 ไปที่ setting



รูปที่ ค.1 หน้าต่างโฮมบนสมาร์ตโฟน

##### 1.2 เลือก More...



รูปที่ ค.2 หน้าต่างการตั้งค่าบนสมาร์ตโฟน

##### 1.3 เลือก Tethering & portable hotspot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.3 หน้าต่าง Wireless & networks บนสมาร์ตโฟน

#### 1.4 เลือก USB tethering



รูปที่ ค.4 หน้าต่างเมนู Tethering & portable hotspot บนสมาร์ตโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 เลือก OK



รูปที่ ค.5 ข้อความยืนยันการ Tethering &amp; portable hotspot บนสมาร์ทโฟน

## 1.6 เปิดฟังก์ชัน USB tethering เรียบร้อย

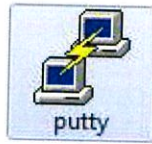


รูปที่ ค.6 หน้าต่างแสดงการเปิดฟังก์ชัน USB tethering บนสมาร์ทโฟนเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

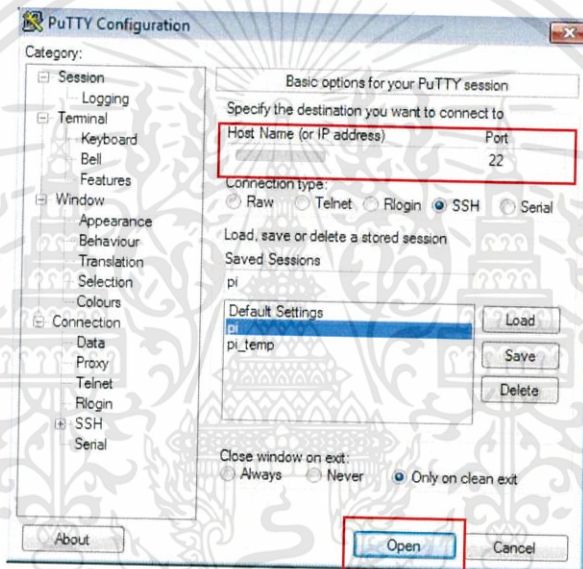
## 2. ขั้นตอนการตั้งค่าเส้นทางเครือข่ายสำหรับใช้งานอินเทอร์เน็ตบน Raspberry Pi จากสมาร์ตโฟนผ่านสายยูเอสบี ด้วยโปรแกรม PuTTY

### 2.1 เปิดโปรแกรม PuTTY เพื่อทำการรีโมตไปใช้งาน SSH Secure Shell บน Raspberry Pi



รูปที่ ค.7 ไอคอนโปรแกรม PuTTY

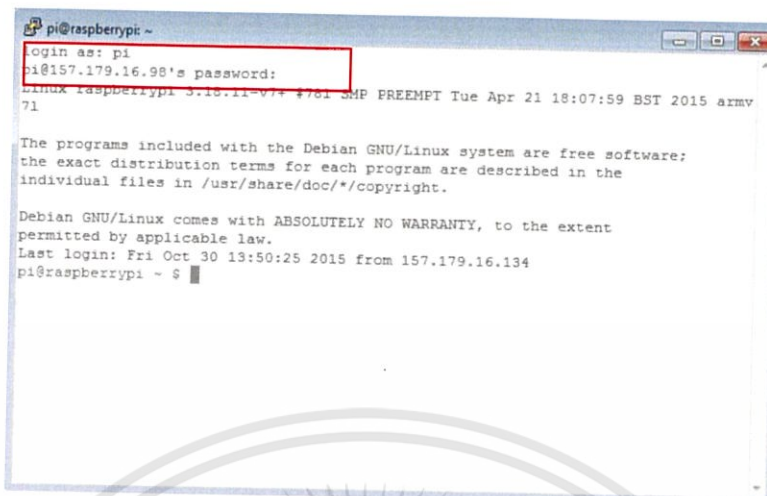
### 2.2 ระบุ Host Name และ Port แล้วกด Open



รูปที่ ค.8 หน้าต่างโปรแกรม PuTTY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3 ใส่ login as และ password แล้วกด Enter



```

pi@raspberrypi ~
login as: pi
pi@157.179.16.98's password:
Linux raspberrypi 3.18.11-v7+ #761 SMP PREEMPT Tue Apr 21 18:07:59 BST 2015 armv7l

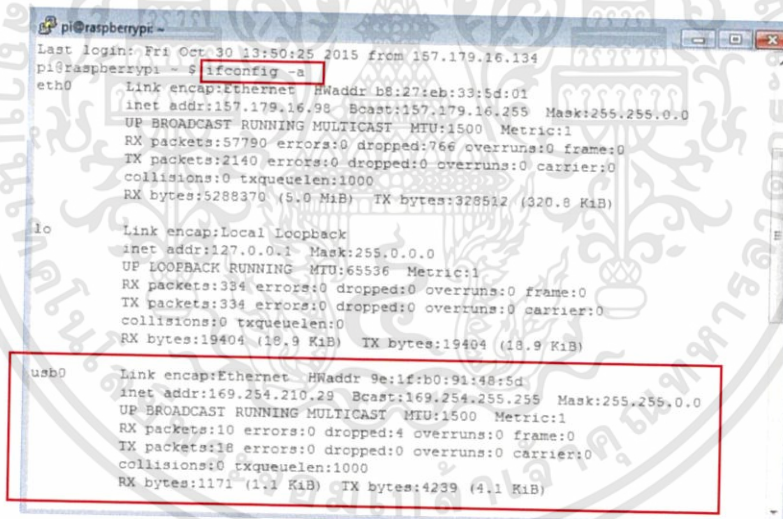
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 30 13:50:25 2015 from 157.179.16.134
pi@raspberrypi ~ $

```

รูปที่ ค.9 หน้าต่างโปรแกรม PuTTY เมื่อเชื่อมต่อ Raspberry PI แล้ว

2.4 ใช้คำสั่ง `ifconfig -a` เพื่อตรวจสอบ network Interface จะพบว่าไม่มีชื่อ `usb0` ซึ่งเป็น network ที่มาจากการเชื่อมต่อกับสมาร์ทโฟนผ่านสายยูเอสบีซี



```

pi@raspberrypi ~
Last login: Fri Oct 30 13:50:25 2015 from 157.179.16.134
pi@raspberrypi ~ $ ifconfig -a
eth0:
  Link encap:Ethernet HWaddr b8:27:eb:33:5d:01
  inet addr:157.179.16.98 Bcast:157.179.16.255 Mask:255.255.0.0
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:57790 errors:0 dropped:766 overruns:0 frame:0
  TX packets:2140 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:5288370 (5.0 MiB) TX bytes:328512 (320.8 KiB)

lo:
  Link encap:Local Loopback
  inet addr:127.0.0.1 Mask:255.0.0.0
  UP LOOPBACK RUNNING MTU:65536 Metric:1
  RX packets:334 errors:0 dropped:0 overruns:0 frame:0
  TX packets:334 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:19404 (18.9 KiB) TX bytes:19404 (18.9 KiB)

usb0:
  Link encap:Ethernet HWaddr 9e:1f:b0:91:48:5d
  inet addr:169.254.210.29 Bcast:169.254.255.255 Mask:255.255.0.0
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:10 errors:0 dropped:4 overruns:0 frame:0
  TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:1171 (1.1 KiB) TX bytes:4239 (4.1 KiB)

```

รูปที่ ค.10 หน้าต่างแสดง network Interface จากคำสั่ง `ifconfig -a`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ใช้คำสั่ง `netstat -r` เพื่อตรวจสอบสถานะเครือข่าย IP Routing table จะพบว่า Gateway ที่เป็น default อยู่ 2 หมายเลข ได้แก่ 157.179.16.1 และ 192.168.42.129

```
pi@raspberrypi ~ $ netstat -r
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default 157.179.16.1 0.0.0.0 UG 0 0 0 eth0
default 192.168.42.129 0.0.0.0 UG 0 0 0 usb0
157.179.16.0 * 255.255.255.0 U 0 0 0 eth0
192.168.42.0 * 255.255.255.0 U 0 0 0 usb0
```

รูปที่ ค.11 หน้าต่างแสดง IP Routing table จากคำสั่ง `netstat -r`

2.6 ใช้คำสั่ง `sudo route del -net 0.0.0.0 gw 157.179.16.1 netmask 0.0.0.0 dev eth0` เพื่อตั้งค่าให้ส่งออกผ่านทาง USB network (192.168.42.129) เท่านั้น

```
pi@raspberrypi ~ $ sudo route del -net 0.0.0.0 gw 157.179.16.1 netmask 0.0.0.0 dev eth0
```

รูปที่ ค.12 คำสั่งลบเส้นทางที่เป็น default gateway

2.7 ใช้คำสั่ง `netstat -r` จะพบว่ามี default destination ที่เป็น USB network เหลือทางเดียวคือ 192.168.42.129

```
pi@raspberrypi ~ $ netstat -r
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default 192.168.42.129 0.0.0.0 UG 0 0 0 usb0
157.179.16.0 * 255.255.255.0 U 0 0 0 eth0
192.168.42.0 * 255.255.255.0 U 0 0 0 usb0
pi@raspberrypi ~ $
```

รูปที่ ค.13 หน้าต่างแสดง IP Routing table หลังจากลบเส้นทางที่เป็น default

2.8 ใช้คำสั่ง traceroute <hostname> (เช่น traceroute www.google.com) เพื่อทดสอบเส้นทางเครือข่ายเน็ตเวิร์คที่ออก จะพบว่าเส้นทางที่ออกคือหมายเลข 192.168.42.129 (USB network)

```

pi@raspberrypi ~ $ traceroute www.google.com
traceroute to www.google.com (173.194.120.145), 30 hops max, 60 byte packets
 1 192.168.42.129 (192.168.42.129)  0.815 ms  0.816 ms  0.878 ms
 2 * * *
 3 10.95.31.13 (10.95.31.13)  731.543 ms  741.353 ms  761.448 ms
 4 10.95.204.14 (10.95.204.14)  761.446 ms * *
 5 * * *
 6 10.95.195.129 (10.95.195.129)  830.175 ms  2531.238 ms  2681.839 ms
 7 te7-1.node254-23.ttw.net.trueinternet.co.th (119.46.13.157)  2870.304 ms  304.591 ms  1311.398 ms
 8 61-91-213-109.static.asianet.co.th (61.91.213.109)  1311.420 ms  1311.370 ms  1311.333 ms
 9 203-144-144-10.static.asianet.co.th (203.144.144.10)  1311.319 ms  1311.286 ms  1311.268 ms
10 103-3-177-202.static.asianet.co.th (103.3.177.202)  1311.214 ms  1311.190 ms  1311.155 ms
11 TIG-Net25-57.trueintergateway.com (122.144.25.57)  1311.148 ms TIG-Net25-193.trueintergateway.com (122.144.25.193)  1311.140 ms tig-net25-197.trueintergateway.com (122.144.25.197)  471.408 ms
12 72.14.218.133 (72.14.218.133)  565.326 ms  825.672 ms  297.321 ms
13 209.85.242.240 (209.85.242.240)  840.035 ms  870.443 ms  859.511 ms
14 72.14.235.171 (72.14.235.171)  870.188 ms  870.170 ms  869.778 ms
15 kul06s08-in-f17.1e100.net (173.194.120.145)  869.996 ms  879.212 ms  879.181 ms
pi@raspberrypi ~ $

```

รูปที่ ค.14 หน้าต่างแสดงผลเส้นทางเครือข่ายเน็ตเวิร์คที่ออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.  
การตั้งค่าโปรแกรม PRTG Network Monitor  
สำหรับทำงานร่วมกับระบบแจ้งเตือนฯ

1. เข้าไปที่เว็บ <https://www.paessler.com/prtg> เพื่อดาวน์โหลดโปรแกรม PRTG Network Monitor มาติดตั้ง



รูปที่ ง.1 หน้าเว็บสำหรับดาวน์โหลดโปรแกรม PRTG Network Monitor

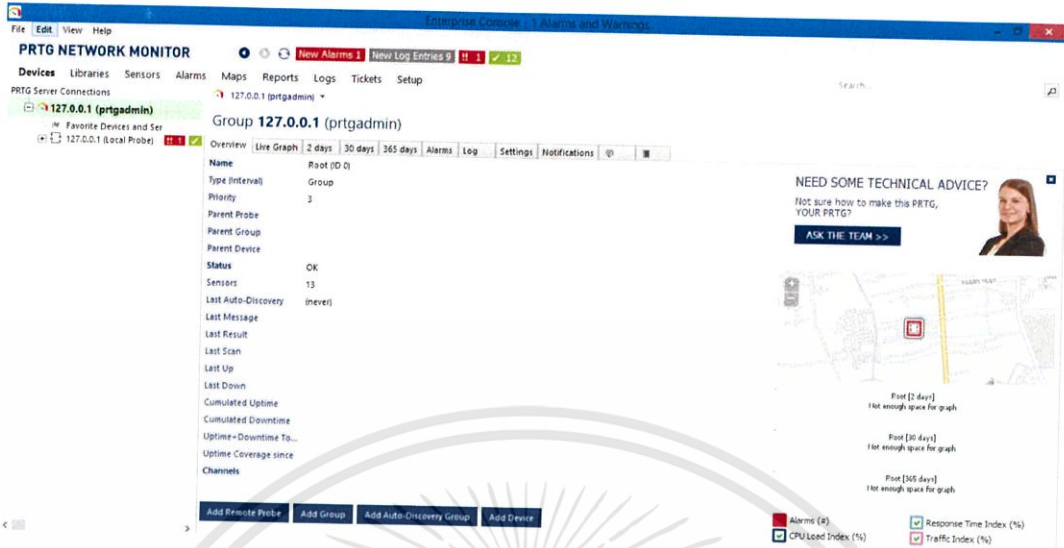
2. หลังจากติดตั้งโปรแกรมเสร็จแล้ว ให้เปิดโปรแกรม PRTG Enterprise Console ขึ้นมา



รูปที่ ง.2 ไอคอนโปรแกรม PRTG Enterprise Console

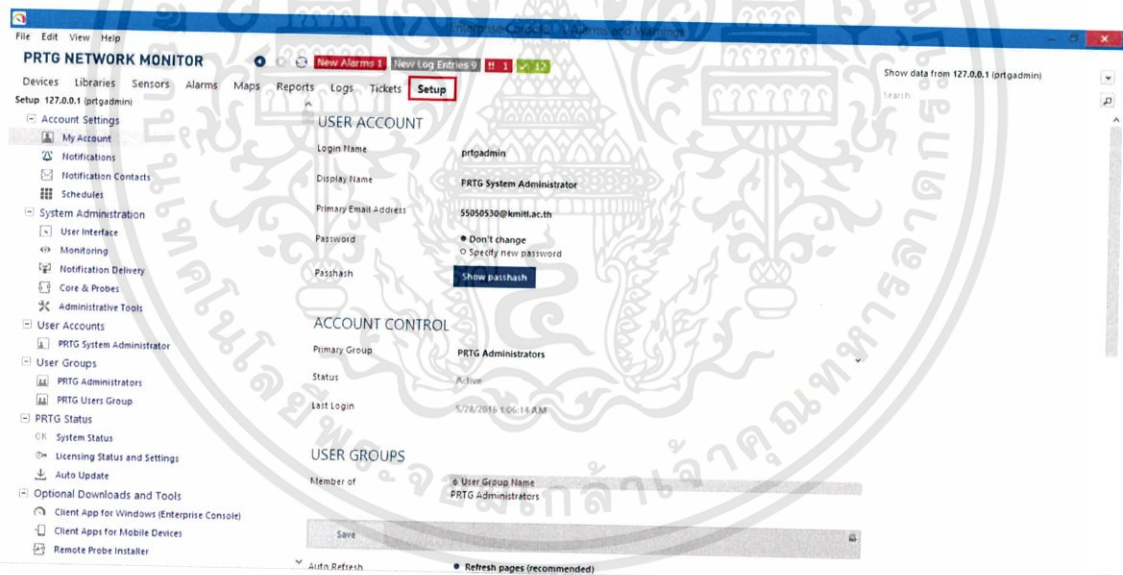
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. โปรแกรมจะแสดงหน้าต่างดังรูปที่ 3



รูปที่ 3.3 หน้าแรกโปรแกรม PRTG Enterprise Console

### 4. ไปที่แถบ Setup



รูปที่ 3.4 หน้าต่างแถบ Setup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5. ไปที่ System Administration เลือก Notification Delivery

Setup 127.0.0.1 (prtgadmin)

- Account Settings
  - My Account
  - Notifications
  - Notification Contacts
  - Schedules
- System Administration
  - User Interface
  - Monitoring
  - Notification Delivery**
  - Core & Probes
  - Administrative Tools
- User Accounts
  - PRTG System Administrator
- User Groups
  - PRTG Administrators
  - PRTG Users Group
- PRTG Status
  - System Status
  - Licensing Status and Settings
  - Auto Update
- Optional Downloads and Tools
  - Client App for Windows (Enterprise Console)
  - Client Apps for Mobile Devices
  - Remote Probe Installer

**SMTP DELIVERY**

SMTP Delivery Mechanism

- Direct delivery using built-in email server (default)
- Use SMTP relay server (recommended inside LANs/NATs)
- Use two SMTP relay servers (primary and fallback server)

Sender E-Mail: 55050530@kmitl.ac.th

Sender Name: PRTG Network Monitor

HELO Ident: PEPCS

**SMS DELIVERY**

Configuration Mode

- Select an SMS provider from a list of providers
- Enter a custom URL for a provider not listed

Service Provider: www.bulksms.co.uk (Port 5567)

User: \_\_\_\_\_

Password: \_\_\_\_\_

API ID / Account: \_\_\_\_\_

Maximum Length of Text: 0

Save

รูปที่ ง.5 หน้าต่างเมนู Notification Delivery

## 6. ไปที่ Configuration Mode เลือกเป็น Enter a custom URL for a provider not listed

Setup 127.0.0.1 (prtgadmin)

- Account Settings
  - My Account
  - Notifications
  - Notification Contacts
  - Schedules
- System Administration
  - User Interface
  - Monitoring
  - Notification Delivery**
  - Core & Probes
  - Administrative Tools
- User Accounts
  - PRTG System Administrator
- User Groups
  - PRTG Administrators
  - PRTG Users Group
- PRTG Status
  - System Status
  - Licensing Status and Settings
  - Auto Update
- Optional Downloads and Tools
  - Client App for Windows (Enterprise Console)
  - Client Apps for Mobile Devices
  - Remote Probe Installer

**SMTP DELIVERY**

SMTP Delivery Mechanism

- Direct delivery using built-in email server (default)
- Use SMTP relay server (recommended inside LANs/NATs)
- Use two SMTP relay servers (primary and fallback server)

Sender E-Mail: 55050530@kmitl.ac.th

Sender Name: PRTG Network Monitor

HELO Ident: PEPCS

**SMS DELIVERY**

Configuration Mode

- Select an SMS provider from a list of providers
- Enter a custom URL for a provider not listed

Custom URL: \_\_\_\_\_

Encoding for SMS

- ANSI local system codepage (default)
- UTF-8
- UTF-16

Maximum Length of Text: 0

Save

รูปที่ ง.6 การตั้งค่า Configuration Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ระบุ Custom URL สำหรับส่งคำร้องขอไปยังระบบแจ้งเตือนฯ ซึ่งมีรูปแบบดังตัวอย่าง

IP address/File name/?from=13013&text=%SMSTEXT&to=%SMSNUMBER

&username=admin&password=admin

หมายเหตุ IP address ในที่นี้เป็นของ Raspberry pi ที่ติดตั้งระบบแจ้งเตือนฯ อยู่

## SMS DELIVERY

Configuration Mode

- Select an SMS provider from a list of providers
- Enter a custom URL for a provider not listed

Custom URL

.....asp?from=13013&text=%SMSTEXT&to=%SMSNUMBER&usernan

Encoding for SMS

- ANSI local system codepage (default)
- UTF-8
- UTF-16

Maximum Length of Text

0

รูปที่ ง.7 การตั้งค่า Custom URL

8. เลือก Encoding for SMS เป็น UTF-8

Encoding for SMS

- ANSI local system codepage (default)
- UTF-8
- UTF-16

รูปที่ ง.8 ตั้งค่า Encoding for SMS

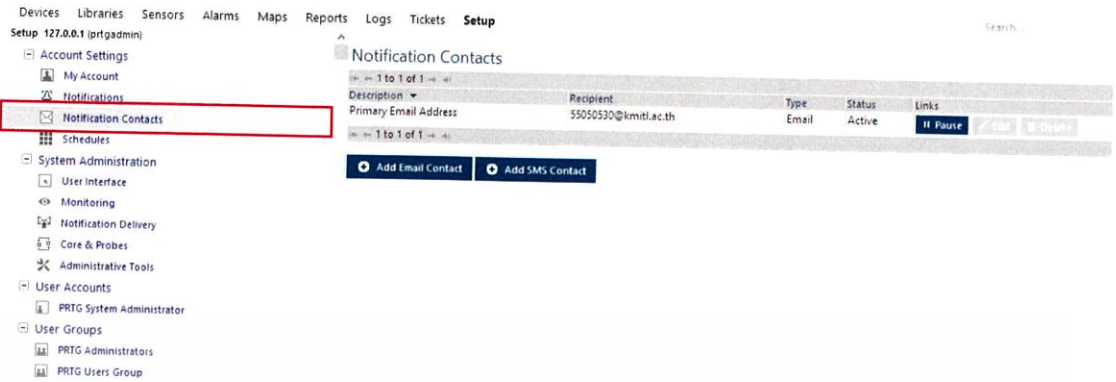
9. กด Save

Save

รูปที่ ง.9 ปุ่มบันทึก

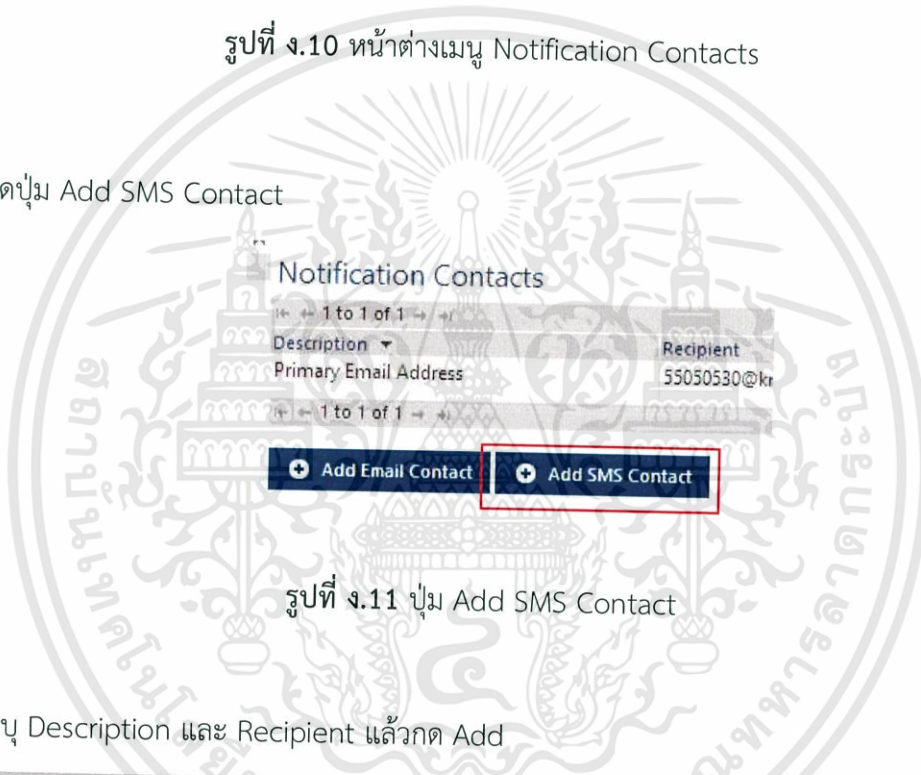
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 10. ไปที่ Notification Contacts



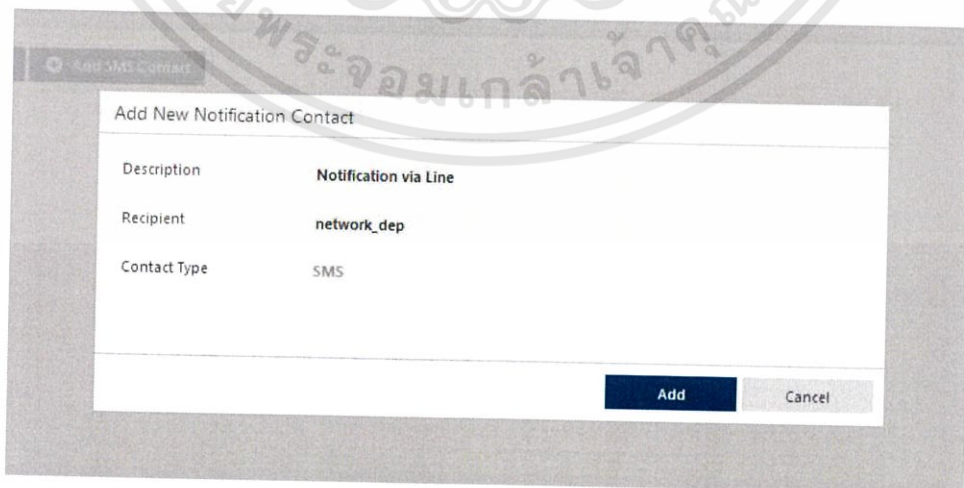
รูปที่ ๑.10 หน้าต่างเมนู Notification Contacts

### 11. กดปุ่ม Add SMS Contact



รูปที่ ๑.11 ปุ่ม Add SMS Contact

### 12. ระบุ Description และ Recipient แล้วกด Add



รูปที่ ๑.12 การสร้าง Notification Contact

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 13. เสร็จสิ้นการตั้งค่า Notification Contacts

Notification Contacts

← 1 to 2 of 2 →

Description	Recipient	Type	Status	Links
Notification via Line	network_dep	SMS	Active	<a href="#">Pause</a> <a href="#">Edit</a> <a href="#">Delete</a>

รูปที่ ง.13 รายการ Notification Contact ที่สร้างขึ้น



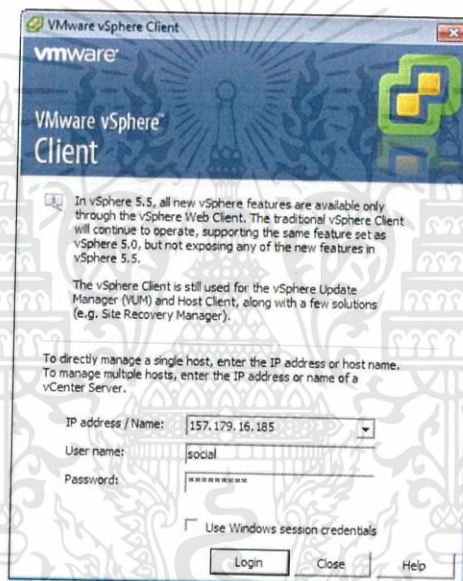
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก จ. การทดสอบปิดการเชื่อมต่อเครือข่าย

ระบบที่ทีมผู้พัฒนาจัดทำขึ้น สามารถใช้งานร่วมกับเครื่องเซิร์ฟเวอร์ที่ติดตั้งโปรแกรม PRTG Network Monitor ซึ่งโปรแกรมดังกล่าวจะทำหน้าที่คอยตรวจจับข้อผิดพลาดที่เกิดขึ้นและทำการร้องขอมายังระบบแจ้งเตือนๆ ของเราเพื่อทำการส่งข้อความแจ้งเตือนในลำดับถัดไป ในส่วนของการทดสอบสามารถศึกษาได้ตามขั้นตอนดังนี้

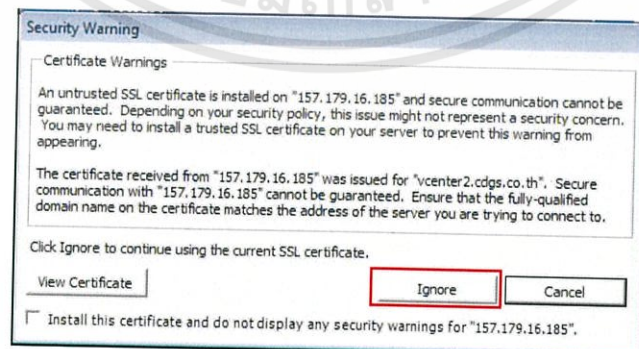
### 1. ทดสอบปิดการเชื่อมต่อเครือข่ายของเครื่องเซิร์ฟเวอร์

1.1 ทำการรีโมตไปใช้งานเครื่องเซิร์ฟเวอร์ที่จำลองขึ้นมาโดยใช้โปรแกรม VMware vSphere Client



รูปที่ จ.1 เริ่มต้นโปรแกรม VMware vSphere Client

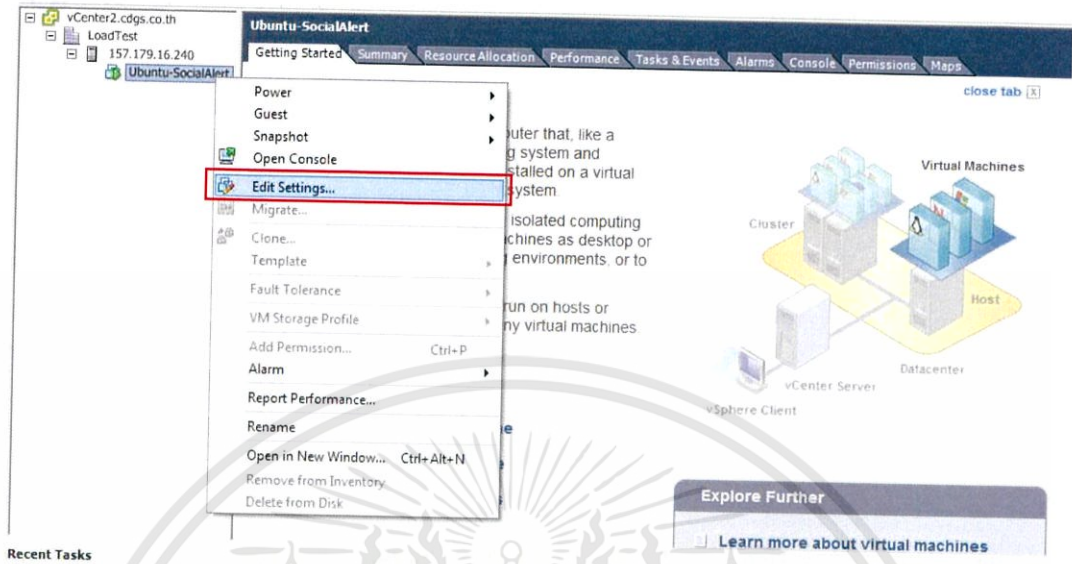
### 1.2 เลือก Ignore



รูปที่ จ.2 ยืนยันการรับรองความปลอดภัยทางอิเล็กทรอนิกส์

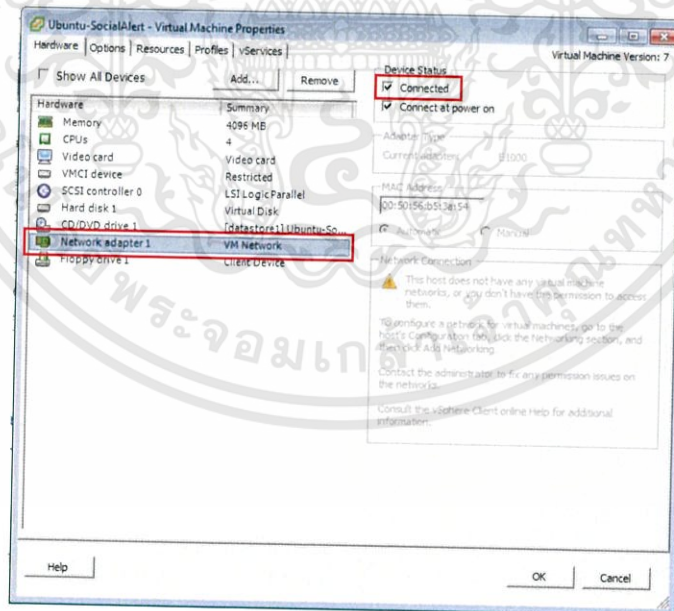
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 จะพบว่าเมื่อคลิกที่ชื่อ Ubuntu-SocialAlert ให้คลิกขวา เลือก “Edit Setting...”



รูปที่ จ.3 หน้าต่างรายละเอียดของ VMware vSphere Client

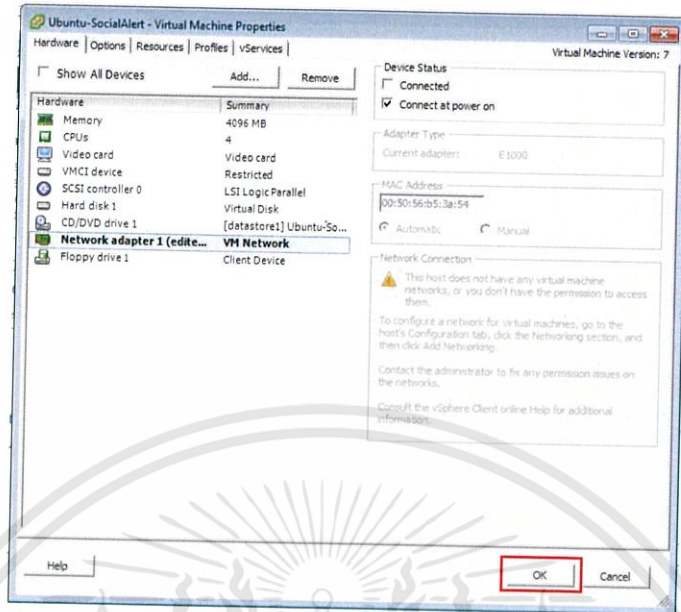
1.4 ปิดการใช้งาน Network adapter ในส่วนของ Device Status ให้เลือก  Connected ออก



รูปที่ จ.4 หน้าต่างการตั้งค่าของ VMware vSphere Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 ยืนยันการแก้ไข เลือก OK



รูปที่ จ.5 หน้าต่างการตั้งค่าของ VMware vSphere Client (หลังเปลี่ยนแปลง)

## 1.6 รอจนกระทั่งสถานะการทำงานเป็น Completed

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Time	Start Time	Completed Time
Reconfigure virtual ma...	Ubuntu-SocialA...	Completed		social	vCenter2.cdgs...	04-12-2015 10:38:15 ...	04-12-2015 10:38:15 ...	04-12-2015 10:38:15 ...

รูปที่ จ.6 แถบสถานะการทำงานหลังการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

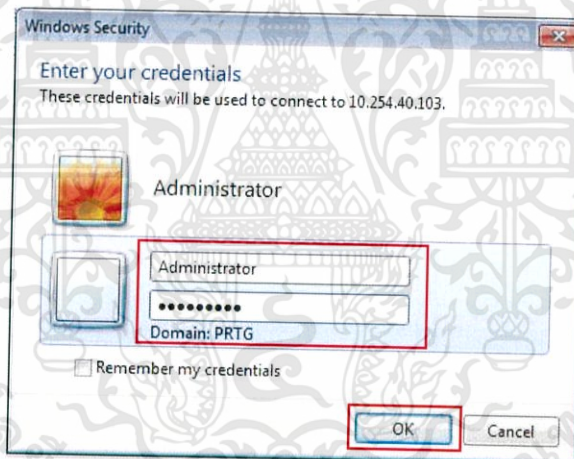
## 2. ตรวจสอบผลการทำงานโดย PRTG NETWORK MONITOR

2.1 ทำการรีโมตไปยังเครื่องเซิร์ฟเวอร์ที่ติดตั้ง PRTG Network Monitor สำหรับทำหน้าที่ดูการทำงานต่างๆ ของเครื่องในเครือข่าย ใส่หมายเลขที่อยู่ แล้วกด “Connect”



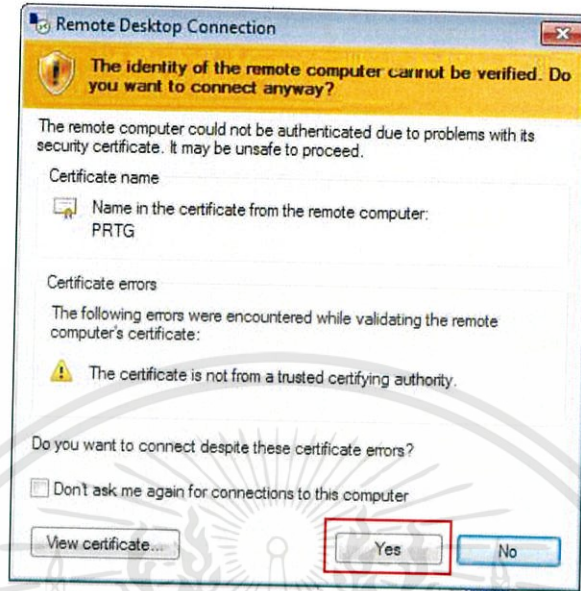
รูปที่ จ.7 เริ่มต้นโปรแกรม Remote Desktop Connection

2.2 กรอกข้อมูลยืนยันผู้ใช้งาน แล้วกด “OK”



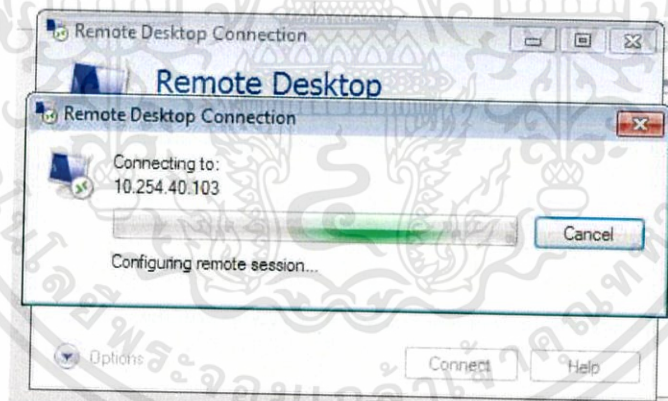
รูปที่ จ.8 หน้าต่างยืนยันข้อมูลผู้ใช้งาน

2.3 ระบบจะแจ้งว่ามีข้อผิดพลาดจากการตรวจสอบใบรองรับการเข้าถึงเซิร์ฟเวอร์ดังกล่าว ให้เลือก Yes เพื่อดำเนินการต่อ



รูปที่ จ.9 หน้าต่างการตรวจสอบใบรองรับการเข้าถึง

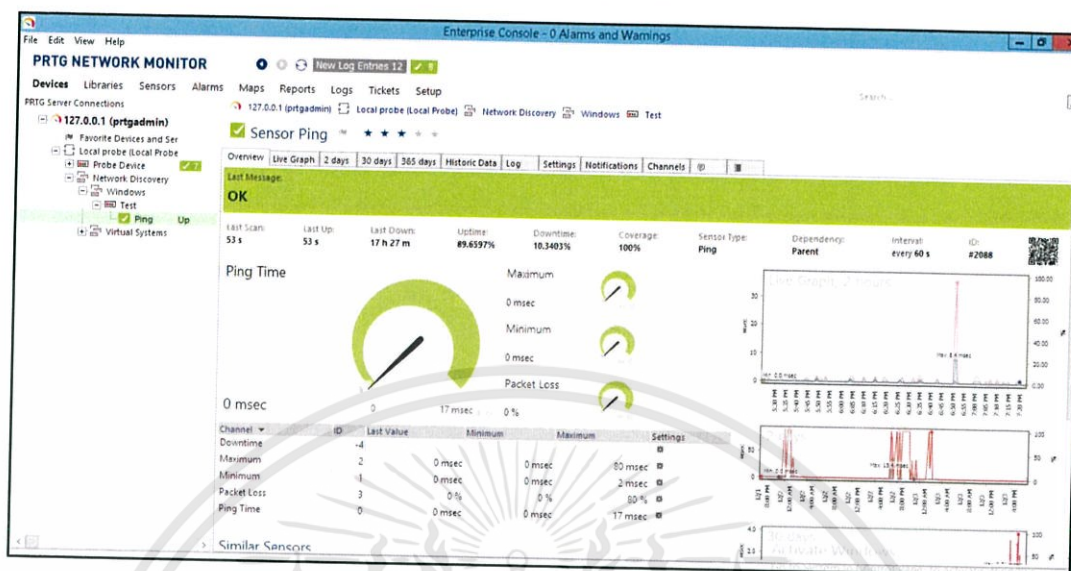
2.4 ทำการเชื่อมต่อไปยังเครื่องเซิร์ฟเวอร์



รูปที่ จ.10 หน้าต่างรอการเชื่อมต่อ

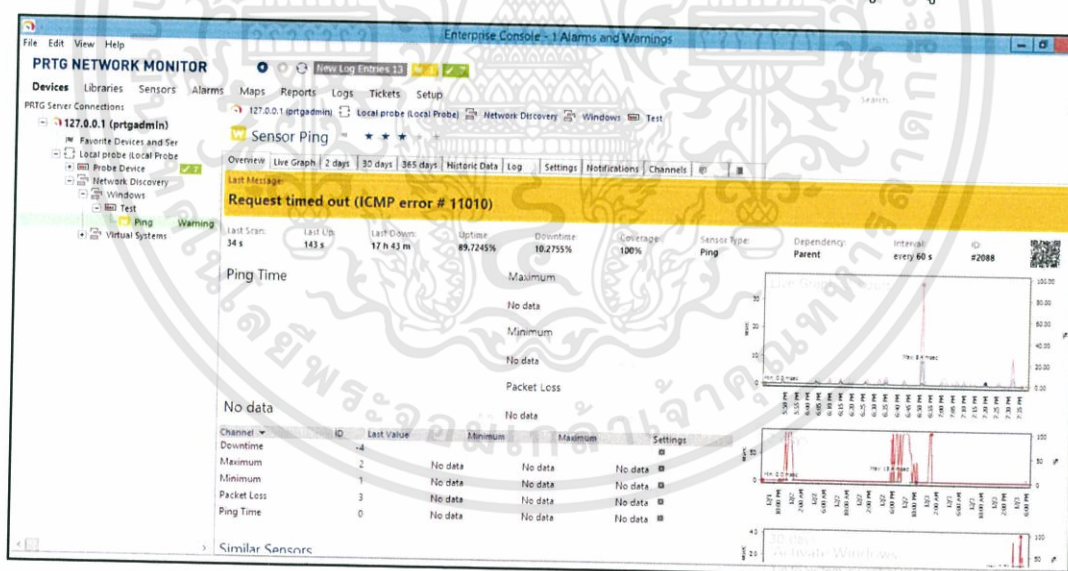
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เปิดโปรแกรม PRTG Network Monitor เพื่อดูสถานะการทำงานบนเครือข่าย หากการเชื่อมต่อบนเครือข่ายทำงานปกติ โปรแกรมจะแสดงข้อมูลดังรูป



รูปที่ จ.11 หน้าต่างโปรแกรม PRTG สถานะการทำงานปกติ (Up)

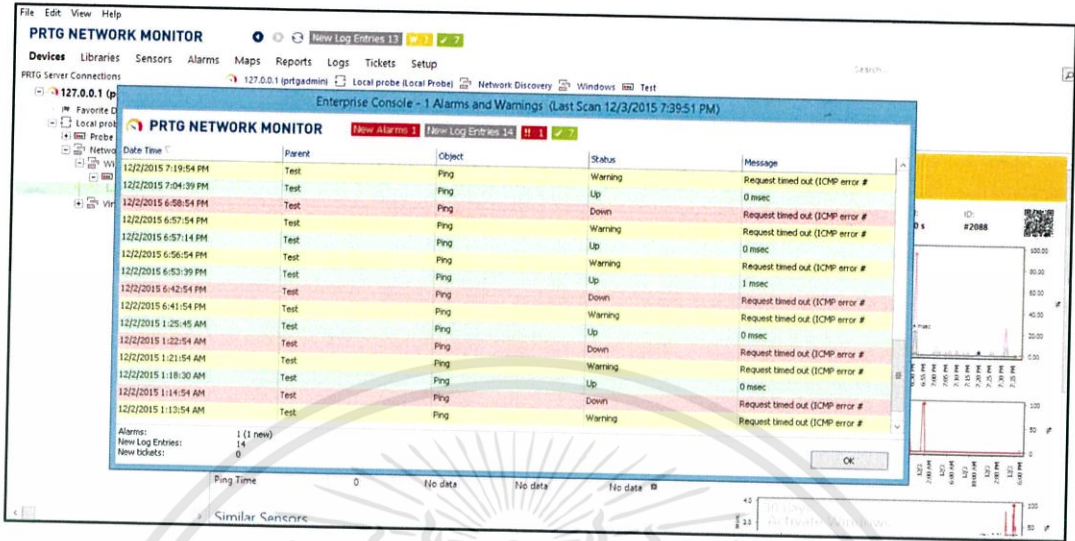
2.6 เมื่อการเชื่อมต่อเครือข่ายเริ่มมีปัญหาเกิดขึ้น โปรแกรมจะแสดงข้อมูลดังรูป



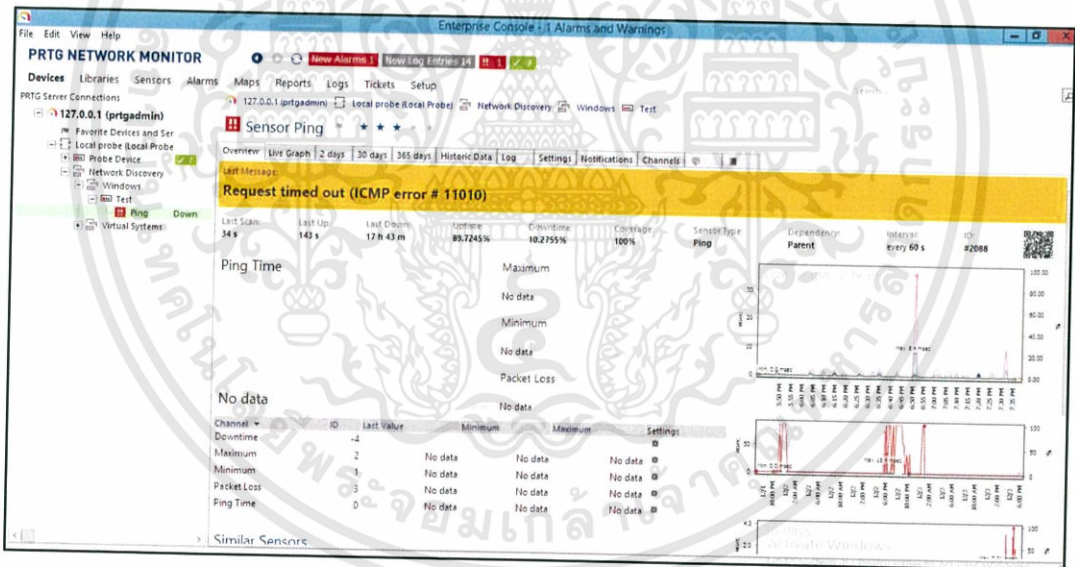
รูปที่ จ.12 หน้าต่างโปรแกรม PRTG สถานะการทำงานเริ่มมีปัญหา (Warning)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 โปรแกรมจะแสดงหน้าต่างป๊อปอัพบอกสถานะการทำงานปัจจุบัน และทำการส่งข้อมูลสำหรับแจ้งเตือนไปยังผู้รับตามที่ได้อัปเดตค่าไว้



รูปที่ จ.13 หน้าต่างโปรแกรม PRTG แสดงสถานะการทำงานปัจจุบัน



รูปที่ จ.14 หน้าต่างโปรแกรม PRTG สถานะการทำงานมีข้อผิดพลาด (Down)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ฉ.

## การใช้งานระบบแจ้งเตือนทางแอปพลิเคชันไลน์ สำหรับอุปกรณ์ในท้องถิ่นเซิร์ฟเวอร์

1. ตั้งค่าไฟล์ alertConf.ini เพื่อระบุที่อยู่ผู้รับสำหรับส่งข้อความในการแจ้งเตือน โดยตำแหน่งไฟล์จะอยู่ที่ไดเรกทอรี demoSMSServer\demoSMSServer\SMSServercmd

bin	25/11/2015 11:34 ...	File folder	
obj	25/11/2015 11:34 ...	File folder	
Properties	25/11/2015 11:34 ...	File folder	
reference	25/11/2015 11:34 ...	File folder	
Service References	26/10/2015 1:50 PM	File folder	
alertConf	27/11/2015 10:43 ...	Configuration sett...	1 KB
AndroidLib.dll	30/8/2015 10:20 AM	Application extens...	2,218 KB
app	16/10/2015 2:48 PM	CONFIG File	1 KB
HtmlAgilityPack.dll	6/8/2012 9:30 PM	Application extens...	132 KB
Program	27/11/2015 2:29 PM	CS File	35 KB
SimpleHttpServer	20/11/2015 4:34 PM	CS File	8 KB
SMSServercmd.csproj	18/11/2015 11:32 ...	CSPROJ File	5 KB
th.co.cdgs.mis.smsserver	31/8/2015 2:20 PM	SMSSERVER File	1,075 KB

รูปที่ ฉ.1 ไดเรกทอรีไฟล์ alertConf.ini

2. เปิดไฟล์ alertConf.ini เพื่อระบุข้อมูลสำหรับส่งข้อความไปยังผู้รับ (รายละเอียดการตั้งค่าอยู่ในหัวข้อ 3.5.1)

```

1  [System]
2  username = social_alert@kmitl.ac.th
3  password = P@ssw0rd
4  time_next = 5
5  time_out = 30
6
7  [database_dep]
8  receiver1 = {John}
9  receiver2 = {Bob}
10 receiver3 = {Cartor}
11
12 [network_dep]
13 receiver1 = {Jerry}
14 receiver2 = {Tony}
15 receiver3 = {Susie}
16 receiver4 = {Deb}
17
18

```

รูปที่ ฉ.2 โครงสร้างไฟล์ alertConf.ini

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. นำไฟล์ระบบแจ้งเตือนฯ ไปติดตั้งบน Raspberry Pi (รายละเอียดอยู่ภาคผนวก ก)
4. หลังจากติดตั้งไฟล์ระบบแจ้งเตือนฯ บน Raspberry Pi เรียบร้อย ให้ทำการรีโมตไปใช้งาน SSH Secure Shell ผ่านโปรแกรม PuTTY ในการสั่งรันโปรแกรม

```

pi@raspberrypi ~
login as: pi
pi@157.179.16.98's password:
Linux raspberrypi 3.18.11-v7+ #781 SMP PREEMPT Tue Apr 21 18:07:59 BST 2015 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov  2 13:43:34 2015 from 157.179.135.50
pi@raspberrypi ~ $

```

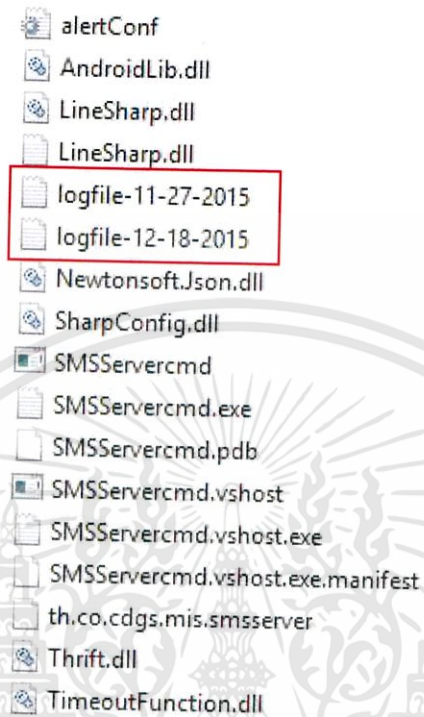
รูปที่ ฉ.3 SSH Secure Shell บน Raspberry Pi

5. ใช้คำสั่ง `cd demoSMSServer\demoSMSServer\SMSServercmd\bin\Debug` เพื่อเปลี่ยนไดเรกทอรีไปยังที่เก็บไฟล์สำหรับรันโปรแกรม
6. ใช้คำสั่ง `sudo mono SMSServercmd.exe` เพื่อสั่งเริ่มต้นโปรแกรมระบบแจ้งเตือนทางแอปพลิเคชันไลน์สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์

```
pi@raspberrypi ~ $ sudo mono SMSServercmd.exe
```

รูปที่ ฉ.4 คำสั่งเริ่มต้นโปรแกรมระบบแจ้งเตือนทางแอปพลิเคชันไลน์สำหรับอุปกรณ์ในห้องเซิร์ฟเวอร์

7. ผู้ใช้สามารถตรวจสอบสถานะการทำงานภายหลังได้จากไฟล์ล็อกสถานะการทำงาน โดยเข้าไปที่ไดเรกทอรี demoSMSServer\demoSMSServer\SMSServercmd\bin\Debug จะพบว่าไฟล์ล็อกสถานะการทำงานจะมีชื่อไฟล์ที่ขึ้นต้นด้วย logfile



รูปที่ ๑.5 ไฟล์ล็อกสถานะการทำงาน