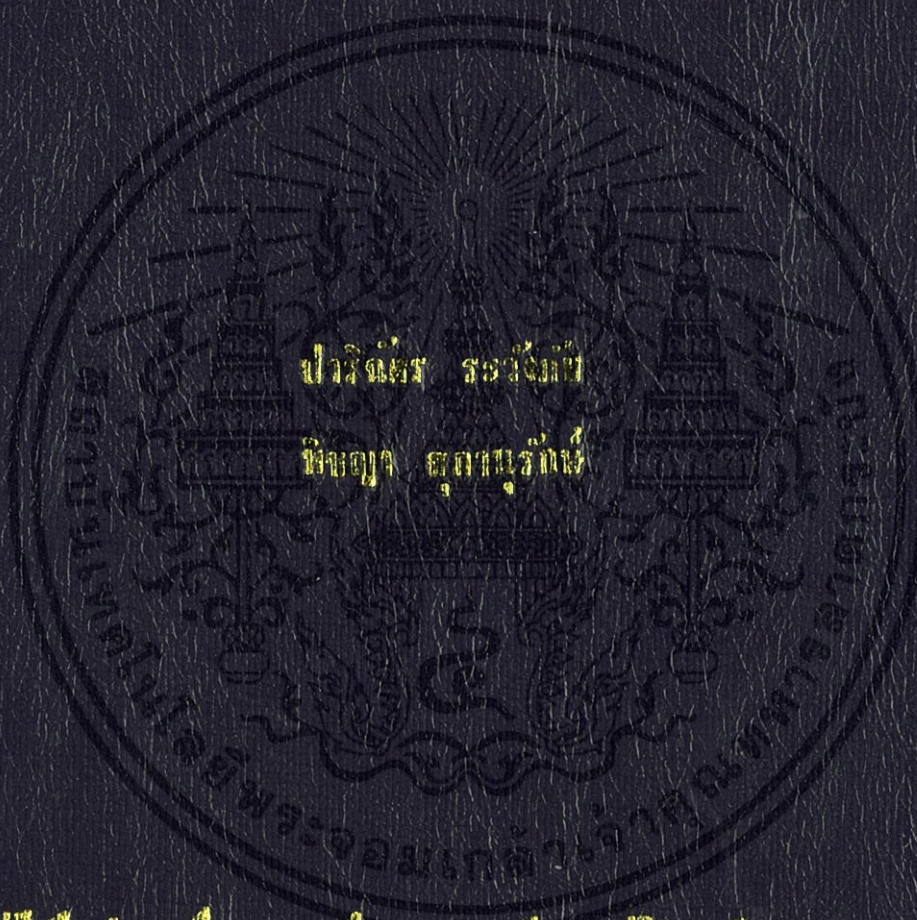


ซอฟต์แวร์ตรวจสอบสถานะเซิร์ฟเวอร์บนแพลตฟอร์มแอนดรอยด์
SERVER MONITOR IN ANDROID APPLICATION



วิทยาลัยเทคโนโลยีและสหวิชาชีพ
วิทยาลัยเทคโนโลยีและสหวิชาชีพ
วิทยาลัยเทคโนโลยีและสหวิชาชีพ

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ปริญญาานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แอปพลิเคชันตรวจสอบสภาพเครื่องแม่ข่ายคอมพิวเตอร์

SERVER MONITOR IN ANDROID APPLICATION

ผู้จัดทำ

1. นางสาวปาริฉัตร ระวังภัย รหัสนักศึกษา 54010814

2. นางสาวพิชญา สุภานุรักษ์ รหัสนักศึกษา 54010912



(อาจารย์เกียรติฉัตร ท่องประเสริฐ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอปพลิเคชันตรวจสอบสภาพเครื่องแม่ข่ายคอมพิวเตอร์

นางสาวปาริฉัตร	ระวิงภัย	54010814
นางสาวพิชญ์ญา	สุภานุรักษ์	54010912
อาจารย์เกียรติฉัตร	ทองประเสริฐ	อาจารย์ที่ปรึกษา

ปีการศึกษา 2557

บทคัดย่อ

แอปพลิเคชันตรวจสอบสภาพเครื่องแม่ข่ายคอมพิวเตอร์ จัดทำขึ้นมาจากแนวคิดที่ต้องการให้สมาร์ตโฟนสามารถติดต่อกับเครื่องแม่ข่ายหรือเครื่องคอมพิวเตอร์ได้ เพื่อใช้สำหรับการตรวจสอบสภาพและการทำงานของเครื่อง เกี่ยวกับด้านฮาร์ดแวร์ ซอฟต์แวร์ และการรีโมทคำสั่งเข้าไปควบคุมเครื่องแม่ข่าย โดยมีอุปกรณ์ที่ใช้ในปฏิญญานิพนธ์นี้คือสมาร์ตโฟนที่ใช้ระบบปฏิบัติการแอนดรอยด์ และเครื่องแม่ข่ายที่ใช้ทดลองคือ เครื่องคอมพิวเตอร์ที่มีระบบปฏิบัติการวินโดวส์ และระบบปฏิบัติการวินโดวส์เซิร์ฟเวอร์

เป็นความจริงที่ว่าบริษัทต่างๆก็มีเครื่องแม่ข่ายที่ใช้งานอยู่มากกว่าหนึ่งเครื่อง จึงมีการเพิ่มฟังก์ชันการทำงานของแอปพลิเคชันให้สามารถติดต่อกับเครื่องคอมพิวเตอร์ได้ทีละหลายเครื่อง การใช้งานแอปพลิเคชันนี้จะต้องคำนึงถึงความปลอดภัยเป็นสำคัญ เมื่อเริ่มต้นการใช้งานแอปพลิเคชันจึงต้องมีการตั้งค่าคีย์ให้ตรงกับค่า Community ใน SNMP Service ของเครื่องแม่ข่าย ซึ่งถือว่าช่วยในการรักษาความปลอดภัยได้ระดับหนึ่ง เมื่อมีการใส่หมายเลขไอพีของเครื่องแต่ละเครื่องแม่ข่ายที่ต้องการตรวจสอบในแอปพลิเคชันแล้ว แอปพลิเคชันจะสามารถตรวจสอบเบื้องต้นได้เลยว่าเครื่องแม่ข่ายนั้นกำลังเปิดใช้งานอยู่หรือไม่(Online or Offline) สามารถเรียกดูรูปภาพของเครื่องแม่ข่ายนั้นๆและ ตรวจสอบได้ว่าเครื่องแม่ข่ายนั้นมีที่ตั้งอยู่ที่ใด โดยแสดงผลออกมาทาง Google Map

ซึ่งแอปพลิเคชันนี้จะช่วยอำนวยความสะดวกให้แก่ผู้ใช้ในกรณีที่ไม่สามารถเดินทางไปตรวจสอบเครื่องแม่ข่ายได้ด้วยตนเอง ช่วยประหยัดเวลาในการทำงานเพราะสามารถตรวจสอบเครื่องหลายๆเครื่องได้พร้อมๆกัน เปรียบเสมือนว่าเครื่องแม่ข่ายหลายๆเครื่องอยู่ใกล้ผู้ใช้ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SERVER MONITOR IN ANDROID APPLICATION

Ms. Parichat Rawangpai 54010814

Ms. Pitchaya Supanuruk 54010912

Asst. Kiatnarong Tongprasert Advisor

Academic Year 2014

ABSTRACT

Server Monitor in Android Application is built for the idea of communication between smart phone and server. This application is used to check status and usage of server in scope of hardware, software and it also can send the command to control server such as shutdown or restart. For this project, we use android operating system device and Microsoft Window server for developing.

In this day, many companies usually have server more than one, so we decide to add function that can monitor several server. In this application, the first time that user run application, we force user to insert key to make this application have more secure. The key has to match field community in SNMP service in server. When user adds IP Address of server, the application will show the status of server that online or offline. User also can take separately photo for each server and know the location for each server that show in Google Map. It very useful for user who cannot check server by themselves. The application save working time because it can monitor many servers at one time.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การศึกษา ออกแบบ และพัฒนา วิชาโครงการนี้สำเร็จได้อย่างดี ด้วยความช่วยเหลือ และ คำปรึกษาจากอาจารย์เกียรติณรงค์ ทองประเสริฐ ซึ่งเป็นอาจารย์ที่ปรึกษาของโครงการนี้ ที่ได้คอย ให้คำแนะนำพร้อมทั้งตรวจสอบความก้าวหน้าของโครงการอย่างสม่ำเสมอตลอดระยะเวลาที่ผ่านมา

ขอขอบพระคุณอาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่างๆ ให้ ข้าพเจ้า ตลอดระยะเวลาที่ผ่านมา

ขอขอบคุณเพื่อนในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้า คุณทหารลาดกระบัง ที่คอยให้คำปรึกษา ดิชม และเป็นผู้ทดลองใช้ที่ดีเสมอมา

ขอกราบขอบพระคุณบิดามารดา และบุคคลภายในครอบครัวที่ให้กำลังใจและการ สนับสนุนในทุกๆ ด้าน

สุดท้ายนี้ขอขอบคุณบุคคลอื่นๆอีกมากมาย ที่ผู้ดำเนินโครงการไม่สามารถกล่าวนามได้ ทั้งหมดในที่นี้ ที่ให้ความช่วยเหลือต่างๆแก่ผู้ดำเนินโครงการ ผู้ดำเนินโครงการรู้สึกซาบซึ้งใน ความปรารถนาดีของทุกท่านเป็นอย่างยิ่ง จึงขอขอบพระคุณ และขอบคุณ ทุกท่านไว้ ณ โอกาสนี้

ปรีฉัตร
พิชญ์

ระวิงภัย
สุภานุรักษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1.....	1
1.1 ความสำคัญและที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีการดำเนินงาน.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 ส่วนประกอบของปริญญานิพนธ์.....	3
บทที่ 2.....	5
2.1 ระบบปฏิบัติการแอนดรอยด์.....	5
2.1.1 ประวัติความเป็นมา.....	5
2.1.2 สถาปัตยกรรมแอนดรอยด์.....	5
2.1.3 ชนิดของโปรแกรมประยุกต์.....	6
2.1.4 เครื่องมือที่สำหรับการพัฒนาแอปพลิเคชันบนระบบแอนดรอยด์.....	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามทำซ้ำหรือดัดแปลงในชื่อของตนเองหรือผู้อื่นโดยไม่ได้รับอนุญาต

สารบัญ(ต่อ)

	หน้า
2.2 Simple Network Management Protocol (SNMP).....	8
2.2.1 ข้อมูลทั่วไปเกี่ยวกับเวอร์ชันของ SNMP	8
2.2.2 เมเนเจอร์และเอเจนต์ (Manager and Agent).....	9
2.2.3 SNMP กับ UDP.....	11
2.2.4 SNMP คอมมูนิตี (SNMP Communities)	12
2.2.5 Simple Network Management Protocol version 2(SNMPv2).....	13
2.3 การสร้างเมสเสจ SNMP	17
2.3.1 ASN.1.....	20
2.3.2 BER (Basic Encoding Rules).....	21
2.3.3 ฟอรัมเมตเมสเสจของ SNMP.....	22
2.4 การให้บริการแผนที่ด้วยกูเกิลแมพ (Google Maps).....	24
บทที่ 3	27
3.1 ภาพรวมของระบบ	27
3.1.1 การทำงานในส่วนหลัก.....	27
3.1.2 การทำงานในส่วนของวิดเจ็ต (Widget).....	28
3.2 Use Case Diagram.....	29
3.3 Class Diagram	30
3.4 Flow Chart.....	31
3.5 Sequence Diagram	38
3.5.1 GetKey Sequence Diagram	38
3.5.2 ServerList Sequence Diagram.....	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ออกทางที่มีให้คิดแบบลงมือทำ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.5.3 Monitor Sequence Diagram	40
3.5.4 ExpandListAdapter Sequence Diagram	41
3.5.5 Map Sequence Diagram	42
3.5.6 DatalistPhoto Sequence Diagram.....	43
บทที่ 4	44
4.1 ทดลองฟังก์ชันการตั้งค่าคีย์เริ่มต้น	44
4.2 ทดลองฟังก์ชันการเพิ่มเครื่องแม่ข่ายที่ต้องการตรวจสอบ.....	45
4.3 ทดลองฟังก์ชันการเปลี่ยนชื่อเครื่องแม่ข่ายที่ต้องการ	47
4.4 ทดลองฟังก์ชันแสดงตำแหน่งบน Google Map	49
4.5 ทดลองสร้างฟังก์ชันในการเข้าสู่เครื่องแม่ข่าย.....	51
4.6 ทดลองสร้างแถบเมนูการทำงานหลัก.....	53
4.7 ทดลองการใช้ฟังก์ชันแสดงรายละเอียดฮาร์ดแวร์	54
4.8 ทดลองการใช้ฟังก์ชันแสดงข้อมูลซอฟต์แวร์	56
4.9 ทดลองการใช้ฟังก์ชันควบคุมเครื่องแม่ข่าย.....	59
4.10 ทดลองการใช้ฟังก์ชันแสดงรูปภาพเครื่องแม่ข่าย.....	60
บทที่ 5	62
5.1 บทสรุป.....	62
5.2 ปัญหาและอุปสรรค	62
5.3 แนวทางการแก้ไข.....	62

สารบัญ(ต่อ)

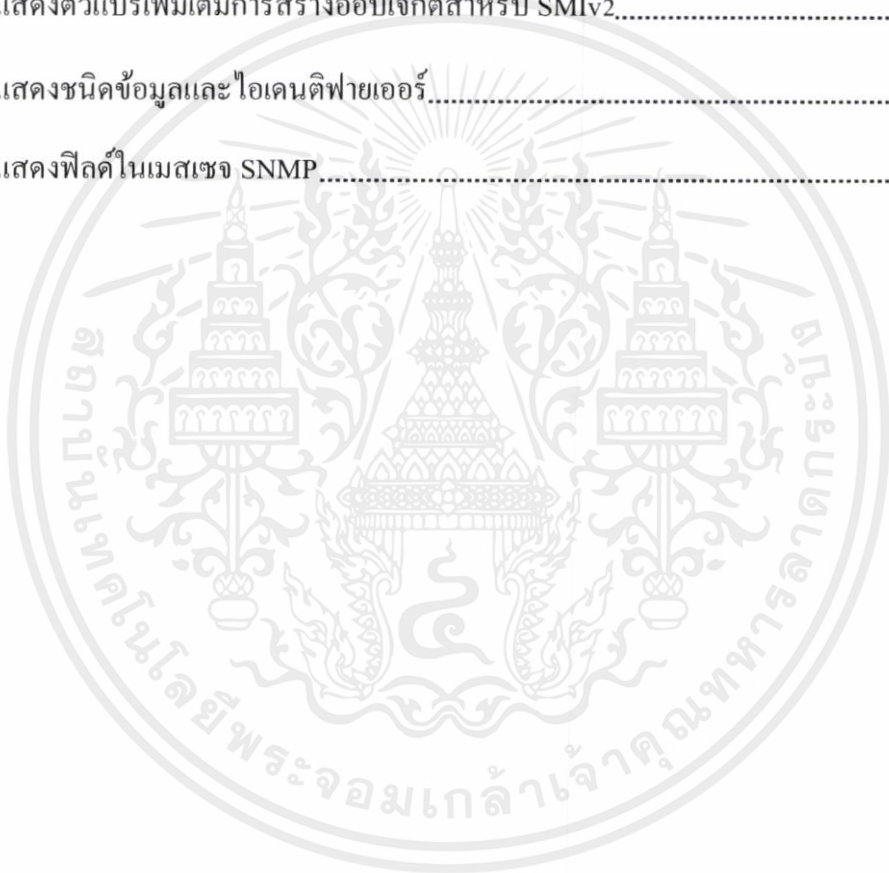
	หน้า
5.4 แนวทางในการพัฒนาต่อ.....	63
เอกสารอ้างอิง	64
ภาคผนวก	65



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงชนิดข้อมูลที่สนับสนุนโดย SMIv2.....	15
2.2 แสดงคำสั่งเพิ่มเติมสำหรับ SMIv2.....	16
2.3 แสดงตัวแปรเพิ่มเติมการสร้างออบเจกต์สำหรับ SMIv2.....	16
2.4 แสดงชนิดข้อมูลและไอเดนติไฟเออร์.....	22
2.5 แสดงฟิลด์ในเมสเซจ SNMP.....	23



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แสดงภาพสถาปัตยกรรมแอนดรอยด์.....	6
2.2 แสดง SNMP Manager และ Agent.....	9
2.3 แสดง SNMP กับ UDP.....	12
2.4 SNMP คอมมิวนิตี.....	13
2.5 โครงสร้างต้นไม้ของ SMI สำหรับ SNMPv2.....	14
2.6 การตำแหน่งออบเจกต์โดยใช้ MIB Browser.....	19
2.7 ชนิดข้อมูลไพรมิทีฟ.....	20
2.8 ชนิดข้อมูลคอมเพล็กซ์.....	21
2.9 พอร์เมตพื้นฐานของ SNMP.....	23
2.10 สถาปัตยกรรมของกูเกิลแมพ เอพีไอ.....	26
3.1 โครงสร้างการทำงานของ SNMP.....	27
3.2 Use Case Diagram.....	29
3.3 Class Diagram.....	30
3.4 Flow Chart ของ Class About.....	31
3.5 Flow Chart ของ Class DetailsPhoto.....	32
3.6 Flow Chart ของ Class Photo.....	32
3.7 Flow Chart ของ Class GetKey.....	33
3.8 Flow Chart ของ Class GetSNMP.....	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณี 3.9 ที่ Flow Chart ของ Class Logo.....และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไป 34

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.10 Flow Chart ของ Class Monitor.....	34
3.11 Flow Chart ของ Class Map.....	35
3.12 Flow Chart ของ Class SetNewKey.....	36
3.13 Flow Chart ของ Class Setting.....	36
3.14 Flow Chart ของ Class ExpandableList.....	37
3.15 Flow Chart ของ Class ServerList.....	37
3.16 GetKey Sequence Diagram.....	38
3.17 ServerList Sequence Diagram.....	39
3.18 Monitor Sequence Diagram.....	40
3.19 ExpandListAdapter Sequence Diagram.....	41
3.20 Map Sequence Diagram.....	42
3.21 DatalistsPhoto Sequence Diagram.....	43
4.1 แสดงการตั้งค่าคีย์เริ่มต้น.....	45
4.2 แสดงหน้ารับชื่อเครื่องหรือหมายเลขไอพีที่ต้องการตรวจสอบ.....	47
4.3 แสดงหน้าการเปลี่ยนชื่อเครื่องแม่ข่ายที่ต้องการ.....	49
4.4 แสดงตำแหน่งของเครื่องแม่ข่ายบน Google Map.....	51
4.5 แสดงฟังก์ชันการเข้าคู่มือเครื่องแม่ข่าย.....	53
4.6 แสดงแถบเมนูการทำงานหลัก.....	54

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.7 แสดงรายละเอียดฮาร์ดแวร์.....	56
4.8 แสดงหัวข้อของข้อมูลซอฟต์แวร์ที่สนใจ.....	57
4.9 แสดงรายชื่อซอฟต์แวร์ที่กำลังทำงานอยู่.....	58
4.10 แสดงรายชื่อซอฟต์แวร์ที่ถูกติดตั้ง.....	58
4.11 แสดงคำสั่งที่สามารถรีโมทไปยังเครื่องแม่ข่าย.....	59
4.12 แสดงรูปภาพที่ถูกถ่ายและบันทึกไว้.....	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

เนื่องจากในปัจจุบันมีการพัฒนาเทคโนโลยีโทรศัพท์มือถือมากขึ้น จากโทรศัพท์มือถือธรรมดา กลายมาเป็นสมาร์ทโฟนที่ทำงานด้วยระบบปฏิบัติการ (Operating System) โดยระบบปฏิบัติการที่มีคนใช้มากที่สุดคือ ระบบปฏิบัติการแอนดรอยด์ บริษัทต่างๆจากเดิมที่ใช้การติดต่อสื่อสารผ่านทางเว็บไซต์ (Web Application) จึงเริ่มมีการเปลี่ยนมาใช้ในการติดต่อสื่อสารผ่านทางแอปพลิเคชันบนโทรศัพท์มือถือ (Mobile Application) ซึ่งมีแอปพลิเคชันที่น่าสนใจมากมาย หนึ่งในนั้นคือ แอปพลิเคชันที่มีความสามารถในการติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ได้

จึงเป็นแนวคิดสำหรับการพัฒนาระบบตรวจสอบสภาพเครื่องคอมพิวเตอร์ในเครือข่ายที่ดูแลอยู่ ผ่านอุปกรณ์ที่มีระบบปฏิบัติการแอนดรอยด์ โดยมีหลักการทำงานคือสามารถตรวจสอบและแสดงผลเกี่ยวกับฮาร์ดแวร์เครื่องคอมพิวเตอร์นั้นๆ ได้แก่ รายละเอียดของ CPU การอัปเดตระบบครั้งล่าสุด การแบ่งฮาร์ดดิสก์ ขนาดของแรมที่เครื่องคอมพิวเตอร์ใช้อยู่ จำนวน โพรเซสส์ที่กำลังประมวลผลอยู่ และขนาดของหน่วยความจำที่เหลือในขณะนั้น ส่วนการตรวจสอบและแสดงผลซอฟต์แวร์ โปรแกรมสามารถแสดงผลข้อมูลซอฟต์แวร์ภายในเครื่องได้ กล่าวคือ สามารถแสดงโปรแกรมที่ถูกติดตั้งลงในเครื่องคอมพิวเตอร์ และสามารถแสดงโปรแกรมที่ถูกเรียกใช้งานอยู่ในขณะนั้นพร้อมทั้งสามารถสั่งปิดการทำงานของโปรแกรมที่กำลังทำงานอยู่ได้ และยังมีความสามารถในการสั่งมอนิเตอร์ควบคุมเครื่องให้สามารถปิดเครื่อง รีสตาร์ทเครื่อง และเข้าสู่โหมดต่างๆได้ และโปรแกรมยังสามารถส่งถ่ายภาพเพื่อบันทึกภาพเครื่องคอมพิวเตอร์นั้นๆได้ โดยเครื่องคอมพิวเตอร์ในเครือข่ายและอุปกรณ์แอนดรอยด์อยู่ภายในเน็ตเวิร์คเดียวกัน ซึ่งแอปพลิเคชันนี้จะช่วยให้สามารถตรวจสอบข้อมูลเครื่องคอมพิวเตอร์ในเครือข่ายทั้งหมดได้ โดยไม่ต้องอยู่หน้าเครื่อง ซึ่งจะมีประโยชน์มากในกรณีที่เครื่องแม่ข่ายคอมพิวเตอร์ในเครือข่ายหลายๆเครื่องถูกตั้งรวมกันอยู่ใน Data Center ซึ่งการจะเข้าไปทำงานหน้าเครื่องจะต้องผ่านกระบวนการรักษาความปลอดภัยมากมาย แอปพลิเคชันนี้ก็จะช่วยให้ประหยัดเวลา มีความปลอดภัยสูง และยังมีฟังก์ชันที่สามารถแสดงตำแหน่งของเครื่องแต่ละเครื่องในรูปแบบแผนที่ ผ่านทาง Google Map ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อศึกษาการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ ให้รองรับการทำงานหลักของผู้ใช้
- 2) เพื่อศึกษาการเขียน โปรแกรมโดยใช้ภาษา Java
- 3) เพื่อศึกษาการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์และอุปกรณ์แอนดรอยด์
- 4) เพื่อพัฒนาแอปพลิเคชันให้สามารถตรวจสอบสภาพเครื่องคอมพิวเตอร์ในเครือข่ายผ่านอุปกรณ์แอนดรอยด์ได้

1.3 ขอบเขตของโครงการ

แอปพลิเคชันสามารถตรวจสอบสภาพเครื่องคอมพิวเตอร์ในเครือข่าย โดยแบ่งการแสดงผลออกเป็น 4 ส่วน คือ ส่วนแสดงข้อมูลฮาร์ดแวร์ (Hardware) ส่วนแสดงข้อมูลซอฟต์แวร์ (Software) ส่วนควบคุมเครื่องคอมพิวเตอร์ในเครือข่าย (Remote Server) และส่วนแสดงรูปภาพ (Gallery)

- 1) ส่วนแสดงผลข้อมูลฮาร์ดแวร์ประกอบด้วย
 - 1.1) แสดงรายละเอียดของฮาร์ดแวร์ (Hardware System Description)
 - 1.2) ชื่อเครื่องแม่ข่าย (System Name)
 - 1.3) ประสิทธิภาพการทำงานของฮาร์ดดิสก์ (Hard disk Performance) ได้แก่ พื้นที่ของฮาร์ดดิสก์ทั้งหมด (Hard disk Space) พื้นที่ของฮาร์ดดิสก์ที่ใช้ไป (Hard disk use space)
 - 1.4) แสดงค่า RAM ของเครื่องแม่ข่าย
 - 1.5) แสดงค่า Processes ที่เครื่องแม่ข่ายใช้อยู่
 - 1.6) แสดงค่า Memory ที่เครื่องแม่ข่ายใช้อยู่
- 2) ส่วนแสดงผลข้อมูล Software ประกอบด้วย
 - 2.1) แสดงรายชื่อซอฟต์แวร์ที่กำลังทำงานอยู่ (Running Software)
 - 2.2) แสดงรายชื่อซอฟต์แวร์ที่ได้ทำการ Install ไว้ (Installed Software)
 - 2.3) สามารถสั่งหยุดการทำงานของโปรแกรมที่กำลังทำงานอยู่
- 3) ส่วนควบคุมเครื่องคอมพิวเตอร์ในเครือข่าย
 - 3.1) สามารถสั่งปิดเครื่อง, รีสตาร์ท, Lock off และ Sleep
- 4) ส่วนแสดงรูปภาพเครื่องคอมพิวเตอร์นั้นๆ
 - 4.1) สามารถถ่ายภาพเพิ่มเข้าไปในโปรแกรมได้และสามารถลบรูปภาพได้

- 5) สามารถรองรับเครื่องภายในเครือข่ายได้ถึง 10 เครื่อง
- 6) รองรับระบบปฏิบัติการแอนดรอยด์ 4.1 เป็นต้นไป

1.4 วิธีการดำเนินงาน

- 1) กำหนดขอบเขตของโครงการ
- 2) ศึกษาเกี่ยวกับทฤษฎีที่เกี่ยวข้อง เช่น การทำงานของระบบปฏิบัติการแอนดรอยด์ การเขียนโปรแกรมเพื่อติดต่อกับเครื่องแม่ข่ายคอมพิวเตอร์
- 3) ออกแบบโครงสร้างโดยรวมของระบบและส่วนติดต่อกับผู้ใช้ (User Interface Design)
- 4) ศึกษาและพัฒนาระบบ
- 5) ทำการทดลองเพื่อสนับสนุนทฤษฎีที่เกี่ยวข้องและหาข้อจำกัดของระบบ
- 6) สรุปผลที่ได้จากการทดลองรวมถึงปัญหาและอุปสรรคที่พบ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รับความรู้และความเข้าใจเกี่ยวกับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์
- 2) ได้รับความรู้ความเข้าใจเกี่ยวกับการเขียนโปรแกรมโดยใช้ภาษา Java
- 3) ได้รับความรู้เกี่ยวกับการเขียนโปรแกรมเพื่อติดต่อกับเครื่องคอมพิวเตอร์ภายในเครือข่าย
- 4) แอปพลิเคชันที่พัฒนาขึ้นมาสามารถนำไปใช้ประโยชน์ได้จริงและสามารถนำไปประยุกต์ใช้ในงานที่เกี่ยวข้องได้
- 5) เกิดระบบต้นแบบที่สามารถพัฒนาต่อยอดในการใช้งานในลักษณะของการติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ภายในเครือข่ายได้

1.6 ส่วนประกอบของปฏิญญานิพนธ์

ปฏิญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บท คือ

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของโครงการ วิธีการดำเนินการ และส่วนประกอบของปฏิญญานิพนธ์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงการ ประกอบด้วย เรื่องของระบบปฏิบัติการแอนดรอยด์ (Android) SNMPv2 และ Google Map API

บทที่ 3 การออกแบบและการพัฒนา กล่าวถึงรายละเอียดของ เครื่องมือที่ใช้ในการ พัฒนาภาพรวมของแอปพลิเคชัน กระบวนการทำงานของโปรแกรมโดยละเอียด

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการทำงานของแอปพลิเคชันบน อุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์

บทที่ 5 บทสรุปและข้อเสนอแนะ กล่าวถึงบทสรุปของโครงการ วิจัยสิ่งที่ ได้รับจากโครงการ ข้อจำกัด รวมถึงปัญหาอุปสรรคต่าง ๆ ของโครงการ และข้อเสนอแนะ สำหรับเป็นแนวทางในการพัฒนาต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบปฏิบัติการแอนดรอยด์

2.1.1 ประวัติความเป็นมา

แอนดรอยด์ถือกำเนิดอย่างเป็นทางการในวันที่ 5 พฤศจิกายน 2550 โดยบริษัท กูเกิล จุดประสงค์ของแอนดรอยด์มีจุดเริ่มต้นมาจากบริษัท Android Inc. ที่ได้นำเอาระบบปฏิบัติการลินุกซ์ (Linux) ซึ่งนิยมนำไปใช้งานกับเครื่องแม่ข่าย (Server) เป็นหลัก นำมาดัดทอนขนาดตัว (แต่ไม่ลดทอนความสามารถ) เพื่อให้เหมาะสมแก่การนำไปติดตั้งบนอุปกรณ์พกพา ที่มีขนาดพื้นที่จัดเก็บข้อมูลที่จำกัด โดยหวังว่า แอนดรอยด์ นั้นจะเป็นหุ่นยนต์ตัวน้อย ๆ ที่คอยช่วยเหลืออำนวยความสะดวกแก่ผู้ที่พกพามัน ไปในทุกที่ ทุกเวลา

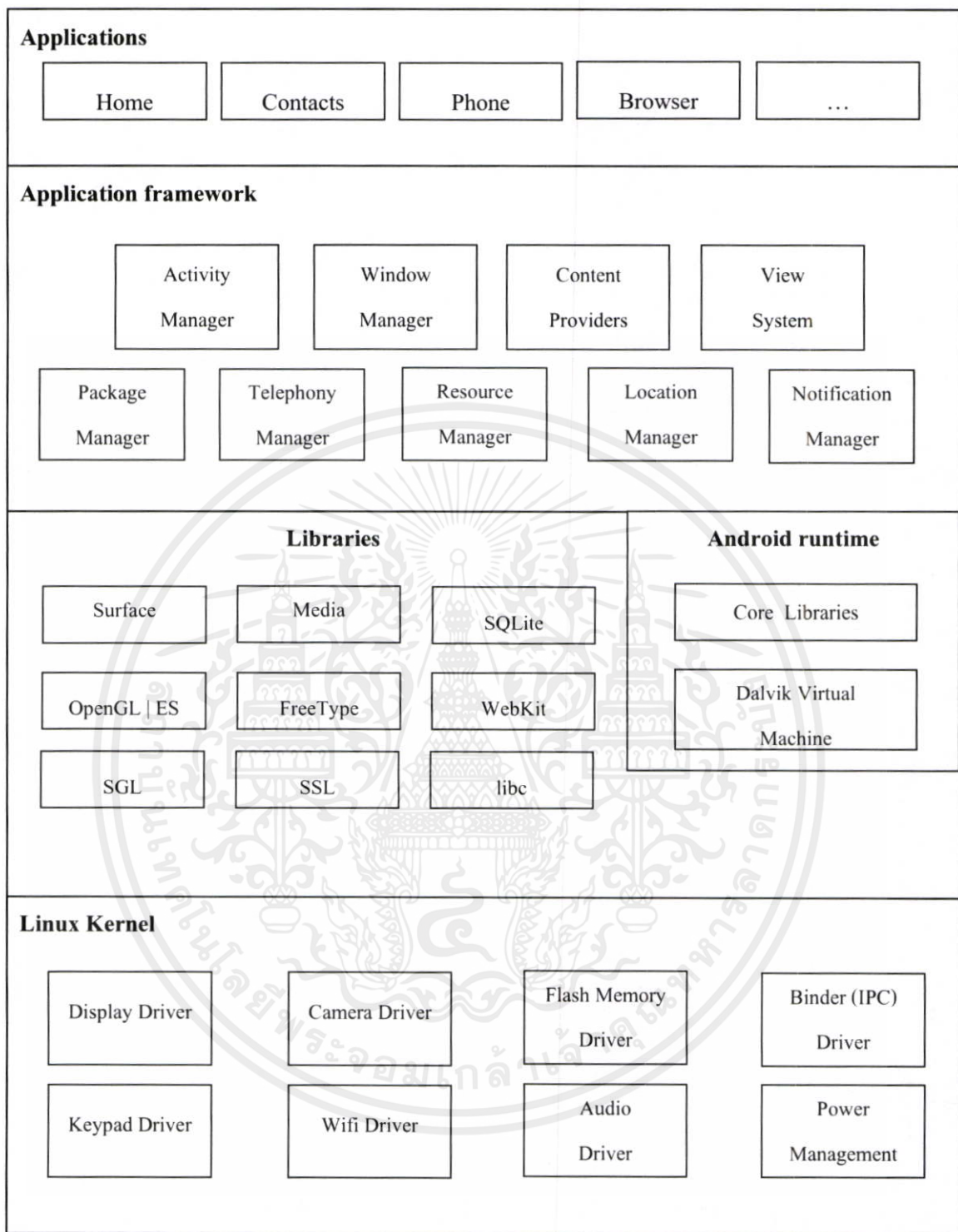
กูเกิลแอนดรอยด์ เป็นชื่อเรียกอย่างเป็นทางการของแอนดรอยด์ เนื่องจากปัจจุบันนี้ บริษัทกูเกิล เป็นผู้ที่ถือสิทธิบัตรในตราสัญญาถักขันธ์ ชื่อ และ รหัสต้นฉบับ (Source Code) ของแอนดรอยด์ ภายใต้เงื่อนไขการพัฒนาแบบ GNU โดยเปิดให้นักพัฒนา (Developer) สามารถนำรหัสต้นฉบับ ไปพัฒนาปรับแต่งได้อย่างเปิดเผย (Open source) ทำให้แอนดรอยด์มีผู้เข้าร่วมพัฒนาเป็นจำนวนมาก และพัฒนาไปได้อย่างรวดเร็ว

2.1.2 สถาปัตยกรรมแอนดรอยด์

แอนดรอยด์เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อนหรือแบบสแต็ก (Stack) ซึ่งรวมเอาระบบปฏิบัติการ (OperatingSystem), มิดเดิลแวร์ (Middleware) และแอปพลิเคชันที่สำคัญเข้าไว้ด้วยกัน เพื่อใช้สำหรับทำงานบนอุปกรณ์พกพาเคลื่อนที่ (MobileDevices) เช่น โทรศัพท์มือถือ เป็นต้น

การทำงานของแอนดรอยด์มีพื้นฐานอยู่บนระบบลินุกซ์เคอร์เนล (LinuxKemel) ซึ่งใช้ AndroidSDK (SoftwareDevelopmentKit) เป็นเครื่องมือสำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android และใช้ภาษา Java ในการพัฒนาสถาปัตยกรรมของแอนดรอยด์ (AndroidArchitecture) นั้นถูกแบ่งออกเป็นลำดับชั้นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงภาพสถาปัตยกรรมแอนดรอยด์

2.1.3 ชนิดของโปรแกรมประยุกต์

1) **Foreground Activity** คือ โปรแกรมประยุกต์ที่จะทำงานและติดต่อกับผู้ใช้ในขณะที่มองเห็นได้เท่านั้น (บน Foreground) ซึ่งเมื่อไม่เห็นโปรแกรมหรือการปิดโปรแกรม ไม่ว่าจะฉับใดๆ ทั้งสิ้น ถูกทงห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ก็สิ้นสุดการทำงาน ผู้ต่างๆ ได้ว่าทำงานเมื่อมองเห็นได้เท่านั้น

2) Background Service คือ โปรแกรมประยุกต์ที่มีการติดต่อกับผู้ใช้น้อยมาก อาจมีบ้างในการตั้งค่าการทำงาน และวงจรการทำงานส่วนใหญ่ของโปรแกรมชนิดนี้จะซ่อนตัวอยู่นั้นคือการทำงานโดยที่เราไม่เห็นหน้าต่างของโปรแกรม เช่น โปรแกรมการแสดงผลภาพของผู้ติดต่อที่โทรเข้าหรือโทรออก หรือโปรแกรมในการส่งข้อความอัตโนมัติ (การทำงานจะทำงานโดยที่เราไม่จำเป็นต้องเห็น ถ้าโดนปิดไป เมื่อมันเริ่มทำงานใหม่เราก็ไม่รู้เพราะทำงานอยู่ที่ background)

3) Intermittent Activity คือ โปรแกรมประยุกต์ที่มีการติดต่อกับผู้ใช้น้อยบ้าง แต่การทำงานส่วนใหญ่จะทำในbackground ซึ่งโดยส่วนใหญ่แล้วโปรแกรมเหล่านี้จะทำงานอย่างเงียบๆ และอาจมีการแจ้งเตือนเมื่อมีความเคลื่อนไหวของโปรแกรม ตัวอย่างที่เห็นได้ชัดคือโปรแกรม media player ซึ่งอาจมีการติดต่อกับผู้ใช้น้อยบ้าง เช่น เลือกเพลงต่างๆ จากนั้นเมื่อผู้ใช้ปิดไป จะทำการซ่อนตัวเองไปทำงานในชั้นbackground

2.1.4 เครื่องมือที่สำหรับการพัฒนาแอปพลิเคชันบนระบบแอนดรอยด์

1) ชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรมจาวา (JDK: Java Development Kit) ชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรมจาวาหรือ JDK คือชุดของเครื่องมือ(tools) ที่ใช้ในการพัฒนาโปรแกรมจาวา ของบริษัทซันไมโครซิสเต็มส์ ซึ่งใครก็ตามที่ต้องการจะพัฒนาโปรแกรมโดยใช้ภาษาจาวา อย่างเช่น Java compiler, Java debugger, Java doc และ Java interpreter หรือ Java VM จะต้องติดตั้งชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรมจาวานี้ ไม่เช่นนั้นจะไม่สามารถ compile และ run java ได้ เวอร์ชันปัจจุบันของชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรมจาวาคือเวอร์ชัน 8 ประกอบไปด้วยโปรแกรมต่างๆ อาทิเช่น โปรแกรมคอมไพเลอร์ (javac.exe) ,โปรแกรมอินเตอร์พรีเตอร์ (java.exe) ,โปรแกรมดีบั๊กเกอร์ แต่จะไม่มีโปรแกรมอิดีเตอร์

2) อีคลิป์ไอดีอี (Eclipse IDE : Eclipse Integrated Development Environment) อีคลิป์เป็นเครื่องมือที่ช่วยในการพัฒนาโปรแกรมโดยมีสิ่งอำนวยความสะดวกต่างๆ เช่น คำสั่ง การคอมไพล์ หรือคำสั่งการรันโปรแกรม ตัวอย่างของ เครื่องมือที่ช่วยในการพัฒนาโปรแกรมเช่นโปรแกรม NetBeans Editplus, JCreator, Eclipse ซึ่งเครื่องมือที่ช่วยในการพัฒนาโปรแกรมเป็นโปรแกรมที่รวมคำสั่ง เมนู และส่วนต่อประสานกราฟิกกับผู้ใช้ (graphical user interface) ต่างๆมาสร้างเป็นโปรแกรมที่มีรูปร่างหน้าต่างเหมาะสมแก่การพัฒนาโปรแกรม หรือ เป็นโปรแกรมที่จะสร้างสภาพแวดล้อม (Environment) ให้เหมาะแก่การพัฒนาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การพัฒนาโปรแกรม

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ซอฟต์แวร์สำหรับพัฒนาแอนดรอยด์ (Android SDK)

ซอฟต์แวร์สำหรับพัฒนาแอนดรอยด์เป็นโปรแกรมสำหรับนักพัฒนาที่เปิดให้เข้ามาพัฒนาแอปพลิเคชัน ซึ่งทางบริษัทกูเกิลได้เปิดให้ดาวน์โหลดซอฟต์แวร์สำหรับพัฒนาแอนดรอยด์ฟรีเพื่อให้เกิดแอปพลิเคชันใหม่ๆเพิ่มขึ้น ภายในชุดพัฒนาแอนดรอยด์มีส่วนจำลองการทำงานทำให้ในระหว่างการพัฒนาแอปพลิเคชันสามารถจำลองการทำงานของมือถือแอนดรอยด์ และทดลองใช้งานแอปพลิเคชันที่ได้ทำการออกแบบโดยไม่ต้องมีโทรศัพท์จริง

4) เครื่องมือพัฒนาแอนดรอยด์ (ADT : Android Development Tools)

เป็นปลั๊กอินใน IDE สำหรับเพิ่มความสามารถในการพัฒนา เป็นเครื่องมือที่ใช้พัฒนาแอนดรอยด์ ในการพัฒนาแอปพลิเคชันบนระบบ Android OS จะใช้ภาษาจาวาโดยต้องติดตั้งส่วนเสริม คือเครื่องมือพัฒนาแอนดรอยด์ (ADT) ซึ่งเป็นส่วนเสริมของ ปลั๊กอินของโปรแกรม Eclipse ใช้ในการพัฒนาโปรแกรม (IDE)

2.2 Simple Network Management Protocol (SNMP)

ปัจจุบันนี้ระบบเครือข่ายมีความซับซ้อนมากขึ้น และมีความวิกฤติมากขึ้น การบริหารจัดการระบบเครือข่ายไม่ใช่แค่เพียงติดตั้ง และใช้งาน แต่ต้องมีการตรวจสอบเฝ้าระวังประสิทธิภาพการทำงานเพื่อทำการบำรุงรักษาให้ระบบทำงานได้อย่างมีประสิทธิภาพและต่อเนื่อง การลดลงหรือถดถอยของประสิทธิภาพการทำงานของระบบเครือข่ายในบางระบบงานอาจทำให้เกิดความเสียหายมูลค่าสูง ทั้งทางการเงิน หรือทรัพย์สิน รวมทั้งชีวิตได้ เช่น ระบบจำหน่ายไฟฟ้า หรือระบบท่อส่งแก๊สและน้ำมัน และนั่นคือที่มาของความต้องการ โพรโทคอลในการบริหารจัดการระบบเครือข่าย หรือ โพรโทคอล SNMP ซึ่งอยู่ในชุดโพรโทคอล TCP/IP (Transmission Control Protocol/Internet Protocol)

โพรโทคอล SNMP ได้ถูกพัฒนาขึ้นในปี พ.ศ.2531 เนื่องจากมีความเจริญเติบโตในการใช้อุปกรณ์ที่สนับสนุนโพรโทคอล TCP/IP อย่างสูง โพรโทคอล SNMP ถูกออกแบบให้มีฟังก์ชันและการทำงานแบบง่าย ๆ เหมาะกับคำว่าซิมเปิล (Simple) ตามชื่อของมัน โดยมีจุดประสงค์หลักเพื่อให้ผู้ดูแลระบบเครือข่ายสามารถเข้ามาจัดการอุปกรณ์เครือข่ายได้จากระยะไกลโดยง่าย

2.2.1 ข้อมูลทั่วไปเกี่ยวกับเวอร์ชันของ SNMP

SNMP Version 1 (SNMPv1) เป็นมาตรฐานปัจจุบันและเป็นที่ยอมรับเพราะความง่ายของโพรโทคอล SNMP ซึ่งถูกระบุใน RFC1157 และได้รับอนุมัติให้เป็นมาตรฐานที่สมบูรณ์ ระดับความปลอดภัย SNMPv1 จะขึ้นอยู่กับคอมมูนิตีสตริง (Community String) ที่ทำหน้าที่เหมือนรหัสผ่าน หรือพาสเวิร์ด (Password) โดยที่จริงแล้วเป็นเพียงข้อความ

แบบธรรมดา (Plain Text) ที่บ่งบอกถึงสิทธิการเข้าไปจัดการอุปกรณ์เครือข่าย โดยปกติคอมมิวนิตีจะมีสามประเภทนั่นคือ อ่านอย่างเดียว (Read-only), อ่านเขียน (Read-write) และแตรป (Trap)

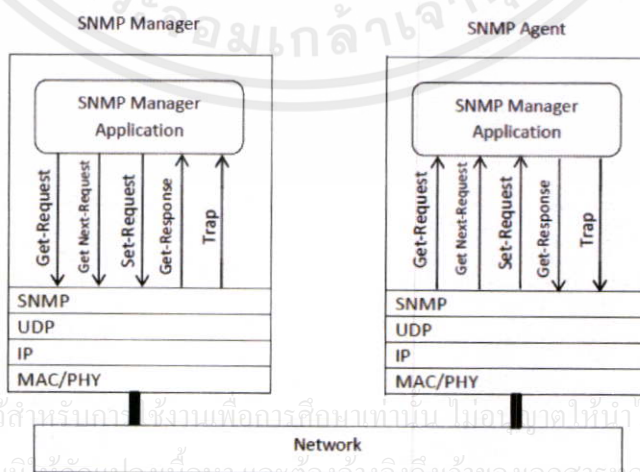
SNMP Version 2 (SNMPv2) คือ เวอร์ชันที่ทำงานบนคอมมิวนิตีที่ได้รับการปรับปรุง ในทางเทคนิคเรียกว่า SNMPv2c ซึ่งระบุใน RFC1905, RFC1906 และ RFC1907 และอยู่ในขั้นตอนทดสอบใช้งาน แต่ก็มีบางผู้ผลิตได้นำมาใช้งานในอุปกรณ์ของพวกเขา SNMPv2 ออกแบบมาเพื่อแก้ไขข้อด้อยของ SNMPv1 ในเรื่องการร้องข้อมูลปริมาณมาก และปัญหาในการส่งข้อมูลแบบแตรป

SNMP Version 3 (SNMPv3) เป็นเวอร์ชันถัดไปของโปรโตคอล SNMP ที่ถูกคาดหวังให้เป็นมาตรฐานที่สมบูรณ์ ซึ่งในปัจจุบันอยู่ในสถานะนำเสนอระบุใน RFC1905, RFC1906, RFC1907, RFC2571, RFC2572, RFC2573, RFC2574 และ RFC2575 โดยมุ่งเน้นการเพิ่มระดับความปลอดภัยของโปรโตคอล SNMP

2.2.2 เมนเนเจอร์และเอเจนต์ (Manager and Agent)

อันที่จริงแล้วทางเทคนิค อุปกรณ์เครือข่ายที่สนับสนุนโปรโตคอล SNMP แบ่งเป็น 2 ประเภทเท่านั้นคือ เมนเนเจอร์ และเอเจนต์

ตัวเมนเนเจอร์โดยทั่วไป คือเซิร์ฟเวอร์ที่รันซอฟต์แวร์ หรือโปรแกรมประยุกต์สำหรับการบริหารจัดการระบบเครือข่าย บ่อยครั้งที่เมนเนเจอร์ ถูกเรียกว่า NMS (Network Management Stations) เมนเนเจอร์มีหน้าที่ร้องขอ (Request บางครั้งเรียกว่า Query) หรือโพลลิ่ง (Polling) หรือรับข้อมูลประเภทแตรป (Trap) ที่ถูกส่งจากตัวเอเจนต์โดยไม่ได้รับร้องขอ



รูปที่ 2.2 SNMP Manager และ Agent

การ โพลล์ คือการร้องขอข้อมูลจากเอเจนต์ เช่น จากเราเตอร์ หรือสวิตช์ ข้อมูลเหล่านี้สามารถใช้ประเมินสภาพการทำงานของระบบเครือข่าย ส่วนแทรก คืออีกวิธีการหนึ่งสำหรับเอเจนต์ในการส่งสัญญาณเตือนไปยังเมนเจอร์ ว่ามีเหตุการณ์สำคัญใดเกิดขึ้น แทรกจะสามารถจะถูส่งเมื่อไรก็ได้โดยไม่ต้องรอการร้องขอจากเมนเจอร์ ตัวเมนเจอร์จะมีการตอบสนองต่อแทรกตามการตั้งค่าของผู้ดูแลระบบเครือข่าย สำหรับตัวอย่าง เมื่อวงจรสื่อสาร T1 เกิดขัดข้อง เราเตอร์สามารถส่งแทรกไปยังเมนเจอร์ หลังจากนั้นเมนเจอร์จะส่ง SMS แจ้งเตือนผู้ดูแลระบบเครือข่ายอีกที ว่ามีวงจรสื่อสาร T1 ขัดข้องเกิดขึ้นในระบบเครือข่ายเป็นต้น

ประเภทอุปกรณ์ชนิดที่สองคือ เอเจนต์ โดยทั่วไปคือ โปรแกรม หรือเฟิร์มแวร์ (Firmware) ที่ติดตั้ง และทำงานบนตัวอุปกรณ์เครือข่ายที่ผู้ดูแลระบบเครือข่ายต้องการจัดการ ซึ่งอาจจะเป็น โปรแกรมเฉพาะ และทำงานเบื้องหลังเป็นแบ็กกราวด์โปรเซส (Background Process) หรือเดมอน (Daemon) เช่น ในไมโครซอฟต์วินโดว หรือยูนิกซ์ หรือเป็นส่วนหนึ่งในระบบปฏิบัติการ เช่น ในเราเตอร์ของ CISCO ซึ่งเป็นเฟิร์มแวร์ระดับต่ำ

ในปัจจุบันอุปกรณ์ที่ใช้โปรโตคอล TCP/IP ส่วนใหญ่จะมาพร้อมกับตัวเอเจนต์ SNMP เพียงแต่อาจต้องมีการเปิดใช้งาน เพราะอันที่จริงแล้วผู้ผลิตต้องการให้ผลิตภัณฑ์ของตนเองง่ายต่อการบำรุงรักษา เอเจนต์จะให้ข้อมูลการทำงานของอุปกรณ์เพื่อการจัดการและบำรุงรักษา สำหรับตัวอย่าง เอเจนต์บนเราเตอร์สามารถบ่งบอกสถานะของแต่ละพอร์ตว่าทำงานปกติ หรือไม่ปกติ โดยเมนเจอร์สามารถร้องขอข้อมูลดังกล่าวผ่านโปรโตคอล SNMP หรืออีกวิธีหนึ่ง ถ้าเกิดมีการขัดข้องรุนแรงเกิดขึ้นในตัวอุปกรณ์ และเอเจนต์ตรวจสอบพบ เอเจนต์จะส่งข้อมูลประเภทแทรกแจ้งไปยังเมนเจอร์โดยไม่ต้องมีการร้องขอ และหลังจากนั้น เมนเจอร์จะดำเนินการต่อข้อมูลดังกล่าวตามการตั้งค่า หรือเซตติ้ง ในบางอุปกรณ์ยังส่งแทรกแจ้งเมนเจอร์เมื่อเหตุที่ขัดข้องได้กลับคืนสภาพปกติ (All Clear) ด้วย ซึ่งก็เป็นวิธีการที่ดี ทำให้สามารถตรวจสอบแก้ไขได้ง่าย และเร็วขึ้น ดังรูปที่ 1 ได้แสดงความสัมพันธ์ระหว่าง เมนเจอร์ และเอเจนต์

สิ่งที่สำคัญที่ควรรับทราบอย่างหนึ่งก็คือ การร้องขอ และการส่งแทรก สามารถเกิดขึ้นได้ในเวลาเดียวกัน ดังนั้นจึงไม่มีปัญหา ถ้ามีการร้องขอข้อมูลในขณะที่กำลังรับแทรกจากตัวเอเจนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 SNMP กับ UDP

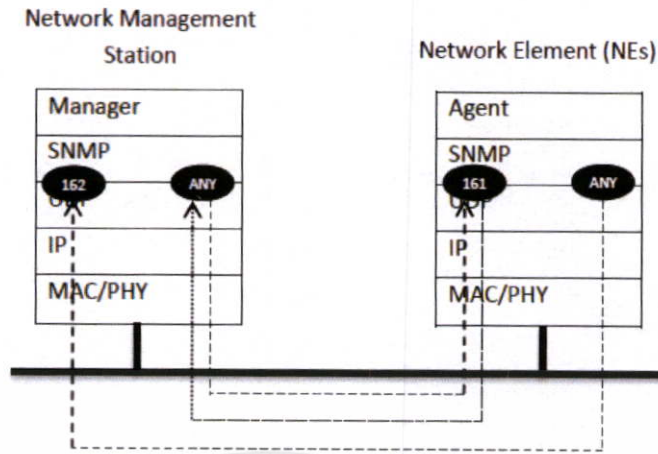
โพรโทคอล SNMP ใช้ UDP เป็นโพรโทคอลในการส่ง และรับข้อมูลระหว่างตัวเมนเจอร์ และเอเจนต์ ถ้ามองว่าทำไมโพรโทคอล UDP ถูกเลือกใช้แทนที่จะเป็น TCP คำตอบก็เพราะว่าโพรโทคอล UDP ใช้การเชื่อมต่อแบบคอนเนกชันเลส (Connectionless) ซึ่งไม่มีการเชื่อมต่อสื่อสารหรือทำแฮนด์เช็กกิ้งก่อนที่จะรับและส่งข้อมูลระหว่างเมนเจอร์และเอเจนต์ ลักษณะการทำงานของโพรโทคอล UDP จริง ๆ มีระดับความเชื่อถือได้ที่ไม่สูง เพราะไม่มีการตอบรับแพ็กเกจถ้าเกิดการสูญหายระหว่างทาง

ดังนั้นจึงตกเป็นหน้าที่ระดับแอปพลิเคชันที่ต้องประเมินว่าแพ็กเกจสูญหายหรือไม่ หรือต้องทำการส่งการร้องขอใหม่หรือไม่ วิธีที่ใช้ทั่วไปคือการใช้ไทมเมอร์ หรือ ไทม์เอาต์ (Timeout) เมนเจอร์ที่ส่งเมสเสจส่งออกไปยังตัวเอเจนต์จะคอยการตอบสนองในระยะเวลาหนึ่ง ซึ่งระยะเวลานั้นขึ้นอยู่กับค่าของผู้ดูแลระบบเครือข่าย ถ้าหมดระยะเวลาการรอคอยแล้วไม่มีการตอบสนองใด ๆ จากเอเจนต์ เมนเจอร์จะประเมินว่าแพ็กเกจเกิดการสูญหาย และจะทำการส่งเมสเสจการร้องขออีกครั้ง จำนวนครั้งที่การร้องขอซ้ำนั้นขึ้นอยู่กับค่าเช่นกัน

อันที่จริงแล้วปัญหาที่เกิดจากตัวโพรโทคอล UDP จากการส่งเมสเสจส่งขอข้อมูลจากเอเจนต์นั้นไม่ได้เป็นประเด็นเลย แต่ที่เป็นประเด็นแท้จริง ก็คือการส่งแทรปจากเอเจนต์ไปยังเมนเจอร์ เนื่องจากแทรปสามารถสูญหายได้โดยที่เมนเจอร์ไม่ได้รับทราบใด ๆ เนื่องจากระบบการทำงานของแทรปไม่ได้ออกแบบให้ยืนยันการได้รับแทรป จึงไม่มีการบวนการส่งแทรปซ้ำ จึงได้เกิดความคิดที่จะใช้ TCP มาแก้ไขปัญหาดังกล่าว

โพรโทคอล SNMP ใช้พอร์ต UDP หมายเลข 161 สำหรับส่งและรับแบบการร้องขอข้อมูล และใช้พอร์ตหมายเลข 162 สำหรับรับเมสเสจชนิดแทรปจากเอเจนต์ของอุปกรณ์เครือข่ายที่ถูกจัดการ ทุกๆ อุปกรณ์เครือข่ายต้องใช้พอร์ตหมายเลขดังกล่าวเป็นหมายเลขดีฟอลต์ แต่อย่างไรก็ตามบางผู้ผลิตอนุญาตให้เปลี่ยนหมายเลขพอร์ต ซึ่งเมนเจอร์ต้องรับทราบ และเปลี่ยนหมายเลขพอร์ตให้ตรงกับอุปกรณ์เครือข่ายเพื่อให้สามารถติดต่อรับส่งเมสเสจได้

รูปแสดงชุดโพรโทคอล TCP/IP ที่เป็นพื้นฐานของการสื่อสารข้อมูลโดยใช้หมายเลข IP ปัจจุบันอุปกรณ์ที่ต้องการเชื่อมต่ออินเทอร์เน็ตจำเป็นต้องใช้โพรโทคอล TCP/IP เท่านั้น โมเดลด้านล่างแสดงการใช้โพรโทคอลสแต็กของ SNMP ซึ่งแต่ละเลเยอร์จะใช้ข้อมูลจากเลเยอร์ด้านล่างมาประมวลผลและให้บริการต่อ เลเยอร์ด้านบน



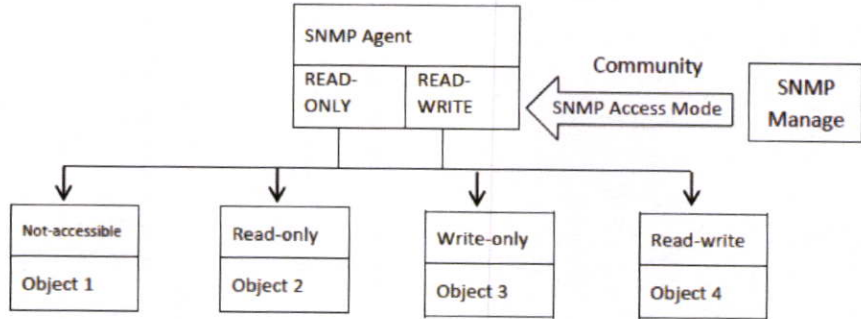
รูปที่ 2.3 แสดง SNMP กับ UDP

2.2.4 SNMP คอมมูนิตี (SNMP Communities)

สำหรับ SNMPv1 และ SNMPv2 ที่เป็นที่นิยม จะใช้ระบบคอมมูนิตีในการสร้างความปลอดภัยในการรับส่งข้อมูลระหว่างเมนเจอร์ และเอเจนต์ โดยทั่วไปเอเจนต์จะถูกตั้งค่าให้มีคอมมูนิตี 3 ประเภทโดยวิธีการตั้งชื่อ นั่นคือ อ่านได้อย่างเดียว สามารถอ่านเขียนและแทรก ชื่อคอมมูนิตี หรือคอมมูนิตีสตริง อันที่จริงทำงานเสมือนเป็นรหัสผ่าน โดยผู้ผลิตทั่วไปจะให้คอมมูนิตีสตริง ชื่อพับลิก (Public) สำหรับการอ่านได้อย่างเดียว คอมมูนิตีสตริง ชื่อไพรเวต (Private) สำหรับการอ่านและเขียน หรือเซตตั้งค่า เป็นสิ่งที่ดีที่จะเปลี่ยนชื่อคอมมูนิตีสตริงที่เป็นดีฟอลต์ให้เป็นชื่อเฉพาะเพื่อเพิ่มระดับความปลอดภัย การสร้างแทรกเพื่อแจ้งผู้ดูแลระบบเครือข่ายเมื่อมีการพยายามเข้ามาตั้งค่า หรือร้องขอข้อมูลจากตัวเอเจนต์ด้วยชื่อคอมมูนิตีสตริงที่ไม่ถูกต้อง หรือไม่ตรงกับที่กำหนดก็เป็นสิ่งที่ดี ก็เพราะเป็นการแจ้งเตือนว่าอาจเกิดมีผู้ไม่หวังดีพยายามเข้ามาเจาะระบบเครือข่าย

เนื่องจากชื่อคอมมูนิตีสตริงเสมือนเป็นรหัสผ่านของระบบจัดการเครือข่าย ดังนั้นการตั้งชื่อคอมมูนิตีสตริงก็ควรตั้งตามกฎการตั้งรหัสผ่านของเซิร์ฟเวอร์ เช่น ไม่เป็นคำในพจนานุกรม คำควรมีทั้งตัวเล็กตัวใหญ่รวมทั้งตัวเลข เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



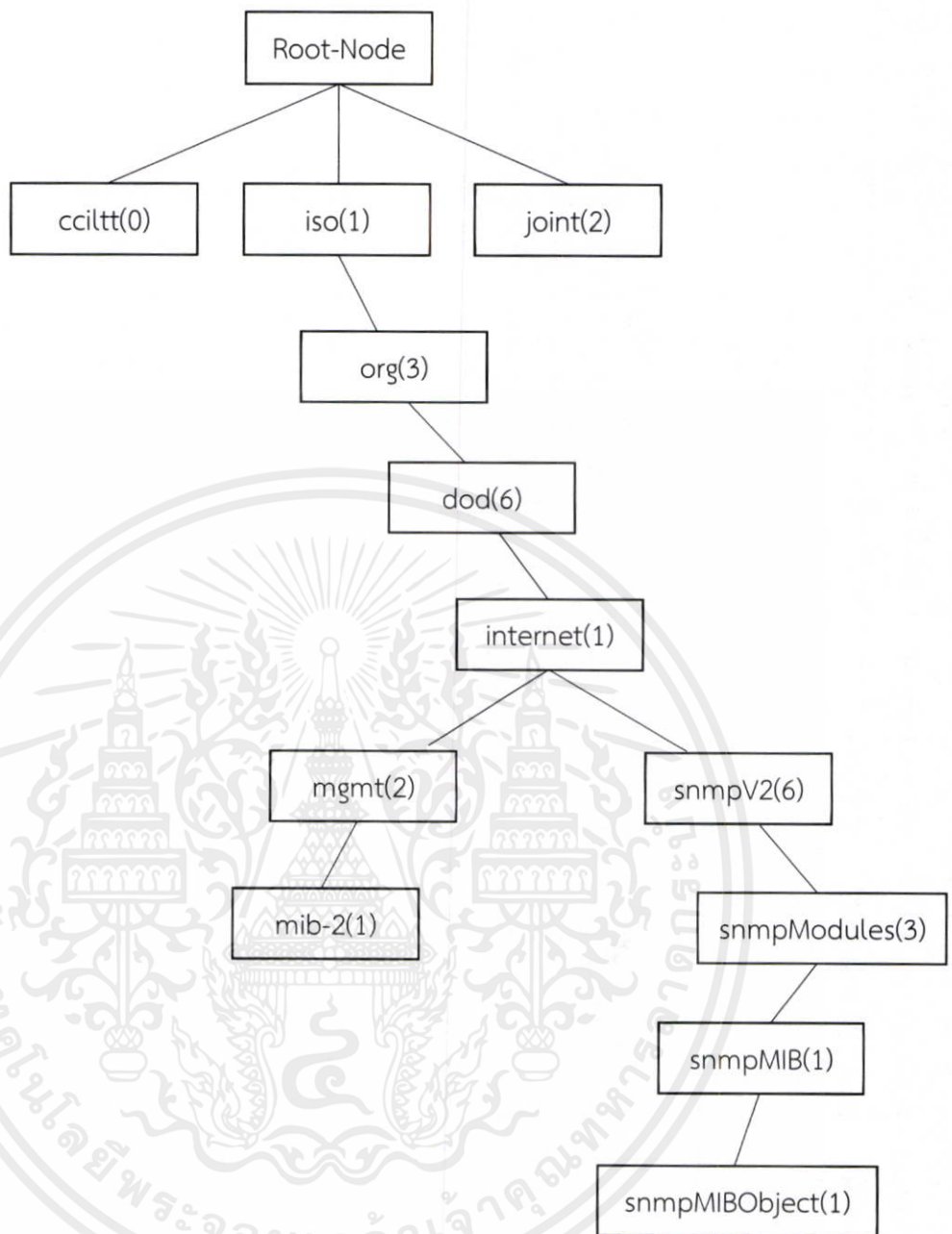
รูปที่ 2.4 SNMP คอมมิวนิตี

แต่อย่างไรก็ตามระดับความปลอดภัยด้วยคอมมิวนิตีที่ตรงก็ยังคงต่ำอยู่ เนื่องจากคอมมิวนิตีที่ตรงถูกส่งแบบข้อความธรรมดาไม่มีการเข้ารหัส ทำให้ง่ายต่อผู้บุกรุกที่มีความชำนาญสูงสามารถทำการดักจับชื่อคอมมิวนิตีที่ตรงได้ และใช้มันเป็นจุดเริ่มต้นในการเจาะเข้าระบบเครือข่าย ดังนั้นจึงได้มีออกข้อกำหนดเพิ่มระดับความปลอดภัยให้สูงขึ้นในมาตรฐาน SNMPv3 แต่อย่างไรก็ตามยังมีวิธีการลดระดับความเสี่ยงต่อการเจาะระบบ ก็คือการติดตั้งไฟร์วอลล์ (Firewall) ซึ่งสามารถกำหนดให้ไฟร์วอลล์อนุญาตเฉพาะโฮสต์ที่รู้จักเข้ามาจัดการระบบเครือข่ายได้เท่านั้น เป็นสิ่งสำคัญที่ต้องตระหนักว่าถ้ามีใครซักคนสามารถอ่านเขียนอุปกรณ์เครือข่ายด้วย SNMP ก็คือสามารถเข้ามาควบคุมระบบเครือข่ายได้ เช่น ปิดพอร์ตเราเตอร์ การเปลี่ยนตารางเราเตอร์ซึ่งทำให้ระบบเครือข่ายล้มเหลวได้ อีกวิธีการหนึ่งในการเพิ่มระดับความปลอดภัย ก็คือการใช้ฟังก์ชัน VPN (Virtual Private Network) เนื่องจากทราฟฟิกของ VPN จะถูกเข้ารหัสไว้ทำให้ดักจับได้ยาก อีกวิธีการหนึ่งแบบง่าย ๆ ก็คือการเปลี่ยนชื่อคอมมิวนิตีที่ตรงอย่างสม่ำเสมอ แต่ก็เป็นการยากสำหรับระบบเครือข่ายขนาดใหญ่ที่จำนวนเมเนเจอร์ และเอเจนต์จำนวนมาก

2.2.5 Simple Network Management Protocol version 2(SNMPv2)

สืบเนื่องจากข้อจำกัดของ SNMPv1 ที่ออกแบบมาให้ใช้งานได้ง่าย แต่ก็มีปัญหาในการใช้งานในเรื่องเอกสารยุคแรกเริ่มไม่เป็นมาตรฐาน หรือเป็นคราฟต์ รวมทั้งข้อจำกัดในเรื่องความปลอดภัย การรายงานความผิดพลาดจากเอเจนต์ที่ยังไม่สมบูรณ์ ชนิดข้อมูลที่ออกแบบไว้มีจำนวนน้อยเกินไปสำหรับงานที่ซับซ้อนขึ้น รวมไปถึงประสิทธิภาพในการทำงานที่ค่อนข้างต่ำในกรณีต้องดึงข้อมูลจำนวนมาก ดังนั้นจึงได้มีการพัฒนาเวอร์ชันใหม่ขึ้นมา นั่นคือ SNMPv2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 โครงสร้างต้นไม้ของ SMI สำหรับ SNMPv2

สำหรับ SNMPv2 นั้นจะมาควบคู่กับมาตรฐานโครงสร้าง SMIv2 ซึ่งเป็นการกำหนดวิธีการสร้าง จากมาตรฐาน SMIv1 ดังรูปที่ 2.5 แสดงตำแหน่งออปเจ็กต์ของ SNMPv2 ในโครงสร้างต้นไม้บรานซ์เพิ่มเติมภายใต้ซัพทรี Internet จากโครงสร้าง SNMPv1 โดยได้มีการสร้างชนิดข้อมูลใหม่เพิ่มเติมมาตรฐาน จะสังเกตได้ว่าหมายเลขOID ของบรานซ์ใหม่จะเท่ากับ 1.3.6.1.6.3.1.1 โดยจะมีชนิดข้อมูลใหม่เพิ่มเติมจาก SNMPv1 ดังตารางชนิดข้อมูลที่สนับสนุนโดย SMIv2 ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงชนิดข้อมูลที่สนับสนุนโดย SMIv2

ชนิดข้อมูล	คำอธิบาย
Integer32	เป็นข้อมูลชนิด 32 บิต โดยจะใช้บรรจุกำที่เป็นตัวเลข หรือถูกอ้างอิงด้วยตัวเลข เช่น สถานการณ์ทำงานของพอร์ตเรเตอร์ เช่น Up, Down หรือ Testing โดยจะถูกระบุเป็นค่าตัวเลขเรียงตามลำดับ 1,2 และ 3
Counter32	เป็นข้อมูลขนาด 32 บิต โดยค่าต่ำสุดเริ่มที่ค่าศูนย์ และค่ามากที่สุดเท่ากับ 4,294,967,295 เมื่อค่าถึงสูงสุดจะกลับมาเริ่มที่ค่าศูนย์โดยอัตโนมัติ และเพิ่มค่าใหม่ ตัว Counter ใช้สำหรับนับจำนวนแพ็กเก็ตที่รับและส่งของแต่ละพอร์ตสื่อสาร รวมทั้งนับจำนวนแพ็กเก็ตที่ผิดพลาด และแพ็กเก็ตที่ถูกดรอปปิ้งโดยอุปกรณ์ ลักษณะการทำงานของ Counter นั้นจะมีแต่การเพิ่มค่าเพียงทิศทางเดียว แต่เมื่อไหร่ก็ตามที่ตัวเอเจนต์เกิดการรีบูต ค่าในแต่ละ Counter ควรถูกเซตให้เท่ากับศูนย์ หรือเริ่มนับใหม่โดยอัตโนมัติ
Gauge32	เป็นข้อมูลขนาด 32 บิต โดยค่าต่ำสุดเริ่มที่ค่าศูนย์ และมากที่สุดมีค่าเท่ากับ 4,294,967,295 มีความแตกต่างจาก Counter ตรงที่ Gauge สามารถเพิ่มและลดค่าได้ แต่จะไม่เกินค่าสูงสุด เช่น การแสดงความเร็ว ณ ปัจจุบันของพอร์ตสื่อสาร
Unsigned32	เป็นข้อมูลเลขฐาน 10 ที่อยู่ในช่วง 0 ถึง $2^{32}-1$
Counter64	จุดประสงค์คล้ายคลึงกับ Counter 32 แต่ค่าสูงสุดอยู่ที่ 18,446,744,073,709,551,615 อันที่จริงแล้ว ตัว Counter64 เป็นตัวนับในอุดมคติจะถูกใช้เฉพาะในกรณีที่ Counter 32 วนกลับสู่ค่าศูนย์เร็วเกินไปเท่านั้น
BITS	เป็นชนิดข้อมูลที่ระบุค่าเป็นแถวของบิต

สำหรับวิธีการนิยามออปเจ็กต์ใน SMIv2 จะมีความแตกต่างจาก SMIv1 เล็กน้อย โดยจะมีฟิลด์ให้เลือกเพิ่มเติมสำหรับการควบคุมการเข้าถึงตัวออปเจ็กต์ และยังอนุญาตให้ทำการขยายขนาดตารางข้อมูลโดยการเพิ่มจำนวนคอลัมน์ ทำให้สามารถใส่ข้อมูล หรือคำอธิบายเพิ่มเติมได้มากขึ้น โดยชนิดข้อมูลที่เพิ่มเติมจะถูกขีดเส้นใต้ เพื่อให้สังเกตเห็นได้ง่าย รวมทั้งแสดงความหมายแสดงในตารางคำสั่งเพิ่มเติมสำหรับ SMIv2 ด้านล่าง

<name> OBJECT-TYPE

เอกสารนี้เป็นเอกสารที่สงวน SYNTAX <datatype> ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 UnitsParts
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งหา MAX-ACCESS ปรึกษาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 STATUS
 DESCRIPTION

“Textual description describing this particular managed object.”
 AUGMENTS {<name of table>}
 ::= {<Unique OID that defines this object>}

ตารางที่ 2.2 แสดงคำสั่งเพิ่มเติมสำหรับ SMIv2

คำสั่ง	คำอธิบาย
UnitsParts	เป็นข้อมูลอธิบายหน่วยของค่าอปเจ็กต์ เช่น วินาที, มิลลิวินาที เป็นต้น
MAX-ACCESS	คล้ายคลึงกับคำสั่ง ACCESS ของ SNMPv1 แต่สำหรับ SNMPv2 คือ MAX-ACCESS โดยค่าที่เลือกได้ของ MAX-ACCESS คือ Read-only, Read-write, Read-create, Not-accessible, และ Accessible-for-notify
STATUS	คำสั่งหรือตัวแปรที่มีไว้เพื่อบ่งบอกสถานะการณั้ถูกใช้งาน เช่น Current, Obsolete และ Deprecated สำหรับสถานะ Current ใน SNMPv2 มีความเดียวกับMandatoryใน SNMPv1
AUGMENT	ในบางกรณีต้องการเพิ่มคอลัมน์ในตารางที่มีอยู่แล้วเพื่อใส่ข้อมูลเพิ่ม คำสั่ง AUGMENT อนุญาตให้เพิ่มคอลัมน์ได้ โดยจะใช้งานควบคู่กับชื่อตารางที่ต้องการขยายขนาด

SMIv2 ยังกำหนดตัวแปรระบบเพิ่มเติมที่ได้ตกลงเป็นทางการในการสร้างอปเจ็กต์โดยใช้หมายเลขมาตรฐาน RFC2579 เป็นตัวกำหนด ดังตารางตัวแปรเพิ่มเติมการสร้างอปเจ็กต์สำหรับ SMIv2 ด้านล่าง

ตารางที่ 2.3 แสดงตัวแปรเพิ่มเติมการสร้างอปเจ็กต์สำหรับ SMIv2

ตัวแปร	คำอธิบาย
DisplayString	แถวสตริงบรรจุข้อมูลแบบแอสกี ที่สามารถยาวได้สูงสุด 255 อักขระ
PhysAddress	หมายเลขที่ในระดับฟิสิกอล หรือระดับสื่อสัญญาณ โดยถูกแทนด้วยข้อมูลชนิดสตริง
MacAddress	หมายเลขที่ระดับ Media Access ตามมาตรฐาน IEEE 802 หรือเรียกอีกอย่างว่าหมายเลขอีเทอร์เน็ต

TruthValue	กำหนดค่าแบบบูลีน True และ False
TestAndIncr	ถูกใช้เพื่อควบคุมการเข้าแก้ไขจัดการอปเจ็กต์จากสองระบบ NMS ในเวลาเดียวกัน
Autonomous Type	เป็นหมายเลข OID ที่ถูกใช้เพื่อกำหนดนิยามซัพพอร์ทที่มีความเกี่ยวข้องกับฐานข้อมูล MIB
variablePointer	ตัวชี้ไปยังออปเจ็กต์
RowPointer	ตัวชี้ไปยังแถวในตาราง
RowStatus	ถูกใช้เพื่อจัดการควบคุมการสร้าง และลบ โรว์ตาราง ตัวแปร RowStatus สามารถใช้ตรวจสอบสถานะของโรว์ในตาราง คำสั่งนี้ถูกออกแบบมาเพื่อสร้างเสถียรภาพให้กับตัวตารางในกรณีที่มีตัวจัดการหรือ NMS มากกว่าหนึ่ง และทำการแก้ไขสร้าง หรือลบ โรว์ในเวลาเดียวกัน ค่าสถานะจะเรียงด้วยค่าตัวเลขดังต่อไปนี้ active(1), notInService(2), notReady(3), CreateAndGo(4), CreateAndWait(5) และ Destroy(6)
TimeStamp	ให้วัดค่าเวลาระหว่างเวลาที่อุปกรณ์เริ่มทำงาน และเกิดเหตุการณ์ต่างๆ
TimeInterval	ใช้วัดเวลาในหน่วยหนึ่งส่วนร้อยวินาที ตัวแปรTimeInterval จะมาในค่าของเลขจำนวนเต็มจาก 0-2147483647
DateAndTime	เป็นข้อความสตริงที่ใช้แสดงข้อมูลวันเวลา
Storage Type	เป็นตัวแปรบอกชนิดหน่วยความจำที่ตัวเอเจนต์ใช้ ส่วนค่าที่เป็นไปได้มีดังนี้ Other(1), volatile(2), notvolatile(3), permanent(4), และ readOnly(5)
TDomain	เป็นตัวแปรบอกชนิดการใช้บริการในระดับทรานสปอร์ต
TAddress	เป็นตัวแปรบอกหมายเลขการใช้บริการในระดับทรานสปอร์ต

2.3 การสร้างเมสเสจ SNMP

จากที่กล่าวมาแล้ว SNMP คือโปรโตคอลที่อนุญาตให้ SNMP Manager หรือ NMS (ซึ่งเป็นตัวควบคุมจัดการ) สามารถควบคุมตัว SNMP เอเจนต์ (ตัวถูกจัดการ) โดยวิธีการส่งแลกเปลี่ยนเมสเสจ และเมสเสจ SNMP จะถูกส่งบนโปรโตคอลระดับทรานสปอร์ตนั่นคือโปรโตคอล UDP/IP โดยใช้พอร์ตหมายเลข 161 เป็นพอร์ตหลัก ดังนั้นถ้าต้องการศึกษาว่าส่งเมสเสจ บน UDP/IP ทำ

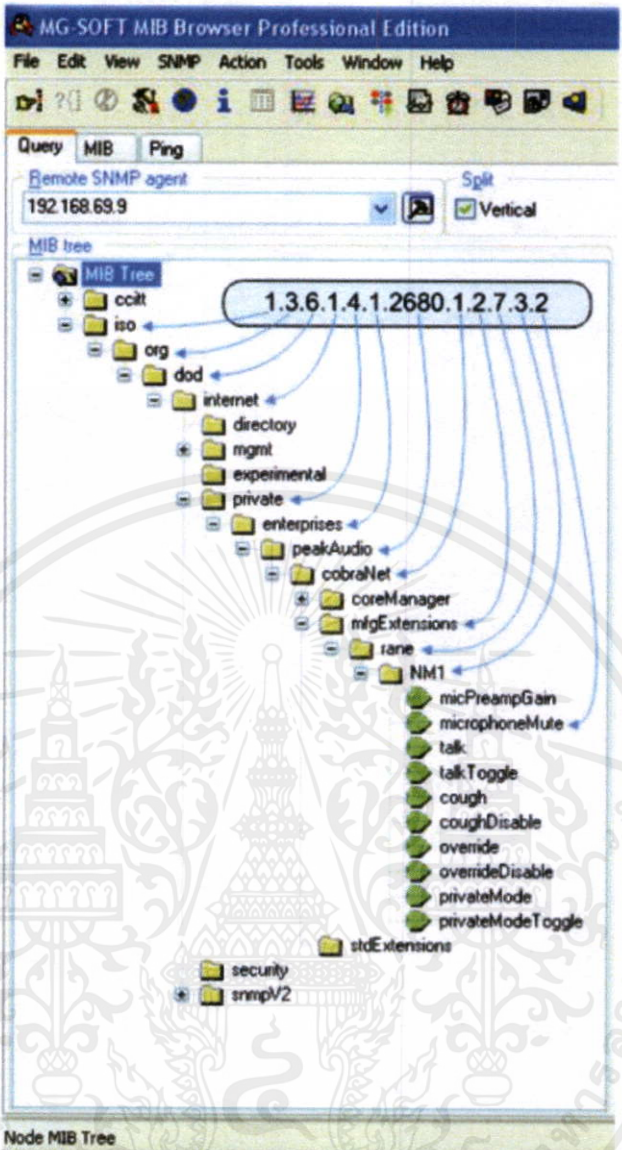
อย่างไร นั้นสามารถหาตัวอย่างการเขียนโปรแกรมได้ทั่วไป เนื่องจาก UDP/IP พิงค์ชั้นนั้นถูกสร้าง และสนับสนุนติดมากับตัวคอมพิวเตอร์อยู่แล้ว

เป้าหมายหลักของการใช้มสเซจ SNMP คือการต้องการควบคุม (คำสั่ง Set) หรือ ต้องการดู ค่าดึงค่า (คำสั่ง Get) จากตัวเอเจนต์นั่นเอง ภายในซอฟต์แวร์ที่สนับสนุน SNMP จะมีค่าพารามิเตอร์ ที่เป็นตัวแทน (Instance) ของออบเจกต์ ยกตัวอย่างสำหรับอุปกรณ์เครื่องเสียงยี่ห้อ RANE ที่ สนับสนุน SNMP จะมีอินสแตนซ์ของออบเจกต์ที่ใช้ควบคุมไมโครโฟนคือ ออบเจกต์ Microphone Mute ซึ่งจะใช้หนึ่งอินสแตนซ์ต่อหนึ่งช่องอินพุตของไมโครโฟน

โดยทั่วไปตัว SMNP Manager สามารถดึงค่าหรือดึงค่าจากแต่ละอินสแตนซ์หรืออุปกรณ์ในที่นี้คือ แต่ละตัวไมโครโฟนนั่นเอง ในตัวเอเจนต์ค่าพารามิเตอร์จะถูกจัดเรียงตามโครงสร้างต้นไม้ SNMP และจะใช้หมายเลข OID (Object Identifier) เพื่อเจาะจงไปยังพารามิเตอร์ที่ต้องการไม่ว่าจะเพื่อตั้ง ค่า หรือตรวจสอบข้อมูล ตัวหมายเลข OID จากที่กล่าวมาแล้วก็คือชุดของตัวเลขที่คั่นด้วย เครื่องหมายจุดเพื่อแยกแยะหาตำแหน่ง

สำหรับในตัวอย่างนี้ OID ที่อ้างอิงที่อยู่ของพารามิเตอร์ในการควบคุมปิดเปิดเสียงของ ไมโครโฟนหรือ Microphone Mute คือ NM1 โดยมีหมายเลข OID ดังต่อไปนี้ 1.3.6.1.4.1.2680.1.2.7.3.2.0 อันที่จริงแล้วหมายเลขOID จะประกอบด้วยสองค่าค่าที่หนึ่งคือค่า อ้างอิงชนิดออบเจกต์นั่นคือ 1.3.6.1.4.1.2680.1.2.7.3.2 และส่วนค่าที่สองคืออ้างอิงถึงอินสแตนซ์ซึ่ง มีไว้เจาะจงอินสแตนซ์หรืออินพุตของไมโครโฟนว่าตัวไหน (Microphone Mute) ในตัวอย่างนี้ค่า อินสแตนซ์มีค่าเท่ากับศูนย์ก็เพราะในตัวอย่างนี้ ตัวอุปกรณ์ควบคุมเครื่องเสียง NM1 ตัวนี้มี ไมโครโฟนเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 การตำแหน่งออบเจกต์โดยใช้ MIB Browser

ในแต่ละตัวเอเจนต์จะมีฐานข้อมูลที่เป็นเสมือนกับสมุดบันทึกตำแหน่งของออบเจกต์ทั้งหมด รวมทั้งหมายเลขและชื่ออ้างอิงที่เรียกว่า MIB (Management Information Base) โดยที่ MIB จะจัดเรียงชื่อ, หมายเลข OID, ชนิดข้อมูล, สิทธิการอ่านและเขียนรวมทั้งคำอธิบายสั้น ๆ สำหรับแต่ละออบเจกต์ที่อยู่ในตัวเอเจนต์ ตัวอย่าง เช่น MIB ของเครื่องเสียง RANE ตัว NMS จำเป็นต้องทราบข้อมูลเกี่ยวกับออบเจกต์จาก MIB รวมทั้งค่าอินสแตนซ์ ซึ่งทำให้ตัว NMS สามารถส่งเมสเซจเพื่อตั้งค่า หรือดึงค่าในตัวเอเจนต์ได้ถูกต้อง อย่างไรก็ตามก็ยังมีสิ่งที่จะต้องศึกษาหรือต้องรับรู้อีกนั่นคือ พฤติกรรม หรือคุณสมบัติเฉพาะของตัวอุปกรณ์ที่สนับสนุน SNMP นั้น ๆ

อุปกรณ์ที่สนับสนุน SNMP จะต้องเข้าใจในความหมายของแต่ละเมสเซจถึงจะทำงานได้อย่างถูกต้องซึ่งอันที่จริงก็เป็นปัญหาที่พบบ่อยในการสื่อสารระหว่างตัวอุปกรณ์ถึงแม้จะมีการ

กำหนดมาตรฐาน ปัญหานี้ที่ยังคงอยู่ก็เพราะมีความแตกต่างของภาษาในการสร้างซอฟต์แวร์ซึ่งอาจมีความแตกต่างในเรื่องประเภทชนิดของข้อมูล (เช่น ขนาดของอินทิเจอร์) ตัวอย่างเช่น ตัว NMS ที่ส่งเมสเซจที่สร้างมาจากภาษา Java อาจจะไม่สื่อสารกับเอเจนต์ที่รับเมสเซจที่สร้างด้วยภาษา C (ปัญหาจะเกิดถ้าโปรแกรมเมอร์ไม่ระมัดระวังหรือไม่มีการตรวจสอบการแจ้งเตือน) และเพื่อแก้ไขปัญหานี้ SNMP จึงกำหนดใช้มาตรฐานการเข้ารหัส ASN.1 (Abstract Syntax Notation One) เพื่อใช้อ้างอิงชนิดข้อมูลมาตรฐานในการสร้างเมสเซจ SNMP เพราะว่า ASN.1 เป็นมาตรฐานอิสระจากภาษาในการโปรแกรมมิ่ง ซึ่งตัวซอฟต์แวร์ ของ NMS หรือ เอเจนต์อาจถูกเขียนด้วยภาษาใดก็ได้ อย่างไรก็ตามถึงจะใช้มาตรฐาน ASN.1 อ้างอิงก็ยังมีปัญหาอื่น ๆ อยู่ เพราะเมื่อข้อมูลถูกส่งบนสายสัญญาณแล้วควรจะถูกเข้ารหัสด้วยมาตรฐานอะไร

ตัวอย่างเช่น ถ้าเป็นข้อความแบบสตริงจะถูกปิดท้ายด้วยอักขระนัล (Null) เหมือนกับการโปรแกรมภาษา C หรือไม่ หรือค่าบูลีนจะมีขนาด 8 บิต เหมือนในภาษา C++ หรือมีขนาด 16 บิตอย่างในภาษาวิซวลเบสิกหรือไม่ เพื่อแก้ไขปัญหาดังกล่าว ไม่ว่าจะใช้ภาษาอะไรในการโปรแกรมมิ่ง ข้อมูลจะแปลงขนาดข้อมูลและถูกเข้ารหัสด้วยวิธีการเดียวกันก่อนที่ส่งลงไปใ้สื่อสัญญาณด้วยมาตรฐาน BER สรุปแล้วทุกข้อมูลในฟิลด์ของเมสเซจต้องเป็นไปตามมาตรฐาน ASN.1 ในเรื่องชนิดของข้อมูลและต้องถูกเข้ารหัสตามมาตรฐาน BER ก่อนที่จะส่งข้อมูลออกไปยังระบบสื่อสาร

2.3.1 ASN.1

การที่จะสร้างเมสเซจ SNMP ต้องมีความรู้ในเรื่องชนิดข้อมูลและขนาดข้อมูลที่ระบุในมาตรฐาน ASN.1 ซึ่งได้จัดเป็นสองกลุ่มคือ ไพรมิทีฟ (Primitive) และคอมเพล็กซ์ (Complex) สำหรับชนิดข้อมูลพื้นฐานหรือไพรมิทีฟของมาตรฐาน ASN.1 ได้แก่ อินทิเจอร์ (Integer), ออกเทต (Octet) หรือ ไบต์ หรือ อักขระ (Character), สตริง (String), นัล (Null), บูลีน (Boolean) รวมทั้ง ออบเจกต์ไเดนติไฟเออร์ (Object Identifier) หรือ OID ซึ่งชนิดข้อมูล OID เป็นสิ่งที่สำคัญมากสำหรับเมสเซจ SNMP ก็เพราะว่าชนิดข้อมูล OID จะบรรจุค่าที่ใช้อ้างอิงถึงพารามิเตอร์ในตัวเอเจนต์ อย่างไรก็ตามเพื่อเพิ่มความสามารถให้กับโปรแกรมเมอร์ในการจัดการข้อมูลมาตรฐาน ดังนั้น ASN.1 ยังอนุญาตให้จัดกลุ่มข้อมูลจากหลายชนิดข้อมูลประเภทไพรมิทีฟให้เป็นชนิดข้อมูลใหม่นั้นคือชนิดข้อมูลประเภทคอมเพล็กซ์

Type	Length	Data
------	--------	------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเปลี่ยนแปลงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 ชนิดข้อมูลไพรมิทีฟ

มาตรฐาน ASN.1 ยังกำหนดชนิดข้อมูลประเภทคอมเพล็กซ์ที่จำเป็นในการสร้างเมสเซจ SNMP นั้นคือชนิดข้อมูลซีเควนซ์ (Sequence) ซึ่งซีเควนซ์มองง่าย ๆ เสมือนเป็นรายการฟิลด์ข้อมูล (List of Data Fields) แต่ละฟิลด์ในซีเควนซ์สามารถประกอบด้วยหลาย ๆ ชนิดข้อมูลที่แตกต่างกัน อีกชนิดข้อมูลประเภทคอมเพล็กซ์ที่ มาตรฐาน ASN.1 กำหนดคือชนิดข้อมูล PDU (Protocol Data Unit) ซึ่งเป็นชนิดข้อมูลประเภทคอมเพล็กซ์ที่ใช้เฉพาะกับโปรโตคอล SNMP ฟิลด์ของ PDU จะใช้บรรจุค่าข้อมูลของเมสเซจ SNMP โดยคำสั่งที่ใช้ชนิดข้อมูล PDU ที่ใช้เป็นหลักคือ คำสั่ง Get Request และ Set Request ที่ใช้สำหรับการดึงค่าและตั้งค่าพารามิเตอร์ตามลำดับ

สรุปได้เลยว่าเมสเซจของ SNMP จะถูกสร้างจากฟิลด์ที่เป็นชนิดข้อมูลตามมาตรฐาน ASN.1 ทั้งหมด อย่างไรก็ตามก็ยังไม่รับประกันว่าชนิดข้อมูลนั้นถูกต้องตามที่ต้องการทั้งฝั่งส่งและฝั่งรับ ก็เพราะว่าเมสเซจของ SNMP เป็นซีเควนซ์ของฟิลด์ที่ประกอบด้วยชนิดข้อมูลที่แตกต่างกันไป แล้วฝั่งผู้รับจะรับทราบได้อย่างไร จุดเริ่มและจุดจบของแต่ละฟิลด์อยู่ตรงไหน หรือแม้กระทั่งชนิดข้อมูลของแต่ละฟิลด์เป็นอย่างไร การแก้ปัญหาเหล่านี้จึงใช้มาตรฐาน BER เป็นตัวแก้ไขปัญหา

Type	Length	Type	Length	Data	Type	Length	Data
------	--------	------	--------	------	------	--------	------

รูปที่ 2.8 ชนิดข้อมูลคอมเพล็กซ์

2.3.2 BER (Basic Encoding Rules)

การสร้างเมสเซจ SNMP ต้องปฏิบัติตามกฎของมาตรฐาน BER เมื่อต้องการวางแต่ละไบต์ข้อมูลลงบนเมสเซจ SNMP กฎพื้นฐานคือ แต่ละฟิลด์จะถูกเข้ารหัสโดยแบ่งเป็นส่วนสามส่วนคือ ชนิด (Type), ความยาว (Length) และ ข้อมูล (Data)

ชนิด (Type) จะระบุชนิดของข้อมูลของฟิลด์โดยมีขนาด 1 ไบต์ สำหรับตัวอย่างให้ดูที่ตารางด้านล่าง ที่จะบอกถึงชนิดข้อมูลและรหัสไอเดนติไฟเออร์ (Identifier) เพื่อใช้อ้างอิง

ความยาว (Length) จะใช้ระบุความยาวของส่วนค่า ในหน่วยไบต์ และสุดท้ายคือ ค่าคือ ค่าหรือข้อมูลจริงที่ต้องการสื่อสาร (เช่น ตัวเลข, สตริงข้อความ, OID เป็นต้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงชนิดข้อมูลและไอเดนติไฟเออร์

Primitive Data Type	Identifier	Complex Data Types	Identifier
Integer	0x02	Sequence	0x30
Octet String	0x04	GetRequest PDU	0xA0
Null	0x05	GetRespond PDU	0xA2
Object Identifier	0x06	SetRequest PDU	0xA3

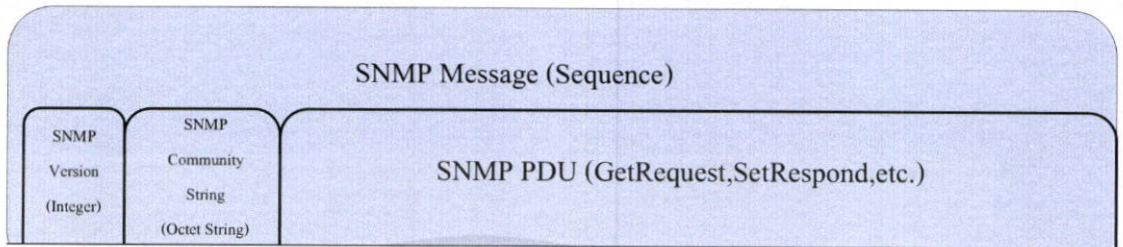
สำหรับบางชนิดข้อมูลจะมีลักษณะเหมือนซีเคิร์ฟหรือ PDU ซึ่งจะถูกสร้างจากฟิลด์เล็ก ๆ หลาย ๆ ฟิลด์ ดังนั้นจึงเป็น ชนิดข้อมูลแบบคอมเพล็กซ์ โดยจะถูกเข้ารหัสเหมือนมีฟิลด์ซ้อนภายในฟิลด์ดังแสดงในรูปที่ 1.3 ยังมีอีกสองกฎพื้นฐานที่จำเป็นต้องทราบในการเข้ารหัสเมสเสจของ SNMP กฎทั้งคู่ถูกใช้ในการเข้ารหัส OID กฎแรกจะถูกใช้เพื่อการเข้ารหัส สองหมายเลขแรกของ OID ตามมาตรฐาน BER สองหมายเลขแรกของ OID (x.y) จะถูกเข้ารหัสรวมให้เป็นค่าเดียวตามสูตร $(40*x) + y$ โดยสองหมายเลขแรกของ OID โดยทั่วไปเริ่มที่ 1.3 เสมอ ดังนั้น สองหมายเลขแรกของ OID จะถูกเข้ารหัสเป็น 43 หรือ $0x2B$ เสมอจาก $(40*1) + 3 = 43$ หลังจากที่สองหมายเลขแรกได้ถูกเข้ารหัส หมายเลขต่อไป ๆ ของ OID จะถูกเข้ารหัสในหน่วยไบต์ อย่างไรก็ตามยังมีกฎพิเศษที่ใช้กับค่าประเภทตัวเลขที่มีค่ามากกว่าหนึ่งไบต์ เนื่องจากหนึ่งไบต์ (8 บิต) สามารถใช้แทนตัวเลขที่ค่าระหว่าง 0-255 เท่านั้น

ตัวอย่างถ้าตัวเลขมีค่าเท่ากับ 2680 ใน Microphone Mute OID 1.3.6.1.4.1.2680.1.2.7.3.2.0 ไม่สามารถเข้ารหัสโดยใช้แค่เพียงหนึ่งไบต์ ตามกฎสำหรับเลขที่มีค่ามากคือ 1 ไบต์จะใช้เพียง 7 บิตสำหรับบรรจุค่าตัวเลข (0-127) และบิตลำดับสูงสุด หรือ MSB จะถูกใช้เป็นแฟล็ก (Flag) เพื่อให้ผู้รับรู้ว่าตัวเลขที่รับดังกล่าวนี้ถูกขยายขนาดมากกว่าหนึ่งไบต์ ดังนั้นค่าตัวเลขใด ๆ ที่มีค่ามากกว่า 127 จะต้องถูกเข้ารหัสโดยใช้จำนวนไบต์มากกว่าหนึ่ง จากกฎดังกล่าว ค่า 2680 ต้องถูกเข้ารหัสโดยใช้ 2 ไบต์คือ 0×94 และ 0×78 เนื่องจากบิต MSB ของไบต์แรก (0×94) มีค่าเท่ากับหนึ่ง ผู้รับจะรู้และถอดค่าจาก 7 บิตที่เหลือของแต่ละไบต์ ($0 \times 14, 0 \times 78$) และถอดรหัสตามเลขฐานสิบหกจนได้ค่า $(0 \times 14*128) + 0 \times 78 = 2,680$

2.3.3 ฟอรัมเมสเสจของ SNMP

ฟอรัมเมสเสจของ SNMP จะระบุว่าฟิลด์ใน SNMP อยู่ตำแหน่งใด และเรียงลำดับอย่างไร ยิ่งกว่านั้นเมสเสจอาจจะมีหลายชั้นหลายเลขอร์ของฟิลด์ที่ซ้อนกัน ณ ที่

ชั้นนอกสุดเมสเสจของ SNMP เป็นฟิลด์เดี่ยวประเภทซีแควนซ์ ภาพรวมของเมสเสจทั้งหมดจะเป็นแถวของฟิลด์ 3 ฟิลด์นั้นคือ ฟิลด์เวอร์ชันของ SNMP (อินทิเจอร์), ฟิลด์คอมมิวนิตีสตริง (ออกเทตสตริง) และ PDU (Get Request, Set Request)



รูปที่ 2.9 ฟอर्मเมตพื้นฐานของ SNMP

ฟิลด์เวอร์ชันและฟิลด์คอมมิวนิตีสตริงจะเป็นชนิดข้อมูลแบบไพรมิทิวที่ที่ไม่ได้สร้างจากฟิลด์เล็กๆหลายฟิลด์ หรือมีหลายชั้น แต่สำหรับฟิลด์ PDU จะเป็นชนิดข้อมูลแบบคอมเพล็กซ์ที่ประกอบด้วยหลายฟิลด์ขนาดเล็ก (หลายเลขเอร์) ภายใน PDU จะประกอบด้วยฟิลด์รีควีสไอดี (Request ID: อินทิเจอร์), ฟิลด์เออร์เรอร์ (Error: อินทิเจอร์), ฟิลด์เออร์เรอร์อินเดกซ์ (Error Index: อินทิเจอร์) และฟิลด์วาร์ไบด์ลิสต์ (Varbind List) วาร์ไบด์จะประกอบด้วยสองฟิลด์ที่ต่อกันป็นคู่ๆ กันไป ฟิลด์แรกคือ OID ที่ระบุที่อยู่ของพารามิเตอร์ที่ต้องการ ฟิลด์ที่สองคือฟิลด์แวลู (Value) คือค่าของพารามิเตอร์ที่ต้องการ สำหรับคำสั่ง Set Request ค่าที่ตั้งค่าหรือเซตต้องเป็นชนิดข้อมูลเดียวกันกับชนิดข้อมูลที่ระบุใน MIB ของอุปกรณ์นั้นๆ สำหรับคำสั่ง Get Request ฟิลด์แวลูจะมีค่าเท่ากับนัล (Null) ด้วยค่าความยาวเท่ากับ 0 x 00 ค่านี้ก็เป็นตัวบ่งบอกตัวเอเจนต์ให้ส่งค่ากลับด้วย Get Request PDU สำหรับวาร์ไบด์ลิสต์คือแถวลำดับของวาร์ไบด์ ในกรณีเมสเสจตั้งค่าหรือดึงค่าเพียงหนึ่งพารามิเตอร์วาร์ไบด์ลิสต์จะมีแค่หนึ่งวาร์ไบด์

ตารางที่ 2.5 แสดงฟิลด์ในเมสเสจ SNMP

Field	Description
SNMP Message	A Sequence representing the entire SNMP message consisting of the SNMP version, Community String, and SNMP PDU.
SNMP Version	An integer that Identifiers version of SNMP. SNMPv1=0
SNMP	An octet String that may contain a string used to add security to SNMP devices.

Community String	
SNMP PDU	An SNMP PDU Contains the body of the SNMP message. They are several types of PDUs. Three common PDUs are GetRequest, GetRespond, and SetRequest
Request ID	An integer set to 0x00 in the request sent by the SNMP manager. The SNMP agent places an error code in this field in the respond message of an error occurred processing the request. Some error code include: 0x00 - No error occurred 0x01 – Respond message too large to transport 0x02 – The name of the request object was not found 0x03 – A data type in the request did not match the data type in SNMP agent 0x04 – The SNMP manager attempted to set a read-only parameter 0x05 – General Error (Some error other than the ones listed above)
Error Index	If an Error occurred, The Error Index holds a pointer to the object that caused the error, otherwise the Error Index is 0x00.
Varbind List	A Sequence of Varbinds.
Varbind	A Sequence of 2 fields, an Object ID and the value foe/from that Object ID.
Object Identifier	An Object Identifier that points to a parameter in the SNMP agent.
Value	SetRequest ODU – Value is applied to the specified OID of the SNMP agent. GetRequest PDU – Value is a Null that act as a placeholder for the return data. GetRespond PDU – The returned Value from the specified OID of the SNMP agent.

2.4 การให้บริการแผนที่ด้วยกูเกิลแมพ (Google Maps)

กูเกิลแมพ คือ บริการแผนที่ออนไลน์จาก Google แผนที่ออนไลน์สามารถใช้งานได้หลายรวมบริการหลายอย่างเข้าไว้ด้วยกัน เช่น ภาพแผนที่บนถนนหนทางต่างๆ ภาพถ่ายดาวเทียมภาพถ่ายจากอากาศยานที่มีความละเอียดของภาพสูง บริการค้นหาสถานที่ห้างร้านต่างๆ บริการ

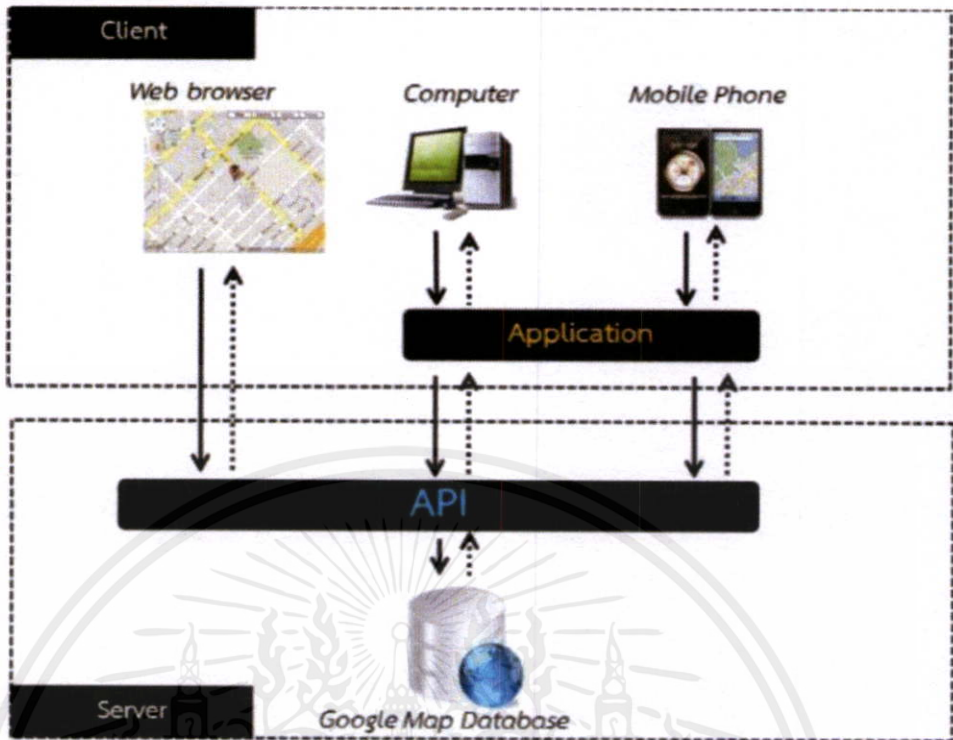
ค้นหาเส้นทางจากสถานที่ต้นทางไปยังสถานที่ปลายทาง บริการภาพถ่ายจากถนนในเมืองสำคัญที่ทำให้ผู้ใช้เห็นสภาพแวดล้อมและอาคารบ้านเรือนที่อยู่สองฝั่งถนน เป็นต้น

Google Maps API ช่วยให้เราสามารถพัฒนาโปรแกรมเพื่อแทรก Google Maps เข้าไปเป็นองค์ประกอบส่วนหนึ่งในเว็บเพจที่ต้องการได้โดยเขียนเป็นรหัส Html และ Javascript ในรูปแบบที่ไม่สลับซับซ้อนมาก สำหรับงานแผนที่แบบง่าย Google Maps API มีความสามารถที่กว้างขวางเน้นในด้านการนำเสนอข้อมูลแผนที่ในลักษณะการหมุดปัก (Push pin / Place marker) ซึ่งสามารถกำหนดให้แสดงข้อมูลประกอบแผนที่เมื่อผู้ใช้คลิกที่ Push pin/Marker นั้นๆ หรือองค์แผนที่แบบเส้น (Polyline) พื้นที่ (Polygon) และภาพ (Ground overlay) บริการด้านแผนที่ของ Google นี้เริ่มต้นตั้งแต่กลางปี ค.ศ. 2005 เป็นบริการฟรี จัดให้แก่ผู้ใช้ทั่วโลกโดยคาดหวังที่จะใช้การโฆษณาบนแผนที่ที่เป็นรายได้กลับคืน แต่ในระยะแรกจะยังไม่มีการโฆษณาดังกล่าวในการจัดบริการแผนที่นี้ ส่วนประกอบพื้นฐานสำคัญที่ดึงดูดให้มีผู้ใช้งานแผนที่ของ Google เป็นอย่างมากคือ แผนที่และภาพถ่ายดาวเทียมคุณภาพดีซึ่งครอบคลุมทั่วพื้นผิวโลกในมาตรา ส่วนต่างๆ ตามความเหมาะสม ทำให้การพัฒนาต่อยอดจากสิ่งที่ Google จัดไว้ให้แล้วเป็นงานที่น่าสนใจ ไม่ต้องลงทุนจัดหาทรัพยากรที่หายากและราคาแพงเองมาใช้ในโครงการอย่างที่เคยเป็นในอดีต

เนื่องจากจัดทำ Google Maps API เป็น โปรแกรมรหัสเปิด (Open source program) ในภาษาจาวาสคริปต์ จึงทำให้ผู้ใช้ที่เป็นนักพัฒนาโปรแกรมสามารถเข้าไปดูรายละเอียดของรหัสโปรแกรมได้สะดวก รวมทั้งสามารถปรับเปลี่ยนแก้ไขโปรแกรมได้ ทำให้ Google Maps API มีผู้ใช้งานอย่างกว้างขวางเหตุผลสำคัญอีก 2 อย่างที่ส่งเสริมให้มีผู้ใช้งานมากคือแผนที่และภาพถ่ายดาวเทียมคุณภาพดีที่ใช้ สนับสนุนการทำแผนที่ให้มีครอบคลุมพื้นที่ต่างๆ อย่างกว้างขวาง และชื่อเสียงของโปรแกรม Google Earth เสริมด้วยบริการ Google Local ที่มีมาก่อน

2.4.1 หลักการทำงาน การทำงานเหมือนกับเว็บแอปพลิเคชันอื่นของกูเกิล กูเกิลแมปใช้ภาษาจาวาสคริปต์ เมื่อผู้ใช้ทำการขยับตำแหน่งของแผนที่ แผนที่ที่ถูกแบ่งส่วนเป็นตารางรูปทรงสี่เหลี่ยมจัตุรัสจะถูกดาวน์โหลดจากเซิร์ฟเวอร์ของกูเกิลและทำการใส่ลงไปในหน้าเว็บเพจ และเมื่อผู้ใช้ทำการค้นหาสถานที่หนึ่งๆ ผลการค้นหาจะถูกดาวน์โหลดอยู่เบื้องหลังการทำงานเพื่อรอการใส่เข้าไปในแถบด้านข้างและการใส่ลงภายในแผนที่ หน้าเว็บเพจจะไม่ถูกทำการโหลดหน้าใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 สถาปัตยกรรมของกูเกิลแมป เอพีไอ (Google Map API Architecture)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบการทำงาน

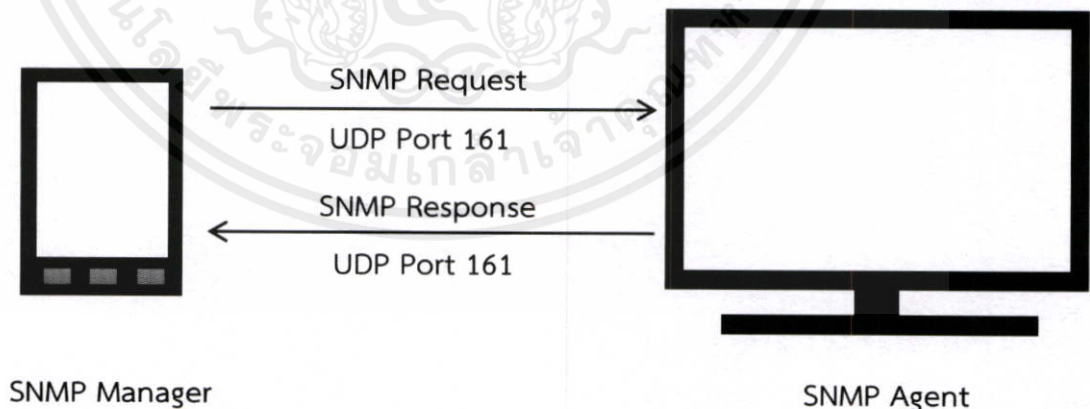
3.1 ภาพรวมของระบบ

การทำงานของโปรแกรมประยุกต์โดยหลักๆ จะแบ่งการทำงานออกเป็น 2 ส่วนคือ

3.1.1 การทำงานในส่วนหลัก

1) การทำงานของโปรแกรมประยุกต์ในส่วนนี้ จะเป็นการทำงานของการใช้โพรโตคอล SNMP ในการดึงข้อมูลของเครื่องแม่ข่ายมา ซึ่งจะใช้ library SNMP4J ในการช่วยการพัฒนา โดยเริ่มต้นจากการสร้างการเชื่อมต่อระหว่าง SNMP Agent และ SNMP Manager ซึ่งใช้โพรโตคอล UDP/IP ผ่านหมายเลขพอร์ต 161 ในการเชื่อมต่อ

2) หลังจากที่ได้มีการสร้างการเชื่อมต่อแล้วตัว Manager จะทำการส่งค่า OID ที่ต้องการทราบค่าไปยังตัว Agent โดยจะต้องมีการกำหนดเป้าหมาย Agent ผ่าน IP Address ซึ่งในที่นี้ได้ใช้ SNMP version 2 และตั้งค่า Community เป็น public จากนั้นทำการส่ง request ไปยัง Agent และได้ค่าต่างๆกลับมา ซึ่งจะนำมาแสดงผลต่อไปในแอปพลิเคชัน เช่น รายละเอียด Hardware, รายชื่อ Software ที่ทำงานอยู่



รูปที่ 3.1 โครงสร้างการทำงานของ SNMP

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้เฉพาะงานวิจัยเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อย่างไรก็ตามจะต้องอ้างอิงถึงงานวิจัยเอกสารฉบับนี้ไว้ด้วย

3) ในแอปพลิเคชันยังมีส่วนในการดูว่าเครื่อง Agent เป้าหมายนั้นมีการเปิดเครื่องทำงานอยู่หรือไม่ โดยจะส่ง Package ไปสอบถาม ซึ่งจะมีการอัปเดตข้อมูลทุกๆ วินาที โดยถ้า Agent เป้าหมายนั้นมีการเปิดเครื่องทำงานอยู่จะมีสถานะเป็นสีเขียว ถ้าไม่จะ

มีสถานะเป็นสีแดง ซึ่งในส่วนนี้สามารถเพิ่มรายชื่อของ Agent โดยการใส่ IP Address ที่ต้องการดูได้ 10 รายการ

4) ส่วนของการรีโมต Agent นั้นจะมีการส่ง Package UDP ผ่านทางพอร์ต 1234 โดยจะมีการส่งคำสั่งไปยัง Agent ซึ่งจะมีอยู่ 4 คำสั่งคือ Shut Down, Restart, Sleep, Log Off จากนั้นจะให้ Agent ซึ่งรันโปรแกรมสำหรับรับคำสั่ง และปฏิบัติตามคำสั่งนั้นๆ โดยโปรแกรมที่อยู่ใน Agent นั้นจะเขียนโดยภาษา C#

5) นอกจากนี้ยังมีฟังก์ชันสำหรับดู Location ของ Agent เป้าหมาย โดยจะใช้ IP Address แล้วนำไปค้นหาใน Website <http://ipinfo.io> จากนั้น นำค่า Latitude และ Longitude ไปพล็อตลงไปใน Google Map version 2 ซึ่งค่าที่ได้จะมีการผิดพลาด ผู้ใช้จะสามารถกำหนด Location ใหม่ได้ โดยการลาก Marker ไปให้ตรงจุดที่ต้องการ

3.1.2 การทำงานในส่วนของวิดเจ็ต (Widget)

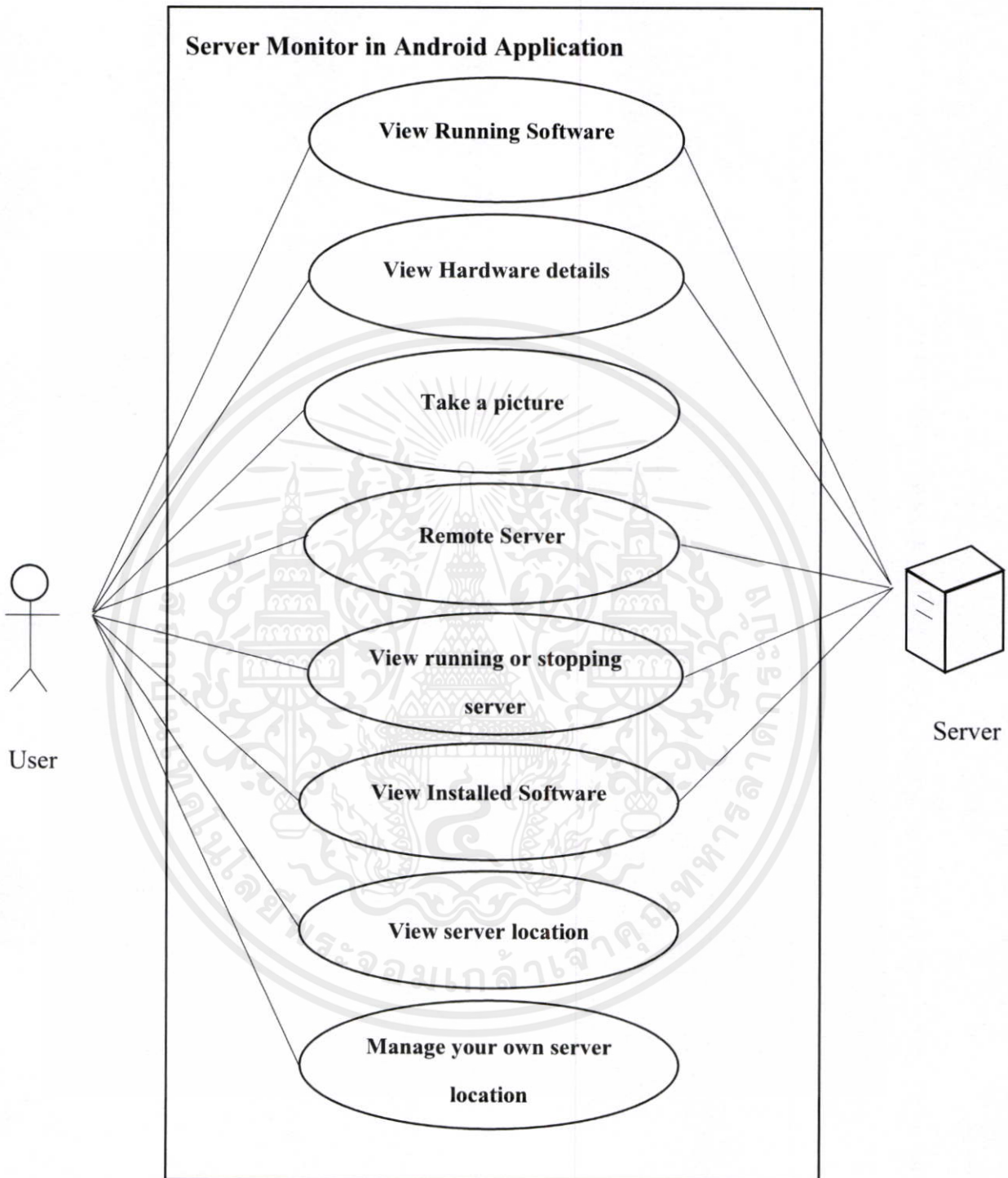
ในส่วนการทำงานของหน้าจอหลัก (Home Screen) จะมีการทำงานดังนี้

- เป็นลักษณะ ListView ที่มีรายละเอียดของ IP Address Agent อยู่ โดยสามารถเพิ่มหรือลบได้ IP Address ได้
- สามารถเปิดไปยังหน้าจอแผนที่ได้ ซึ่งแสดงผลโดย Google Map

จากนั้นเมื่อทำการคลิก IP ที่ต้องการดูจะพบกับ Tab Widget ซึ่งมีอยู่ 4 อันด้วยกัน คือ

- 1) การแสดงผลในส่วนหน้า Hardware จะแสดงเป็นลักษณะของ Table
- 2) การแสดงผลในส่วนหน้า Software จะแสดงเป็นลักษณะของ Expandable Listview ซึ่งแบ่งเป็น 2 หัวข้อใหญ่ๆ คือ Running Software และ Installed Software
- 3) การแสดงผลในส่วนหน้า Remote จะแสดงเป็นลักษณะของ Button 4 ปุ่ม
- 4) การแสดงผลในส่วนหน้า Gallery จะแสดงเป็นลักษณะของ GridView

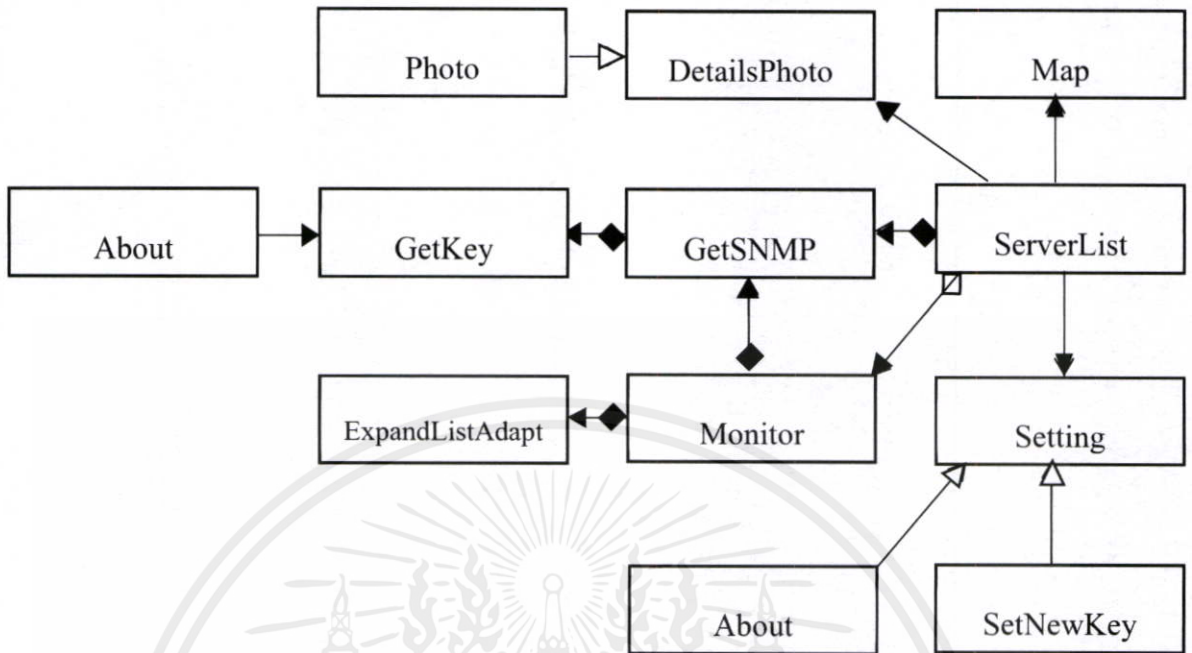
3.2 Use Case Diagram



รูปที่ 3.2 Use Case Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Class Diagram



รูปที่ 3.3 Class Diagram

ในโปรแกรมนี้ประกอบไปด้วย 12 คลาสคือ

- **About**
เป็นคลาสสำหรับแสดงรายละเอียดเกี่ยวกับแอปพลิเคชันและผู้พัฒนา
- **DetailsPhoto**
เป็นคลาสสำหรับแสดงรูปภาพของแต่ละเครื่องแม่ข่าย และลบรูปภาพ
- **ExpandListAdapter**
เป็นคลาสสำหรับการปรับแต่ง Expandable List ที่แสดงอยู่ในหน้าจอของรายละเอียด software
- **GetKey**
เป็นคลาสสำหรับการตั้งค่าคีย์ในตอนเริ่มแอปพลิเคชัน
- **Logo**
เป็นคลาสสำหรับการแสดง logo ตอนเข้าแอปพลิเคชัน
- **GetSNMP**
เป็นคลาสสำหรับการรับและส่งแพคเกจของ SNMP

- **Monitor**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคลาสสำหรับการจัดการข้อมูลและนำมาแสดงในส่วนของ Hardware และ Software นอกจากนี้ยังมีมีส่วนสำหรับส่งแพ็คเกจ ของการ monitor

- **Map**

เป็นคลาสสำหรับการแสดงตำแหน่งของเครื่องแม่ข่ายใน Google Map

- **Photo**

เป็นคลาสสำหรับถ่ายภาพของแอปพลิเคชัน

- **ServerList**

เป็นคลาสสำหรับหน้ารายการแสดง IP Address โดยจะมีการจัดการรายละเอียดต่างๆเช่น การแสดงสถานะ การเปลี่ยนชื่อเครื่อง การลบรายการ และการเพิ่มรายการ เป็นต้น

- **SetNewKey**

เป็นคลาสสำหรับการตั้งค่าคีย์ใหม่สำหรับแอปพลิเคชัน

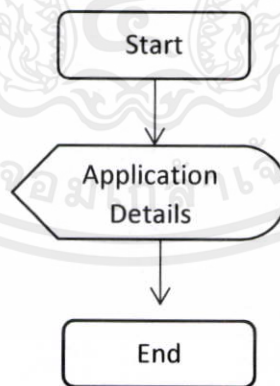
- **Setting**

เป็นคลาสสำหรับการเข้าหน้าตั้งค่าเพื่อที่จะไปตั้งค่าคีย์หรือหน้ารายละเอียดเกี่ยวกับแอปพลิเคชัน

3.4 Flow Chart

Flow Chart ของแต่ละคลาส

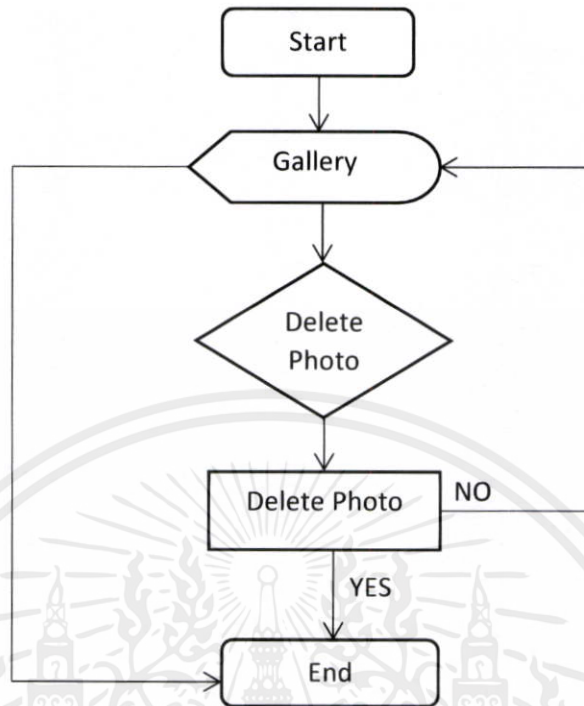
1. About



รูปที่ 3.4 Flow Chart ของ Class About

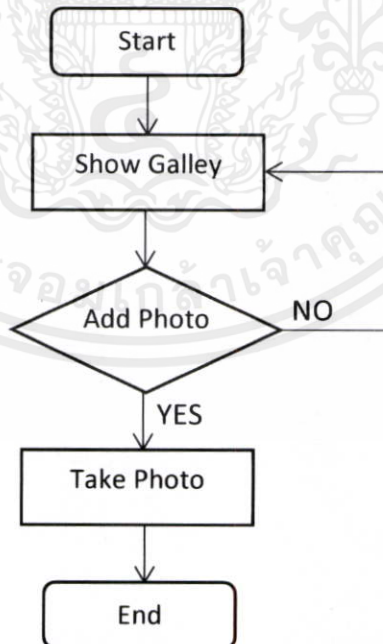
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. DetailsPhoto



รูปที่ 3.5 Flow Chart ของ Class DetailsPhoto

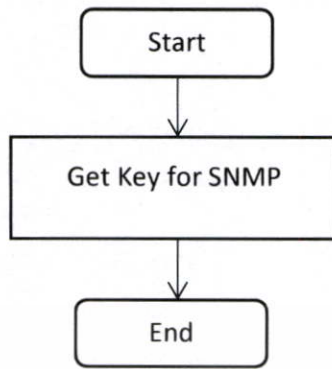
3. Photo



รูปที่ 3.6 Flow Chart ของ Class Photo

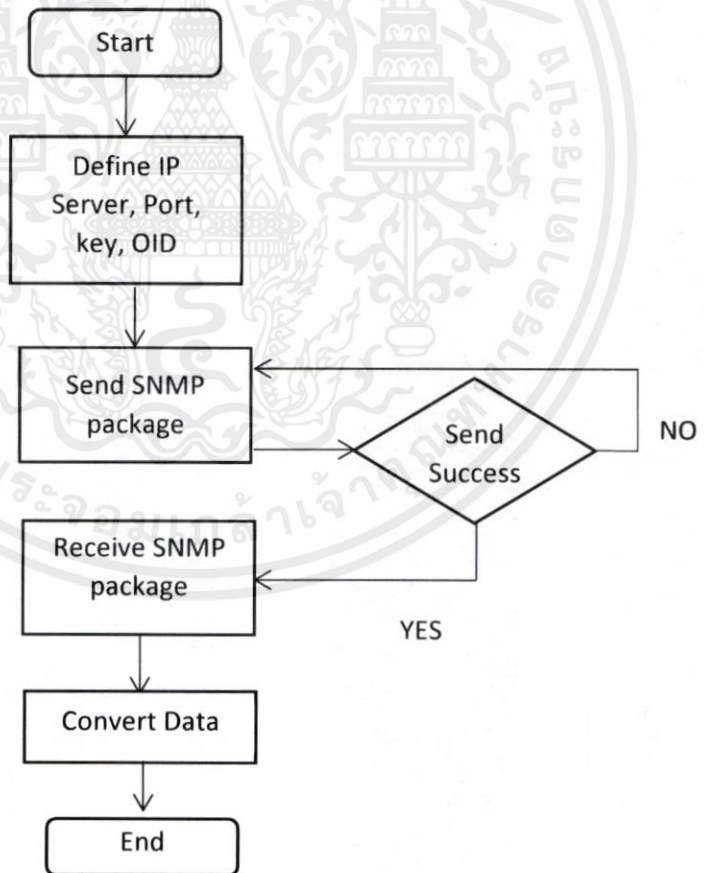
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกสิ่งนี้ออกไปและต้องขออนุญาตเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. GetKey



รูปที่ 3.7 Flow Chart ของ Class GetKey

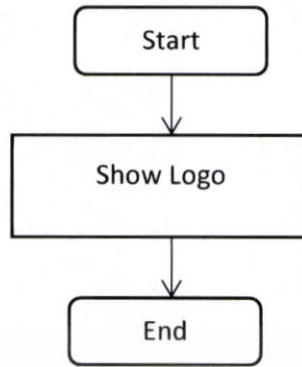
5. GetSNMP



รูปที่ 3.8 Flow Chart ของ Class GetSNMP

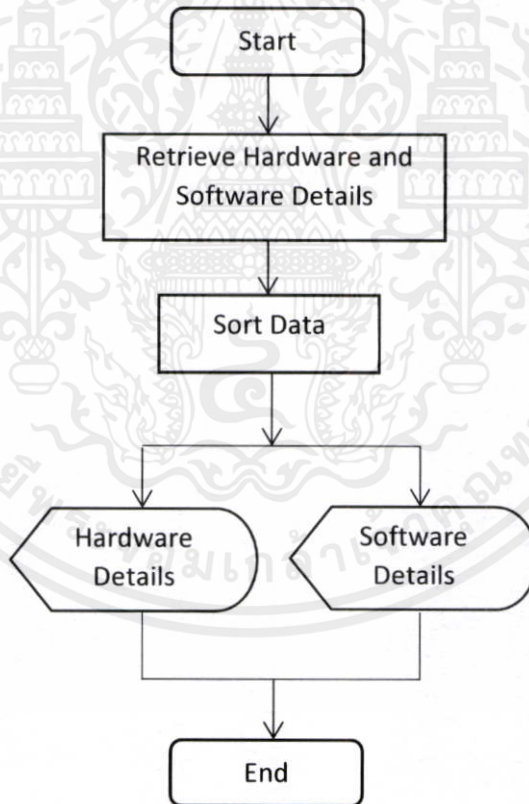
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

6. Logo



รูปที่ 3.9 Flow Chart ของ Class Logo

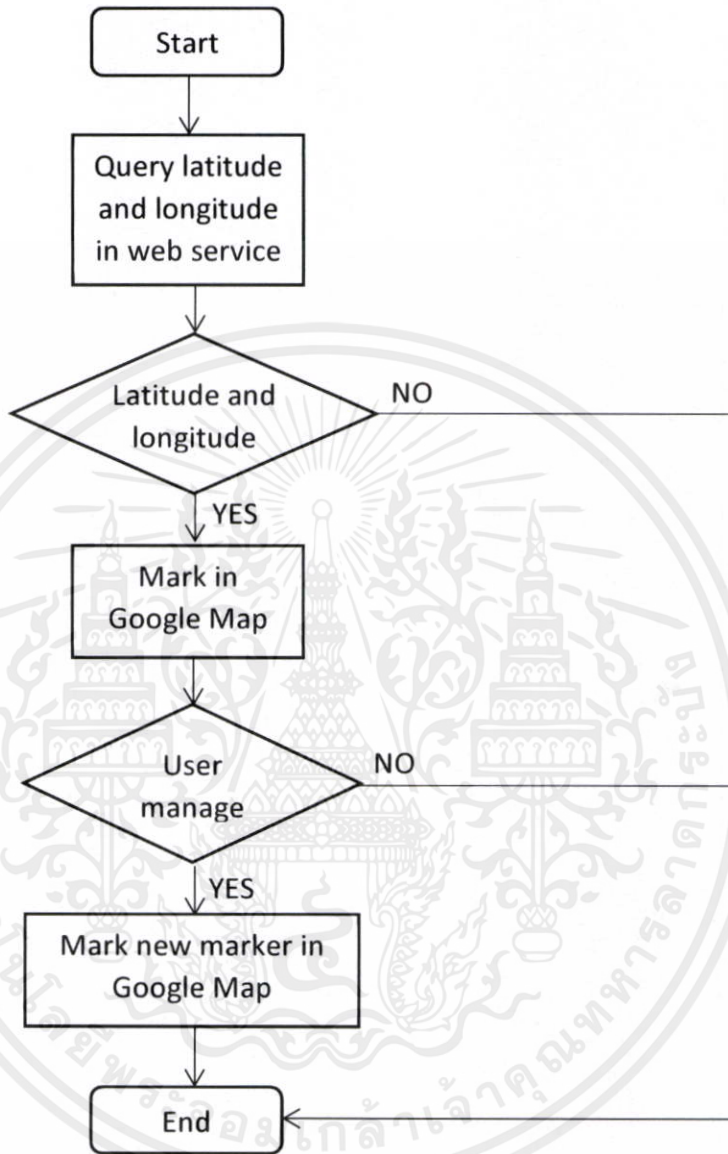
7. Monitor



รูปที่ 3.10 Flow Chart ของ Class Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

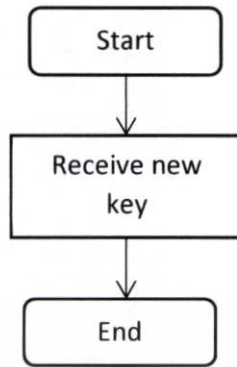
8. Map



รูปที่ 3.11 Flow Chart ของ Class Map

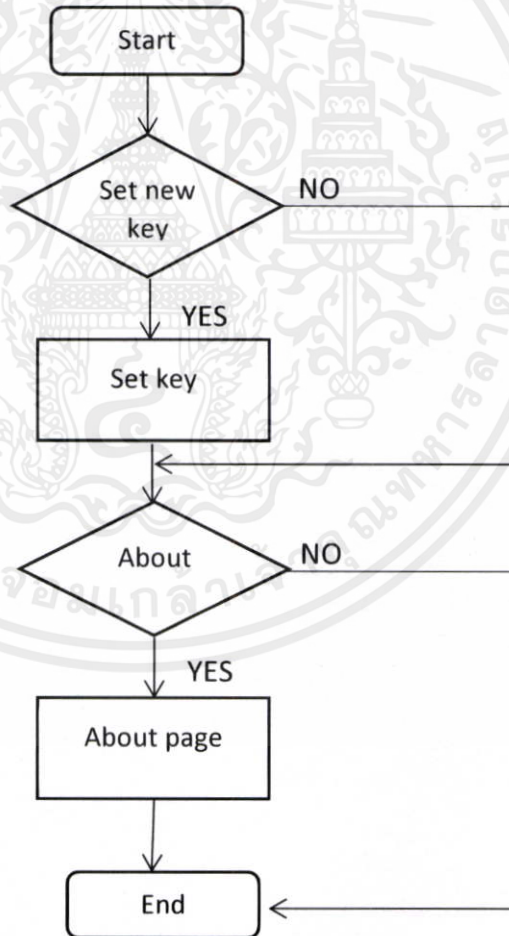
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. SetNewKey



รูปที่ 3.12 Flow Chart ของ Class SetNewKey

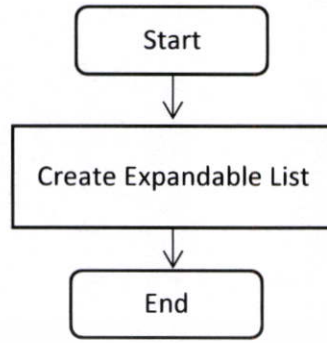
10. Setting



รูปที่ 3.13 Flow Chart ของ Class Setting

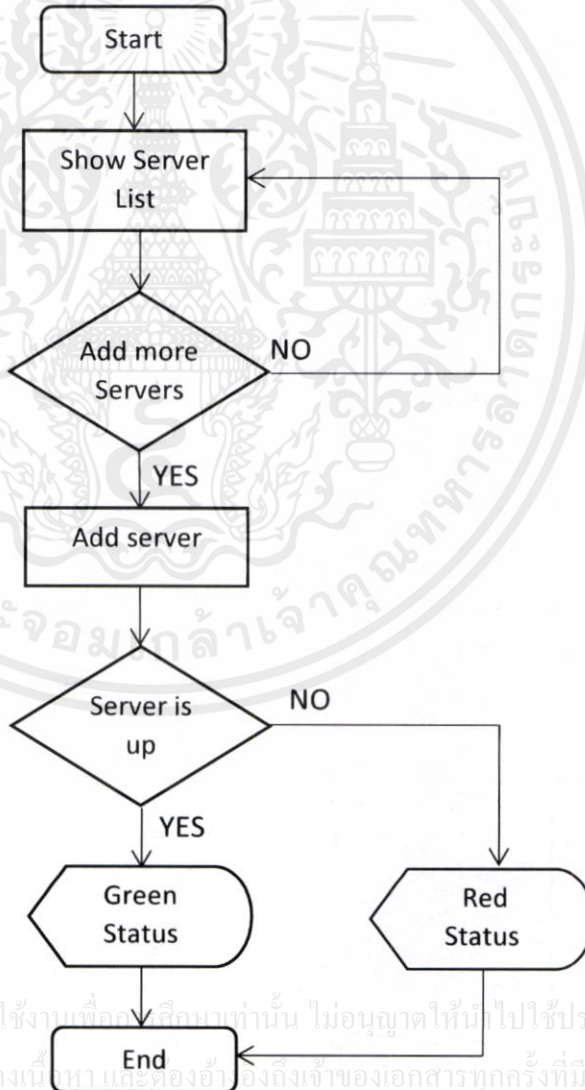
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเผยแพร่โดยไม่ได้รับอนุญาตของเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ExpandableList



รูปที่ 3.14 Flow Chart ของ Class ExpandableList

12. ServerList

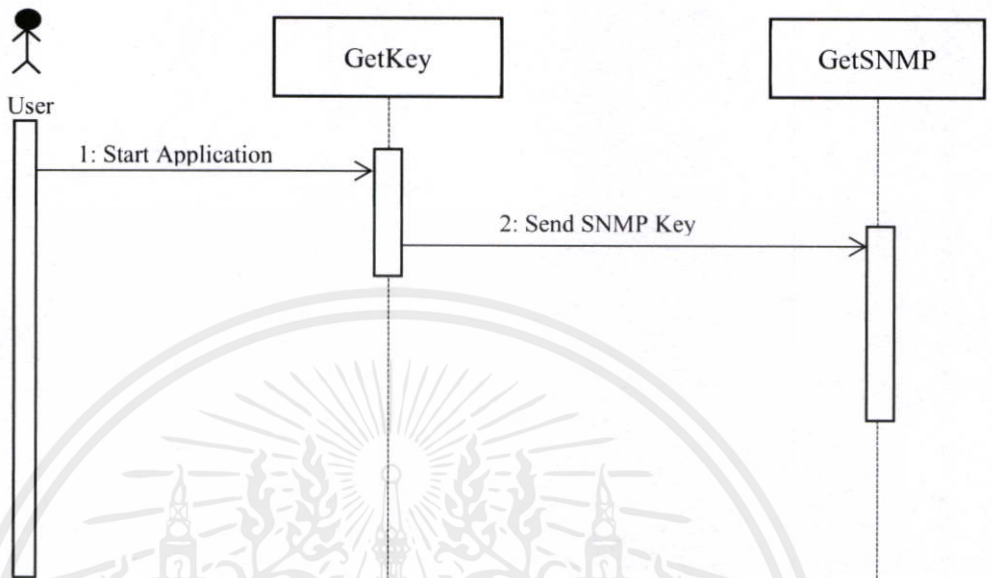


รูปที่ 3.15 Flow Chart ของ Class ServerList

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อสื่อความเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และข้อมูลอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Sequence Diagram

3.5.1 GetKey Sequence Diagram

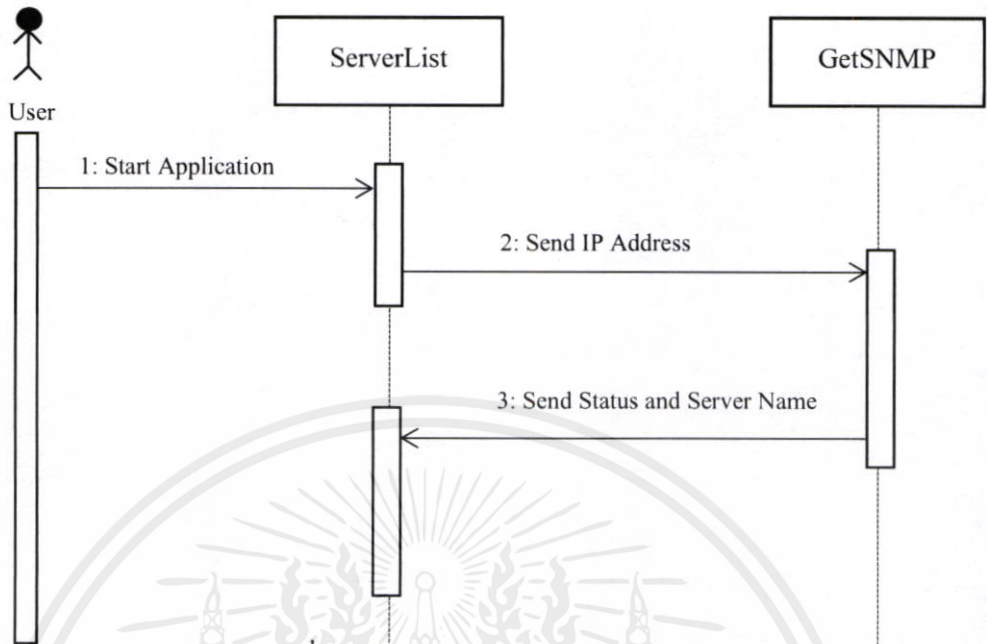


รูปที่ 3.16 GetKey Sequence Diagram

- 1) เมื่อผู้ใช้เปิดแอปพลิเคชันในครั้งแรก จะมีหน้าต่างขึ้นมาให้ผู้ใช้ทำการใส่คีย์ลงไป (คลาส GetKey) เพื่อให้สามารถใช้งาน SNMP service ได้
- 2) โปรแกรมจะนำคีย์ไปใช้ในคลาส GetSNMP เพื่อเริ่มต้นการส่งแพคเกจไปยังเครื่องเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 ServerList Sequence Diagram

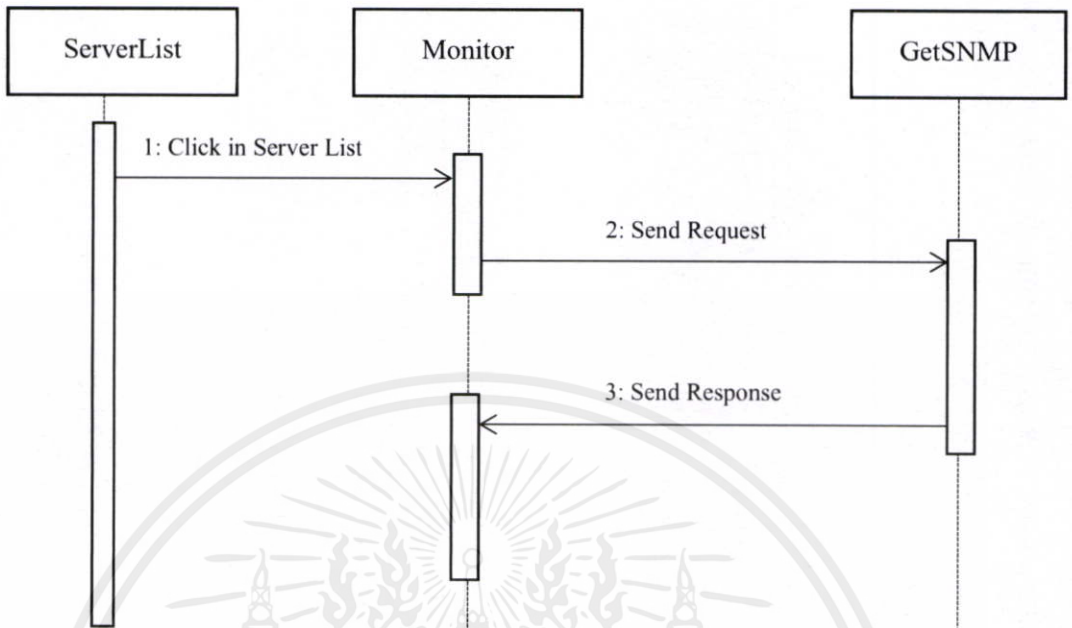


รูปที่ 3.17 ServerList Sequence Diagram

- 1) เมื่อผู้ใช้เปิดแอปพลิเคชัน จะเข้าไปยังหน้ารายการเครื่องเซิร์ฟเวอร์ (คลาส ServerList) ซึ่งในส่วนนี้จะมีสถานะบอกว่าเครื่องเซิร์ฟเวอร์นั้นเปิดการใช้งานอยู่หรือไม่
- 2) โปรแกรมจะส่งแพ็คเกจ โดยจะส่งเป็น IP Address ไปยังคลาส GetSNMP เพื่อดูว่าสถานะของเครื่องเซิร์ฟเวอร์นั้นเป็นอย่างไร
- 3) คลาส GetSNMP จะทำการส่งแพ็คเกจกลับมายังคลาส ServerList เพื่อแสดงผลสถานะของเครื่องเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 Monitor Sequence Diagram

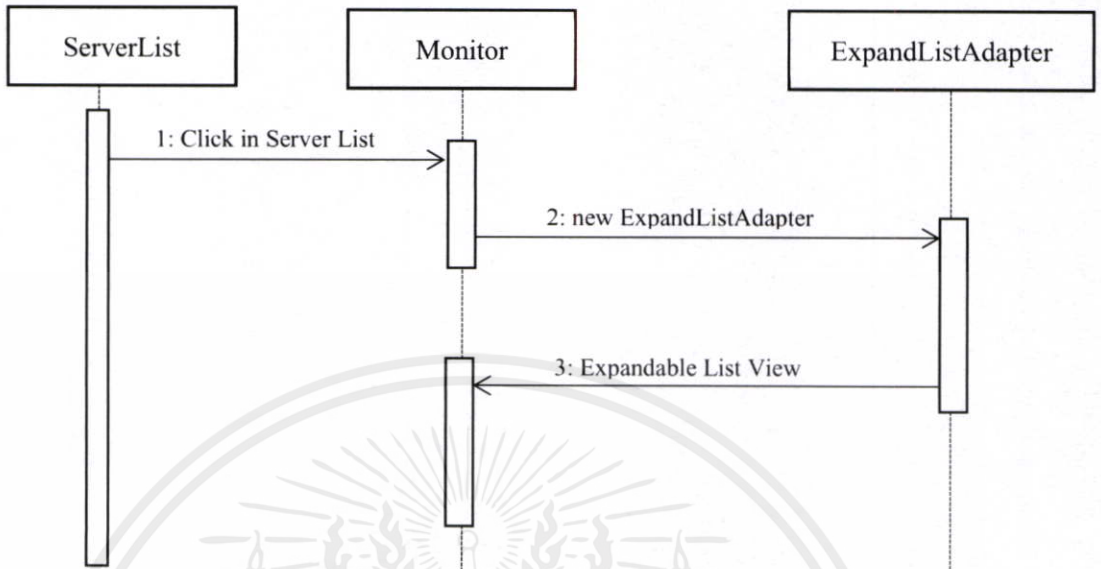


รูปที่ 3.18 Monitor Sequence Diagram

- 1) เมื่อผู้ใช้คลิกที่รายชื่อเซิร์ฟเวอร์ในหน้าคลาส ServerList เลือกเมนู monitor โปรแกรมจะแสดงผลหน้าคลาส Monitor เพื่อแสดงรายละเอียดเกี่ยวกับ Hardware, Software ต่างๆ
- 2) จากนั้นคลาส monitor จะทำการส่งแพ็คเกจไปยังคลาส GetSNMP เพื่อร้องขอข้อมูลรายละเอียดของเครื่องเซิร์ฟเวอร์
- 3) คลาส GetSNMP จะส่งแพ็คเกจเกี่ยวกับรายละเอียดเซิร์ฟเวอร์กลับมายังคลาส monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 ExpandListAdapter Sequence Diagram

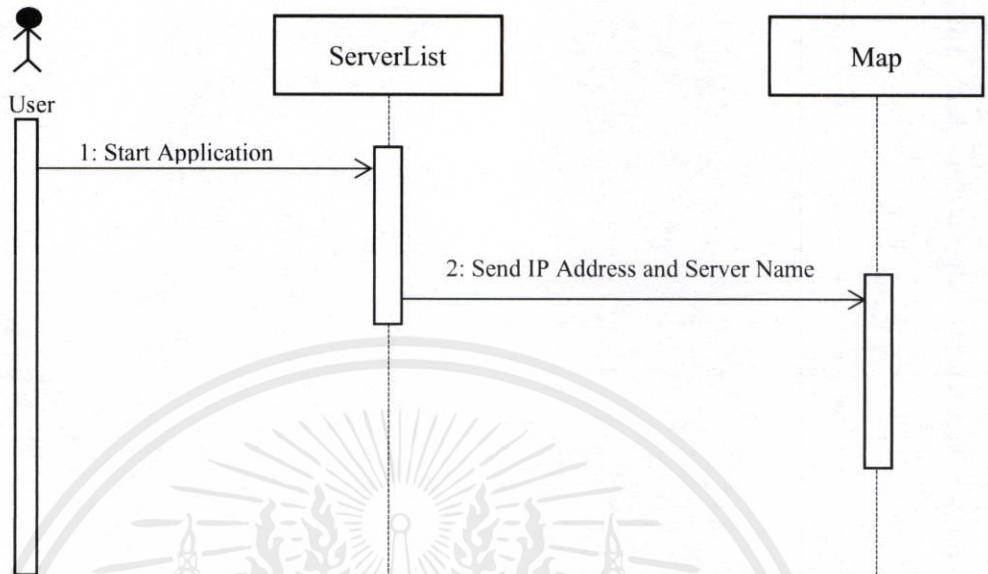


รูปที่ 3.19 ExpandListAdapter Sequence Diagram

- 1) เมื่อผู้ใช้คลิกที่รายชื่อเซิร์ฟเวอร์ในหน้าคลาส ServerList และเลือกเมนู monitor โปรแกรมจะแสดงผลหน้าคลาส Monitor เพื่อแสดงรายละเอียดเกี่ยวกับ Hardware, Software ต่างๆ
- 2) ในส่วนของการแสดงผล Software นั้นจะมีการแสดงผลเป็นลักษณะ Expandable List ซึ่งจะมีคลาส ExpandListAdapter มาช่วยในการสร้าง List
- 3) คลาส ExpandListAdapter ทำการสร้าง Expandable List ขึ้นมาในหน้าแสดงผล software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.5 Map Sequence Diagram

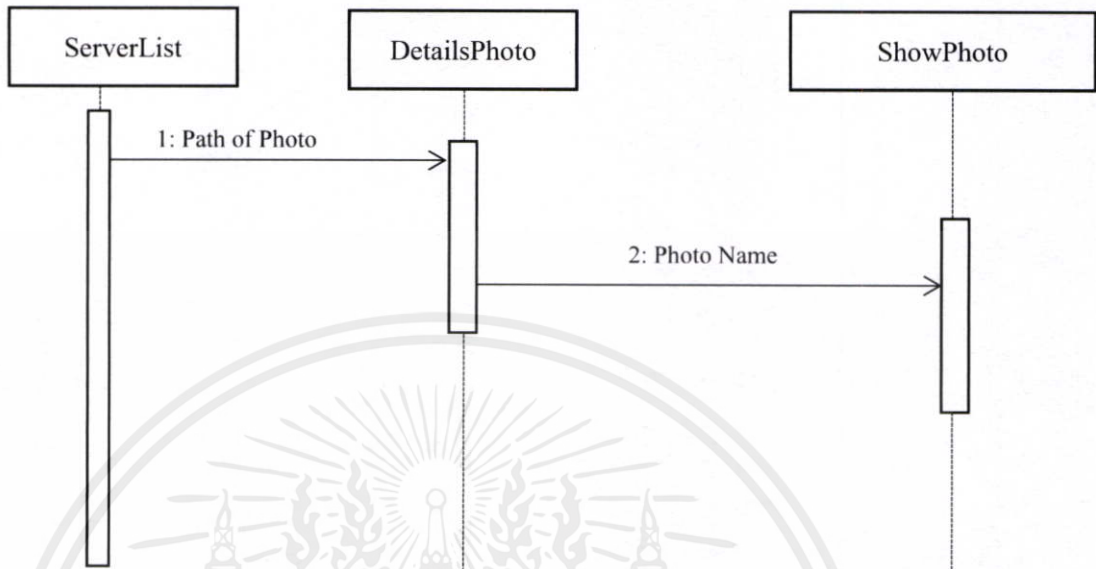


รูปที่ 3.20 Map Sequence Diagram

- 1) เมื่อผู้ใช้เปิดแอปพลิเคชันจะเข้าไปยังหน้ารายการเครื่องเซิร์ฟเวอร์ (คลาสServerList) ซึ่งในส่วนนี้จะมีสถานะบอกว่าเครื่องเซิร์ฟเวอร์นั้นเปิดการใช้งานอยู่หรือไม่
- 2) เมื่อผู้ใช้ทำการคลิกไปยังหน้าแผนที่ คลาส ServerList จะรายชื่อ IP Address และ Host Name ของเซิร์ฟเวอร์แต่ละอันไปยังคลาส Map เพื่อนำไปหาตำแหน่งของเครื่องเซิร์ฟเวอร์นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6 DetailsPhoto Sequence Diagram



รูปที่ 3.21 DetailsPhoto Sequence Diagram

- 1) เมื่อผู้ใช้คลิกที่รายชื่อเซิร์ฟเวอร์ในหน้าคลาส ServerList เลือกเมนู gallery โปรแกรมจะแสดงผลหน้าคลาส DetailsPhoto ก็จะแสดงภาพของเครื่องเซิร์ฟเวอร์นั้นๆ ที่ผู้ใช้ได้ถ่ายภาพไว้เป็นลักษณะ Gridview โดยจะแสดงรูปภาพตาม IP Address ของเครื่องนั้นๆ ซึ่งคลาส ServerList จะส่ง path ของไฟล์เคอร์รูปภาพใน IP Address นั้นๆ ไป
- 2) เมื่อผู้ใช้ทำการคลิกที่แต่ละรูปภาพ จะคลาส DetailsPhoto จะส่ง path ของรูปภาพนั้นๆ เพื่อเปิดรูปภาพนั้นขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

แอปพลิเคชันนี้พัฒนาโดยโปรแกรม Eclipse IDE ซึ่งเป็นเครื่องมือที่ช่วยในการพัฒนาซอฟต์แวร์ที่สามารถประยุกต์ใช้กับ Android Development Kit (ADK) และ Android Software Development Kit (Android SDK) และมีโปรแกรมจำลอง Genymotion ที่ใช้ในการแสดงผลการทำงานของแอปพลิเคชันที่พัฒนาขึ้นได้

โดยการทดลองนี้นอกจากทำการทดลองในโปรแกรมจำลอง Genymotion แล้ว ยังได้มีการนำไปทดลองในอุปกรณ์จริง โดยอุปกรณ์ที่นำมาทำการทดลอง คือ

โทรศัพท์มือถือยี่ห้อ ZTE รุ่น V887 ซึ่งมีรายละเอียดของอุปกรณ์ดังนี้

- 1) ระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 4.1
- 2) หน่วยประมวลผล : Dual Core ความเร็ว : 1 GHz
- 3) หน่วยความจำ 4 GB (ตัวเครื่อง) RAM 512MB
- 4) การ์ดหน่วยความจำ microSD - สูงสุด 32 GB

โดยขอบเขตของการทดลองจะทำการทดลองในหัวข้อต่างๆดังนี้

4.1 ทดลองฟังก์ชันการตั้งค่าคีย์เริ่มต้น

ทดลองสร้างฟังก์ชันตั้งค่าคีย์เริ่มต้น โดยใช้สำหรับตั้งค่าให้ตรงกับ Security ของ SNMP Service ของเครื่องแม่ข่าย เปรียบเสมือนเป็นรหัสผ่านเพื่อให้แอปพลิเคชันสามารถติดต่อกับเครื่องแม่ข่ายได้

วิธีการดำเนินการ

- 1) ใช้ภาษา Java และ XML ในการเขียนโปรแกรม โดยใช้ EditText ในการรับค่าคีย์เข้ามาและทำการบันทึกไว้การตั้งค่าคีย์โดยมีโค้ดแสดงดังนี้

```
ok.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        r = key.getText().toString();  
        try {  
            savedata();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
});
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องแจ้งให้ทราบล่วงหน้าทุกครั้งที่มีการนำไปใช้

```

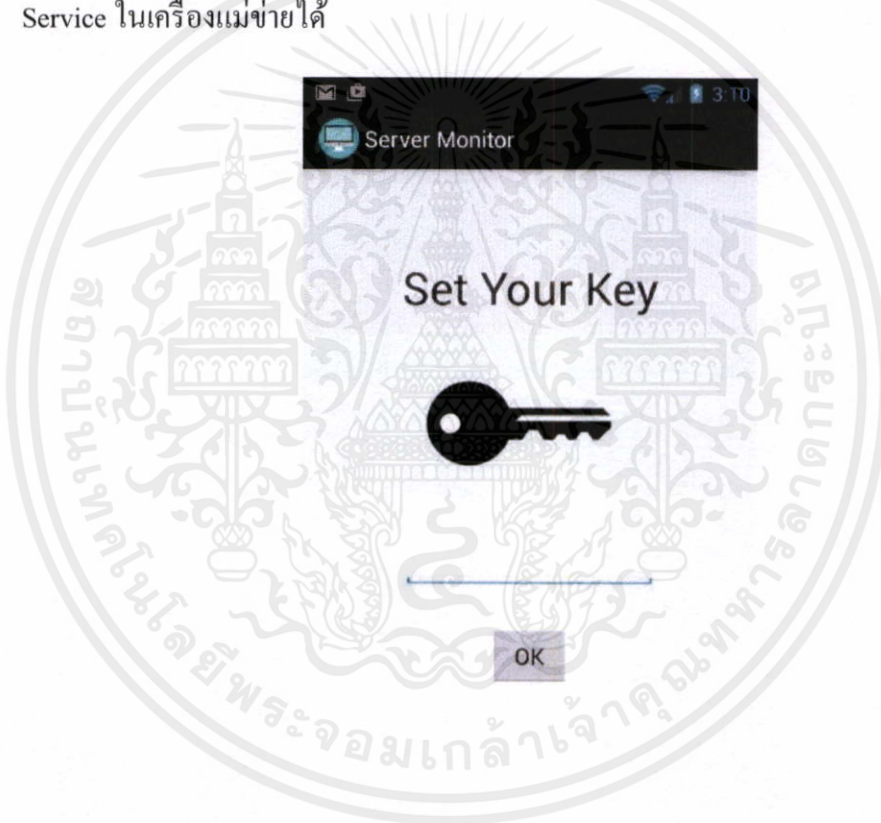
Intent intent1 = new Intent(GetKey.this,
                           ServerList.class);
startActivity(intent1);
}
});

```

- 2) หลังจากตรวจสอบโค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องเครื่องโทรศัพท์มือถือ
- 3) รันโปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

จากการทดลองโปรแกรมสามารถตั้งค่าคีย์ให้ตรงกับค่า Community ของ SNMP Service ในเครื่องแม่ข่ายได้



รูปที่ 4.1 แสดงการตั้งค่าคีย์เริ่มต้น

4.2 ทดลองฟังก์ชันการเพิ่มเครื่องแม่ข่ายที่ต้องการตรวจสอบ

ทดลองสร้างฟังก์ชันการเพิ่มเครื่องแม่ข่ายที่ต้องการตรวจสอบ โดยการทำเป็นลิสต์ของเครื่องต่างๆ สามารถเพิ่มเครื่องที่ต้องการตรวจสอบได้สูงสุดถึง 10 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อเผยแพร่ให้ผู้อื่นใช้โดยไม่ได้รับอนุญาตเป็นการฝ่าฝืนหรือไม่ว่ากรณีใดๆ พึงสนธิสัญญาที่มีผลบังคับใช้และต้องปฏิบัติตามของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการดำเนินการ

- 1) ใช้ภาษา Java และ XML ในการเขียนโปรแกรม โดยใช้ ListView และ SimpleAdapter ในการแสดงรายการ และใช้ class getSNMP ในการตรวจสอบว่าเครื่องแม่ข่ายนั้น ทำงานอยู่หรือไม่โดยมีโค้ดแสดงดังนี้

```
mAdapter = new SimpleAdapter(this, arlist, R.layout.list_single, new
    String[] {"title", "subtitle", "status" }, new
int[] {
    R.id.txtTitle, R.id.txtDetail, R.id.image
})
    setListAdapter(mAdapter);
listView = getListView();

class getName extends AsyncTask<Void, Void, String> {
    protected String doInBackground(Void... something) {
        try {
            for (int i = 0; i < count; i++) {
                getSNMP1[i].sendSnmRequest(valueip[i]);
                sysup1[i] = getSNMP1[i].getsysup();
                sysname1[i] =
                getSNMP1[i].getsysname();
                if (sysup1[i] == null) {
                    sysup1[i] = 0;
                }
                if (sysname1[i] == null) {
                    sysname1[i] = "null";
                }
            }
        } catch (Exception e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }

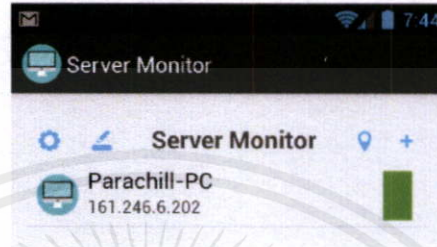
        if (buttonclick == 1) {
            try {
                addr =
                InetAddress.getBy_name(valueip[count]);
            } catch (UnknownHostException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        return null;
    }
}
```

- 2) หลังจากตรวจสอบโค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องเครื่องโทรศัพท์มือถือ

- 3) รันโปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

จากการทดลองโปรแกรมสร้างลิสต์ของเครื่องแม่ข่ายที่ผู้ใช้เพิ่มเข้าไปโดยใช้ IP Address ซึ่งแต่ละเครื่องจะมีแถบสีสถานะเพื่อเป็นตัวบอกว่าเครื่องนี้กำลังออนไลน์อยู่หรือไม่ และเครื่องที่กำลังออนไลน์อยู่มีชื่อเครื่องว่าอย่างไร และสามารถแก้ไขชื่อเครื่องเพื่อให้ง่ายต่อการจดจำของผู้ใช้ และสามารถเชื่อมต่อไปยังหน้าแผนที่ได้อีกด้วย



รูปที่ 4.2 แสดงหน้ารับชื่อเครื่องหรือหมายเลขไอพีที่ต้องการตรวจสอบ

4.3 ทดลองฟังก์ชันการเปลี่ยนชื่อเครื่องแม่ข่ายที่ต้องการ

ทดลองสร้างฟังก์ชันการเปลี่ยนชื่อเครื่องแม่ข่ายที่ต้องการ โดยการกดค้างที่รายชื่อเครื่องแม่ข่ายที่ต้องการเปลี่ยนชื่อ และใช้ EditText ในการรับข้อมูลใหม่เข้ามา และนำไปแสดง

วิธีการดำเนินการ

- 1) ใช้ภาษา Java และ XML ในการเขียนโปรแกรม โดยใช้ EditText ในการรับค่า และกำหนดค่าใหม่ใน Array ที่เก็บชื่อไว้

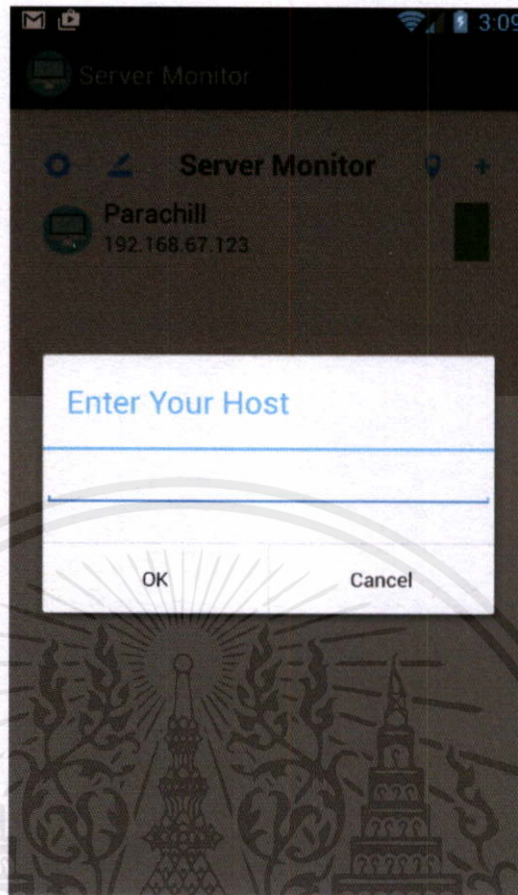
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นกรณีพิเศษที่ได้รับอนุญาตจากทางผู้จัดทำเอกสาร

```

listView.setOnItemClickListener(new OnItemClickListener() {
    public boolean onItemClick(AdapterView<?>
parent, View v, int position, long id) {
        final int position_ = position;

```

รูปที่ 4.3 แสดงหน้าการเปลี่ยนชื่อเครื่องแม่ข่ายที่ต้องการ

4.4 ทดลองฟังก์ชันแสดงตำแหน่งบน Google Map

ทดลองสร้างฟังก์ชันแสดงตำแหน่งของเครื่องแม่ข่ายบน Google Map

วิธีการดำเนินการ

- 1) ใช้ภาษา Java และ XML ในการเขียนโปรแกรม โดยใช้ไลบรารี (Library) Google Map ในการแสดงผลแผนที่ โดยค้นหา latitudes และ longitudes ในเว็บเซอร์วิส จากนั้นนำมาพล็อตลงบนแผนที่ โดยมีโค้ดแสดงดังนี้

```
private void initilizeMap() {
    if (googleMap == null) {
        googleMap = ((MapFragment)
            getFragmentManager().
            findFragmentById(.id.map)).getMap();
        // check if map is created successfully or not
        if (googleMap == null) {
            Toast.makeText(getApplicationContext(),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำเอกสารนี้ไปเผยแพร่หรือแจกจ่ายแก่บุคคลอื่นโดยไม่ได้รับอนุญาต

```

        "Sorry! unable to create maps",
        Toast.LENGTH_SHORT)
        .show();
    }
}

public String getLatLng(String ip) {

    Log.d("doInBackground", "getLatLng now");

    String uri = "http://ipinfo.io/" + ip + "/loc";
    HttpGet httpGet = new HttpGet(uri);
    HttpClient client = new DefaultHttpClient();
    HttpResponse response;
    StringBuilder stringBuilder = new StringBuilder();

    try {
        response = client.execute(httpGet);
        HttpEntity entity = response.getEntity();
        InputStream stream = entity.getContent();
        int b;
        while ((b = stream.read()) != -1) {
            stringBuilder.append((char) b);
        }
        temp = stringBuilder.toString();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return temp;
}

```

- 2) หลังจากตรวจสอบโค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องโทรศัพท์มือถือรัน โปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

สามารถแสดงตำแหน่งของเครื่องแม่ข่ายบน Google Map ได้ และเมื่อทราบว่าตำแหน่งที่ระบุมีความคลาดเคลื่อนผู้ใช้สามารถระบุตำแหน่งที่ถูกต้องใหม่ได้ด้วยตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงตำแหน่งของเครื่องแม่ข่ายบน Google Map

4.5 ทดลองสร้างฟังก์ชันในการเข้าดูเครื่องแม่ข่าย

ทดลองสร้างฟังก์ชันในการเข้าดูเครื่องแม่ข่ายแต่ละเครื่อง โดยมี 2 เมนู ให้เลือก คือ การเข้าชมภาพถ่ายใน Gallery และ การ Monitor เพื่อดูรายละเอียดต่างๆของเครื่องแม่ข่าย

วิธีการดำเนินการ

- 1) ใช้ภาษา Java และ XML ในการเขียน โปรแกรม โดยใช้ alert dialog ในการแสดงเมนูต่างๆ โดยมีโค้ดแสดงดังนี้

```
AlertDialog.Builder builderSingle = new AlertDialog.Builder(
    ServerList.this);
```

```
final ArrayAdapter<String> arrayAdapter = new
ArrayAdapter<String>(ServerList.this,
android.R.layout.select_dialog_item);
arrayAdapter.add("Gallery");
arrayAdapter.add("Monitor");
```

```
builderSingle.setNegativeButton("cancel",
new DialogInterface.OnClickListener() {
```

```
@Override สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
public void onClick(DialogInterface dialog, int which) {
    dialog.dismiss();
}
```

```
});
```

```

builderSingle.setAdapter(arrayAdapter,new
DialogInterface.OnClickListener() {

@Override
public void onClick(DialogInterface dialog, int which) {

String strName = arrayAdapter.getItem(which);

    if(strName == "Gallery"){

String a = (String)
arrlist.get(position_).get("subtitle");

Intent intent2 = new Intent(ServerList.this, Photo.class);
intent2.putExtra("send_ipaddress", a);
startActivity(intent2);

    }
    else if(strName == "Monitor"){

String a = (String)
arrlist.get(position_).get("subtitle");

Intent intent1 = new Intent(ServerList.this, Monitor.class);
intent1.putExtra("send_ipaddress", a);
intent1.putExtra("send_run", 1);
startActivity(intent1);
    });
builderSingle.show();
}
}

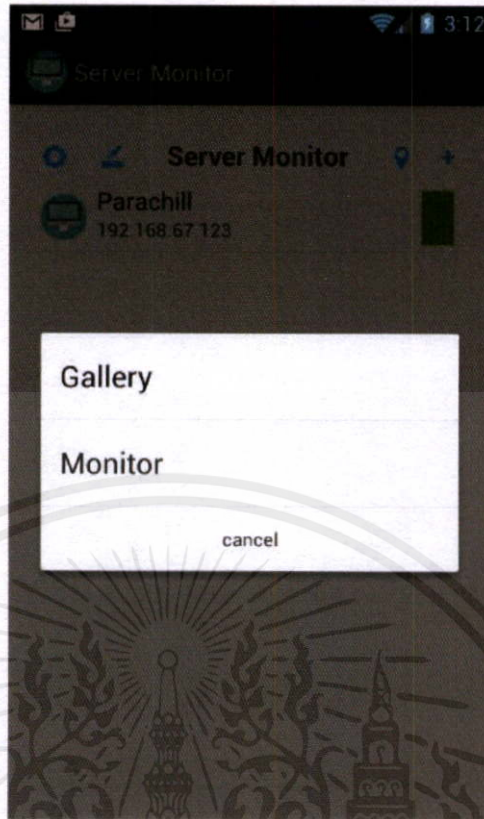
```

- 2) หลังจากตรวจสอบโค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องเครื่องโทรศัพท์มือถือ
- 3) รันโปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

จากการทดลองโปรแกรมสร้างฟังก์ชันขึ้นมาให้เลือก 2 เมนู คือ Gallery และ Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงฟังก์ชันการเข้าดูเครื่องแม่ข่าย

4.6 ทดลองสร้างแถบเมนูการทำงานหลัก

ทดลองสร้างแถบเมนูการทำงานหลักของแอปพลิเคชันตรวจสอบสภาพเครื่องแม่ข่าย โดยแบ่งการแสดงผลออกเป็น 4 ส่วน คือ ส่วนแสดงข้อมูลฮาร์ดแวร์ (Hardware) ส่วนแสดงข้อมูลซอฟต์แวร์ (Software) ส่วนควบคุมเครื่องแม่ข่าย (Remote Server) และส่วนแสดงรูปภาพ (Gallery)

วิธีการดำเนินการ

- 1) ใช้ภาษา Java และ XML ในการเขียนโปรแกรม โดยใช้ widget TabHost ในการแสดงเมนูต่างๆ โดยมีโค้ดแสดงดังนี้

```
final TabHost tabHost = (TabHost) findViewById(R.id.tabhost);
tabHost.setup();
```

```
TabSpec spec1 = tabHost.newTabSpec("Tab1");
spec1.setContent(R.id.tab1);
spec1.setIndicator("", null);
```

```
TabSpec spec2 = tabHost.newTabSpec("Tab2");
spec2.setContent(R.id.tab2);
spec2.setIndicator("", null);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

```

TabSpec spec3 = tabHost.newTabSpec("Tab3");
spec3.setContent(R.id.tab3);
spec3.setIndicator("", null);

tabHost.addTab(spec1);
tabHost.addTab(spec2);
tabHost.addTab(spec3);

setTabIcon(tabHost, 0, R.drawable.hw);
setTabIcon(tabHost, 1, R.drawable.sw);
setTabIcon(tabHost, 2, R.drawable.rm);

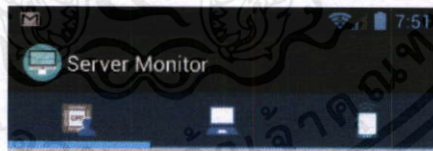
tabHost.setOnTabChangeListener(new OnTabChangeListener() {
    @Override
    public void onTabChanged(String arg0) {
        setTabColor(tabHost);}
});
setTabColor(tabHost);

```

- 2) หลังจากตรวจสอบโค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องโทรศัพท์มือถือ
- 3) รันโปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

จากการทดลองโปรแกรมสร้างแถบเมนูขึ้นมา 4 แถบ ให้สามารถเลือกหน้าแสดงข้อมูลในเมนูต่างๆ ได้คือเมนู Hardware, Software และ remote



รูปที่ 4.6 แสดงแถบเมนูการทำงานหลัก

4.7 ทดลองการใช้ฟังก์ชันแสดงรายละเอียดฮาร์ดแวร์ (Hardware System Description)

ทดลองการใช้ฟังก์ชันแสดงรายละเอียดฮาร์ดแวร์ของเครื่องแม่ข่ายคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการดำเนินการ

- 1) ใช้ภาษา Java ในการเขียนโปรแกรม ใช้ package UDP ในการรับและส่งข้อมูล SNMP ผ่านทาง port 161 ซึ่งจะระบุค่า OID ที่ต้องการทราบค่าลงไปโดยมีโค้ดแสดงดังนี้

```

ip = "udp:" + ip_sv + "/161";
TransportMapping transport = new DefaultUdpTransportMapping();
snmp = new Snmp(transport);
USM usm = new USM(SecurityProtocols.getInstance(), new
OctetString(MPv3.createLocalEngineID()),0);
SecurityModels.getInstance().addSecurityModel(usm);
transport.listen();

sysDescr = getAsString(new OID(".1.3.6.1.2.1.1.1.0"));
sysUpdate = getAsString(new OID(".1.3.6.1.2.1.1.3.0"));
driveC = getAsString(newOID(".1.3.6.1.2.1.25.2.3.1.3.1"));
driveD = getAsString(newOID(".1.3.6.1.2.1.25.2.3.1.3.2"));
ProLoad = getAsLong(new OID(".1.3.6.1.2.1.25.3.3.1.2"));
RAM = getAsInt(new OID(".1.3.6.1.2.1.25.2.2.0"));
Process = getAsLong(new OID(".1.3.6.1.2.1.25.1.6.0"));
StoCapacity1 = getAsInt(new OID(".1.3.6.1.2.1.25.2.3.1.5.1"));
StoCapacity2 = getAsInt(new OID(".1.3.6.1.2.1.25.2.3.1.5.2"));
StoAllo1 = getAsInt(new OID(".1.3.6.1.2.1.25.2.3.1.6.1"));
StoAllo2 = getAsInt(new OID(".1.3.6.1.2.1.25.2.3.1.6.2"));

```

- 2) หลังจากตรวจสอบ โค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องเครื่องโทรศัพท์มือถือ
- 3) รัน โปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

จากการทดลองโปรแกรมเรียกแสดงข้อมูลได้ดังนี้

- 1) ชื่อเครื่องแม่ข่าย (System Name)
- 2) อับเตศระบบปฏิบัติการครั้งล่าสุด
- 3) พื้นที่ของฮาร์ดดิสก์ทั้งหมด (Hard disk Space) พื้นที่ของฮาร์ดดิสก์ที่ใช้ไป (Hard disk use space)
- 4) แสดงค่า RAM ของเครื่องคอมพิวเตอร์
- 5) แสดงค่า Processes ที่เครื่องคอมพิวเตอร์ใช้อยู่
- 6) แสดงค่า Memory ที่เครื่องคอมพิวเตอร์ใช้อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงรายละเอียดฮาร์ดแวร์

4.8 ทดลองการใช้ฟังก์ชันแสดงข้อมูลซอฟต์แวร์ (Software Information)

ทดลองการใช้ฟังก์ชันแสดงข้อมูลซอฟต์แวร์ของเครื่องแม่ข่ายคอมพิวเตอร์

วิธีการดำเนินการ

- 1) ใช้ภาษา Java ในการเขียนโปรแกรม ใช้ package UDP ในการรับและส่งข้อมูล SNMP ผ่านทาง port 161 ซึ่งจะระบุค่า OID ที่ต้องการทราบค่าลงไปโดยมีโค้ดแสดงดังนี้

```
while (run) {
    if (Swrun.contains("1.3.6.1.2.1.25.4.2.1.3")) {
        run = false;
    }

    PDU pdu = new PDU();
    pdu.add(new VariableBinding(new OID("." +
Swrun)));
    pdu.setRequestID(new Integer32(1));
    pdu.setType(PDU.GETNEXT);

    ResponseEvent response = snmp.getNext(pdu, getTarget());
    PDU responsePDU = response.getResponse();
    if (Swrun.contains("1.3.6.1.2.1.25.4.2.1.2")) {
        Swrun1[count] = responsePDU.get(0).toValueString();
        count++;
    }
}
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

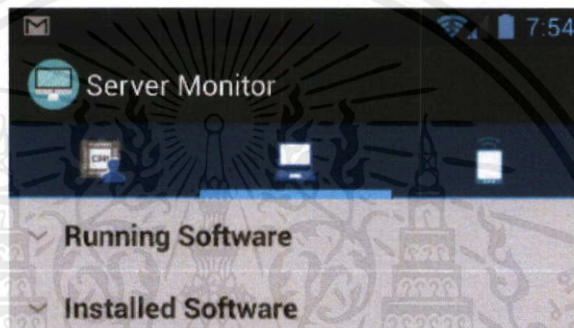
Swrun =
responsePDU.get(0).getOid().toString();
    }
}

```

- 2) หลังจากตรวจสอบโค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องเครื่องโทรศัพท์มือถือรันโปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

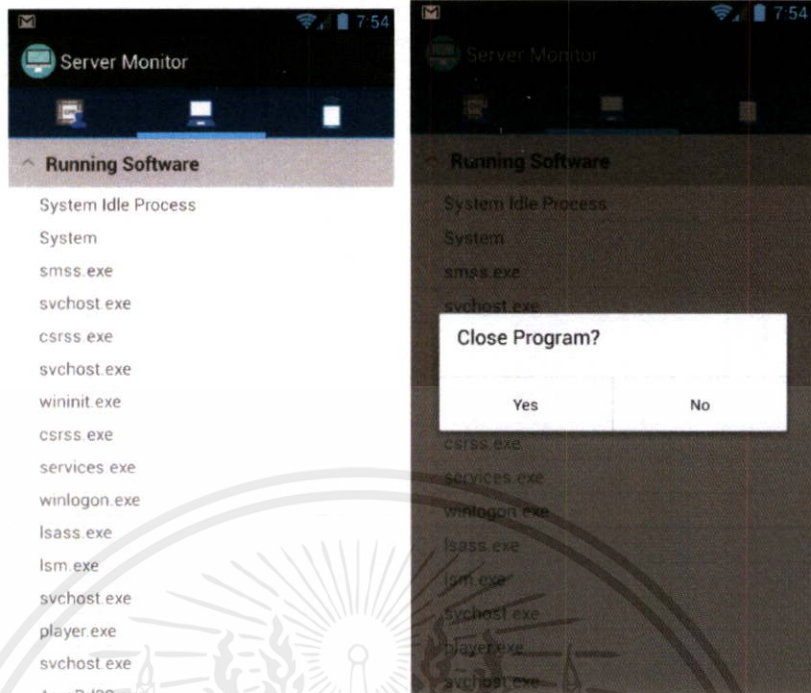
จากการทดลองโปรแกรมเรียกแสดงข้อมูลได้ดังนี้



รูปที่ 4.8 แสดงหัวข้อของข้อมูลซอฟต์แวร์ที่สนใจ

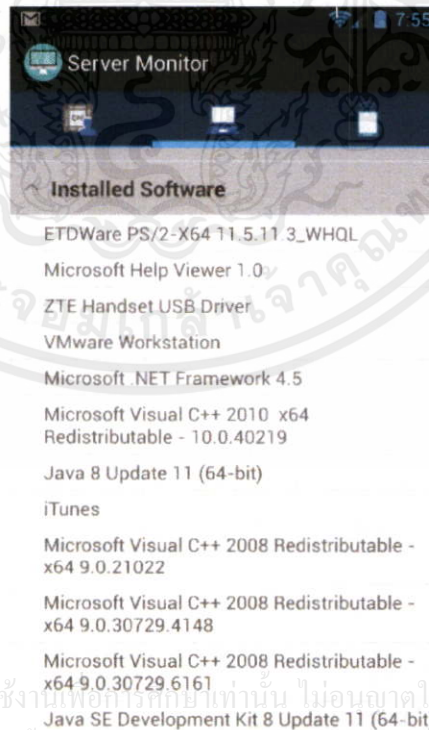
- 1) แสดงรายชื่อซอฟต์แวร์ที่กำลังทำงานอยู่บนเครื่องแม่ข่ายคอมพิวเตอร์ (Running Software) และสามารถสั่งปิดโปรแกรมที่กำลังทำงานอยู่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงรายชื่อซอฟต์แวร์ที่กำลังทำงานอยู่

- 2) แสดงรายชื่อซอฟต์แวร์ที่ได้ทำการ Install ไว้บนเครื่องแม่ข่ายคอมพิวเตอร์ (Installed Software)



รูปที่ 4.10 แสดงรายชื่อซอฟต์แวร์ที่ถูกติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่นอนญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาก่อนและห้องซึ่งจึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 ทดลองการใช้ฟังก์ชันควบคุมเครื่องแม่ข่าย (Remote Server)

ทดลองการใช้ฟังก์ชันสร้างคำสั่งเพื่อควบคุมเครื่องแม่ข่ายคอมพิวเตอร์

วิธีการดำเนินการ

- 1) ใช้ภาษา Java ในการเขียนโปรแกรม โดยใช้ Package UDP ผ่านทาง port 123 ส่งไปยังเครื่องแม่ข่าย จากนั้นจะมีโปรแกรมในเครื่องข่ายส่งการอีกครั้ง โดยที่โค้ดแสดงดังนี้

```
InetAddress serverAddr =
InetAddress.getBy_name(getip);

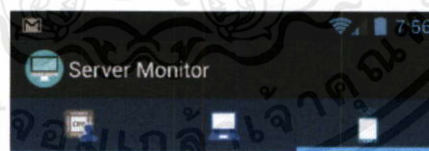
DatagramSocket socket = new DatagramSocket();
byte[] buf;
buf = (remote_cm).getBytes();
DatagramPacket packet = new
DatagramPacket(buf,buf.length,
serverAddr, SERVERPORT);

socket.send(packet);
```

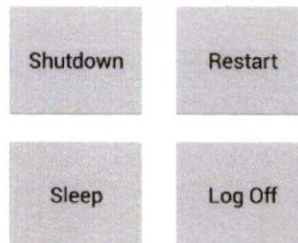
- 2) หลังจากตรวจสอบโค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องเครื่องโทรศัพท์มือถือรัน โปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

จากการทดลองโปรแกรมสามารถรีโมทคำสั่งไปได้ดังนี้



Remote Your Server



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.11 แสดงคำสั่งที่สามารถรีโมทไปยังเครื่องแม่ข่าย

4.10 ทดลองการใช้ฟังก์ชันแสดงรูปภาพเครื่องแม่ข่าย (Gallery)

ทดลองการใช้ฟังก์ชันแสดงรูปภาพของเครื่องแม่ข่ายคอมพิวเตอร์

วิธีการดำเนินการ

- 1) ใช้ภาษา Java และ XML ในการเขียนโปรแกรม ใช้ widget GridView ในการแสดงรูปภาพและแยกภาพเครื่องแม่ข่ายตาม IP Address โดยมีโค้ดแสดงดังนี้

```
final GridView gridView = (GridView) findViewById(R.id.gridview);
myImageAdapter = new ImageAdapter(this);
gridView.setAdapter(myImageAdapter);

String ExternalStorageDirectoryPath = Environment
    .getExternalStorageDirectory().getAbsolutePath();

File sdCard = Environment.getExternalStorageDirectory();
dir = new File(sdCard.getAbsolutePath() +
    "/Server_Monitor/photo/" + getip);
if(!dir.exists()){
    dir.mkdirs();
}
String targetPath = ExternalStorageDirectoryPath
    + "/Server_Monitor/photo/" + getip;
targetDirector = new File(targetPath);

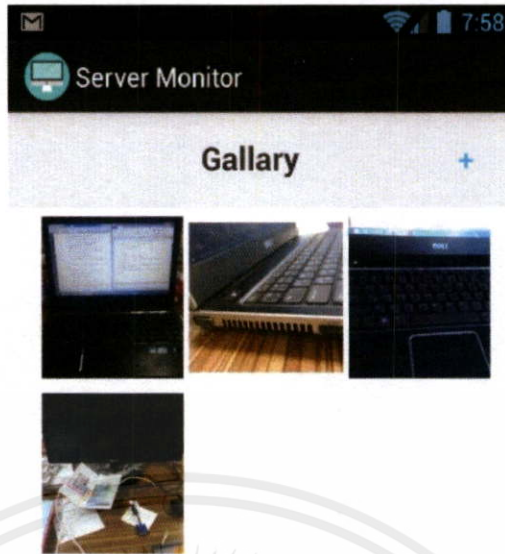
File[] files = targetDirector.listFiles();
for (File file : files) {
    myImageAdapter.add(file.getAbsolutePath());
}
```

- 2) หลังจากตรวจสอบโค้ดเสร็จเรียบร้อยแล้ว ก็ทำการเชื่อมต่อโทรศัพท์กับเครื่องคอมพิวเตอร์ จากนั้นทำการส่งออกโปรแกรมลงเครื่องเครื่องโทรศัพท์ที่มีठीอรันโปรแกรมที่เครื่องโทรศัพท์

ผลการทดลอง

จากการทดลองโปรแกรมเรียกแสดงรูปภาพที่ถูกบันทึกไว้ และสามารถเพิ่มรูปภาพได้โดยการสั่งเปิดกล้องเพื่อถ่ายภาพได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงรูปภาพที่ถูกถ่ายและบันทึกไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ปริญญาานิพนธ์นี้เป็นการพัฒนาระบบตรวจสอบสภาพเครื่องแม่ข่ายคอมพิวเตอร์ผ่านอุปกรณ์แอนดรอยด์ ตามที่ได้ออกแบบระบบ และรูปแบบการใช้งานไว้ โดยแอปพลิเคชันบนแอนดรอยด์นี้จะเชื่อมต่อกับเครื่องแม่ข่ายในเครือข่ายเดียวกันผ่านทางไวไฟ (WIFI) ซึ่งสามารถแสดงผลข้อมูลเกี่ยวกับฮาร์ดแวร์ของเครื่องแม่ข่ายและแสดงซอฟต์แวร์ที่มีการติดตั้งภายในเครื่องรวมถึงซอฟต์แวร์ที่กำลังทำงานอยู่ สามารถรีโมทไปยังเครื่องแม่ข่ายเพื่อส่งการต่าง และแสดงรูปภาพเกี่ยวกับเครื่องแม่ข่ายแต่ละเครื่อง รวมถึงบอกตำแหน่งของเครื่องแม่ข่ายว่าอยู่ที่ใดผ่านทาง Google Map ซึ่งล้วนเป็นข้อมูลที่เป็นประโยชน์ให้ผู้ใช้งานทราบ

5.2 ปัญหาและอุปสรรค

- 1) เนื่องจากระบบปฏิบัติการแอนดรอยด์ (Android) มีคลาสและไลบรารี (Library) เป็นของตนเอง จึงจำเป็นที่จะต้องศึกษาการทำงานของคลาส (Class) และไลบรารีต่างๆ ใหม่
- 2) ทดสอบแอปพลิเคชันใน simulator และในอุปกรณ์จริงได้ผลไม่เหมือนกัน
- 3) ความเร็วการทำงานแอปพลิเคชันขึ้นอยู่กับอินเทอร์เน็ตที่เชื่อมต่อ
- 4) ออกแบบระบบได้ไม่ดีพอ จึงทำให้แยกต่อการเพิ่มเติมฟังก์ชันต่างๆ
- 5) เมื่อเริ่มรันแอปพลิเคชันครั้งแรก ListView จะแสดงผลช้า
- 6) การหาตำแหน่ง Location โดยใช้ IP Address ไม่มีความแม่นยำ
- 7) Custom ListView ระบบปฏิบัติการแอนดรอยด์สามารถใช้งานได้ยาก ใช้เวลาศึกษานาน

5.3 แนวทางในการแก้ไข

- 1) ศึกษาค้นหาและทดลองการทำงานของคลาสและไลบรารีรวมถึงการค้นหาจากอินเทอร์เน็ตเพราะระบบปฏิบัติการแอนดรอยด์เป็น Open Source ทำให้สามารถค้นหาข้อมูลและแนวทางการพัฒนาได้ง่าย

- 2) ศึกษาข้อมูลเพิ่มเติมเกี่ยวกับ AndroidManifest.xml ทำให้สามารถแก้ไขข้อผิดพลาดที่แอปพลิเคชันใน simulator และในอุปกรณ์จริงได้ผลไม่เหมือนกัน โดยทำการเพิ่มการอนุญาตในไฟล์ดังกล่าว ดังนี้ `<category android:name="android.intent.category.LAUNCHER" />`
- 3) ต้องออกแบระบบทุกครั้งก่อนเริ่มเขียนโค้ด
- 4) ผู้ใช้สามารถเลื่อนตำแหน่งให้ถูกต้องแบบ manual ได้
- 5) ศึกษาข้อมูลเพิ่มเติมเกี่ยวกับ Widget ต่างๆในระบบปฏิบัติการแอนดรอยด์

5.4 แนวทางในการพัฒนาต่อ

แอปพลิเคชันนี้สามารถนำไปพัฒนาต่อได้ โดยมีแนวทางในการพัฒนาคือ

- 1) ทำให้ระบบสามารถแสดงผลได้ real-time ขึ้นในส่วนของคุณรายละเอียด Hardware และ Software
- 2) ออกแบระบบในบางคลาสใหม่เนื่องจากออกแบบไว้ได้ไม่ดีพอ ทำให้มีการทำงานที่ซ้ำ
- 3) หา Database Location ที่สามารถระบุ Location ได้เที่ยงตรงกว่า
- 4) สามารถตรวจสอบเครื่องแม่ข่ายได้มากกว่า 10 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] ไทยแลนด์อินดัสตรีคอตคอม. “มารู้จักโปรโตคอล SNMP.” [Online].
Available : <http://www.thailandindustry.com/guru/view.php?id=14747.2554>.
- [2] Android. “OS Android.” [Online].
Available : <http://osandroid99.blogspot.com/2012/01/os-android-sms-short-message-service.html>. 2555.
- [3] Frank Fock. “Class Snmp.” [Online].
Available : <http://www.snmp4j.org/doc/org/snmp4j/Snmp.html>. 2557.
- [4] GitHub. “Android SNMP Client.” [Online].
Available : <https://gist.github.com/issamux/6709513>. 2556.
- [5] Ravi Tamada. “Android Combining Tab Layout and List View.” [Online].
Available : <http://www.androidhive.info/2012/05/android-combining-tab-layout-and-list-view/#>. 2555.
- [6] TechDive.in. “Snmp Set Example.” [Online].
Available : <http://techdive.in/snmp/snmp4j-snmp-set-example>. 2553.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

คลาสการทำงานของโปรแกรม

1. Class Monitor

Monitor
-context: Context
-getSNMP: getSNMP
-message: String
-getip: String
-sysDescr : String
-sysName : String
-sysUpdate : String
-driveC : String
-driveD : String
-SWrun : String[]
-SWrun1 : String[]
-SWins : String[]
-SWins1 : String[]
-ProLoad : long
-RAM : int
-Process : long
-StoCapacity1 : int
-StoCapacity2 : int
-StoAllo1 : int
-StoAllo2 : int
-p_allocate1 : int
-p_allocate2 : int
-drived : int

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

-drivec : int
-summem : float
-tp_sysDescr : TextView
-t_sysDescr : TextView
-tp_sysName : TextView
-t_sysName : TextView
-tp_sysUpdate : TextView
-t_sysUpdate : TextView
-tp_drive : TextView
-t_drive : TextView
-tp_RAM : TextView
-t_RAM : TextView
-tp_Process : TextView
-t_Process : TextView
-tp_mem : TextView
-t_mem : TextView
-btt_shutdown: Button
-btt_restart: Button
-btt_sleep: Button
-btt_logoff: Button
-ExpAdapter : ExpandListAdapter
-ExpandList : ExpandableListView
-row1: TableRow
-row2: TableRow
-row3: TableRow
-SERVERPORT: int
-start_sd: Boolean
-Handler: Handler
-running: int
-dir: File
-TargetDirectory: File

```

```

-count : int
-count1 : int
-builder1 : StringBuilder
-builder2 : StringBuilder
-listDataChild : HashMap<String, List<String>>
-listDataHeader : List<String>

```

```

+onCreate(savedInstanceState : Bundle) : void
+onResume() : void
+ onActivityResult(request Code) : void
+ getPath(uri : Uri, activity : Activity) : String
+ setTabIcon(tabHost : TabHost, tabIndex : int, iconResource : int) : void
+ setTabColor(tabhost : TabHost) : void
+ onRestoreInstanceState(savedInstanceState : Bundle) : void
+ onBackPressed()
+ doInBackground() : void
+ onPreExcute () : void
+ onPostExcute () : void

```

2. Class GetSNMP

GetSNMP

```

-snmp : Snmp
-comtarget : CommunityTarget
-pdu : PDU
-oid : OID
-req : VariableBinding
-run : Boolean
-run1 : Boolean
-run2 : Boolean
-SWrun : String
-SWins : String

```

<pre>-mem: String -sysup : Integer -sysname : String -ip : String</pre>
<pre>+get(oids[] : OID) : ResponseEvent +getsysup() : Integer +getsysname() : String +getTarget() : Target +getAsString(oid : OID) : String +getAsInt(oid : OID) : int +getAsLong(oid : OID) : long +sendSnmpRequest() : void</pre>

3. Class ExpandListAdapter

ExpandListAdapter
<pre>-context : Context -listDataHeader_ : List<String> -listDataChild_ : HashMap<String, List<String>></pre>
<pre>+ExpandListAdapter(context : Context, listDataHeader : List<String>, listDataChild : HashMap<String, List<String>>) +getChild(groupPosition : int, childPosititon : int) : object + getChildId(groupPosition : int, childPosition : int) : long + getChildView(groupPosition : int, childPosition : final int, isLastChild : boolean, convertView : View, parent : ViewGroup) : View + getChildrenCount(groupPosition : int) : int + getGroup(groupPosition : int) : Object + getGroupCount() : int + getGroupId(groupPosition : int) : long + getGroupView(groupPosition : int, isExpanded : boolean, convertView</pre>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
View, parent : ViewGroup) : View
+ hasStableIds() : Boolean
+ isChildSelectable(groupPosition : int, childPosition : int) : boolean
```

4. Class ServerList

ServerList
-NUM : int
-countbut : int
-addServer : TextView
-edit : TextView
-valueip : String[]
-host : String[]
-status : Object[]
-listview : ListView
-addr : InetAddress
-arrlist : ArrayList<Map<String, Object>>
-mAdapter : SimpleAdapter
-sysup1 : Integer[]
-printorder : Integer[]
-check : String[]
-count : int
-sysname1 : String[]
-running : int
-getSNMP1 : getSNMP
-nowdel : int
-editstate : String
-getRun : MainActivity
-buttonclick : int
-Statuscolor : ImageView

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีข้อเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-map : ImageView

-editclick : String

-View : View

-lefttoright : Animation

-righttoleft : Animation

+onCreate(savedInstanceState : Bundle) : Void

+savedata() : void

+retrievedata() : void

+onBackPressed() : void

+onListItemClick() : void

+findnowdel() : int

+doInBackground() : String

+onPostExecute() : void

5. Class Map

Map

-googleMap : GoogleMap

-lng : double[]

-lat : double[]

-s_lng1 : double[]

-s_lat1 : double[]

-dumb : int

-temp : String

-value : String[]

-index : int

-num : int

-mymarker : Marker[]

-ServerList : ServerList

-name : String[]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

+onCreate(savedInstanceState : Bundle) : Void
+initilizeMap() : void
+onBackPressed() : void
+onResume() : void
+splitLatLng(latLng String : index : int) : void
+savadata() : void
+retrievedata() : void
+getLatLng(ip String) : String
+doInBackground() : String
+onPostExecute() : void

```

6. DetailsPhoto Class

DetailsPhoto
-imgFile : File
-myBitmap : Bitmap
+onCreate(savedInstanceState : Bundle) : Void

7. GetKey Class

GetKey
-r :String
-key : EditText
-set : TextView
+onCreate(savedInstanceState : Bundle) : Void
+savadata() : Void

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Class About

About
+onCreate(savedInstanceState : Bundle) : Void + onBackPressed() : Void

9. Class Logo

Logo
+onCreate(savedInstanceState : Bundle) : Void + onBackPressed() : Void

10. Class Photo

Photo
-itemList : ArrayList<String> -add_photo : TextView -dir : File -targetDirectory : File -getip : String
+onCreate(savedInstanceState : Bundle) : Void + onBackPressed() : Void +getPath(Uri uri, Activity activity) : String +onResume() : Void +onActivityResult(int requestCode, int resultCode, Intent data) : Void +onRestoreInstanceState(Bundle savedInstanceState) : Void

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. Class SetNewKey

SetNewKey
-r : String
-key : EditText
-set : TextView
-lefttoright : Animation
-k : ImageView
+onCreate(savedInstanceState : Bundle) : Void
+ savedata() : Void

12. Class Setting

Setting
-value : String[]
-listView : ListView
+onCreate(savedInstanceState : Bundle) : Void
+ onBackPressed() : Void

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้