

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบชำระค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธ

Electronics Toll Correction System via Bluetooth Technology



โดย

นางสาวสิริธัญญา ปัญญาไว

นางสาวอภิรดี ถิ่นแดงสงวน

เลขหมู่.....

เลขทะเบียน..... 72238

วัน,เดือน,ปี..... 12 ส.ย. 2550

b..... 112 ๖๕๕๕๕

i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Electronics Toll Correction System via Bluetooth Technology**

**By**

**Ms. Sirinya Panyawai**

**Ms. Apiradee Limdaengsanguan**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2006**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ระบบชำระค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธ
THESIS	Electronics Toll Correction System via Bluetooth Technology
ชื่อนักศึกษา	นางสาวสิริญา ปัญญาไว รหัสนักศึกษา 46012205 นางสาวอภิรดี ลี้มแดงสงวน รหัสนักศึกษา 46012210
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ศศ. มนต์ชัย แซ่มชัย
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2549

วิทยานิพนธ์ฉบับนี้ได้รับการอนุมัติเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

(ศศ. มนต์ชัย แซ่มชัย)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ระบบชำระค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธ
THESIS	Electronics Toll Correction System via Bluetooth Technology
ชื่อนักศึกษา	นางสาวสิริญา ปัญญาไว รหัสนักศึกษา 46012205 นางสาวอภิรดี ถิ่นแดงสงวน รหัสนักศึกษา 46012210
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ศส. มนต์ชัย แซ่มซ้อย
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2549

### บทคัดย่อ

โทรศัพท์เคลื่อนที่กลายมาเป็นปัจจัยที่สำคัญในการดำรงชีวิต นอกจากจะใช้ในการติดต่อสื่อสารทั่วไปแล้ว ปัจจุบันนี้โทรศัพท์เคลื่อนที่ยังมีเทคโนโลยีมากมายที่นำเข้ามาประยุกต์ใช้อาทิเช่น เทคโนโลยีบลูทูธ โดยโครงการนี้ ได้ศึกษาและพัฒนาทางด้านระบบการชำระค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธ ในส่วนของโครงการ ทางผู้จัดทำได้ทำการศึกษาการเขียนโปรแกรมในส่วนโปรแกรมที่เป็นส่วนติดต่อกับผู้ใช้ ใช้การพัฒนาจาวา บนโทรศัพท์เคลื่อนที่ และในส่วนของบลูทูธจะเป็นการใช้ระบบฝังตัว สำหรับการส่งข้อมูลผ่านเครือข่ายไร้สาย วัตถุประสงค์ของโครงการนี้ก็คือ เพื่อเป็นอีกทางเลือกหนึ่งในการชำระค่าผ่านทางให้ผู้ใช้บริการได้รับความสะดวกสบายในการชำระค่าผ่านทางยิ่งขึ้น

<b>THESIS TITLE</b>	Electronics Toll Correction System via Bluetooth Technology	
<b>STUDENT</b>	Miss Sirinya Panyawai	No.46012205
	Miss Apiradee Limdaengsanguan	No.46012210
<b>ADVISOR</b>	Assoc.Prof. Monchai Chamchoy	
<b>COURSE</b>	Bachclor of Information Engineering	
<b>DEPARTMENT</b>	Information Engineering	
<b>YEAR</b>	2006	

### Abstract

Nowadays, mobile phone is become more important. More than just a phone, it also works with many technologies. For example Bluetooth technology, it's now more useful. In this project, we study about electronics toll correction via Bluetooth technology. By this, Client will be charged from their mobile account. For this application, we will focus on user interface how to communicate with client by using J2ME .About Bluetooth, we will use embedded system for transferring data. It will be an alternative way to pay a toll and it will be more convenient.

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้ไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลคนแรกที่ต้องกล่าวถึงก็คือ ศศ. มนต์ชัย แซ่มะซ้อ อาจารย์ที่ปรึกษาปริญญาบัตร ที่ได้ให้ความช่วยเหลือในหลายๆ ด้าน หากไม่ได้อาจารย์ปริญญาบัตรฉบับนี้อาจไม่สามารถสำเร็จลงได้

ขอขอบคุณภาควิชาวิศวกรรมศาสตรสาสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับหลายๆ สิ่ง โดยเฉพาะสิ่งอำนวยความสะดวกต่าง ๆ ห้องวิจัย และอินเตอร์เน็ตสำหรับการค้นคว้าหาข้อมูลที่สำคัญต่างๆ ในการทำปริญญาบัตรฉบับนี้

ขอขอบพระคุณอาจารย์ทุกท่าน รวมทั้งที่เคยให้ความรู้มาก่อนจะเข้ามาเรียน ณ สถาบันแห่งนี้ หากไม่ได้ทุกท่าน คงไม่มีปริญญาบัตรฉบับนี้ และคงไม่มีวันนี้ขึ้นมาได้

ขอกราบขอบพระคุณ บิดา มารดา ครอบครัว ปัญญาไว และ ครอบครัว ลี้มแดงสงวนทุกท่าน ที่ทำให้มีวันนี้ขึ้นมา ขอขอบคุณสำหรับกำลังใจดีๆ ความรักและความเป็นห่วง พร้อมทั้งให้โอกาสทางการศึกษาอย่างเต็มที่อันหาที่เปรียบมิได้

สุดท้ายขอขอบคุณเพื่อนๆ ทุกคน ที่อยู่ด้วยกันมาจนถึงทุกวันนี้ ขอขอบคุณสำหรับกำลังใจดีๆ และการช่วยเหลือต่างๆ รวมถึงมิตรภาพดีๆ ที่มีให้กันมา

คุณค่าและประโยชน์อันพึงมาจากปริญญาบัตรฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

สิริัญญา ปัญญาไว

อภิรดี ลี้มแดงสงวน

## สารบัญ

หัวข้อ	หน้า
บทคัดย่อภาษาไทย	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ช
สารบัญตาราง	ฅ
<b>บทที่ 1 บทนำ</b>	
1.1 ที่มาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
1.4 ขอบเขตของโครงการ	2
1.5 ขั้นตอนการดำเนินโครงการ	2
1.6 ส่วนประกอบของปริญญานิพนธ์	2
<b>บทที่ 2 เทคโนโลยีที่ใช้ในโครงการ</b>	
2.1 เทคโนโลยีบลูทูธ	4
2.1.1 ความเป็นมาของเทคโนโลยีบลูทูธ	4
2.1.2 กำเนิดเทคโนโลยีบลูทูธ	5
2.1.3 บลูทูธคืออะไร	6
2.1.4 กำเนิดเทคโนโลยีบลูทูธ	6
2.1.5 สถาปัตยกรรมของบลูทูธ	7
2.1.6 งานประยุกต์ที่ใช้เทคโนโลยีบลูทูธ	8
2.1.7 โพรโตคอลสแต็กสำหรับระบบบลูทูธ	9
2.1.8 ชั้นสื่อสาร Radio Layer ในระบบบลูทูธ	11
2.1.9 ชั้นสื่อสาร Baseband Layer ในบลูทูธ	12
2.1.10 ชั้นสื่อสาร L2CAP ในระบบบลูทูธ	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

หัวข้อ	หน้า
<b>บทที่ 2 เทคโนโลยีที่ใช้ในโครงการ (ต่อ)</b>	
2.1.11 โครงสร้างเฟรมในบลูทูธ	13
<b>2.2 เทคโนโลยี J2ME</b>	15
2.2.1 สถาปัตยกรรมของ J2ME	17
2.2.2 สถาปัตยกรรมของ MIDP	20
2.2.3 MIDlet	21
<b>2.3 JSR 82 Bluetooth API และ OBEX</b>	23
2.3.1 Interface DiscoveryListener	24
2.3.2 Interface L2CAPConnection	30
2.3.3 Interface L2CAPConnectionNotifier	34
2.3.4 Interface ServiceRecord	36
2.3.5 Class DataElement	43
2.3.6 Class DeviceClass	56
2.3.7 Class DiscoveryAgent	58
2.3.8 Class LocalDevice	66
2.3.9 Class RemoteDevice	73
2.3.10 Class UUID	81
2.3.11 BluetoothConnectionException	83
2.3.12 Class BluetoothStateException	86
2.3.13 Class ServiceRegistrationException	87
<b>บทที่ 3 การออกแบบและพัฒนา</b>	
3.1 การออกแบบและพัฒนาทางฝั่ง Client	89
3.2 การออกแบบและพัฒนาทางด้าน Server	90
3.2.1 คอมพิวเตอร์ (Computer)	91
3.2.2 Database Server	91

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

หัวข้อ	หน้า
<b>บทที่ 3 การออกแบบและพัฒนา (ต่อ)</b>	
3.2.3 Customer	91
3.3 โฟลว์ชาร์ต (Flow Chart) การทำงานของระบบชำระ ค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธ	92
3.4 โฟลว์ชาร์ต (Flow Chart) แสดงการทำงานที่ฝั่ง Client	94
3.5 โฟลว์ชาร์ต (Flow Chart) การทำงานที่ฝั่ง Server	96
<b>บทที่ 4 การทดลองและผลการทดลอง</b>	
4.1 ทดลองสร้าง Mobile Application	98
4.2 ทดลองสร้างตารางแสดงฐานข้อมูลของระบบ ETC	103
<b>บทที่ 5 สรุปผลการทดลองและปัญหาที่พบ</b>	
5.1 สรุปผลการทดลอง	105
5.2 ปัญหาที่พบในการทดลอง	105
5.3 แนวทางการแก้ไขปัญหา	106
<b>บรรณานุกรม</b>	<b>107</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
2.1 กษัตริย์ Harald Bluetooth ปี ค.ศ. 940-981	4
2.2 การเชื่อมต่ออุปกรณ์ต่างๆ ด้วยบลูทูธ	5
2.3 พีโคเน็ต 2 ระบบ สามารถนำมาสร้างเป็นสแกตเตอร์เน็ตได้	7
2.4 สถาปัตยกรรมของมาตรฐาน 802.15 หรือบลูทูธที่ได้รับการปรับปรุงแล้ว	10
2.5 โครงสร้างเฟรมข้อมูลในระบบบลูทูธ	14
2.6 เปรียบเทียบ Java editions เวอร์ชันต่าง ๆ	16
2.7 สถาปัตยกรรมของ MIDP	20
2.8 สถานะต่างๆของ MIDlet	22
2.9 รูปแสดงขั้นตอนของการพัฒนา midlet	23
3.1 ภาพรวมของระบบ	89
3.2 การออกแบบและพัฒนาทางฝั่ง Client	90
3.3 การออกแบบและพัฒนาทางฝั่ง Server	90
3.4 โฟลว์ชาร์ต (Flow Chart) การทำงานของระบบชำระค่าผ่านทาง โดยใช้เทคโนโลยีบลูทูธ	92
3.5 โฟลว์ชาร์ต (Flow Chart) แสดงการทำงานที่ฝั่ง Client	94
3.6 โฟลว์ชาร์ตแสดงการทำงานที่ฝั่ง Server	96
4.1 หน้าจอโทรศัพท์เคลื่อนที่เมื่อเข้ามายังหน้าแรกของแอปพลิเคชัน	98
4.2 กด Start Service เพื่อทำการเริ่มใช้บริการของแอปพลิเคชัน	99
4.3 ทำการค้นหาการให้บริการและอุปกรณ์บลูทูธ	99
4.4 ทำการแสดงรายชื่อบริการและอุปกรณ์บลูทูธที่ค้นเจอ ทำการเลือกจากรายชื่อที่แสดง	100
4.5 ทำการเชื่อมต่อไปยังผู้ให้บริการ โดยใช้เทคโนโลยีบลูทูธ	100
4.6 ทำการส่งค่าหมายเลขโทรศัพท์เข้าไปยังเซิร์ฟเวอร์	101
4.7 กดปุ่ม Exit เพื่อออกจากโปรแกรม	101
4.8 หน้าจอของเซิร์ฟเวอร์เมื่อทำการรับค่าจากโทรศัพท์เคลื่อนที่	102
4.9 เป็นการรับข้อมูลจากผู้ให้บริการเมื่อมีผู้ใช้ติดต่อขอให้บริการ	102
4.10 ตารางระบบฐานข้อมูลของระบบ ETC	103

## สารบัญรูป(ต่อ)

รูปที่	หน้า
4.11 แสดงหน้าหลักของฐานข้อมูลการใช้งาน	103
4.12 แสดงหน้าจอการเรียกดูข้อมูลลูกค้า	104
4.13 แสดงหน้าจอเมื่อมีการทำการเพิ่มหรือแก้ไขข้อมูล แล้วทำการบันทึก	104



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
2.1 ไปรไฟล์สำหรับบลูทูธ	8
2.2 รายละเอียดในส่วนของฟิลด์ (Field) ของอินเตอร์เฟซ DiscoveryListener	24
2.2 รายละเอียดในส่วนของฟิลด์ (Field) ของอินเตอร์เฟซ DiscoveryListener(ต่อ)	25
2.3 รายละเอียดในส่วนของเมธอด (Method) ของอินเตอร์เฟซ DiscoveryListener	25
2.4 ค่า respCode	29
2.5 รายละเอียดในส่วนของฟิลด์ ของอินเตอร์เฟซ L2CAPConnection	30
2.6 รายละเอียดในส่วนของเมธอด ของอินเตอร์เฟซ L2CAPConnection	31
2.7 รายละเอียดในส่วนของเมธอด ของอินเตอร์เฟซ L2CAPConnectionNotifier	34
2.8 หมายเลขของ ID แอทริบิวท์ของการให้บริการ	36
2.9 ค่าสตริงชดเชยของแอทริบิวท์ IDs ที่ใช้บ่อยใน ServiceRecord	37
2.10 รายละเอียดในส่วนของฟิลด์ ของอินเตอร์เฟซ ServiceRecord	37
2.11 รายละเอียดในส่วนของเมธอด ของอินเตอร์เฟซ ServiceRecord	38
2.12 รายละเอียดในส่วนของฟิลด์ ของอินเตอร์เฟซ DataElement	44
2.12 รายละเอียดในส่วนของฟิลด์ ของอินเตอร์เฟซ DataElement (ต่อ)	45
2.13 รายละเอียดในส่วนของคอนสตรัคเตอร์ DataElement	45
2.14 รายละเอียดในส่วนของเมธอด DataElement	46
2.15 ค่าตามกฎสำหรับ valueType และค่าแอทริบิวท์ที่คล้ายกัน	51
2.16 ค่าตามกฎสำหรับ valueType และค่าแอทริบิวท์ที่คล้ายกัน	52
2.17 ชนิดของข้อมูลดาต้าอีลิเมนต์และชนิดของข้อมูลจาวา	55
2.18 รายละเอียดของ service class, major device class และ minor device class	56
2.19 รายละเอียดในส่วนของคอนสตรัคเตอร์ DeviceClass	57
2.20 รายละเอียดในส่วนของเมธอด DeviceClass	57
2.21 รายละเอียดในส่วนของฟิลด์ DiscoveryAgent	59
2.22 รายละเอียดในส่วนของเมธอด DiscoveryAgent	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
2.23 รายละเอียดในส่วนของเมธอด LocalDevice	66
2.23 รายละเอียดในส่วนของเมธอด LocalDevice (ต่อ)	67
2.24 รายละเอียดของชื่อพรีอเพอดี	69
2.25 รายละเอียดของคอนสตรัคเตอร์ RemoteDevice	73
2.26 รายละเอียดของเมธอด RemoteDevice	73
2.26 รายละเอียดของเมธอด RemoteDevice (ต่อ)	74
2.27 รายละเอียดของคอนสตรัคเตอร์ UUID	81
2.28 รายละเอียดของเมธอด UUID	81
2.29 รายละเอียดของฟิลด์ BluetoothConnectionException	83
2.30 รายละเอียดของคอนสตรัคเตอร์ BluetoothConnectionException	84
2.31 รายละเอียดของเมธอด BluetoothConnectionException	84
2.32 รายละเอียดของคอนสตรัคเตอร์ BluetoothStateException	87
2.33 รายละเอียดของคอนสตรัคเตอร์ ServiceRegistrationException	88

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบันนี้การสื่อสารเข้ามามีบทบาทในชีวิตเป็นอย่างมาก กลายมาเป็นปัจจัยสำคัญในการดำรงชีวิต และกิจกรรมต่างๆ การสื่อสารได้มีการพัฒนาไปอย่างรวดเร็วและมีประสิทธิภาพ จะเห็นได้ว่าประโยชน์ในการสื่อสารนั้นทำให้เราสามารถติดต่อถึงกันได้สะดวกและรวดเร็วยิ่งขึ้น แม้ว่าจะอยู่ระยะไกลกันเพียงใดก็ตามก็สามารถเชื่อมต่อถึงกันได้

นอกเหนือจากนั้นแล้ว ยังมีการพัฒนาเทคโนโลยีของการสื่อสารเพิ่มขึ้นเป็นอย่างมากในปัจจุบัน มีการนำเทคโนโลยีเหล่านั้นเข้ามาประยุกต์ใช้ เช่น อินฟราเรด อินเทอร์เน็ต บลูทูธ เป็นต้น โดยเทคโนโลยีบลูทูธ ได้ถูกนำมาใช้เป็นที่แพร่หลายเป็นจำนวนมากในปัจจุบัน เนื่องจากสะดวกในการใช้ ไม่ยุ่งยาก และไม่ต้องเสียค่าบริการในการใช้บริการเทคโนโลยีประเภทนี้

โครงการนี้ได้ทำการศึกษาเกี่ยวกับเทคโนโลยีบลูทูธ มาปรับปรุงให้เกิดประโยชน์ยิ่งขึ้น โดยได้นำมาประยุกต์ใช้กับการชำระค่าบริการในการใช้เส้นทางพิเศษ โดยจะใช้โทรศัพท์เคลื่อนที่เป็นตัวเชื่อมต่อกับเทคโนโลยีบลูทูธ เพื่อที่จะใช้ในการชำระค่าบริการ โดยจะเป็นอีกทางเลือกหนึ่งสำหรับการชำระค่าผ่านทาง เพื่อให้ผู้ใช้จะได้รับความสะดวกสบายในการชำระค่าผ่านทางยิ่งขึ้น

### 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อวิเคราะห์และพัฒนาระบบการชำระค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธ
- 1.2.2 เพื่อศึกษาการใช้ J2ME ในการเขียน Application บนโทรศัพท์เคลื่อนที่
- 1.2.3 เพื่อนำ Application ที่สร้างขึ้นมา สามารถนำไปใช้งานได้จริง
- 1.2.4 เพื่อศึกษาการทำระบบฐานข้อมูล (Database) เพื่อใช้สำหรับเก็บข้อมูลของผู้ใช้บริการ
- 1.2.5 เพื่อเป็นอีกทางเลือกหนึ่งในการชำระค่าผ่านทาง

### 1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 เพื่อได้รับความรู้ความเข้าใจถึงเทคโนโลยีบลูทูธ
- 1.3.2 เพื่อได้รับความรู้ความเข้าใจในการใช้ภาษา VB.NET ในการเชื่อมต่อบลูทูธในการรับส่งข้อมูลระหว่างโทรศัพท์มือถือและคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.3 เพื่อได้รับความรู้ความเข้าใจในการใช้ภาษา J2ME สำหรับการพัฒนา Application บนโทรศัพท์เคลื่อนที่

1.3.4 เพื่อได้รับความรู้ความเข้าใจในการจัดทำระบบฐานข้อมูล (Database)

#### 1.4 ขอบเขตของโครงการงาน

1.4.1 ออกแบบระบบการชำระค่าผ่านทาง ด้วยโทรศัพท์เคลื่อนที่ผ่านเทคโนโลยีบลูทูธ

1.4.2 สามารถใช้ภาษา J2ME สำหรับการสร้าง Application บนโทรศัพท์เคลื่อนที่ เพื่อเชื่อมต่อระหว่าง Application และคอมพิวเตอร์โดยใช้เทคโนโลยีบลูทูธ

1.4.3 ศึกษาการพัฒนาโปรแกรมบนระบบฝังตัวโดยจะเป็นการสร้างการเชื่อมต่อระหว่างโทรศัพท์เคลื่อนที่และคอมพิวเตอร์ และสร้างการเชื่อมต่อระหว่างคอมพิวเตอร์ สำหรับการติดต่อไปยังฐานข้อมูลของผู้ให้บริการ

1.4.4 ในส่วนของฐานข้อมูลจะเป็นการจัดเก็บข้อมูลหมายเลข Mac Address ของผู้ใช้บริการ

#### 1.5 ขั้นตอนการดำเนินโครงการงาน

1.5.1 ศึกษาและรวบรวมข้อมูล

1.5.2 ออกแบบระบบสำหรับการชำระค่าผ่านทางโดยผ่านเทคโนโลยีบลูทูธ

1.5.3 ศึกษาและทดลองเขียน Application บนโทรศัพท์เคลื่อนที่

1.5.4 ศึกษาและเขียนโปรแกรมทดสอบการเชื่อมต่อเทคโนโลยีบลูทูธบนโทรศัพท์เคลื่อนที่

1.5.5 ศึกษาและเขียนโปรแกรมทดสอบการเชื่อมต่อเทคโนโลยีบลูทูธระหว่างโทรศัพท์เคลื่อนที่และคอมพิวเตอร์

1.5.6 ศึกษาและเขียนโปรแกรมสำหรับการจัดเก็บข้อมูลบนคอมพิวเตอร์

1.5.7 ทดสอบระบบการชำระค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธผ่านโทรศัพท์เคลื่อนที่

1.5.8 ปรับปรุงและแก้ไขเพื่อเพิ่มประสิทธิภาพ

1.5.9 จัดทำรายงานและสรุปผลโครงการงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.1 ส่วนประกอบของปริญญาโท

### 1.1.1 บทที่ 1 บทนำ

- ที่มาและความสำคัญของโครงการ
- วัตถุประสงค์ของโครงการ
- ประโยชน์ที่คาดว่าจะได้รับ
- ขอบเขตของโครงการ
- ขั้นตอนการดำเนินโครงการ
- ประโยชน์ที่คาดว่าจะได้รับ

### 1.1.2 ทฤษฎีพื้นฐานที่ใช้ในโครงการ

- Bluetooth
- J2ME
- JSR 82 Bluetooth API and OBEX API

### 1.1.3 การออกแบบและพัฒนาโครงการ

- พัฒนาในส่วนของ Application
- พัฒนาในส่วนของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### เทคโนโลยีที่ใช้ในโรงงาน

#### 2.1 เทคโนโลยีบลูทูธ (Bluetooth)

เทคโนโลยีใหม่ๆ ที่ได้มีการถูกสร้างขึ้นมา เพื่อรองรับการพัฒนาเทคโนโลยีและควมทันสมัยบนอุปกรณ์ต่างๆ มักจะช่วยอำนวยความสะดวก ประหยัดเวลาให้แก่ผู้ใช้บริการ อีกทั้งช่วยให้การเชื่อมต่อมีความสะดวกยิ่งขึ้น ในโรงงานนี้ได้เสนอในเรื่องเทคโนโลยีบลูทูธ (Bluetooth Technology) หรือ “ฟันสีฟ้า” โดยเทคโนโลยีนี้เป็นเทคโนโลยีขนาดเล็กที่ช่วยทำให้ชีวิตของเราสะดวกยิ่งขึ้น

##### 2.1.1 ความเป็นมาของเทคโนโลยีบลูทูธ

คำว่าบลูทูธ (Bluetooth) หรือฟันสีฟ้า ความจริงแล้วเป็นนามของกษัตริย์ประเทศเดนมาร์ก ที่มีชื่อว่า “Harald Bluetooth” (ภาษาเดนมาร์ก Harald Blåtand) ในช่วงปี ค.ศ. 940-981 หรือประมาณเมื่อ 1,000 กว่าปีมาก่อน กษัตริย์องค์นี้ได้ปกครองประเทศเดนมาร์กและนอร์เวย์ในยุคของไวกิง และต้องการรวบรวมประเทศให้เป็นหนึ่งเดียวกัน นอกจากนั้น ยังทรงเป็นผู้นำเอาศาสนาคริสต์เข้าสู่ประเทศเดนมาร์กอีกด้วย



รูปที่ 2.1 กษัตริย์ Harald Bluetooth ปี ค.ศ. 940-981

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเพื่อเป็นการรำลึกถึงเกียรติยศ Bluetooth ผู้ปกครองประเทศกลุ่มสแกนดิเนเวีย ซึ่งในปัจจุบันเป็นกลุ่มผู้นำในด้านการผลิตโทรศัพท์มือถือป้อนสู่ตลาดโลก และระบบ Bluetooth นี้ ก็ถูกสร้างขึ้นมาเพื่อใช้กับโทรศัพท์เคลื่อนที่ และเริ่มต้นจากประเทศในแถบนี้ด้วยเช่นกัน

### 2.1.2 กำเนิดเทคโนโลยีบลูทูธ

ปี 1994 บริษัท อีริคสัน โบบาย คอมมูนิเคชั่น เริ่มต้นที่จะค้นคว้าวิจัยเป็นไปได้ในการนำคลื่นสัญญาณวิทยุ มาใช้ระหว่างโทรศัพท์มือถือและอุปกรณ์ต่างๆ และเป็นผู้นำชื่อ Bluetooth มาใช้

ปี 1998 กลุ่มผู้พัฒนาวิจัยระบบ Bluetooth ได้ถูกก่อตั้งขึ้น โดยเกิดจากการรวมตัวของบริษัทยักษ์ใหญ่อย่าง Ericsson, Nokia, IBM, Toshiba และ Intel ในกลุ่มที่ใช้ชื่อว่า Special Interest Group (SIG) ซึ่งในกลุ่มจะประกอบด้วย กลุ่มผู้นำทางด้านโทรศัพท์เคลื่อนที่, คอมพิวเตอร์ ฯลฯ ซึ่งกลุ่มเหล่านี้ได้ประเมินว่า ภายในปี 2002 ในอุปกรณ์การสื่อสาร, เครื่องใช้, คอมพิวเตอร์ จะถูกติดตั้ง Bluetooth ที่จะใช้เชื่อมต่อระหว่างอุปกรณ์ต่างๆ อย่างแพร่หลาย



รูปที่ 2.2 การเชื่อมต่ออุปกรณ์ต่างๆ ด้วยบลูทูธ

โดยในปีเดียวกัน บริษัทเหล่านี้ ได้ประกาศ การรวมตัวกัน และเชิญชวนบริษัทอื่นๆ ให้เข้าร่วม ในลักษณะของการนำเทคโนโลยีนี้ไปใช้ โดยในปี 1999 ได้ทำการเผยแพร่ Bluetooth specification Version 1.0 และได้สมาชิกเพิ่มขึ้น ดังนี้ Microsoft, Lucent, 3Com, Motorola

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3 บลูทูธคืออะไร

บลูทูธ คือ ระบบสื่อสารของอุปกรณ์อิเล็กทรอนิกส์แบบสองทาง ด้วยคลื่นวิทยุระยะสั้น (Short-Range Radio Links) โดยปราศจากการใช้สายเคเบิล หรือ สายสัญญาณเชื่อมต่อ และไม่จำเป็นต้องใช้การเดินทางแบบเส้นตรงเหมือนกับอินฟราเรด ซึ่งถือว่าเพิ่มความสะดวกมากกว่า การเชื่อมต่อแบบอินฟราเรด ที่ใช้ในการเชื่อมต่อระหว่างโทรศัพท์มือถือ กับอุปกรณ์ ใน โทรศัพท์เคลื่อนที่รุ่นก่อนๆ และในการวิจัย ไม่ได้มุ่งเฉพาะการส่งข้อมูลเพียงอย่างเดียว แต่ยังศึกษา ถึงการส่งข้อมูลที่เป็นเสียง เพื่อใช้สำหรับ Headset บน โทรศัพท์เคลื่อนที่ด้วย

### 2.1.4 การทำงานของเทคโนโลยีบลูทูธ

Bluetooth จะใช้สัญญาณวิทยุความถี่สูง 2.4 GHz. (กิกะเฮิรตซ์) แต่จะแยกย่อยออกไปตามแต่ละประเทศ อย่างในแถบยุโรปและอเมริกา จะใช้ช่วง 2.400 ถึง 2.4835 GHz. แบ่งออกเป็น 79 ช่องสัญญาณ และจะใช้ช่องสัญญาณที่แบ่งนี้ เพื่อส่งข้อมูลสลับช่องไปมา 1,600 ครั้งต่อ 1 วินาที ส่วนที่ญี่ปุ่นจะใช้ความถี่ 2.402 ถึง 2.480 GHz. แบ่งออกเป็น 23 ช่อง ระยะเวลาทำการของ Bluetooth จะอยู่ที่ 5-10 เมตร โดยมีระบบป้องกันโดยใช้การป้อนรหัสก่อนการเชื่อมต่อ และ ป้องกันการดักสัญญาณระหว่างสื่อสาร โดยระบบจะสลับช่องสัญญาณไปมา จะมีความสามารถในการเลือกเปลี่ยนความถี่ที่ใช้ในการติดต่อเองอัตโนมัติ โดยที่ไม่จำเป็นต้องเรียงตามหมายเลขช่อง ทำให้การดักฟังหรือลักลอบขโมยข้อมูลทำได้ยากขึ้น

โดยหลักของบลูทูธจะถูกออกแบบมาเพื่อใช้กับอุปกรณ์ที่มีขนาดเล็ก เนื่องจากใช้การขนส่งข้อมูลในจำนวนที่ไม่มาก อย่างเช่น ไฟล์ภาพ, เสียง, แอปพลิเคชันต่างๆ และสามารถเคลื่อนย้ายได้ง่าย เพียงแค่อยู่ในระยะที่กำหนดไว้เท่านั้น (ประมาณ 5-10 เมตร) นอกจากนี้ยังใช้พลังงานต่ำ กินไฟน้อย และสามารถใช้งานได้นาน โดยไม่ต้องนำไปชาร์จไฟบ่อยๆ ด้วย

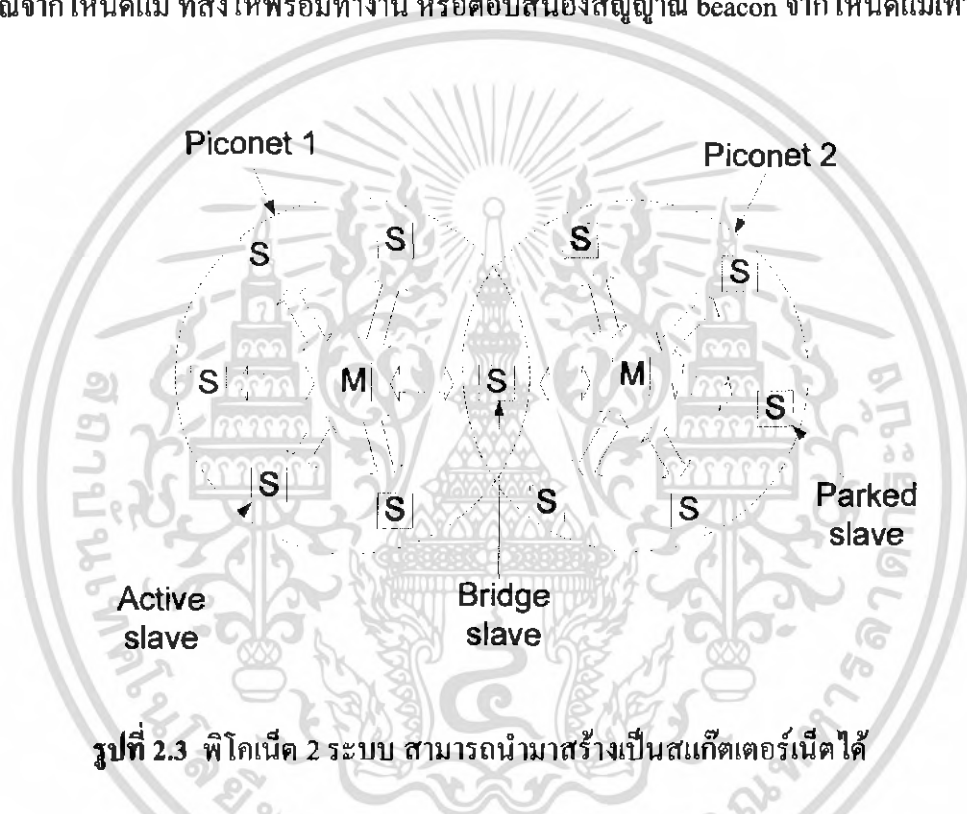
ส่วนความสามารถในการส่งถ่ายข้อมูลของ Bluetooth อยู่ที่ 1 Mbps (1 เมกกะบิตต่อวินาที) และไม่มีปัญหาอะไรมากสำหรับขนาดของไฟล์ที่ใช้กันบนโทรศัพท์เคลื่อนที่ หรือ การใช้งานแบบทั่วไป แต่ถ้าเป็นข้อมูลที่มีขนาดใหญ่ อาจจะช้าเกินไป และถ้าเปรียบเทียบกับ Wireless LAN (WLAN) แล้ว ซึ่งในส่วนของ WLAN ก็ยังมีระยะเวลาการรับ-ส่งที่ไวกว่า แต่ข้อได้เปรียบของ Bluetooth จะอยู่ที่ขนาดที่เล็กกว่า การติดตั้งทำได้ง่ายกว่า และที่สำคัญ การใช้พลังงานก็น้อยกว่ามาก อยู่ที่ 0.1 วัตต์

### 2.1.5 สถาปัตยกรรมของบลูทูธ

ขอบเขตพื้นฐานของระบบสื่อสารของบลูทูธเรียกว่า “พีโคเน็ต” (Piconet) ซึ่งประกอบด้วย โหนดแม่ (master) และ โหนดลูก (slave) อีกไม่เกิน 7 โหนด ที่สื่อสารกันภายในระยะทางไม่เกิน 10 เมตร เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมตร พิคเน็ตหลายๆ ระบบสามารถที่จะมีอาณาเขตซ้อนทับกันได้ภายในห้องทำงานแห่งหนึ่ง หรือแม้แต่กระทั่งสามารถสื่อสารข้ามระบบกันได้ผ่านสะพานสื่อสารดังแสดงในรูป ระบบพิคเน็ตที่มีอาณาเขตซ้อนทับกันและทำงานร่วมกันนี้เรียกว่า “สแก็ตเตอร์เน็ต” (Scatternet)

นอกเหนือจากโหนดลูกจำนวน 7 โหนดแล้วในพิคเน็ตอาจยังมี parked โหนดได้มากกว่าถึง 255 โหนด อุปกรณ์ที่เป็น parked โหนดนี้เป็นอุปกรณ์ที่โหนดแม่ได้สั่งให้อยู่ในสถานะที่ใช้พลังงานต่ำเพื่อเป็นการประหยัดพลังงาน โหนดกลุ่มนี้จะไม่สามารถทำอะไรได้นอกจากตอบสนองสัญญาณจากโหนดแม่ ที่สั่งให้พร้อมทำงาน หรือตอบสนองสัญญาณ beacon จากโหนดแม่เท่านั้น



รูปที่ 2.3 พิคเน็ต 2 ระบบ สามารถนำมาสร้างเป็นสแก็ตเตอร์เน็ตได้

เหตุผลที่ระบบได้ถูกออกแบบมาในลักษณะ โหนดแม่โหนดลูก คือผู้ออกแบบมีเจตนาที่จะทำให้การสร้างระบบบลูทูธขึ้นมาใช้งานด้วยอุปกรณ์ที่มีราคาไม่เกิน 5 เหรียญสหรัฐเท่านั้น ผลจากการตกลงใจนี้ทำให้โหนดลูกเป็นแบบง่าย ไม่มีความสามารถพิเศษใดๆ โดยที่จะทำตามโหนดแม่สั่งการมาเท่านั้นระบบเครือข่ายพิคเน็ตจึงเป็นระบบรวมศูนย์แบบ TDM (ผสมสัญญาณด้วยวิธีแบ่งเวลาการทำงาน) โดยมีโหนดแม่เป็นผู้ควบคุมสัญญาณนาฬิกา และเป็นผู้กำหนดว่าอุปกรณ์ใดจะได้รับมอบหมายให้ใช้ช่วงเวลาใด การสื่อสารทั้งหมดเกิดขึ้นระหว่างโหนดแม่กับโหนดลูกเท่านั้น โดยที่โหนดลูกจะไม่สามารถสื่อสารโดยตรงระหว่างกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.6 งานประยุกต์ที่ใช้เทคโนโลยีบลูทูธ

โพรโทคอลสำหรับระบบเครือข่ายส่วนมากจะทำหน้าที่จัดเตรียมช่องสื่อสารระหว่างส่วนที่สื่อสารถึงกันไว้และปล่อยให้ผู้ออกแบบงานประยุกต์ทั้งหลายเป็นผู้กำหนดเอาเองว่าจะนำช่องสื่อสารนี้มาใช้งานใด ตัวอย่างเช่น มาตรฐาน 802.11 ไม่ได้กำหนดว่าผู้ใช้ควรที่จะใช้เครื่องโน้ตบุ๊กในการอ่านอีเมลล์ การท่องเว็บ หรืออย่างอื่นใด ในทางกลับกันมาตรฐานของบลูทูธรุ่น 1.1 ได้กำหนดชนิดของงานประยุกต์ที่พามาใช้ไว้เป็น 13 ประเภทซึ่งจะได้สร้างโพรโทคอลสแต็กไว้สนับสนุนการทำงานเป็นการเฉพาะงานประยุกต์ทั้ง 13 ชนิดเรียกว่าโปรไฟล์ (profiles) แสดงในตาราง

ตารางที่ 2.1 โปรไฟล์สำหรับบลูทูธ

Name	Description
Generic access	Procedures for link management
Service discovery	Protocol for discovery offered services
Serial port	Replacement for a serial port cable
Generic object exchange	Defines client-server relationship for object movement
LAN access	Protocol between a mobile computer and a fixed LAN
Dial-up networking	Allows a notebook computer to call via a mobile phone
Fax	Allows a mobile fax machine to talk to a mobile phone
Cordless telephony	Connects a handset and its local base station
Intercom	Digital walkie-talkie
Headset	Allows hand-free voice communication
Object push	Provides a way to exchange simple objects
File transfer	Provide a more general file transfer facility
Synchronization	Permits a PDA to synchronization with another computer

โปรไฟล์ Generic access ไม่ได้เป็นงานประยุกต์โดยตรงแต่ทำหน้าที่เป็นส่วนประกอบพื้นฐานสำหรับการสร้างงานประยุกต์อื่นๆ ที่จะนำไปใช้ มีหน้าที่หลักในการสนับสนุนวิธีการจัดตั้งและรักษาการเชื่อมต่อช่องสื่อสารที่ปลอดภัย ระหว่างโหนดแม่และโหนดลูก โปรไฟล์ service discovery ก็ถูกสร้างขึ้นมาเพื่อใช้งานทั่วไป ซึ่งจะถูกอุปกรณ์โหนดลูกนำไปใช้งานได้บ้าง อุปกรณ์เอกสารนี้เป็นเอกสารที่ส่งวนเวียนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบบถูททั้งหมดจะต้องสามารถใช้งาน โปรไฟล์ทั้ง 2 อย่างนี้ได้ ส่วนโปรไฟล์ที่เหลือนั้นสามารถเลือกใช้ได้ตามต้องการ

โปรไฟล์ Serial port เป็นโปรโตคอลในชั้นสื่อสารนำส่งข้อมูล ซึ่งจะถูกโปรไฟล์ที่เหลือเรียกใช้งาน โปรไฟล์นี้จะทำให้ช่องสื่อสารเป็นเสมือนสายสื่อสารแบบอนุกรม (serial line) ซึ่งเป็นสายสื่อสารชนิดที่งานประยุกต์ส่วนใหญ่ต้องการนำไปใช้

โปรไฟล์ Generic object exchange กำหนดความสัมพันธ์แบบผู้ให้บริการและผู้ใช้บริการ (Client/Server) สำหรับการเคลื่อนย้ายข้อมูลในระบบเครือข่าย ผู้ใช้บริการจะเป็นผู้กระตุ้นให้เกิดการทำงานและโหนดลูกอาจเป็นผู้ให้บริการหรือผู้ใช้บริการก็ได้ โปรไฟล์นี้ก็เป็นอีกอันหนึ่งที่ถูกนำไปใช้ในการสร้างงานประยุกต์

กลุ่มของโปรไฟล์ต่อมาอีก 3 โปรไฟล์ (LAN access, Dial-up networking และ Fax) ถูกจัดเตรียมไว้สำหรับการสื่อสารบนระบบเครือข่าย โปรไฟล์ LAN access ช่วยให้อุปกรณ์บถูทสามารถเชื่อมต่อเข้ากับระบบเครือข่ายเฉพาะบริเวณได้ จำทำหน้าทีคล้ายกับมาตรฐาน 802.11 โปรไฟล์ Dial-up networking นั้นเปรียบเสมือนกับเป็นจุดเริ่มต้นของโปรเจกตบถูท เป็นตัวที่ช่วยให้เครื่องโน้ตบุ๊กสามารถเชื่อมต่อเข้ากับโทรศัพท์มือถือ ทำให้การสื่อสารแบบไร้สายเกิดขึ้นได้ โปรไฟล์ Fax มีลักษณะคล้ายคลึงกับโปรไฟล์ Dial-up networking ยกเว้นแต่ได้ถูกออกแบบมาสำหรับการใช้งานกับเครื่องแฟกซ์แบบไร้สาย ทำให้สามารถรับและส่งแฟกซ์ผ่านโทรศัพท์เคลื่อนที่

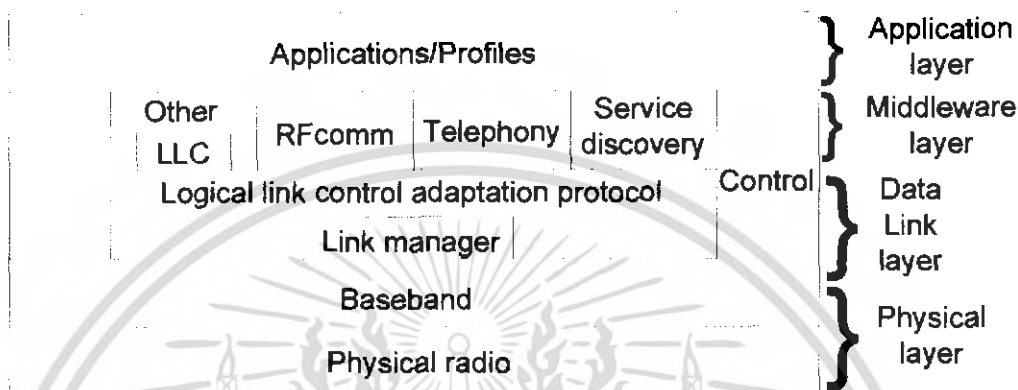
กลุ่มของโปรไฟล์ต่อมาอีก 3 โปรไฟล์ (Cordless telephony, Intercom และ Handset) ใช้สำหรับงานที่เกี่ยวข้องกับโทรศัพท์ (Telephony) โปรไฟล์ Cordless telephony ช่วยให้ชุดหูฟังโทรศัพท์สามารถเชื่อมต่อเข้ากับตัวโทรศัพท์โดยไม่ต้องใช้สาย โปรไฟล์ Intercom ช่วยให้โทรศัพท์สองเครื่องเชื่อมต่อกันได้ในลักษณะเดียวกันกับเครื่องรับส่งวิทยุมือถือ (Walkie-Talkie) และโปรไฟล์ Headset ช่วยให้อุปกรณ์ประเภท hard-free สามารถสื่อสารกับเครื่องฐานได้ (เช่นชุด hand-free ที่ใช้กับโทรศัพท์เคลื่อนที่)

โปรไฟล์ที่เหลือทั้ง 3 แบบ (Object push, File transfer และ Synchronization) นั้นเป็นส่วนที่ทำหน้าที่ในการแลกเปลี่ยนอ็อบเจกตระหว่างอุปกรณ์ไร้สาย 2 เครื่อง ตัวอย่างของอ็อบเจกต เช่นนามบัตร รูปถ่าย หรือพื้มข้อมูล โปรไฟล์ Synchronization ใช้สำหรับการส่งสำเนาข้อมูลจากเครื่องโน้ตบุ๊กเข้าไปเก็บไว้เครื่องพีดีเอ (PDA; Personal Data Assistant)

### 2.1.7 โพรโตคอลลตแตคสำหรับระบบบถูท

มาตรฐานบถูท ประกอบด้วยโปรโตคอลลจำนวนมากที่รวมตัวกันอย่างหลวมๆ ในแต่ละชั้นตอนการสื่อสาร โครงสร้างของชั้นสื่อสารไม่ได้เป็นไปตามมาตรฐาน OSI TCP/IP หรือเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน 802 หรือมาตรฐานใดๆ อย่างไรก็ตาม องค์กร IEEE ได้ทำการปรับปรุงมาตรฐานเพื่อให้สามารถเข้าไปเป็นหนึ่งในมาตรฐานในตระกูล 802 ดังรูป แสดงสถาปัตยกรรมของมาตรฐานที่ได้รับการปรับปรุงแล้ว



รูปที่ 2.4 สถาปัตยกรรมของมาตรฐาน 802.15 หรือมาตรฐานที่ได้รับการปรับปรุงแล้ว

ชั้นการสื่อสารต่ำสุดคือชั้นการสื่อสาร Physical radio layer ซึ่งตรงกับชั้นการสื่อสารกายภาพในมาตรฐาน OSI ทำงานเกี่ยวกับการรับ-ส่งสัญญาณคลื่นวิทยุและการแปลงสัญญาณ เป้าหมายในการสร้างชั้นการสื่อสารนี้ คือ การทำให้มีความง่ายที่สุด เพื่อให้สามารถมีต้นทุนต่ำที่ต่ำมากในการทำตลาด

ชั้นการสื่อสาร baseband layer นั้นทำงานคล้ายกับชั้นการสื่อสารย่อยควบคุมการใช้สื่อ (MAC sublayer) แต่ได้รวบรวมความสามารถบางส่วนของชั้นการสื่อสารกายภาพไว้ด้วย ชั้นการสื่อสารนี้เกี่ยวกับการทำงานของโหนดแม่ในการควบคุมการใช้ช่องสื่อสาร และการรวมของกลุ่มเวลาสื่อสารเข้าเป็นเฟรมข้อมูล

ชั้นต่อไป เป็นกลุ่มของโปรโตคอลที่ทำหน้าที่สัมพันธ์ซึ่งกันและกัน โปรโตคอล Link manager จัดตั้งช่องสื่อสารตรรกะขึ้นระหว่างอุปกรณ์ รวมทั้งการบริหารการใช้พลังงานของอุปกรณ์ การตรวจสอบสิทธิผู้ใช้และคุณภาพของการบริการ โปรโตคอล logical link control adaptation หรือ L2CAP ซ่อนความซับซ้อนของการถ่ายทอดสัญญาณไว้จากชั้นการสื่อสารในระดับบน ซึ่งเปรียบเทียบกับ ชั้นการสื่อสารย่อย LLC ในมาตรฐาน 802 แต่ใช้เทคโนโลยีที่แตกต่างกัน โปรโตคอล Audio และ Control ทำงานเกี่ยวกับเสียงและการควบคุมโปรแกรมประยุกต์สามารถเรียกใช้โปรโตคอลนี้ได้โดยตรงไม่ต้องเรียกผ่านโปรโตคอล L2CAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นสื่อสารในระดับต่อไปคือ Middleware layer ประกอบด้วยโปรโตคอลจำนวน 6 ชั้น ชั้นสื่อสารย่อย LLC ถูกใส่เข้าไปเพื่อทำให้โครงสร้างมีลักษณะเหมือนกับระบบเครือข่าย 802 แบบอื่นๆ โปรโตคอล RFcomm (Radio Frequency communication) เป็นโปรโตคอลที่จำลองการทำงานของ serial port ที่มีไซในเครื่อง PC ทัวไปเอาไว้ใช้สำหรับเชื่อมต่อกับแป้นพิมพ์ เมาส์ โมเด็มและอื่นๆ ถูกออกแบบมาเพื่อให้โปรแกรมรุ่นเก่ายังคงสามารถใช้งานได้โดยไม่มีปัญหา โปรโตคอล telephony ทำงานแบบ Real-time เตรียมไว้สำหรับทำงานร่วมกับ โปรไฟล์ที่เกี่ยวข้องจำนวน 3 โปรไฟล์ ดังที่กล่าวมาแล้ว รวมทั้งการจัดการเรื่องการใช้โทรศัพท์ในการติดต่อและการยกเลิกการโทร โปรโตคอลสุดท้ายคือ service discovery นำมาใช้ในการค้นหาบริการต่างๆ ที่มีให้ในระบบเครือข่าย

ชั้นสื่อสารชั้นบนสุดคือ Application layer เป็นที่อยู่ของโปรแกรมประยุกต์และโปรไฟล์ โปรโตคอลในชั้นนี้จะทำงานโดยการเรียกใช้บริการจากโปรโตคอลในชั้นล่าง งานประยุกต์แต่ละงานจะมีโปรโตคอลย่อยสำหรับการติดต่ออยู่ในตัวเอง

### 2.1.8 ชั้นสื่อสาร Radio Layer ในระบบบลูทูธ

ชั้นสื่อสาร Radio Layer ทำการเคลื่อนย้ายบิตข้อมูลระหว่างโหนดแม่กับโหนดลูก เป็นระบบที่ใช้พลังงานน้อยที่มีรัศมีการส่งเพียง 10 เมตรและทำงานในย่านความถี่ 2.4 GHz ของ ISM band ช่องสื่อสารจะถูกแบ่งออกเป็น 79 ช่อง โดยมีความกว้างช่องละ 1 MHz และใช้วิธีการเปลี่ยนแปลงสัญญาณแบบ frequency shift keying ที่ส่งข้อมูลออกไป 1 บิตต่อ Hz ทำให้มีความเร็วรวมในการส่งข้อมูล 1 Mbps แต่ก็ต้องเสียความสามารถในการส่งข้อมูลส่วนมากไปให้กับการส่งข้อมูลส่วนหัวของเฟรมข้อมูลการกำหนดใช้ช่องสัญญาณใช้เทคนิคแบบ frequency hopping spread spectrum ซึ่งมีความเร็วในการเปลี่ยนช่องสัญญาณใช้เทคนิคแบบ 1600 ครั้งต่อวินาที และมีระยะเวลาที่ใช้ในแต่ละช่องสัญญาณ (dwell time) เป็น 625 ไมโครวินาที โหนดทุกโหนดในฟิโคเน็ทจะเปลี่ยนช่องสัญญาณไปพร้อมๆ กันทั้งหมดโดยมีโหนดแม่เป็นผู้กำหนดรายละเอียดในการเปลี่ยนช่องสัญญาณ

เนื่องจากทั้งมาตรฐาน 802.11 และบลูทูธทำงานในย่านความถี่ 2.4 GHz ของช่อง ISM band และแบ่งออกเป็น 79 ช่องสัญญาณเหมือนกันจึงเกิดการรบกวนซึ่งกันและกัน เนื่องจากบลูทูธมีอัตราความเร็วในการเปลี่ยนช่องสัญญาณเร็วกว่า 802.11 มาก จึงเป็นไปได้ว่าระบบบลูทูธจะรบกวนให้การสื่อสารในระบบ 802.11 เสียหายมากกว่าที่จะเป็นผู้เสียหายเอง และเนื่องจากทั้ง 802.11 และ 802.15 เป็นมาตรฐานที่กำหนดโดย IEEE ด้วยกันทั้งคู่ องค์กร IEEE จึงพยายามหาทางแก้ไขปัญหาที่เกิดขึ้นนี้ มาตรฐาน 802.11a ใช้ย่านความถี่ที่ 5 GHz ISM band แต่มีระยะเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งสัญญาณสั้นกว่า 802.11b มาก (เนื่องจากคุณสมบัติทางกายภาพของคลื่นความถี่ที่ใช้) ดังนั้นการใช้มาตรฐาน 802.11a จึงไม่ใช่วิธีการแก้ปัญหาสำหรับทุกกรณี องค์กรบางแห่งแก้ปัญหาด้วยการห้ามใช้บลูทูธเด็ดขาด

### 2.1.9 ชั้นสื่อสาร Baseband Layer ในบลูทูธ

ชั้นสื่อสาร Baseband เป็นส่วนที่มีความคล้ายคลึงกับชั้นสื่อสารย่อยควบคุมใช้สื่อมากที่สุดทำหน้าที่ในการเปลี่ยนข้อมูลจากกระแสบิตที่ได้รับให้เป็นเฟรมข้อมูลและกำหนดรูปแบบที่มีความสำคัญให้ด้วย ในรูปแบบที่ง่ายที่สุด โหนดแม่ของระบบพีโคเน็ตแต่ละระบบจะกำหนดช่องเวลาขนาด 625 ไมโครวินาทีจำนวนหนึ่งโดยที่โหนดแม่จะเริ่มการถ่ายทอดสัญญาณแรกในช่องสัญญาณเลขคู่และให้โหนดลูกส่งสัญญาณผ่านช่องเวลาที่เลขคี่ วิธีการนี้เป็นวิธีการพื้นฐานของการส่งสัญญาณในแบบการเปลี่ยนแปลงสัญญาณ โดยการแบ่งช่วงเวลา (time division multiplexing) โดยแบ่งช่วงเวลาการส่งสัญญาณระหว่างโหนดแม่และโหนดลูกโหนดละครึ่งหนึ่งพอดีเฟรมข้อมูลแต่ละเฟรมอาจมีความยาว 1, 3 หรือ 5 ช่องเวลา

ระยะเวลาที่กำหนดไว้สำหรับการเปลี่ยนช่องสัญญาณ ได้ทำให้มีระยะเวลาในแต่ละช่องสัญญาณประมาณ 250-260 ไมโครวินาทีซึ่งมากพอที่จะทำให้เกิดสภาพเสถียรขึ้นในวงจรสัญญาณ การกำหนดให้มีระยะเวลาที่สั้นกว่านี้ก็สามารทำได้แต่จะทำให้อุปกรณ์มีราคาสูงขึ้นสำหรับเฟรมที่มีความยาวเพียงหนึ่งช่องของเวลา หลังจากที่มีการส่งสัญญาณมีความเสถียรแล้ว จะทำให้เหลือเวลาสำหรับการส่งข้อมูลเพียง 366 บิตจากทั้งหมด 625 บิต ในจำนวนนี้ 126 บิตจะถูกใช้เป็นโค้ดสำหรับการใช้งานและข้อมูลควบคุมอื่นๆ (ข้อมูลส่วนหัว) เหลืออีก 240 บิตสำหรับการส่งข้อมูลจริงสำหรับเฟรมที่มีความยาว 5 ช่องเวลาจะใช้เวลาในการทำให้ช่วงเวลาเสถียรเพียงครั้งเดียวเท่านั้น ดังนั้นจากปริมาณข้อมูลที่ส่งออกไปทั้งหมด 315 บิต (5 x 625) จะมีปริมาณบิตข้อมูลสำหรับ Baseband layer มากถึง 2781 บิตดังนั้น เฟรมที่มีขนาดใหญ่จะมีประสิทธิภาพในการทำงานสูงกว่าเฟรมขนาดเล็ก

เฟรมแต่ละเฟรมจะถูกส่งผ่านช่องสื่อสารตรรกะ (Logical channel) เรียกว่า link ซึ่งเกิดขึ้นระหว่างโหนดแม่และโหนดลูกลิงค์แบ่งออกเป็น 2 ชนิดคือ ACL และ SCO ลิงค์แบบ ACL (Asynchronous Connection-Less) ถูกนำมาใช้สำหรับการส่งข้อมูลแบบ Packet switching ที่มีใช้งานเป็นครั้งคราว ข้อมูลนี้ถูกส่งมาจากชั้นสื่อสาร L2CAP จากทางฝั่งผู้ส่ง และถูกส่งต่อไปให้ชั้นสื่อสาร L2CAP ทางฝั่งผู้รับ ข้อมูลแบบ ACL จะถูกส่งในสถานะ best-efforts service ซึ่งไม่รับประกันการนำส่งข้อมูล เฟรมข้อมูลอาจสูญหายและจะต้องถูกจัดส่งใหม่ โหนดลูกสามารถมีลิงค์ ACL ได้เพียงลิงค์เดียวไปยังโหนดแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิงค์แบบที่ 2 คือ SCO (Synchronous Connection Oriented) ใช้สำหรับการส่งข้อมูลแบบ Real-time เช่นการเชื่อมต่อโทรศัพท์สำหรับการสนทนา ช่องสื่อสารประเภทนี้จะถูกกำหนดให้ใช้ช่องเวลาที่แน่นอนในแต่ละทิศทางการส่งข้อมูล เนื่องจาก ข้อจำกัดในเรื่องเวลาการทำงานของลิงค์ SCO ทำให้ไม่มีการส่งเฟรมข้อมูลซ้ำ ซึ่งอาจแก้ไขได้ด้วยการใช้เทคนิคการแก้ไขข้อมูลแบบ forward error correction โหนดลูกสามารถมีลิงค์ SCO ได้มากถึง 3 ลิงค์ไปยังโหนดแม่เพียงโหนดเดียว ลิงค์ SCO สามารถถ่ายทอดสัญญาณเสียงได้ 64,000 bps ผ่านทางช่องสื่อสาร

### 2.1.10 ชั้นสื่อสาร L2CAP ในระบบบลูทูธ

ชั้นสื่อสาร L2CAP มีหน้าที่หลัก 3 ประการ ประการแรก ทำหน้าที่รับ Packet ข้อมูลจากชั้นสื่อสารชั้นบนซึ่งมีขนาดไม่เกิน 64 กิโลไบต์แล้วจัดการแบ่งออกเป็นเฟรมเพื่อส่งต่อไปยังผู้รับทางฝั่งผู้รับก็จะจัดการรวบรวมเฟรมต่างๆ ที่ส่งมาแล้วประกอบกลับเป็น Packet เพื่อส่งต่อไปให้ชั้นสื่อสารชั้นบนต่อไป

ประการที่สอง จัดการผสมสัญญาณ และแยกสัญญาณของ Packet ข้อมูลที่ถูกส่งมาจากหลายแหล่ง เมื่อจัดการสร้าง Packet ข้อมูลขึ้นมาใหม่ทางฝั่งผู้รับ ชั้นสื่อสาร L2CAP จะทำการสืบค้นว่าโปรโตคอลใดเป็นเจ้าของ Packet นั้นและจัดการส่งคืนให้

ประการที่สาม ชั้นสื่อสาร L2CAP จัดการเกี่ยวกับคุณภาพของการให้บริการ ทั้งในช่วงที่กำลังจัดตั้งลิงค์และในระหว่างการทำงานปกติ ในระหว่างการจัดตั้งลิงค์ก็จะสามารถทำการต่อรองขนาดของปริมาณข้อมูลที่จะส่งเข้าไปในข้อมูลเฟรม เพื่อป้องกันไม่ให้เฟรมขนาดใหญ่สร้างปัญหาให้แก่อุปกรณ์ขนาดเล็ก คุณสมบัติข้อนี้ จึงมีความจำเป็นอย่างยิ่งเพราะอุปกรณ์ส่วนหนึ่งไม่สามารถรับเฟรมข้อมูล ขนาด 64 กิโลไบต์ได้

### 2.1.11 โครงสร้างเฟรมในบลูทูธ

รูปแบบของเฟรมนั้นมีอยู่มากมาย รูป แสดงรูปแบบเฟรมข้อมูลที่ถูกนำมาใช้งานมากที่สุด เฟรมเริ่มต้นด้วย Access code ซึ่งใช้บอกโหนดแม่ทราบว่าโหนดลูกที่อยู่ในขอบเขตการกระจายสัญญาณวิทยุที่มีโหนดแม่อยู่ 2 โหนดจะได้ทราบว่าข้อมูลใดเป็นข้อมูลสำหรับโหนดแม่โหนดใด ส่วนต่อไป คือ ข้อมูลส่วนหัว (header) ขนาด 54 บิต ซึ่งประกอบด้วยเขตข้อมูลในชั้นสื่อสารย่อยควบคุมการใช้สื่อ ต่อมาคือ เขตข้อมูลจริง (Data field) มีขนาดไม่เกิน 2744 บิต (สำหรับเฟรมขนาด 5 ช่องเวลา) สำหรับเฟรมขนาดหนึ่งช่องเวลา (Single time slot) ก็มีรูปแบบเฟรมเช่นเดียวกัน เพียงแต่มีขนาดของเขตข้อมูลจริงเพียง 240 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

72	54	0-2744
Access code	Header	Data

3	4	1	1	1	8
Addr	Type	F	A	S	Checksum

The 18-bit header is repeated three times for a total of 54 bits

### รูปที่ 2.5 โครงสร้างเฟรมข้อมูลในระบบบลูทูธ

พิจารณาข้อมูลส่วนหัวของเฟรมข้อมูล เขตข้อมูลที่อยู่ใน Addr (Address) บอกให้ทราบว่าอุปกรณ์ใดใน 8 อย่างที่เป็นเจ้าของเฟรมนั้นเขตข้อมูล Type ใช้บอกชนิดของเฟรม (ACL, SCO หรือ null) บอกชนิดของการแก้ไขความถูกต้องของข้อมูลที่ใช้ในเขตข้อมูล (data field) และบอกให้ทราบว่าเฟรมนั้นยาวกี่ช่วงเวลา บิต flow ถูกใส่เข้ามาโดยโหนดลูกเมื่อบัฟเฟอร์ของโหนดลูกเต็ม และไม่สามารถรองรับข้อมูลได้อีก นี่คือรูปแบบพื้นฐานของเทคนิคการควบคุมการไหลเวียนของข้อมูล บิต Acknowledgement ใช้ในการยืนยัน ACK เข้าไปในเฟรมข้อมูล (โดยไม่ต้องส่งเฟรม Acknowledgement เป็นการเฉพาะ) บิต sequence ใช้บอกหมายเลขของเฟรมเพื่อตรวจหาเฟรมที่มีการส่งซ้ำ เนื่องจากโปรโตคอลนี้ทำงานในระบบ stop-and-wait การตรวจสอบเฟรมจึงต้องการใช้เพียง 1 บิตเท่านั้น ส่วนสุดท้ายคือ ข้อมูลสำหรับการตรวจสอบความถูกต้องของข้อมูลส่วนหัว (Checksum) ขนาด 8 บิตข้อมูลส่วนหัวมีขนาดความยาวรวม 18 บิตจะถูกส่งซ้ำ 3 ครั้งรวมเป็น 54 บิต ดังแสดงในรูป ทางด้านข้อมูลผู้รับ จะใช้วงจรอิเล็กทรอนิกส์แบบง่ายๆ ในการตรวจสอบบิตข้อมูลส่วนหัวทั้ง 3 สำเนา ซึ่งถ้าเหมือนกันทั้งหมดจะรับข้อมูลนี้ไว้ แต่ถ้าไม่เหมือนกันก็จะเลือกใช้ข้อมูลที่เหมือนกัน 2 ใน 3 สำเนา ดังนั้น จากข้อมูลทั้งหมด 54 บิตถูกนำมาใช้ส่งข้อมูลส่วนหัวขนาดเพียง 10 บิต (Address, Type, Acknowledgement และ Sequence) เท่านั้น เหตุผลก็เนื่องมาจาก ต้องการส่งข้อมูลที่มีความไว้วางใจได้สูงในสิ่งแวดล้อมที่มีการรบกวนสูง โดยใช้อุปกรณ์ราคาถูก ใช้พลังงานน้อย และมีความสามารถในการคำนวณที่จำกัด จึงจำเป็นจะต้องส่งข้อมูลที่มีความซ้ำซ้อนสูงมาเป็นพิเศษ

สำหรับรูปแบบของเขตข้อมูลจริง (Data field) นั้นมีอยู่หลายรูปแบบใน ACL เฟรม ในขณะที่ SCO เฟรมนั้นมีรูปแบบที่ง่ายกว่าเพราะเขตข้อมูลจริงจะมีขนาด 240 บิตเสมอ รูปแบบที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

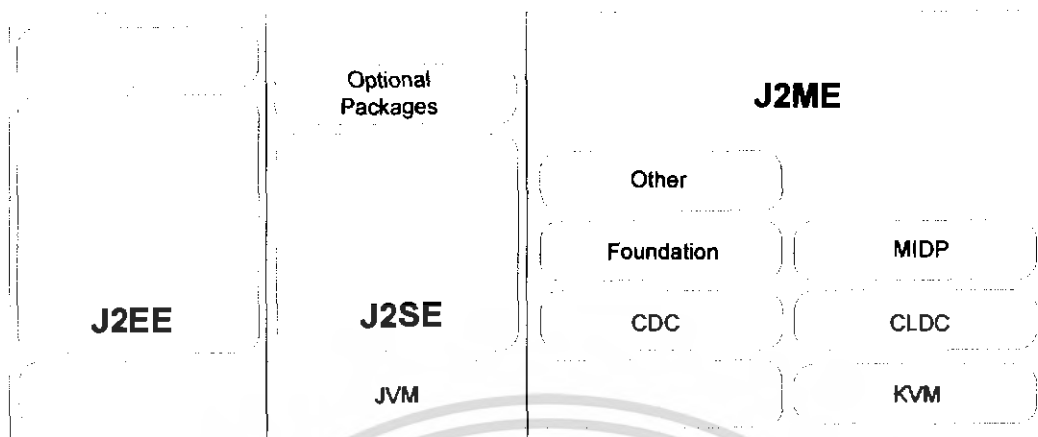
ใช้ใน ACL เฟรมแบ่งออกเป็น 3 ชนิดมีขนาด 80,160 และ 240 บิต (ขนาดของข้อมูลจริง) ส่วนที่เหลือถูกนำมาใช้สำหรับการตรวจสอบความถูกต้องของข้อมูล ชนิดที่มีความไว้วางใจมากที่สุด (ขนาด 80 บิต) จะส่งสำเนาของข้อมูลจริงมา 3 ชุดเช่นเดียวกับข้อมูลส่วนตัว

เนื่องจากโหนดลูกจะสามารถส่งข้อมูลออกมาได้ทางช่องเวลาที่มียหมายเลขที่เท่ากัน จึงมีจำนวนช่องเวลาเป็น 800 ช่องต่อวินาที เช่นเดียวกับที่โหนดแม่มีใช้งาน ด้วยข้อมูลจริงขนาด 80 บิต ความจุของช่องสัญญาณจากโหนดลูกและจากโหนดแม่ คือ 64,000 bps ซึ่งพอดีสำหรับการส่งข้อมูลแบบ 2 ทางสมบูรณจำนวนหนึ่งช่องสัญญาณสำหรับข้อมูล PCM voice channel หมายความว่าช่องสื่อสารในพีโคเน็ตนี้มีความอิ่มตัวแล้ว แม้ว่าจะมีความสามารถในการส่งข้อมูลได้สูงถึง 1 Mbps ก็ตาม แต่สำหรับการส่งข้อมูลที่มีความไว้วางใจได้ในระดับต่ำ (เฟรมขนาด 240 บิตโดยไม่มี การส่งข้อมูลซ้ำ) จะสามารถสร้างช่องสื่อสารแบบสองทางสมบูรณได้ 3 ช่องซึ่งเป็นเหตุผลที่ทำให้สามารถสร้างลิงค์ SCO ได้ 3 ช่องสำหรับโหนดลูกแต่ละโหนด

## 2.2 เทคโนโลยี J2ME

จุดมุ่งหมายที่สำคัญของการออกแบบภาษาจาวาคือ โปรแกรมต้องทำงานบนเครื่องต่างระบบกันได้ โดยเรียกคุณสมบัตินี้ว่า "ไม่ขึ้นกับระบบ" ซึ่งเป็นภาษาเชิงออบเจกต์ที่สามารถทำงานได้ ทุกระบบปฏิบัติการ "Write Once, Run Anywhere" เทคโนโลยีจาวาแบ่งกลุ่มออกเป็น 3 กลุ่มเพื่อใช้งานกับอุปกรณ์ที่มีคุณสมบัติที่แตกต่างกันดังนี้คือ

1. Standard Edition (J2SE): เป็นเทคโนโลยีจาวา ที่ออกแบบเพื่อนำมาใช้พัฒนางานบนเครื่องคอมพิวเตอร์เดสก์ทอปและคอมพิวเตอร์เวิร์กสเตชัน
2. Enterprise Edition (J2EE): ถูกออกแบบมาเพื่อให้สามารถใช้งานบนระบบเครื่องเซิร์ฟเวอร์รองรับการทำงานร่วมกับ Servlets, JSP และ XML
3. Micro Edition (J2ME): มีเป้าหมายสำหรับพัฒนาแอปพลิเคชันที่ ทำงานบนอุปกรณ์เครื่องใช้ที่มีทรัพยากรจำกัดไม่ว่าจะเป็น ขนาดของหน่วยความจำ ความสามารถในการประมวลผล เช่น โทรศัพท์มือถือ เพจเจอร์ ปาล์ม PDA เป็นต้น



รูปที่ 2.6 เปรียบเทียบ Java editions เวอร์ชันต่าง ๆ

J2ME หรือ Java 2 Micro Edition มีจุดมุ่งหมายสำหรับอุปกรณ์ที่มีขนาดเล็กซึ่งมีทรัพยากรจำกัดทางด้านพลังงานความสามารถในการประมวลผล หน่วยความจำ อุปกรณ์จำพวกนี้มีมากมาย เช่น เพจเจอร์ โทรศัพท์เคลื่อนที่ ปาล์ม พีดีเอ เป็นต้น นอกจากนี้ J2ME ยังสามารถใช้พัฒนาแอปพลิเคชันให้ทำงานบนอุปกรณ์ที่ไม่ได้เป็นอุปกรณ์ไร้สายได้อีกด้วยถ้าไม่มีข้อจำกัดเพื่อความหนาแน่นและติดตั้งซอฟต์แวร์เช่น กล้องรับสัญญาณดาวเทียมสำหรับทีวี อินเทอร์เน็ตทีวี เป็นต้น

ในการออกแบบโปรแกรม J2ME มีการแบ่งออกเป็นเอ디션ย่อย ก็เพื่อความเหมาะสมกับการเลือกไปพัฒนาแอปพลิเคชันตามขนาดและ ลักษณะของงานซึ่งอาจจะแบ่งแยกออกเป็นกลุ่มๆ เหมือนกับเป็นการจัดแบ่งกลุ่มของคลาสและแพคเกจของแต่ละเอ디션

J2ME หรือ JAVA 2 Micro Edition คือ เทคโนโลยีหนึ่งของตระกูล JAVA ซึ่งเป็นทั้งสถานะแวดล้อมของการพัฒนาและของตัวรันไทม์ (Run Time) ไปพร้อม ๆ กัน ซึ่งเป้าหมายของ J2ME คือการนำซอร์ฟแวร์ของ JAVA เข้าไปรัน และทำงานอยู่บนอุปกรณ์อิเล็กทรอนิกส์อย่างเช่น โทรศัพท์มือถือ เป็นต้น ซึ่งอุปกรณ์เหล่านี้มีขนาดเล็ก หน่วยความจำน้อย และมีขีดความสามารถในการประมวลผลต่ำ J2ME จึงช่วยสร้างแอปพลิเคชันไร้สายยุคใหม่ เช่น เกมสับน โทรศัพท์มือถือ , พานิชย์ของ โทรศัพท์มือถือ และแอปพลิเคชันของไคลเอนต์/เซิร์ฟเวอร์

อุปกรณ์ที่สามารถใช้เทคโนโลยีได้ J2ME แบ่งออกเป็น 2 กลุ่มดังนี้ คือ

1. **Personal, mobile, connected, information devices** เช่น โทรศัพท์มือถือ เพจเจอร์ และออบแกในเซอร์ เป็นต้น ซึ่งเป็นอุปกรณ์ที่มีการอินเตอร์เฟสแบบง่าย ๆ มีหน่วยความจำประมาณ 128 - 512 กิโลไบต์และมีแบนด์วิดธ์ที่ต่ำ ส่วนการเชื่อมต่อของอุปกรณ์ไม่ได้ใช้

โพรโตคอล TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2. **Shared, fixed, connected, information devices** เช่น อุปกรณ์เซตท็อป(set-top boxes) อินเทอร์เน็ตทีวี ความบันเทิงในรถยนต์และระบบการเดินเรือ เป็นต้น อุปกรณ์จำพวกนี้มี ยูสเซอร์อินเทอร์เน็ตให้ใช้หลายแบบ มีหน่วยความจำประมาณ 2 – 16 เมกะไบต์ ส่วนการ เชื่อมต่อกับเน็ตเวิร์กนั้น โดยมากแล้วใช้โปรโตคอล TCP/IP ที่มีแบนด์วิดค์ค่อนข้างสูง

### 2.2.1 สถาปัตยกรรมของ J2ME

การออกแบบโครงสร้างเทคโนโลยีจาวาตระกูล J2ME มีการ จัดแบ่งหน้าที่ทำงานออกเป็น 4 ระดับชั้นด้วยกัน เพื่อรองรับกับการใช้งานกับอุปกรณ์ต่างๆ คือ Operating System ,Java Virtual Machine ,Layer Configuration และ Layer Profile Layer

- **Host Operating System** จะเป็นส่วนของระบบปฏิบัติการ เช่นเดียวกับบนเครื่อง Desktop หรือ Laptop ที่มี Windows เป็นระบบปฏิบัติการ บน โทรศัพท์มือถือก็มีด้วยเช่นกัน อาทิ Nokia 7650, 3650 จะมี Symbian OS เป็นระบบปฏิบัติการ, เครื่อง Palm จะมี Palm OS เป็นระบบปฏิบัติการ
- **Java Virtual Machines (JVM)** จะเป็นส่วนของระบบจัดการ ที่ควบคุม และทำงาน ให้สามารถ ทำงานร่วมกันได้ ระหว่าง Java กับ Host Operating System โดยมากจะเป็นการแปลงจาก code Java ไปเป็นคำสั่ง ที่ Host Operating System เข้าใจ และทำงานร่วมกันได้
- **Configuration** เป็นกลุ่มของ Class Library (คลัง Class) ที่ครอบคลุม ถึงอุปกรณ์ต่างๆ ที่อยู่ใน กลุ่ม
- **Profiles** เป็นกลุ่มของ คำสั่ง,API (Application Programming Interface) ที่ใช้สำหรับอุปกรณ์ แต่ละประเภทโดยเฉพาะ

หัวใจหลักของสถาปัตยกรรมหรือโครงสร้างของ J2ME สามารถแบ่งออกเป็น 2 ส่วน คือ คอนฟิกูเรชัน (Configurations) และ โพรไฟล์ (Profiles) ซึ่งมีลักษณะเป็น โมดูลที่มีความยืดหยุ่น พร้อมกับ สามารถนำไปใช้งานกับอุปกรณ์ที่ต่างกัน ได้ โดยสามารถปรับแต่งได้ตามความต้องการ ของผู้บริโภคและนอกจากนี้ผู้ผลิต หรือผู้พัฒนาซอฟต์แวร์บนอุปกรณ์ ยังสามารถขยายหรือเพิ่ม ความสามารถใน Virtual Machine ได้ ดังนั้นจึง ไม่ได้เป็นแพลตฟอร์ม เฉพาะกับอุปกรณ์ชนิดใด ชนิดหนึ่ง แต่ก็ยังมีเงื่อนไขตามแนวที่กำหนดของ J2ME อยู่

#### 2.2.1.1 Configuration

เป็นกลุ่มของ API ระดับต่ำ ( Low-level API ) ทำหน้าที่เป็นตัวกำหนดคุณลักษณะใน ระหว่างการรันแอปพลิเคชันปัจจุบันได้แบ่ง Configuration สำหรับ J2ME ออกเป็น 2 ประเภทคือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง 72238 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการจัดกลุ่มจัดโดยแบ่งตาม device ซึ่งมีความคล้ายกันในเรื่องของขนาดของ memory และ processing power

- CLDC (Connected Limit Device Configuration)

เป็นคอนฟิกูเรชันสำหรับอุปกรณ์ที่มีหน่วยความจำน้อย มีกำลังของหน่วยประมวลผลต่ำ กินไฟน้อยและมีข้อจำกัดในด้านการเชื่อมต่อเครือข่าย ซึ่งคอนฟิกูเรชัน CLDC นี้จะใช้ Kilobyte Virtual Machine หรือ KVM เป็น Virtual Machine ซึ่งอุปกรณ์ที่จะใช้คอนฟิกูเรชันนี้ได้แก่ โทรศัพท์มือถือ , PDA เป็นต้น

- CDC (Connect Device Configuration)

เป็นคอนฟิกูเรชันสำหรับอุปกรณ์ที่มีความสามารถสูงกว่า CLDC โดยสนับสนุนอุปกรณ์ที่มีหน่วยความจำตั้งแต่ 512 กิโลไบต์จนถึง 2 เมกะไบต์ โดยใช้ C-Virtual Machine หรือ CVM เป็น Virtual Machine ตัวอย่างอุปกรณ์ที่ใช้คอนฟิกูเรชันนี้ได้แก่ Pocket PC และอุปกรณ์จำพวก set-top box ของทีวี เป็นต้น

### 2.2.1.2 Profile

เป็นข้อกำหนดทางด้านคุณลักษณะของอุปกรณ์ที่จะรัน J2ME ซึ่งอาจจะเป็นโพรไฟล์สำหรับอุปกรณ์หนึ่งชิ้นโดยเฉพาะ หรือเป็นโพรไฟล์ที่ทำงานคล้ายๆกันก็ได้ ตัวอย่างข้อกำหนดของโพรไฟล์ เช่น อุปกรณ์มีอินเทอร์เน็ตเฟสติดต่อกับผู้ใช้อย่างไร (ในการรับและแสดงข้อมูล) , อุปกรณ์จะติดต่อกับเครือข่ายอย่างไรหรือเก็บข้อมูลไว้อย่างไร เป็นต้น ดังนั้นในการพัฒนาแอปพลิเคชัน J2ME จำเป็นจะต้องเลือกโพรไฟล์อย่างน้อย 1 ตัว เพื่อเป็นข้อกำหนดแอปพลิเคชันที่พัฒนาขึ้น ว่าสามารถจะนำไปรันกับอุปกรณ์ใดได้บ้าง โพรไฟล์ที่ได้รับความนิยมมากที่สุดคือ MIDP (Mobile Information Device Profile)

#### ตัวอย่าง Profile สำหรับอุปกรณ์แต่ละชนิด

Mobile Information Device Profile (MIDP) เป็นโพรไฟล์สำหรับโทรศัพท์มือถือ ดังนั้นจึงมีข้อจำกัดหลายอย่าง ได้แก่ ขนาดหน่วยความจำทั้งแรมและรอมรวมกันไม่เกิน 512 กิโลไบต์ ใช้แบตเตอรี่เป็นตัวให้พลังงาน เชื่อมต่อกับเน็ตเวิร์กไร้สายด้วยแบนด์วิดท์ที่จำกัด จอภาพขนาดเล็ก นอกจากนี้ส่วนใหญ่ยังเป็นจอภาพขาวดำ พวกจอภาพสีราคายังค่อนข้างสูง

**PDA Profile** เป็นโพรไฟล์สำหรับพีดีเอ มีหน่วยความจำแรมและรอมรวมกันไม่น้อยกว่า 512 กิโลไบต์สำหรับจาวารัน ไทม์และไลบรารี อย่างไรก็ตามหน่วยความจำส่วนใหญ่มีไม่เกิน 16 เมกะไบต์ พีดีเอยังมีข้อจำกัดเรื่องพลังงานที่ใช้จากแบตเตอรี่ แต่ก็มี ข้อดีเรื่องจอภาพขนาดใหญ่และความ

ละเอียดสูงกว่า MIDP คือไม่น้อยกว่า 20,000 พิกเซล สามารถบันทึกข้อมูลโดยการเขียนด้วยสไตลัส หรือคีย์บอร์ดขนาดเล็ก

**Personal Profile** เป็นโปรไฟล์ที่ต่อยอดมาจาก PersonJava Application Environment อุปกรณ์ในกลุ่มนี้ส่วนใหญ่จะมีความสามารถในการเชื่อมต่ออินเทอร์เน็ตค่อนข้างสูง หน่วยความจำรวมอย่างต่ำ 2.5 เมกะไบต์ หน่วยความจำแรม 1 เมกะไบต์ นอกจากนี้ยังใช้กราฟิกอินเตอร์เฟซที่สวยงาม

**Foundation Profile** เป็นโปรไฟล์อีกกลุ่มสำหรับอุปกรณ์ที่ต้องการใช้การเชื่อมต่อเน็ตเวิร์กที่มีประสิทธิภาพมาก แต่ไม่ต้องการความสามารถเรื่องกราฟิกอินเตอร์เฟซ ส่วนใหญ่มีหน่วยความจำรวม 1 เมกะไบต์ หน่วยความจำแรม 512 กิโลไบต์

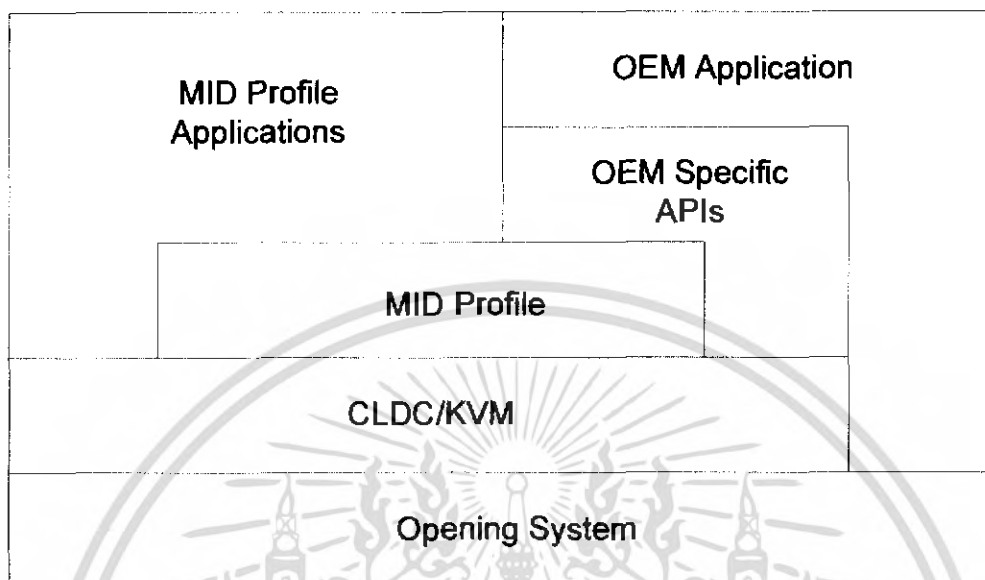
### 2.2.1.3 Java Virtual Machines

JVM เป็นตัวกลางที่ภาษาจาวานำมาใช้เพื่อทำการแปลโปรแกรมแล้วส่งต่อไปให้ระบบปฏิบัติการ จากนั้นก็จะทำแปลงคลาสให้เป็นภาษาเครื่องและทำงานต่อไป โดยโปรแกรมภาษาจาวาทุกตัวจะต้องทำงานอยู่ในภายใต้ Java Virtual Machine ซึ่งมีอยู่ในตัวไม่ว่าจะเป็น Applet servlet และอื่นๆ นอกจากนี้แล้ว JVM ยังมีหน้าที่ความรับผิดชอบเตรียมความพร้อมสำหรับความปลอดภัยอีกด้วย สำหรับ CDC มีการกำหนดให้สามารถใช้ virtual machine ชุดเดียวกันกับ J2SE แต่ CLDC ไม่ได้กำหนดให้ใช้กลุ่มคลาสเดียวกันทั้งหมด ทาง Sun จึงได้พัฒนาเครื่องมือสำหรับสำหรับ virtual machine ขึ้นมาใหม่เรียกว่า K Virtual Machine หรืออาจจะเรียกอีกอย่างหนึ่งว่า KVM สำหรับ virtual machine นี้ได้รับการออกแบบพัฒนาเป็นพิเศษเหมาะสำหรับอุปกรณ์ประเภทมือถือ โดย KVM เป็นจาวาเวอร์ชวลแมชีน ขนาดเล็กที่พัฒนา Java virtual machine ขึ้นมาใหม่ ซึ่งมีคุณสมบัติดังนี้

- virtual machine ต้องการหน่วยความจำ 40 และ 80 กิโลไบต์เท่านั้น
- ต้องการหน่วยความจำแบบไดนามิก 20-40 กิโลไบต์ เท่านั้น
- สามารถรับบน 16 บิต โดยมีความเร็วในการประมวลผล 25 เมกะเฮิร์ตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2.2 สถาปัตยกรรมของ MIDP



รูปที่ 2.7 สถาปัตยกรรมของ MIDP

Mobile Information Device Profile หรือ MIDP เป็นโพรไฟล์ที่ Sun ได้พัฒนาขึ้นมาสำหรับอุปกรณ์ที่มีหน้าจอขนาดเล็กหรือโดยเป็นแบบสัมผัสบนหน้าจอ คีย์บอร์ด สามารถติดต่อ Mobile network ด้วยแบนด์วิดธ์ต่ำหรือค่อนข้างจำกัด ส่วนแอปพลิเคชันที่สร้างขึ้นมาจากใช้ MIDP นี้จะเรียกว่า MIDlet เป็นหัวใจหลักอันสำคัญของ Java Phone เป็นแอปพลิเคชันที่พัฒนาและมีความสัมพันธ์กันกับ CLDC KVM สำหรับ โครงสร้างมีทั้งหมด 5 เลเยอร์ด้วยกัน MID คือ hardware layer Native system software layer KVM layer CLDC layer และ MIDP layer ซึ่งในแต่ละชั้นก็มีหน้าที่ที่แตกต่างกัน อุปกรณ์ในแต่ละคลาสโอบาริจะมีคุณสมบัติเฉพาะและในรุ่นนั้นๆ CLDC จะเป็นแพลตฟอร์มพื้นฐานที่ทำงานร่วมกับ MIDP APIs โดยเป็นกลุ่มคลาสที่อยู่ใน CLDC เป็นคลาสที่ทำงานที่มีมาตรฐาน

**MIDP (Mobile Information Device Profile) ได้จัดเตรียมเกี่ยวกับ**

- User Interface จัดการเกี่ยวกับการแสดงผล
- Persistent storage จัดการเกี่ยวกับการเก็บข้อมูล และฐานข้อมูล
- Networks จัดการเกี่ยวกับการเชื่อมต่อ เน็ตเวิร์ค
- Application life-cycle จัดการเกี่ยวกับ ลำดับขั้นตอนการทำงาน
- Event handling จัดการเกี่ยวกับ อีเวนต์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### MIDP Packages จะมีอยู่ 3 Packages คือ

- javax.microedition.midlet ,เป็น API ในการสร้างโปรแกรมหลัก
- javax.microedition.lcdui ,เป็น API ในการจัดการ User Interface
- javax.microedition.rms ,RMS (Record Management System) เป็นส่วนของการเก็บข้อมูล เช่นเดียวกับฐานข้อมูล

#### 2.2.3 MIDlet

การพัฒนาแอปพลิเคชันด้วยจาวานั้น สำหรับ J2ME แล้วแอปพลิเคชันที่ได้ออกมาจะเรียกว่า MIDlet ซึ่งจะนำไปรันบนอุปกรณ์ต่างๆที่สนับสนุนเทคโนโลยี J2ME ซึ่ง MIDlet ที่พัฒนาขึ้นจะประกอบด้วยไฟล์ 2 ไฟล์ คือไฟล์นามสกุล .jar และไฟล์นามสกุล .jad ซึ่งไฟล์นามสกุล .jar นั้นเป็นที่เก็บคลาสไฟล์ต่างๆ (นามสกุล .class) ของ MIDlet นั้น ส่วนไฟล์ .jad (Java Description) เป็นไฟล์ข้อความธรรมดาที่ใช้บรรยายหรืออธิบายไฟล์ .jar ว่ามีชื่ออะไร หรือมีขนาดไฟล์เท่าไร เป็นต้น MIDlet ยังเป็น class ที่อยู่ใน Package ของ javax.microedition.midlet และเป็นโปรแกรมที่พัฒนาขึ้นจาก MIDP และสืบทอดจาก MIDlet

Application ที่เราเขียนขึ้นมาเพื่อใช้กับอุปกรณ์ที่เป็น MIDP device เราจะเรียกว่า MIDlet ซึ่งลักษณะของมันก็จะเหมือนกับ Java Applet ที่คนเขียนจาวาบนเว็บ รู้จักการเป็นอย่างดี ใน midlet จะต้อง extend javax.microedition.midlet.MIDlet class และ impliment 3 abstract method คือ startApp(), pauseApp() และ destroyApp()

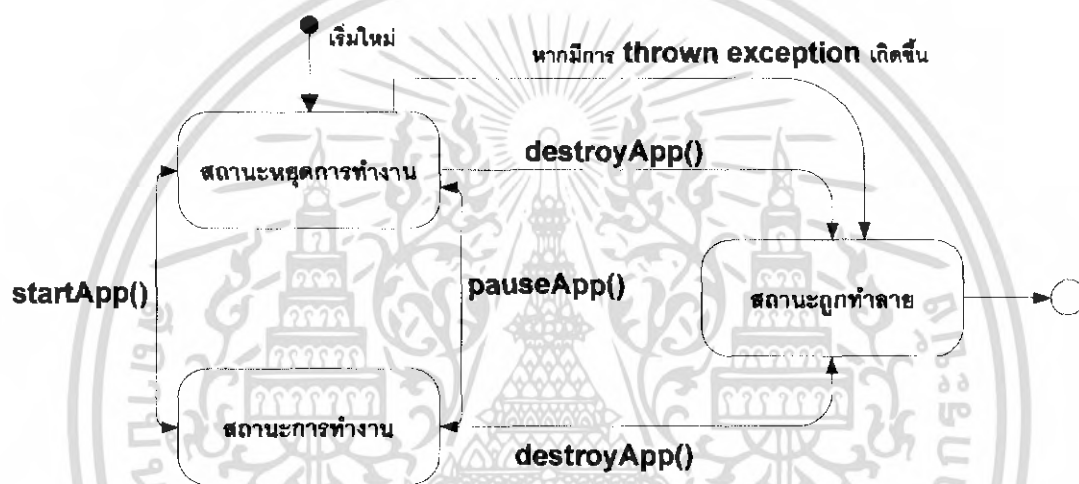
เมื่อได้ทำการสร้าง class ต่างๆที่ได้ทำการสืบทอดมาจาก MIDlet Package ที่ชื่อ javax.microedition.midlet.MIDlet แล้ว, class จะมีวงจร การทำงานเริ่มจากการสร้าง หลังจากนั้นจะอยู่ในสถานะ Paused(หยุด) เมื่อเรา ต้องการให้มีการทำงานเกิดขึ้น เราก็ต้องเรียกใช้ method ที่ชื่อ startApp() สถานะของคลาสก็เปลี่ยนไปเป็นสถานะ Active(ทำงาน) ดังรูป, และหากต้องการให้ class หยุดการทำงาน สามารถทำได้โดย เรียกใช้ Method pauseApp(), สุดท้ายก็คือการ จบการทำงาน โดยเรียกใช้ Method destroyApp() โปรแกรมก็จะเข้าสู่สถานะ Destroyed(ถูกทำลาย) สถานะต่างๆของ MIDlet มีอยู่ 3 สถานะด้วยกัน คือ

1. Paused state (สถานะหยุดการทำงาน) จะเกิดขึ้น ตอนแรกที่เราทำการเรียก run โดยการ ใช้ method startApp(), มีการเรียกใช้ method pauseApp() หรือ notifyPaused()
2. Active state (สถานะการทำงาน) จะเกิดขึ้น ตอนแรกๆที่เริ่มต้นโปรแกรม, มีการเรียกใช้ method startApp() หรือ มีการเรียกใช้ method resumeRequest() ในตอนที่สถานะการก ทำงานยังเป็น pauseApp()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Destroyed state (สถานะถูกทำลาย) จะเกิดขึ้นหลังจากที่มีการใช้ method `destroyApp()` หรือ `notifyDestroy()`

ในการเปลี่ยนสถานะ (state) ของการทำงาน จะใช้ method 3 อันด้วยกัน คือ  
`resumeRequest()` ใช้เพื่อ ให้กลับมาอยู่ในสถานะ Active state อีกครั้ง  
`notifyPaused()` ใช้เพื่อ สั่งหยุดการทำงาน  
`notifyDestroy()` ใช้เพื่อ สั่งทำลาย เพื่อจบการทำงาน

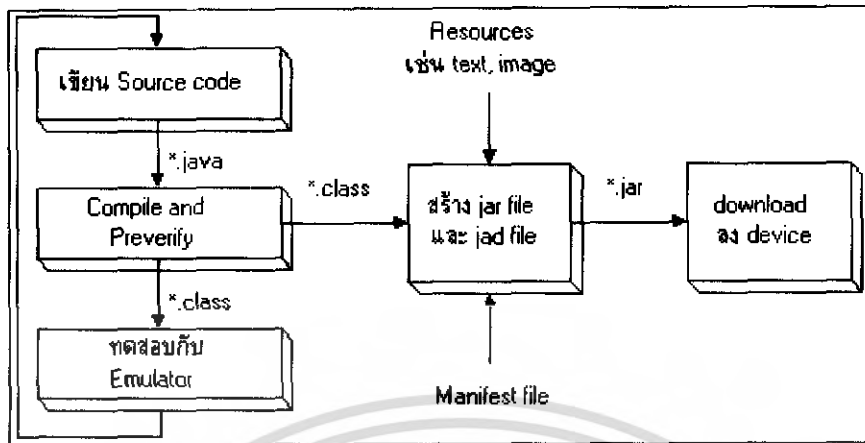


รูปที่ 2.8 สถานะต่างๆของ MIDlet

ขั้นตอนของการพัฒนา midlet

1. เขียน source code
2. compile java code
3. run class ที่ได้จากการ compile ผ่าน preverify tool
4. ใส่ class ที่ผ่านการ verify และ resource file ทั้งหมดลงใน JAR file
5. ใช้ emulator ทดสอบโปรแกรม
6. download application ลงใน device

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ขั้นตอนของการพัฒนา midlet

### 2.3 JSR 82 Bluetooth API และ OBEX API

เป็นเอกสารอ้างอิงในการเขียนโค้ดเกี่ยวกับบลูทูธในการเขียนโปรแกรมโดยใช้ภาษา Java โดยจะประกอบไปด้วย javax.bluetooth และ javax.obex

โดยในที่นี้เราได้ทำการศึกษาและพัฒนาในส่วนของ Bluetooth ดังนั้นเราจะอธิบายในส่วนของ javax.bluetooth โดยมีรายละเอียดดังต่อไปนี้

#### 1. Interface Summary

DiscoveryListener อินเตอร์เฟสนี้เป็นตัวอนุญาตให้แอปพลิเคชันต่างๆ รับผิดชอบต่อการค้นหาอุปกรณ์และการให้บริการของบลูทูธ

L2CAPConnection อินเตอร์เฟสนี้เป็นตัวแทนของช่องสัญญาณแบบเชื่อมต่อตลอดเวลาของ L2CAP

L2CAPConnectionNotifier เป็นอินเตอร์เฟสที่มีไว้เพื่อเป็นตัวแจ้งการเชื่อมต่อของ L2CAP

ServiceRecord อินเตอร์เฟสนี้ใช้อธิบายคุณลักษณะต่างๆ ของบริการของบลูทูธ

#### 2. Class Summary

DataElement สำหรับคลาสนี้จะเป็นการกำหนดชนิดข้อมูลต่างๆ ที่คุณลักษณะของบริการบลูทูธจะมี

DeviceClass สำหรับคลาสนี้เป็นตัวแทนของ the class of device record (CoD) ดังถูกกำหนดโดยข้อกำหนดของบลูทูธ

DiscoveryAgent ในคลาสนี้แสดงเมธอดต่างๆ เพื่อแสดงอุปกรณ์และค้นหาการให้บริการ

LocalDevice เป็นคลาสที่กำหนดฟังก์ชันพื้นฐานของการจัดการบลูทูธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RemoteDevice เป็นคลาสที่เป็นตัวแทนของการควบคุมอุปกรณ์บลูทูธ  
 UUID เป็นคลาสที่กำหนด Universally Unique Identifiers

### 3. Exception Summary

BluetoothConnectionException จะถูก thrown ก็ต่อเมื่อการเชื่อมต่อบลูทูธไม่สามารถสร้างได้อย่างสมบูรณ์

BluetoothStateException จะถูก thrown ก็ต่อเมื่อการร้องขอไปยังระบบบลูทูธนั้นไม่สามารถรองรับสถานะปัจจุบันได้

ServiceRegistrationException จะถูก thrown ก็ต่อเมื่อมีข้อผิดพลาดในการเพิ่ม service record ไปยังฐานข้อมูลของ local Service Discovery (SDDDB) หรือเพื่อแก้ไขการบริการที่มีอยู่แล้วใน SDDDB

#### 2.3.1 Interface DiscoveryListener

**public interface DiscoveryListener**

สำหรับอินเตอร์เฟซนี้ อนุญาตให้แอปพลิเคชันได้รับผลจากการค้นพบบริการและอุปกรณ์ ในอินเตอร์เฟซนี้จะประกอบไปด้วย เมธอดทั้งหมด 4 เมธอด ประกอบไปด้วยการค้นหาอุปกรณ์ 2 เมธอดและการค้นหาการให้บริการอีก 2 เมธอด

ตารางที่ 2.2 รายละเอียดในส่วนของฟิลด์ (Field) ของอินเตอร์เฟซ DiscoveryListener

Field Summary	
static int	INQUIRY_COMPLETE แสดงถึงการเสร็จสิ้นของการค้นพบอุปกรณ์
static int	INQUIRY_ERROR แสดงให้เห็นว่าการตรวจสอบว่าการร้องขอนั้นมีความผิดพลาดที่จะทำให้เสร็จสมบูรณ์แต่จะไม่ถูกยกเลิก
static int	INQUIRY_TERMINATED แสดงถึงการค้นพบอุปกรณ์ได้ถูกยกเลิกโดยแอปพลิเคชันและการค้นพบอุปกรณ์นั้นจะไม่เสร็จสมบูรณ์
static int	SERVICE_SEARCH_COMPLETE แสดงถึงการเสร็จสิ้นของการค้นหาการให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 รายละเอียดในส่วนของฟิลด์ (Field) ของอินเทอร์เฟซ DiscoveryListener (ต่อ)

Field Summary	
static int	SERVICE_SEARCH_DEVICE_NOT_REACHABLE แสดงให้เห็นว่าการค้นหาการให้บริการนั้นไม่สามารถทำได้เพราะการควบคุมอุปกรณ์ ชี้ให้เห็นว่า DiscoveryAgent.searchServices() ไม่สามารถเชื่อมต่อได้
static int	SERVICE_SEARCH_ERROR แสดงถึงการยุติเนื่องจากมีข้อผิดพลาดในการค้นหาการให้บริการ
static int	SERVICE_SEARCH_NO_RECORDS แสดงถึงการเสร็จสิ้นการค้นหาการให้บริการด้วยบริการที่ไม่ได้ถูกบันทึกไว้ในอุปกรณ์
static int	SERVICE_SEARCH_TERMINATED แสดงให้เห็นถึงการค้นหาการให้บริการได้ถูกยกเลิกโดยแอปพลิเคชันและการค้นหาการ ให้บริการนั้นจะไม่เสร็จสมบูรณ์

ตารางที่ 2.3 รายละเอียดในส่วนของเมธอด (Method) ของอินเทอร์เฟซ DiscoveryListener

Method Summary	
void	deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) จะถูกเรียกก็ต่อเมื่ออุปกรณ์ถูกค้นพบในระหว่างการตรวจสอบ
void	inquiryCompleted(int discType) จะถูกเรียกก็ต่อเมื่อการตรวจสอบนั้นเสร็จสมบูรณ์
void	servicesDiscovered(int transID, ServiceRecord[] servRecord) จะถูกเรียกก็ต่อเมื่อมีการค้นหาการให้บริการขณะที่มีค้นหา
void	serviceSearchCompleted(int transID, int respCode) จะถูกเรียกก็ต่อเมื่อเสร็จสิ้นการค้นหาหรือถูกยุติเมื่อมีข้อผิดพลาดเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.1.1 รายละเอียดในส่วนของ Field

#### - INQUIRY\_COMPLETED

static final int INQUIRY\_COMPLETED

แสดงถึงการเสร็จสมบูรณ์ของการค้นพบอุปกรณ์ ใช้ร่วมกับเมธอด inquiryCompleted()

ค่าของ INQUIRY\_COMPLETED มีค่าเท่ากับ 0x00 (0)

#### - INQUIRY\_TERMINATED

static final int INQUIRY\_TERMINATED

แสดงถึงการค้นพบอุปกรณ์ได้ถูกยกเลิกโดยแอปพลิเคชันและการค้นพบอุปกรณ์นั้นจะไม่เสร็จสมบูรณ์ ใช้ร่วมกับเมธอด inquiryCompleted()

ค่าของ INQUIRY\_TERMINATED มีค่าเท่ากับ 0x00 (5)

#### - INQUIRY\_ERROR

static final int INQUIRY\_ERROR

แสดงให้เห็นว่าการตรวจสอบว่าการร้องขอนั้นมีความผิดพลาดที่จะทำให้เสร็จสมบูรณ์ แต่จะไม่ถูกยกเลิก

ค่าของ INQUIRY\_ERROR มีค่าเท่ากับ 0x07 (7)

#### - SEVICE\_SEARCH\_COMPLETED

static final int SEVICE\_SEARCH\_COMPLETED

แสดงถึงการเสร็จสิ้นของการค้นพบการให้บริการ ใช้ร่วมกับเมธอด serviceSearchCompleted()

ค่าของ SEVICE\_SEARCH\_COMPLETED มีค่าเท่ากับ 0x01 (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- SERVICE\_SEARCH\_TERMINATED**

static final int **SERVICE\_SEARCH\_TERMINATED**

แสดงให้เห็นถึงการค้นหาการให้บริการได้ถูกยกเลิกโดยแอปพลิเคชันและการค้นหาการให้บริการนั้นจะไม่เสร็จสมบูรณ์ ใช้ร่วมกับเมธอด serviceSearchCompleted()

ค่าของ *SERVICE\_SEARCH\_TERMINATED* มีค่าเท่ากับ 0x02 (2)

**- SERVICE\_SEARCH\_ERROR**

static final int **SERVICE\_SEARCH\_ERROR**

แสดงถึงการยุติเนื่องจากมีข้อผิดพลาดในการค้นหาการให้บริการ ใช้ร่วมกับเมธอด serviceSearchCompleted()

ค่าของ *SERVICE\_SEARCH\_ERROR* มีค่าเท่ากับ 0x03 (3)

**- SERVICE\_SEARCH\_NO\_RECORDS**

static final int **SERVICE\_SEARCH\_NO\_RECORDS**

แสดงถึงการเสร็จสิ้นการค้นหาการให้บริการด้วยบริการที่ไม่ได้ถูกบันทึกไว้ในอุปกรณ์ ใช้ร่วมกับเมธอด serviceSearchCompleted()

ค่าของ *SERVICE\_SEARCH\_NO\_RECORDS* มีค่าเท่ากับ 0x04 (4)

**- SERVICE\_SEARCH\_DEVICE\_NOT\_REACHABLE**

static final int **SERVICE\_SEARCH\_DEVICE\_NOT\_REACHABLE**

แสดงให้เห็นว่าการค้นพบการให้บริการนั้นไม่สามารถทำได้เพราะการควบคุมอุปกรณ์ชี้ให้เห็นว่า DiscoveryAgent.searchServices() ไม่สามารถเชื่อมถึงได้ ใช้ร่วมกับเมธอด serviceSearchCompleted()

ค่าของ *SERVICE\_SEARCH\_DEVICE\_NOT\_REACHABLE* มีค่าเท่ากับ 0x06 (6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.1.2 รายละเอียดในส่วนของ Method

#### - deviceDiscovered

void **deviceDiscovered** (remoteDevice btDevice, DeviceClass cod)

จะถูกเรียกก็ต่อเมื่ออุปกรณ์ถูกค้นพบในระหว่างการตรวจสอบ เป็นการตรวจสอบการค้นหาลำดับอุปกรณ์ที่ถูกค้นพบ โดยอุปกรณ์เดียวกันอาจจะให้ผลตอบกลับมาเป็นค่าที่ซ้ำกันก็ได้

#### ตัวแปร

**btDevice** เป็นอุปกรณ์ที่ถูกพบระหว่างการตรวจสอบ

**cod** เป็นคลาสการบริการ, คลาสอุปกรณ์หลัก, และอุปกรณ์รองของอุปกรณ์ควบคุม

#### - serviceDiscovered

void **servicesDiscovered**(int transID, ServiceRecord[] servRecord)

จะถูกเรียกก็ต่อเมื่อมีการค้นพบการให้บริการขณะที่มีค้นหา

#### ตัวแปร

**transID** คือ transaction ID ของการค้นหาบริการ

**servRecord** คือรายชื่อของบริการที่ถูกพบระหว่างการร้องขอการค้นหา

#### - serviceSearchComplete

void **serviceSearchCompleted**(int transID, int respCode)

จะถูกเรียกก็ต่อเมื่อเสร็จสิ้นการค้นหาหรือถูกขุดเมื่อมีข้อผิดพลาดเกิดขึ้น ค่าสถานะตามกฎในข้อเท็จจริงของ respCode นั้นยังรวมไปถึง

SEVICE\_SEARCH\_COMPLETED

SERVICE\_SEARCH\_TERMINATED

SERVICE\_SEARCH\_ERROR

SERVICE\_SEARCH\_NO\_RECORDS

SERVICE\_SEARCH\_DEVICE\_NOT\_REACHABLE

ต้องการอธิบายค่าที่ได้แสดงตามตารางดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ค่า respCode

respCode	Reason
SEVICE_SEARCH_COMPLETED	ถ้าเสร็จสิ้นของการค้นหาการให้บริการ
SERVICE_SEARCH_TERMINATED	ถ้าการค้นหาการให้บริการได้ถูกยกเลิกโดยแอปพลิเคชันและการค้นหาการให้บริการนั้นจะไม่เสร็จสมบูรณ์
SERVICE_SEARCH_ERROR	ถ้าการยุติเนื่องจากมีข้อผิดพลาดในการค้นหาการให้บริการ
SERVICE_SEARCH_NO_RECORDS	ถ้าการเสร็จสิ้นการค้นหาการให้บริการด้วยบริการที่ไม่ได้ถูกบันทึกไว้ในอุปกรณ์
SERVICE_SEARCH_DEVICE_NOT_REACHABLE	ถ้าการค้นหาการให้บริการนั้นไม่สามารถทำได้ เพราะการควบคุมอุปกรณ์ชี้ให้เห็นว่า <code>DiscoveryAgent.searchSevices()</code> ไม่สามารถเชื่อมถึงได้

**ตัวแปร**

**transID** ค่า transaction ID ระบุนำร่องขอซึ่งจะถูกเริ่มเมื่อการค้นหาบริการ

**repsCode** ค่า response code ซึ่งบ่งบอกถึงสถานะของการดำเนินการ

**- inquiryCompleted**

`void inquiryCompleted(int discType)`

จะถูกเรียกก็ต่อเมื่อการตรวจสอบนั้นเสร็จสมบูรณ์ `discType` จะเป็น `INQUIRY_COMPLETED` หากการตรวจสอบสิ้นสุดลงปกติ หรือเป็น `INQUIRY_TERMINATED` หากการตรวจสอบถูกยกเลิกโดยการเรียกไปยัง `DiscoveryAgent.cancelInquiry()` ตัว `discType` จะเป็น `INQUIRY_ERROR` ถ้ามีข้อผิดพลาดเกิดขึ้นขณะที่กำลังดำเนินการตรวจสอบทำให้การตรวจสอบสิ้นสุดลงอย่างไม่เป็นปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตัวแปร

**discType** เป็นชนิดของการร้องขอซึ่งจะถูกทำให้สมบูรณ์รวมทั้ง INQUIRY\_COMPLETED, INQUIRY\_TERMINATED หรือ INQUIRY\_ERROR

### 2.3.2 Interface L2CAPConnection

#### public interface L2CAPConnection

อินเทอร์เฟซ L2CAPConnection เป็นตัวแทนของช่องสัญญาณ L2CAP แบบเชื่อมโยงตลอดเวลา อินเทอร์เฟซนี้เคยเป็นส่วนหนึ่งของ CLDC Generic Connection Framework

เพื่อที่จะสร้างลูกข่ายการติดต่อ โปรโตคอลจะเป็น btL2cap เป้าหมายก็คือเป็นการรวมแอดเดรสของอุปกรณ์บลูทูธเพื่อที่จะติดต่อไปยัง Protocol Service Multiplexor (PSM) ของบริการค่าของ PSM นั้นใช้โดย L2CAP เพื่อที่จะกำหนดว่าโปรโตคอลระดับที่สูงขึ้นหรือแอปพลิเคชันจะเป็นผู้ที่ได้รับข้อความที่เลเซอร์ได้รับมา

ตัวแปรที่ได้ถูกกำหนดสำหรับ L2CAP นั้นเป็น ReceiveMTU (Maximum Transmission Unit (MTU)) และ TransmitMTU ทั้ง ReceiveMTU และ TransmitMTU เป็นตัวเลือก ReceiveMTU นั้นชี้ให้เห็นถึงขนาดความจุสูงสุดที่การเชื่อมต่อจะรับได้ และสำหรับ TransmitMTU ชี้ให้เห็นถึงขนาดความจุสูงสุดที่การเชื่อมต่อจะส่งไปได้

ตารางที่ 2.5 รายละเอียดในส่วนของฟิลด์ ของอินเทอร์เฟซ L2CAPConnection

Field Summary	
static int	DEFAULT_MTU ค่า DEFAULT_MTU สำหรับช่องสัญญาณที่เชื่อมโยงแบบตลอดเวลาแบบ 672 ไบต์
static int	MINIMUM_MTU ค่า MINIMUM_MTU สำหรับช่องสัญญาณที่เชื่อมโยงแบบตลอดเวลาแบบ 48 ไบต์

ตารางที่ 2.6 รายละเอียดในส่วนของเมธอด ของอินเทอร์เฟซ L2CAPConnection

Method Summary	
int	getReceiveMTU() คืนค่า ReceiveMTU ที่สนับสนุนการเชื่อมต่อ
int	getTransmitMTU() คืนค่า MTU ที่สนับสนุนอุปกรณ์ควบคุม
boolean	ready() กำหนดหากมีแพ็กเก็ตที่สามารถอ่านได้ระหว่างการเรียกไปยัง receive()
int	receive(byte[] inBuf) อ่านแพ็กเก็ตเกิดของข้อมูล
void	send(byte[] data) ร้องขอข้อมูลที่ถูกส่งไปยังอุปกรณ์ควบคุม

### 2.3.2.1 รายละเอียดในส่วนของ Field

#### - DEFAULT\_MTU

Static final int DEFAULT\_MTU

ค่า DEFAULT\_MTU สำหรับช่องสัญญาณที่เชื่อมโยงแบบตลอดเวลาแบบ 672 ไบต์

ค่าของ DEFAULT\_MTU มีค่าเท่ากับ 0x02A0 (672)

#### - MINIMUN\_MTU

static final int MINIMUN\_MTU

ค่า MINIMUN\_MTU สำหรับช่องสัญญาณที่เชื่อมโยงแบบตลอดเวลาแบบ 48 ไบต์

ค่าของ MINIMUN\_MTU มีค่าเท่ากับ 0x30 (48)

### 2.3.2.2 Method Detail

#### - `getTransmitMTU`

`int getTransmitMTU()` throws `java.io.IOException`

จะคืนค่า MTU ที่อุปกรณ์ควบคุมสนับสนุน ค่านี้จะได้รับหลังจากการเชื่อมต่อได้ถูกกำหนด หากแอปพลิเคชันได้กำหนด `TransmitMTU` ใน `Connector.open()` string ดังนั้นค่านี้ก็ควรจะมีค่าเท่ากับ หากแอปพลิเคชันไม่ได้กำหนดค่า `TransmitMTU` ใดๆเลย ดังนั้นค่านี้ก็ควรจะมีค่าน้อยกว่าหรือเท่ากับ `ReceiveMTU` โดยอุปกรณ์ควบคุมจะแจ้งไว้ระหว่างการโครงสร้างของช่องสัญญาณ

#### Returns

ค่าสูงสุดของจำนวนไบต์ที่สามารถส่งการเรียก `send()` โดยปราศจากการสูญเสียข้อมูลใดๆเลย

#### Throws

`java.io.IOException` หากการเชื่อมต่อได้ถูกปิดลง

#### - `getReceiveMTU`

`Int getReceiveMTU()` throws `java.io.IOException`

จะคืนค่า `ReceiveMTU` ที่สนับสนุนการเชื่อมต่อ หากสตรีมการเชื่อมต่อไม่ได้กำหนด `ReceiveMTU` ค่าที่ได้กลับมามีค่าน้อยกว่าหรือเท่ากับ `DEFAULT_MTU` เช่นกันนั้นหากการสตรีมเชื่อมต่อไม่ได้กำหนด `MTU` ค่านี้จะมีค่าน้อยกว่าหรือเท่ากับค่าที่ได้ถูกกำหนดในสตรีมการเชื่อมต่อ

#### Returns

ค่าสูงสุดของจำนวนไบต์ที่สามารถอ่านได้ในการเรียกไปยัง `receive()`

#### Throws

`java.io.IOException` หากการเชื่อมต่อได้ถูกปิดลง

#### - `send`

`void send (byte[] data)` throws `java.io.IOException`

จะทำการร้องขอข้อมูลที่ถูกส่งไปยังอุปกรณ์ควบคุม `TransmitMTU` กำหนดจำนวนข้อมูลที่ สามารถส่งได้อย่างสมบูรณ์ในการทำการส่งในครั้งเดียว หากขนาดของ `data` มีค่ามากกว่า เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TransmitMTU ดังนั้นเพียง TransmitMTU ไบต์แรกของแพ็กเก็ตจะถูกส่ง และที่เหลือก็จะถูกทิ้งไป ถ้า data มีความยาวเท่ากับ 0 แพ็กเก็ตของ L2CAP ที่ว่างนั้นจะเป็นตัวถูกส่งไป

### ตัวแปร

**data** ข้อมูลที่ถูกส่ง

### Throws

java.io.IOException หาก data ไม่สามารถส่งได้อย่างสมบูรณ์หรือปิดการเชื่อมต่อ

java.lang.NullPointerException หาก data นั้นว่างเปล่า

### - receive

int receive (byte[] inBuf) throws java.io.IOException

ทำการอ่านแพ็กเก็ตของข้อมูล จำนวนของข้อมูลที่ได้รับในการดำเนินการนี้ มีความสัมพันธ์กับค่าของ ReceiveMTU หากขนาดของ inBuf มีค่ามากกว่าหรือเท่ากับ ReceiveMTU ดังนั้นไม่มีข้อมูลใดเลยที่จะสูญหาย ต่างจาก read() ใน java.io.InputStream หากขนาดของ inbuf อนมีค่าน้อยกว่า ReceiveMTU ดังนั้นสัดส่วนของความจุ L2CAP นั้นจะพอดีกับ inBuf ส่วนที่เกินมาก็จะถูกทิ้งไป หากแอปพลิเคชันได้มีการจำกัดจำนวนไบต์ (ให้มีค่าน้อยกว่า ReceiveMTU) มันจะรับการถ่ายโอนใดๆ ก็ตาม ดังนั้นขนาดของ inBuf สามารถที่จะมีค่าน้อยกว่า ReceiveMTU และจะ ไม่มีการสูญเสียข้อมูล ถ้าความยาวของ inBuf มีค่าเท่ากับ 0 ข้อมูลทั้งหมดที่ถูกส่งมาในแพ็กเก็ตเดียวจะสูญหายเว้นแต่ความยาวของแพ็กเก็ตมีค่าเท่ากับ 0

### ตัวแปร

**inBuf** เป็นอะไรที่ใช้เก็บข้อมูลที่ได้รับมา

### Returns

จำนวนไบต์ที่อ่านได้จริง ; จะเป็น 0 หากความยาวแพ็กเก็ตที่ได้รับมาเป็น 0 หรือ inBuf ความยาวเท่ากับ 0

### Throws

java.io.IOException หาก I/O เกิดข้อผิดพลาดขึ้นหรือปิดการเชื่อมต่อ

java.io.InterruptedIOException หากการร้องขอเกิด Timed out

java.lang.NullPointerException หาก inBuf ว่างเปล่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - ready()

boolean **ready()** throws java.io.IOException

จะกำหนดหากมีแพ็กเก็ตที่สามารถอ่านได้ระหว่างการเรียกไปยัง receive() หากมีค่าเป็น true การเรียกไปยัง receive() จะไม่เกิดแอพลิเคชัน

### Returns

true หากมีข้อมูลให้อ่านได้

false หากไม่มีข้อมูลให้อ่าน

### Throws

Java.io.IOException หากปิดการเชื่อมต่อ

### 2.3.3 Interface L2CAPConnectionNotifier

**public interface L2CAPConnectionNotifier**

เพื่อที่จะสร้างแม่ข่ายการเชื่อมต่อ โปรโตคอลควรเป็น bd2cap เป้าหมายประกอบด้วย "localhost:" และ UUID ของการบริการ ตัวแปรนั้นจะเป็น ReceiveMTU และ TransmitMTU ตัวแปรเดียวกันกับลูกข่ายการเชื่อมต่อ

การเรียกไปยัง Connector.open() ด้วยสตริงนี้จะคืนค่าวัตถุประสงค์เป็น javax.bluetooth.L2CAPConnectionNotifier วัตถุประสงค์ของตัว L2CAPConnection นั้นได้รับมาจาก L2CAPConnectionNotifier โดยการเรียกเมธอด acceptAndOpen()

### ตารางที่ 2.7 รายละเอียดในส่วนของเมธอด ของอินเตอร์เฟส L2CAPConnectionNotifier

Method Summary	
L2CAPConnection	acceptAndOpen() รอลูกข่ายเพื่อจะติดต่อไปยังบริการของ L2CAP

#### 2.3.3.1 รายละเอียดในส่วนของ Method

##### - acceptAndOpen

L2CAPConnection **acceptAndOpen()** throws java.io.IOException

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รอถูกข่ายเพื่อจะติดต่อไปยังบริการของ L2CAP ในการเชื่อมต่อจะคืนค่า L2CAPConnection นั้นจะสามารถใช้เพื่อติดต่อกับถูกข่ายได้

Service Record มีความเชื่อมโยงกับการเชื่อมต่อนี้จะถูกเพิ่มเข้าไปใน SDDB ซึ่งจะมีความเชื่อมโยงกับวัตถุประสงค์ L2CAPConnectionNotifier นี้ด้วย หากมีอันที่ยังไม่มีใน SDDB เมธอดนี้จะทำการใส่ local device ในโหมคความสามารถในการติดต่อดังนั้นมันอาจสามารถพยายามที่จะติดต่อโดยถูกข่ายทั้งหลายได้

ตามที่ได้มีการตรวจสอบเงื่อนไขต่อไปเพื่อพิสูจน์ว่าการปรับปรุงใดๆที่ทำจากแอปพลิเคชัน ไปยัง record service หลังจากสร้างโดย Connector.open() ไม่ได้สร้าง service record ที่ใช้ไม่ได้ หากข้อใดเกิดข้อผิดพลาดใดๆ ขึ้นมา ServiceRegistrationException จะถูก throw

- ServiceClassIDList และ ProtocolDescriptorList เป็นแอทริบิวต์การบริการที่ได้รับคำสั่งสำหรับ service record ของ bt2cap จะต้องถูกแสดงใน service record
- L2CAP จะต้องอยู่ใน ProtocolDescriptorList
- ค่า PSM จะต้องไม่มีการเปลี่ยนแปลงใน Service Record

เมธอดนี้จะไม่ทำให้แน่ใจได้ว่า Service record ที่ถูกสร้างขึ้นนี้เป็น Service record ที่ถูกต้องสมบูรณ์ มันเป็นความรับผิดชอบของแอปพลิเคชันที่จะทำให้แน่ใจได้ว่า Service record ได้เป็นไปตามการปรับไปใช้ได้ของกฎและทฤษฎีต่างๆ สำหรับความถูกต้องของ Service record

### Returns

การเชื่อมต่อที่สื่อสารกับถูกข่าย

### Throws

java.io.IOException หากการแจ้งได้ถูกปิดก่อนที่ acceptAndOpen() จะถูกเรียก  
ServiceRegistrationException หากโครงสร้างนี้ที่ถูกเชื่อมโยงกับ Service Record นั้นใช้ไม่ได้หรือหาก Service record นั้นไม่สามารถเพิ่มใน SDDB ได้อย่างสมบูรณ์ โครงสร้างของ service record นี้จะใช้ไม่ได้หาก service record นี้มีแอทริบิวต์ที่มีคำสั่งใดๆ หายไป หรือได้เปลี่ยนค่าใดๆ ก็ตามของค่าที่ได้อธิบายไว้ข้างต้นซึ่งค่าเหล่านั้นจะถูกกำหนดไว้แล้วและไม่สามารถที่จะเปลี่ยนได้ ความล้มเหลวต่างๆที่เพิ่มเข้าไปใน SDDB นั้นสามารถที่จะเพิ่มไปได้เนื่องจากพื้นที่ดิสก์ที่ไม่เพียงพอ  
ฐานข้อมูลถูกล็อก เป็นต้น

BluetoothStateException หากอุปกรณ์แม่ข่ายไม่สามารถที่จะใส่ไว้ในโหมคการเชื่อมต่อเพราะว่าผู้ใช้อุปกรณ์ได้กำหนดให้อุปกรณ์เป็นโหมคที่ไม่สามารถเชื่อมต่อได้

### 2.3.4 Interface ServiceRecord

#### public interface ServiceRecord

อินเตอร์เฟซ ServiceRecord ได้อธิบายคุณลักษณะของบริการบลูทูธ ServiceRecord ประกอบด้วย เซตของแอทริบิวต์การบริการ โดยที่แต่ละการบริการจะเป็นคู่กัน (ID, และค่าต่างๆ) และ ID ของแอทริบิวต์บลูทูธเป็น 16 บิต และค่าแอทริบิวต์เป็น DataElement

ServiceRecords ได้ถูกทำขึ้นสำหรับแอปพลิเคชันลูกข่าย ผ่านข้อเท็จจริงของเมธอด servicesDiscovered ของอินเตอร์เฟซ DiscoveryListener ServiceRecords นั้นสำหรับแอปพลิเคชันแม่ข่ายผ่านเมธอด getRecord() บน LocalDevice

เอกสารการกำหนดหมายเลขของบลูทูธได้กำหนดการกำหนดหมายเลขของ ID แอทริบิวต์ของการให้บริการ นี่เป็นตัวอย่าง IDs ที่ถูกใช้บ่อยและชนิดของมัน

ตารางที่ 2.8 หมายเลขของ ID แอทริบิวต์ของการให้บริการ

Attribute Name	Attribute ID	Attribute Value Type
ServiceRecordHandle	0x0000	32 บิต ชนิด unsigned integer
ServiceClassIDList	0x0001	DATSEQ ของ UUIDs
ServiceRecordState	0x0002	32 บิต ชนิด unsigned integer
ServiceID	0x0003	UUID
ProtocolDescriptorList	0x0004	DATSEQ ของ DATSEQ ของ UUID และเป็นตัวแปรเลือก
BrowseGroupList	0x0005	DATSEQ ของ UUIDs
LanguageBasedAttributeIDList	0x0006	DATSEQ ของ DATSEQ จำนวนสามส่วน
ServiceInfoTimeToLive	0x0007	32 บิต ชนิด unsigned integer
ServiceAvailability	0x0008	8 บิต ชนิด unsigned integer
BluetoothProfileDescriptorList	0x0009	DATSEQ ของ DATSEQ จำนวนคู่
DocumentationURL	0x000A	URL
ClientExecutableURL	0x000B	URL
IconURL	0x000C	URL
VersionNumberList	0x0200	DATSEQ ของ 16 บิต ชนิด unsigned integer
ServiceDatabaseState	0x0201	32 บิต ชนิด unsigned integer

ตารางดังแสดงต่อไปนี้เป็นค่าสตริงชดเชยของแอทริบิวต์ IDs ที่ใช้บ่อยใน ServiceRecord ค่าชดเชยเหล่านี้จะต้องถูกเพิ่มเข้าไปในค่าพื้นฐานเพื่อรับค่า Service ID จริง

ตารางที่ 2.9 ค่าสตริงชดเชยของแอทริบิวต์ IDs ที่ใช้บ่อยใน ServiceRecords

Attribute Name	Attribute ID Offset	Attribute Value Type
ServiceName	0x0000	สตริง
ServiceDescription	0x0001	สตริง
ProviderName	0x0002	สตริง

ตารางที่ 2.10 รายละเอียดในส่วนของฟิลด์ ของอินเตอร์เฟส ServiceRecord

Field Summary	
static int	AUTHENTICATE_ENCRYPT การกำหนดสิทธิ์และการเข้ารหัสจะถูกร้องขอสำหรับการติดต่อในการบริการนี้
static int	AUTHENTICATE_NOENCRYPT การกำหนดสิทธิ์นี้จะถูกร้องขอสำหรับการติดต่อในบริการนี้ แต่ไม่ต้องเข้ารหัส
static int	NONAUTHENTICATE_NOENCRYPT การกำหนดสิทธิ์และการเข้ารหัสนั้นไม่จำเป็นสำหรับการติดต่อในการบริการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.11 รายละเอียดในส่วนของเมธอด ของอินเทอร์เฟซ ServiceRecord

Method Summary	
int[]	getAttributeIDs() คืนค่า IDs แอทริบิวต์การให้บริการซึ่งค่าเหล่านั้นสามารถที่จะได้กลับมาโดยการเรียกไปยัง getAttributeValue()
DataElement	getAttributeValue(int attrID) คืนค่าของ ID แอทริบิวต์การให้บริการแสดงให้เห็นว่ามียู่ใน service record นอกจากนั้นเมธอดนี้จะคืนค่ากลับมาเป็น null
java.lang.String	getConnectionURL(int requiredSecurity, Boolean mustBeMaster) คืนค่าสตริงรวมถึงตัวแปรเลือกที่สามารถใช้ได้โดยถูกข้ายเพื่อติดต่อไปยังการให้บริการ อธิบายได้โดย ServiceRecord
RemoteDevice	getHostDevice() คืนค่าการควบคุมอุปกรณ์บลูทูธที่มีอยู่ใน service record กับค่าแอทริบิวต์
boolean	populateRecord(int[] attrIDs) รับค่ากลับมา โดยการติดต่อกับการควบคุมอุปกรณ์บลูทูธสำหรับจำนวนเซตของ IDs แอทริบิวต์ของการให้บริการที่สามารถใช้งานได้ใอุปกรณ์บลูทูธ
boolean	setAttributeValue(int attrID, DataElement attrValue) เป็นส่วนขยาย ServiceRecord นี้เพื่อที่จะจำกัดแอทริบิวต์การให้บริการกำหนดโดยค่าแอทริบิวต์เป็นคู่ๆ (attrID, attrValue)
void	setDeviceServiceClasses(int classes) ใช้โดยแอปพลิเคชันแม่ข่ายเพื่อที่จะแสดงบิตคลาสการให้บริการหลัก ที่ควรจะใช้ได้ใน DeviceClass ของแม่ข่ายเมื่อ ServiceRecord ถูกเพิ่มเข้าไปใน SDDB

#### 2.3.4.1 รายละเอียดในส่วนของ Field

##### - NONAUTHENTICATE\_NOENCRYPT

static final int NONAUTHENTICATE\_NOENCRYPT

การกำหนดสิทธิ์และการเข้ารหัสนั้นไม่จำเป็นสำหรับการติดต่อในการบริการนี้ ใช้ร่วมกับ

เมธอด getConnectionURL()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*NONAUTHENTICATE\_NOENCRYPT* จะถูกเซตเป็นค่าคงที่มีค่าเป็น 0x00 (0)

#### - AUTHENTICATE\_NOENCRYPT

static final int AUTHENTICATE\_NOENCRYPT

การกำหนดสิทธิ์จะถูกร้องขอสำหรับการติดต่อในบริการนี้ แต่ไม่ต้องเข้ารหัส ใช้ร่วมกับ  
เมธอด getConnectionURL()

*AUTHENTICATE\_NOENCRYPT* จะถูกเซตเป็นค่าคงที่มีค่าเป็น 0x01 (1)

#### - AUTHENTICATE\_ENCRYPT

static final int AUTHENTICATE\_ENCRYPT

การกำหนดสิทธิ์และการเข้ารหัสจะถูกร้องขอสำหรับการติดต่อในการบริการนี้ ใช้ร่วมกับ  
เมธอด getConnectionURL()

*AUTHENTICATE\_ENCRYPT* จะถูกเซตเป็นค่าคงที่มีค่าเป็น 0x02 (2)

### 2.3.4.2 รายละเอียดในส่วนของ Method

#### - getAttributeValue

DataElement getAttributeValue(int attrID)

คืนค่าของ ID แอททริบิวท์การให้บริการแสดงให้เห็นว่ามีอยู่ใน service record นอกจากนั้น  
เมธอดนี้จะคืนค่ากลับมาเป็น null

#### ตัวแปร

**attrID** เป็นแอททริบิวท์ซึ่งเป็นค่าที่ถูกคืนกลับมา

#### Returns

ค่าของแอททริบิวท์ ID ถ้าปรากฏใน service Record นอกเหนือจากนั้นจะเป็น null

#### Throws

Java.lang.IllegalArgumentException หาก attrID เป็นค่าลบหรือมากกว่าหรือเท่ากับ  $2^{16}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - `getHostDevice`

`RemoteDevice` `getHostDevice()`

คืนค่าการควบคุมอุปกรณ์บลูทูธที่มีอยู่ใน service record กับค่าแอทริบิวต์ สำคัญมากที่จะสังเกตไว้ว่าอุปกรณ์บลูทูธจะแสดงให้เห็นว่าค่านั้นอาจไม่สามารถเข้าถึงได้ต่อไป ตั้งแต่อุปกรณ์เหล่านั้นเคลื่อนที่ได้ ปิดได้ หรือเปลี่ยนโหมดการรักษาความปลอดภัยของมันเอง จะปฏิเสธการดำเนินการต่างๆ

### Returns

คืนค่าการควบคุมอุปกรณ์บลูทูธที่มีอยู่ใน service record หรือค่า null หากมี local device อยู่ใน ServiceRecord นี้

### - `getAttributeIDs`

`int[]` `getAttributeIDs()`

คืนค่า IDs แอทริบิวต์การให้บริการซึ่งค่าเหล่านั้นสามารถที่จะได้กลับมาโดยการเรียกไปยัง `getAttributeValue()` รายชื่อของแอทริบิวต์ที่ถูกคืนค่าไม่เป็นจัดเป็นชนิดและรวมไปถึงแอทริบิวต์ที่เป็นตัวเลือกอัตโนมัติ

### Returns

จะคืนค่าอาร์เรย์ของ IDs แอทริบิวต์ของการให้บริการที่อยู่ในเป้าหมายนี้ และมีค่าสำหรับทั้งหมด หากไม่มีแอทริบิวต์ IDs ที่มีค่า เมธอดนี้จะคืนค่าอาร์เรย์ของความยาวศูนย์

### - `populateRecord`

`boolean` `populateRecord` (`int[]` `attrIDs`) throws `java.io.IOException`

รับค่ากลับมาโดยการติดต่อกับการควบคุมอุปกรณ์บลูทูธสำหรับจำนวนเซตของ IDs แอทริบิวต์ของการให้บริการที่สามารถใช้งานได้ใอุปกรณ์บลูทูธ

### ตัวแปร

**`attrIDs`** เป็นรายชื่อของ IDs แอทริบิวต์ของการให้บริการซึ่งค่าจะถูกตรวจสอบ จำนวนของแอทริบิวต์ไม่สามารถขยายพื้นที่ของ `Bluetooth.sd.attr.retrieveable.max` ได้ แอทริบิวต์ในการร้องขอนั้น ต้องมีค่าถูกต้องตามกฎหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Returns**

true หากการร้องขอสำเร็จในการตรวจสอบค่าแอททริบิวต์ไม่ว่าจะเป็นบางส่วนหรือทั้งหมด

false หากไม่สำเร็จในการตรวจสอบไม่ว่าค่าใดก็ตาม

**Throws**

java.io.IOException หาก local device ไม่สามารถติดต่อไปยังอุปกรณ์ควบคุมบลูทูธที่เป็นแหล่งข้อมูลของ service record นี้, หาก Service record ได้ถูกลบไปจาก SDDB ของอุปกรณ์ควบคุม java.lang.IllegalArgumentException, หากขนาดของ attrIDs ขยายระบบที่จำกัดตั้งได้กำหนดโดย Bluetooth.sd.attr.retrieveable.max, หากความยาวแอททริบิวต์ attrIDs เป็น 0 ,หากค่าใดๆ ก็ตามไม่อยู่ในช่วงของ  $[0, 2^{16}-1]$ , หาก attrIDs มีค่าเป็น null

java.lang.NullPointerException หาก attrIDs มีค่าเป็น null

java.lang.RuntimeException หาก ServiceRecord นี้อธิบายการให้บริการใน local device มากกว่าให้บริการบน Remote device

**- getConnectionURL**

java.lang.String getConnectionURL (int requiredSecurity, Boolean mustBeMaster)

คืนค่าสตริงรวมถึงตัวแปรเลือกที่สามารถใช้ได้โดยลูกข่ายเพื่อติดต่อไปยังการให้บริการอธิบายได้โดย ServiceRecord การคืนค่าสามารถใช้เป็นอาร์กิวเมนต์แรกไปยัง Connection.open() หากเมธอดนั้นถูกเรียกบน ServiceRecord จะถูกคืนค่าจาก LocalDevice.getRecord() มันจะคืนสตริงการเชื่อมต่อที่ Remote Device จะใช้เพื่อติดต่อไปยังการให้บริการ

**ตัวแปร**

**requiredSecurity** ใช้ตรวจสอบไม่ว่าจะกำหนดสิทธิ์หรือเข้ารหัสได้ถูกขอสำหรับการเชื่อมต่อ

**mustBeMaster** เป็น true แสดงว่าอุปกรณ์นี้ต้องเป็นมาสเตอร์ในการเชื่อมต่อไปยังการให้บริการนี้ และเป็น false แสดงว่า local device ต้องการที่จะเป็นทั้งมาสเตอร์และสเลฟ

**Returns**

สตริงที่สามารถใช้เพื่อติดต่อไปยังการให้บริการ หรือ null หาก ProtocolDescriptorList ใน ServiceRecord นี้ไม่ถูกทำเป็นรูปแบบเนื่องจากข้อจำกัดของบลูทูธ

**Throws**

java.lang.IllegalArgumentException หาก requiredSecurity ไม่ใช่หนึ่งในค่าคงที่  
NONAUTHENTICATE\_NOENCRYPT, AUTHENTICATE\_NOENCRYPT หรือ  
AUTHENTICATE\_ENCRYPT

**- setDeviceServiceClasses**

void setDeviceServiceClasses(int classes)

ใช้โดยแอปพลิเคชันแม่ข่ายเพื่อที่จะแสดงบิตคลาสการให้บริการหลัก ที่ควรจะใช้ได้ใน DeviceClass ของแม่ข่ายเมื่อ ServiceRecord ถูกเพิ่มเข้าไปใน SDDB เมื่ออุปกรณ์ถูกข่ายได้ทำการค้นหาอุปกรณ์ DeviceClass ของแม่ข่ายได้แสดงดังเป็นหนึ่งในอาร์กิวเมนต์ของเมธอด deviceDiscovered ของอินเทอร์เฟซ DiscoveryListener อุปกรณ์ถูกข่ายสามารถค้นหาข้อมูล DeviceClass ของอุปกรณ์แม่ข่ายเพื่อที่จะได้รับไอเดียทั่วไปของอุปกรณ์ชนิดนี้ และคลาสการให้บริการหลักที่มันเสนอให้ แอปพลิเคชันแม่ข่ายควรใช้เมธอด setDeviceServiceClasses เพื่อที่จะอธิบายการให้บริการของมันเทอมของคลาสการให้บริการหลัก ด้วยการนี้ได้อนุญาตให้ถูกข่ายได้รับ DeviceClass สำหรับแม่ข่ายที่อธิบายทั้งหมดของการให้บริการได้ถูกเสนออย่างถูกต้อง

**ตัวแปร**

classes เป็นจำนวนเต็มซึ่ง ใบนารีเป็นตัวแทนแสดงถึงบิตของคลาสการให้บริการหลักที่ควรจะมีปฏิกิริยาตอบสนอง

**Throws**

Java.lang.IllegalArgumentException หาก classes นั้นไม่เป็นหนึ่งในหรือมากกว่านั้นของคลาสการให้บริการหลัก จำนวนเต็มในเอกสารการกำหนดหมายเลขบิตทูลุข ขณะที่โหมด Limited Discoverable นั้นถูกรวมอยู่ในรายชื่อของคลาสการให้บริการหลัก บิตของตัวเองได้ตอบสนองโดยการวางอุปกรณ์ในโหมด Limited Discoverable ดังนั้น หากบิตที่ 13 นั้นถูกเซต exception ก็จะถูก throws

java.lang.RuntimeException หาก ServiceRecord ได้รับข้อความ ซึ่งได้รับมาจาก remote device

**- setAttributeValue**

Boolean setAttributeValue(int attrID, DataElement attrValue)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นส่วนขยาย ServiceRecord นี้เพื่อที่จะจำกัดแอตทริบิวต์การให้บริการกำหนดโดยค่าแอตทริบิวต์เป็นคู่ๆ (attrID, attrValue) หาก attrID ไม่มีอยู่ใน ServiceRecord ค่าแอตทริบิวต์คู่นี้จะถูกเพิ่มเข้าไปใน ServiceRecord ถ้าหากมีอยู่ใน ServiceRecord แล้ว ค่าของแอตทริบิวต์ก็จะไปเปลี่ยน attrValue ถ้า attrValue เป็น null แอตทริบิวต์กับ attribute ID ของ attrID จะถูกย้ายออกไปจาก ServiceRecord หาก attrValue เป็น null และไม่มีค่า attrID เมธอดนี้จะทำการคืนค่าเป็น false

#### ตัวแปร

**attrID** เป็น ID ของแอตทริบิวต์การให้บริการ

**attrValue** เป็น DataElement ซึ่งเป็นค่าของแอตทริบิวต์การให้บริการ

#### Returns

true หากแอตทริบิวต์การให้บริการนั้นเพิ่ม ย้าย หรือปรับเปลี่ยนได้อย่างสมบูรณ์

false หาก attrValue เป็น null และ attrID ไม่มีอยู่ในจุดประสงค์นี้

#### Throws

java.lang.IllegalArgumentException หาก attrID ไม่ได้เป็นแสดง 16-bit ชนิด unsigned integer และหาก attrID ไม่ใช่ค่าของ ServiceRecordHandle (0x0000)

java.lang.RuntimeException หากเมธอดนี้ถูกเรียกบน ServiceRecord ที่ถูกสร้างโดยการเรียกไปยัง DiscoveryAgent.searchServices()

#### 2.3.5 Class DataElement

```
public class DataElement extends java.lang.Object
```

ตารางที่ 2.12 รายละเอียดในส่วนของฟิลด์ ของอินเตอร์เฟซ DataElement

Field Summary	
static int	BOOL กำหนดชนิดของข้อมูลเป็น BOOL
static int	DATALT กำหนดชนิดของข้อมูลเป็น DATALT
static int	DATSEQ กำหนดชนิดของข้อมูลเป็น DATSEQ
static int	INT_1 กำหนดจำนวนเต็มชนิด signed ของขนาด 1 ไบต์
static int	INT_16 กำหนดจำนวนเต็มชนิด signed ของขนาด 16 ไบต์
static int	INT_2 กำหนดจำนวนเต็มชนิด signed ของขนาด 2 ไบต์
static int	INT_4 กำหนดจำนวนเต็มชนิด signed ของขนาด 4 ไบต์
static int	INT_8 กำหนดจำนวนเต็มชนิด signed ของขนาด 8 ไบต์
static int	NULL กำหนดชนิดของข้อมูลเป็น null
static int	STRING กำหนดชนิดของข้อมูลเป็น String
static int	U_INT_1 กำหนดจำนวนเต็มชนิด unsigned ของขนาด 1 ไบต์
static int	U_INT_16 กำหนดจำนวนเต็มชนิด unsigned ของขนาด 16 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.12 รายละเอียดในส่วนของฟิลด์ ของอินเทอร์เฟซ DataElement (ต่อ)

Field Summary	
static int	U_INT_2 กำหนดจำนวนเต็มชนิด unsigned ของขนาด 2 ไบต์
static int	U_INT_4 กำหนดจำนวนเต็มชนิด unsigned ของขนาด 4 ไบต์
static int	U_INT_8 กำหนดจำนวนเต็มชนิด unsigned ของขนาด 8 ไบต์
static int	URL กำหนดชนิดของข้อมูลเป็น URL
static int	UUID กำหนดชนิดข้อมูลเป็น UUID

ตารางที่ 2.13 รายละเอียดในส่วนของคอนสตรัคเตอร์ DataElement

Constructor Summary	
DataElement (boolean bool)	สร้าง DataElement ซึ่งชนิดข้อมูลเป็น Bool ซึ่งมีค่าเท่ากับ bool
DataElement (int valueType)	สร้าง DataElement ของชนิด NULL, DATALT หรือ DATSEQ
DataElement (int valueType, longValue)	สร้าง DataElement ซึ่งครอบคลุมค่าของขนาดจำนวนเต็ม U_INT_1, U_INT_2, U_INT_4, INT_1, INT_2, INT_4, และ INT_8.
DataElement(int valueType, java.lang.Object value)	สร้าง DataElement ว่างชนิดข้อมูลถูกให้โดย valueType และค่าซึ่งกำหนดเฉพาะโดยอาร์กิวเมนต์ value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.14 รายละเอียดในส่วนของเมธอด DataElement

Method Summary	
void	addElement(DataElement elem) เพิ่ม DataElement ไปยัง DATALT หรือ DATSEQ ของ DataElement object
boolean	getBoolean() คืนค่าของ DataElement หากใช้แทน Boolean ได้
int	getDataType() คืนชนิดของข้อมูลของออบเจกต์ที่ใช้แทน DataElement นี้
long	getLong() คืนชนิดของข้อมูลของ DataElement หากมันสามารถใช้แทน long ได้
int	getSize() คืนค่าจำนวนของ DataElement ซึ่งแสดงในออบเจกต์ของ DATALT หรือ DATSEQ นี้
java.lang.object	getValue() คืนค่าของ DataElement เหมือนเป็นออบเจกต์
void	insertElementAt(DataElement elem, int index) เพิ่ม DataElement ในที่ที่กำหนด
boolean	removeElement(DataElement elem) ย้าย DataElement ที่ปรากฏขึ้นเป็นตัวแรกจากออบเจกต์นี้

### 2.3.5.1 รายละเอียดในส่วนของ Field

- NULL

public static final int NULL

กำหนดชนิดของข้อมูลเป็น NULL ค่าของชนิดข้อมูล DataElement.NULL

ค่าของ NULL มีค่าเท่ากับ 0x00 (0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- U\_INT\_1**

public static final int U\_INT\_1

กำหนดจำนวนเต็มชนิด unsigned ของขนาด 1 ไบต์

ค่าของค่าคงที่ U\_INT\_1 มีค่าเท่ากับ 0x08 (8)

**- U\_INT\_2**

public static final int U\_INT\_2

กำหนดจำนวนเต็มชนิด unsigned ของขนาด 2 ไบต์

ค่าของค่าคงที่ U\_INT\_2 มีค่าเท่ากับ 0x09 (9)

**- U\_INT\_4**

public static final int U\_INT\_4

กำหนดจำนวนเต็มชนิด unsigned ของขนาด 4 ไบต์

ค่าของค่าคงที่ U\_INT\_4 มีค่าเท่ากับ 0x0A(10)

**- U\_INT\_8**

public static final int U\_INT\_8

กำหนดจำนวนเต็มชนิด unsigned ของขนาด 8 ไบต์

ค่าของค่าคงที่ U\_INT\_8 มีค่าเท่ากับ 0x0B (11)

**- U\_INT\_16**

public static final int U\_INT\_16

กำหนดจำนวนเต็มชนิด unsigned ของขนาด 16 ไบต์

ค่าของค่าคงที่ U\_INT\_16 มีค่าเท่ากับ 0x0C (12)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- INT\_1****public static final int INT\_1**

กำหนดจำนวนเต็มชนิด signed ของขนาด 1 ไบต์

ค่าของค่าคงที่ *INT\_1* มีค่าเท่ากับ *0x10 (16)***- INT\_2****public static final int INT\_2**

กำหนดจำนวนเต็มชนิด signed ของขนาด 2 ไบต์

ค่าของค่าคงที่ *INT\_2* มีค่าเท่ากับ *0x11 (17)***- INT\_4****public static final int INT\_4**

กำหนดจำนวนเต็มชนิด signed ของขนาด 4 ไบต์

ค่าของค่าคงที่ *INT\_4* มีค่าเท่ากับ *0x12 (18)***- INT\_8****public static final int INT\_8**

กำหนดจำนวนเต็มชนิด signed ของขนาด 8 ไบต์

ค่าของค่าคงที่ *INT\_8* มีค่าเท่ากับ *0x13 (19)***- INT\_16****public static final int INT\_16**

กำหนดจำนวนเต็มชนิด signed ของขนาด 16 ไบต์

ค่าของค่าคงที่ *INT\_16* มีค่าเท่ากับ *0x14 (20)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- URL**

public static final int URL

กำหนดชนิดข้อมูลเป็น URL

ค่าของค่าคงที่ *URL* มีค่าเท่ากับ *0x40* (64)

**-UUID**

public static final int UUID

กำหนดชนิดข้อมูลเป็น UUID

ค่าของค่าคงที่ *UUID* มีค่าเท่ากับ *0x18* (24)

**-BOOL**

public static final int BOOL

กำหนดชนิดข้อมูลเป็น BOOL

ค่าของค่าคงที่ *BOOL* มีค่าเท่ากับ *0x28* (40)

**-STRING**

public static final int STRING

กำหนดชนิดข้อมูลเป็น STRING

ค่าของค่าคงที่ *STRING* มีค่าเท่ากับ *0x20* (32)

**-DATSEQ**

public static final int DATSEQ

กำหนดชนิดข้อมูลเป็น DATSEQ ค่าแอทริบิวต์ของการให้บริการซึ่งข้อมูลที่เป็นชนิดนี้  
ต้องพิจารณาองค์ประกอบทั้งหมดในลิสต์

ค่าของค่าคงที่ *DATSEQ* มีค่าเท่ากับ *0x30* (48)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**-DATALT**

public static final int **DATALT**

กำหนดชนิดข้อมูลเป็น DATALT ค่าแอมทริบิวท์ของการให้บริการซึ่งข้อมูลที่เป็นชนิดนี้ ต้องพิจารณาเพียงแค่องค์ประกอบเดียวที่มีอยู่ในกลุ่มเท่านั้น  
ค่าของค่าคงที่ *DATALT* มีค่าเท่ากับ 0x38 (56)

**2.3.5.2 รายละเอียดในส่วนของ Construtor****-DataElement**

public **DataElement**(int valueType)

สร้าง DataElement ของชนิด NULL, DATALT หรือ DATSEQ

**ตัวแปร**

**valueType** เป็นชนิดของ DataElement ที่จะสร้าง: NULL, DATALT หรือ DATSEQ

**Throws**

java.lang.IllegalArgumentException หาก valueType ไม่มีค่าเป็น NULL, DATALT, DATSEQ

**-DataElement**

public **DataElement**(boolean bool)

สร้าง DataElement ซึ่งชนิดข้อมูลเป็น Bool ซึ่งมีค่าเท่ากับ bool

**ตัวแปร**

**bool** เป็นค่าของ DataElement ของชนิด BOOL

**-DataElement**

public **DataElement**(int valueType, long value)

สร้าง DataElement ซึ่งครอบคลุมค่าของขนาดจำนวนเต็ม U\_INT\_1, U\_INT\_2, U\_INT\_4, INT\_1, INT\_2, INT\_4, และ INT\_8. ค่าตามกฎสำหรับ valueType และค่าแอมทริบิวท์ที่คล้ายกันมีค่าดังต่อไปนี้

ตารางที่ 2.15 ค่าตามกฎสำหรับ valueType และค่าเอทริบิวท์ที่คล้ายกัน

Value Type	Value Range
U_INT_1	$[0, 2^8-1]$
U_INT_2	$[0, 2^{16}-1]$
U_INT_4	$[0, 2^{32}-1]$
INT_1	$[-2^7, 2^7-1]$
INT_2	$[-2^{15}, 2^{15}-1]$
INT_4	$[-2^{31}, 2^{31}-1]$
INT_8	$[-2^{63}, 2^{63}-1]$

#### ตัวแปร

**valueType** เป็นชนิดข้อมูลของออบเจกต์ที่ถูกสร้างขึ้น ต้องเป็นหนึ่งในดังต่อไปนี้ U\_INT\_1, U\_INT\_2, U\_INT\_4, INT\_1, INT\_2, INT\_4, or INT\_8

**value** เป็นค่าของออบเจกต์ที่ถูกสร้างขึ้นมา ต้องอยู่ในช่วงของ value

#### -DataElement

`public DataElement(int valueType, java.lang.Object value)`

สร้าง DataElement ว่างชนิดข้อมูลลูกให้โดย valueType และค่าซึ่งกำหนดเฉพาะโดยอาร์กิวเมนต์ value ค่าตามกฎสำหรับ valueType และค่าเอทริบิวท์ที่คล้ายกันมีค่าดังต่อไปนี้

ตารางที่ 2.16 ค่าตามกฎสำหรับ valueType และค่าแอททริบิวต์ที่คล้ายกัน

Value Type	Java Type/ Value Range
URL	java.lang.String
UUID	javax.bluetooth.UUID
STRING	java.lang.String
INT_16	$[-2^{127}, 2^{127}-1]$ เป็นไบต์อาร์เรย์ซึ่งต้องมีความยาวเท่ากับ 16
U_INT_8	$[0, 2^{64}-1]$ เป็นไบต์อาร์เรย์ซึ่งต้องมีความยาวเท่ากับ 8
U_INT_16	$[0, 2^{128}-1]$ เป็นไบต์อาร์เรย์ซึ่งต้องมีความยาวเท่ากับ 16

#### ตัวแปร

**valueType** เป็นชนิดข้อมูลของออบเจกต์ที่ถูกสร้างขึ้น ต้องเป็นหนึ่งในดังต่อไปนี้ U\_INT\_1, U\_INT\_2, U\_INT\_4, INT\_1, INT\_2, INT\_4, or INT\_8

**value** เป็นค่าของออบเจกต์ที่ถูกสร้างขึ้นมา ต้องอยู่ในช่วงของ value

#### Throws

java.lang.IllegalArgumentException หาก value ไม่เป็นชนิดของ valueType หรือไม่อยู่ในช่วงที่กำหนด หรือมีค่าเป็น null

#### 2.3.5.3 รายละเอียดในส่วนของ Method

##### -addElement

```
public void addElement(DataElement elem)
```

เพิ่ม DataElement ไปยัง DATALT หรือ DATSEQ ของ DataElement object ตัว elem จะถูกเพิ่มเข้าไปเป็นครั้งสุดท้ายในลิสต์ elem สามารถเป็น DataElement ใดๆ ก็ได้

#### ตัวแปร

**elem** เป็น DataElement ออบเจกต์ที่ถูกเพิ่มเข้าไป

#### Throws

java.lang.ClassCastException ถ้าเมธอดถูกเรียกใน DataElement ซึ่งไม่เป็นทั้ง DATALT หรือ DATSEQ

java.lang.NullPointerException ถ้า elem เป็น null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - insertElementAt

```
public void insertElementAt(DataElement elem, int index)
```

เพิ่ม DataElement ในที่ที่กำหนด เมธอดนี้สามารถก่อให้เกิดได้แก่บน DATALT หรือ DATSEQ DataElement เท่านั้น elem สามารถเป็น DataElement ชนิดใดก็ได้ ในส่วนของ index ต้องมากกว่าหรือเท่ากับ 0 หรือน้อยกว่าหรือเท่ากับขนาดของมันเอง ดังนั้น DATALT และ DATSEQ เป็นออบเจกต์ที่เป็นฐานมีค่าเป็น 0

### ตัวแปร

**elem** เป็น DataElement ที่เพิ่มเข้าไป

**index** เป็นที่ที่ DataElement ถูกเพิ่มเข้าไป

### Throws

java.lang.ClassCastException หากเมธอดถูกเรียกในตัวอย่างของ DataElement ซึ่งไม่เป็นทั้ง DATALT หรือ DATSEQ

java.lang.IndexOutOfBoundsException หาก index เป็นค่าลบหรือมีขนาดมากกว่า DATALT หรือ DATSEQ

java.lang.NullPointerException หาก elem มีค่าเป็น null

### - getSize

```
public int getSize()
```

คืนค่าจำนวนของ DataElement ซึ่งแสดงในออบเจกต์ของ DATALT หรือ DATSEQ นี้ มันเป็นไปได้ว่าจำนวนของ elements จะมีค่าเท่ากับ 0

### Returns

จำนวนของ element ใน DATALT หรือ DATSEQ

### Throws

java.lang.ClassCastException หากออบเจกต์ไม่เป็นชนิดของ DATALT หรือ DATSEQ

### - removeElement

```
public boolean removeElement(DataElement elem)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย้าย DataElement ที่ปรากฏขึ้นเป็นตัวแรกจากออบเจกต์นี้ elem นี้จะเป็นชนิดใดก็ได้ เฉพาะออบเจกต์แรกในลิสต์ที่มีค่าเท่ากับ elem เท่านั้นจะถูกย้ายออกไป

### ตัวแปร

**elem** เป็น DataElement ที่ถูกย้ายออกไป

### Returns

true ถ้าค่าอินพุตถูกพบและถูกย้ายออกไป นอกจากนั้นจะเป็น false

### Throws

java.lang.ClassCastException หากออบเจกต์ไม่เป็นชนิดของ DATALT หรือ DATSEQ

java.lang.NullPointerException หาก elem มีค่าเป็น null

### - getDataType

public int **getDataType()**

คืนชนิดของข้อมูลของออบเจกต์ที่ใช้แทน DataElement นี้

### Returns

ชนิดข้อมูลของออบเจกต์ DataElement นี้ ค่าตามกฎที่จะได้คืนมาจะมี URL, NULL, BOOL, UUID, STRING, DATSEQ, DATALT, U\_INT\_1, U\_INT\_2, U\_INT\_4, U\_INT\_8, U\_INT\_16, INT\_1, INT\_2, INT\_4, INT\_8 หรือ INT\_16

### - getLong

public long **getLong()**

คืนชนิดของข้อมูลของ DataElement หากมันสามารถใช้แทน long ได้ ค่าที่จะได้คืนมาจะต้องเป็น U\_INT\_1, U\_INT\_2, U\_INT\_4, U\_INT\_8, U\_INT\_16, INT\_1, INT\_2, INT\_4, INT\_8 หรือ INT\_16

### Returns

ค่าของ DataElement ชนิด long

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Throws**

java.lang.ClassCastException หากออบเจกต์ไม่เป็นชนิดของ U\_INT\_1, U\_INT\_2, U\_INT\_4, U\_INT\_8, U\_INT\_16, INT\_1, INT\_2, INT\_4, INT\_8 หรือ INT\_16

**- getBoolean**

public Boolean **getBoolean()**

คืนค่าของ DataElement หากใช้แทน Boolean ได้

**Returns**

ค่า Boolean ของออบเจกต์ DataElement

**Throws**

java.lang.ClassCastException หากออบเจกต์ไม่เป็นชนิดของ BOOL

**- getValue**

public java.lang.Object **getValue()**

คืนค่าของ DataElement เหมือนเป็นออบเจกต์ เมธอดนี้คืนค่าของ จาวา ออบเจกต์ที่เหมาะสมตามชนิดดังต่อไปนี้ URL, UUID, STRING, DATSEQ, DATALT, U\_INT\_8, U\_INT\_16, INT\_16 ตารางต่อไปนี้แสดงคู่ของชนิดข้อมูลและชนิดของออบเจกต์ที่ได้ค่าคืนกลับ

ตารางที่ 2.17 ชนิดของข้อมูลค้ำอีลิเมนต์และชนิดของข้อมูลจาวา

DataElement Data Type	Java Data Type
URL	java.lang.String
UUID	javax.bluetooth.UUID
STRING	java.lang.String
DATSEQ	java.util.Enumeration
DATALT	java.util.Enumeration
U_INT_8	byte[] ความยาว 8
U_INT_16	byte[] ความยาว 16
INT_16	byte[] ความยาว 16

**Returns**

ค่าของออบเจกต์นี้

**Throws**

java.lang.ClassCastException หากออบเจกต์ไม่เป็นชนิดของ URL, UUID, STRING, DATSEQ, U\_INT\_8, U\_INT\_16 หรือ INT\_16

**2.3.6 Class DeviceClass**

public class DeviceClass extends java.lang.Object

คลาส DeviceClass แสดงข้อมูล class of device (CoD) ซึ่งถูกกำหนดจากลักษณะของบลูทูธ ข้อมูล class of device (CoD) นี้จะประกอบไปด้วยข้อมูลของชนิดของอุปกรณ์และบริการที่อุปกรณ์นั้นๆ ให้บริการอยู่ ซึ่งข้อมูล class of device (CoD) ถูกกำหนดไว้ในเอกสาร Bluetooth Assigned Numbers

เอกสาร Bluetooth Assigned ได้กำหนดรายละเอียดของ service class, major device class และ minor device class ไว้ ดังตัวอย่างที่แสดงไว้ในตารางข้างล่างนี้

ตารางที่ 2.18 รายละเอียดของ service class, major device class และ minor device class

Method	Return Value	Class of Device
getServiceClasses()	0x22000	Networking and Limited Discoverable Major Service Classes
getServiceClasses()	0x100000	Object Transfer Major Service Class
getMajorDeviceClass()	0x00	Miscellaneous Major Device Class
getMajorDeviceClass()	0x200	Phone Major Device Class
getMinorDeviceClass()	0x0C	With a Computer Major Device Class, Laptop Minor Device Class
getMinorDeviceClass()	0x04	With a Phone Major Device Class, Cellular Minor Device Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.19 รายละเอียดในส่วนของคนสร้าง DeviceClass

Constructor Summary	
DeviceClass(int record)	สร้าง DeviceClass จากข้อมูล class of device (CoD) ที่รับมา

ตารางที่ 2.20 รายละเอียดในส่วนของเมธอด DeviceClass

Method Summary	
int	getMajorDeviceClass() คืนค่า major device class
int	getMinorDeviceClass() คืนค่า minor device class
int	getServiceClasses() คืนค่า major service class

### 2.3.6.1 รายละเอียดของ Constructor

#### - DeviceClass

public DeviceClass(int record)

สร้าง DeviceClass จากข้อมูล class of device (CoD) ที่รับมา ซึ่ง record ที่รับมานั้นจะอยู่ในรูปของข้อมูล class of device

#### ตัวแปร

**record** เป็นชนิดของ class of device

#### Throws

java.lang.IllegalArgumentException หาก record มีขนาดระหว่าง 24-31 บิต

### 2.3.6.2 รายละเอียดของ Method

#### - getMajorDeviceClass()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public int getMajorDeviceClass()
```

จะคืนค่า major device class ซึ่งอุปกรณ์แต่ละตัวจะมีค่า major device class เพียงค่าเดียว

**Returns**

ค่า major device class

```
- getMinorDeviceClass()
```

```
public int getMinorDeviceClass()
```

จะคืนค่า minor device class

**Returns**

ค่า minor device class

```
- getServiceClasses()
```

```
public int getServiceClasses()
```

จะคืนค่า major service class ซึ่งอุปกรณ์แต่ละตัวอาจจะมีค่า major service class ได้หลาย

ค่า

**Returns**

ค่า major service class

### 2.3.7 Class DiscoveryAgent

```
public class DiscoveryAgent extends java.lang.Object
```

DiscoveryAgent มีเพื่อการค้นหาอุปกรณ์และบริการบนอุปกรณ์ LocalDevice หนึ่งตัวต่อ DiscoveryAgent หนึ่งตัว เราสามารถเชื่อมต่อ DiscoveryAgent ก็ได้เมื่อเราใช้ระบบเชื่อมต่อผ่านทาง LocalDevice

#### การค้นหาอุปกรณ์

มีสองวิธีในการค้นหาอุปกรณ์ วิธีที่หนึ่งคือใช้ startInquiry เพื่อเรียกการค้นหาอุปกรณ์ อุปกรณ์ที่ถูกค้นพบจะถูกส่งกลับมาด้วยการใช้วิธีค้นหาอุปกรณ์ของ DiscoveryListener วิธีที่สองคือการหาอุปกรณ์โดยการใช้ retrieveDevices วิธีนี้ อุปกรณ์ที่ถูกค้นพบจะเป็นอุปกรณ์ที่เราสามารถเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค้นหาได้ก่อนหน้านี โดยอุปกรณ์ตัวนี้จะได้รับการบันทึกในบลูทูธเรียบร้อยแล้ว retrieveDevices ไม่ได้เป็นขั้นตอนหรือกระบวนการเพื่อค้นหาอุปกรณ์แต่เป็นเพียงแค่การ โขว์รายชื่ออุปกรณ์ที่อยู่ในบริเวณที่เราค้นหาเท่านั้น

### การค้นหาการให้บริการ

DiscoveryAgent ยังเอื้อประโยชน์ในการให้ข้อมูลของการค้นหาการให้บริการในการค้นหาการให้บริการ ของอุปกรณ์ หนึ่งตัว เราควรใช้ searchServices แต่ถ้าไม่สนใจว่าการให้บริการ อันใดที่สามารถใช้งานได้ แต่เพียงต้องการค้นหาการให้บริการที่อยู่ในบริเวณรอบๆ เราสามารถใช้ selectService

ตารางที่ 2.21 รายละเอียดในส่วนของฟิลด์ DiscoveryAgent

Field Summary	
static int	CACHED ใช้ร่วมกับเมธอด retrieveDevices() เพื่อที่จะคืนค่าอุปกรณ์ที่เจอในการตรวจสอบก่อนหน้า
static int	GIAC เป็นการตรวจสอบรหัสการเข้าใช้สำหรับ General/Unlimited Inquiry Access Code (GIAC)
static int	LIAC เป็นการตรวจสอบรหัสการเข้าใช้สำหรับ Limited Dedicated Inquiry Access Code (LIAC)
static int	NOT_DISCOVERABLE เอาอุปกรณ์ออกจากโหมดการค้นหา
static int	PREKNOWN ใช้ร่วมกับเมธอด retrieveDevices() เพื่อที่จะคืนค่าอุปกรณ์ที่ได้กำหนดไว้เป็นอุปกรณ์ประเภท pre-known

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.22 รายละเอียดในส่วนของเมธอด DiscoveryAgent

Method Summary	
boolean	<b>cancelInquiry</b> (DiscoveryListener listener) ย้ายอุปกรณ์ออกจากโหมดการตรวจสอบ
boolean	<b>cancelServiceSearch</b> (int transID) ยกเลิกการดำเนินการค้นหาบริการที่มี ID การดำเนินการที่เฉพาะเจาะจง
RemoteDevice[]	<b>retrieveDevices</b> (int option) คืนค่าอาร์เรย์ของอุปกรณ์บลูทูธที่ถูพบโดยทั้ง local device ระหว่างร้องขอการตรวจสอบก่อนหน้าหรือเป็นถูกกำหนดเป็นอุปกรณ์ประเภท pre-known ขึ้นอยู่กับอาร์กิวเมนต์
int	<b>searchServices</b> (int[] attrSet, UUID[] uuidSet, RemoteDevice btDev, DiscoveryListener discListener) ค้นหาบริการบนอุปกรณ์ควบคุมบลูทูธซึ่งมี UUIDs ที่ถูกกำหนดอยู่ใน uuidSet ทั้งหมด
java.lang.String	<b>selectService</b> (UUID uuid, int security, boolean master) พยายามที่จะหาคำแหน่งการบริการซึ่งประกอบไปด้วย uuid ใน ServiceClassIDList ของ service record ของมันเอง
boolean	<b>startInquiry</b> (int accessCode, DiscoveryListener listener) นำอุปกรณ์เข้าไปในโหมดการตรวจสอบ

## 2.3.7.1 รายละเอียดของ Field

## - NOT\_DISCOVERABLE

public static final int NOT\_DISCOVERABLE

เอาอุปกรณ์ออกจากโหมดการค้นหา

ค่าของ NOT\_DISCOVERABLE มีค่าเท่ากับ 0x00 (0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - GIAC

public static final int **GIAC**

เป็นการตรวจสอบรหัสการเข้าใช้สำหรับ General/Unlimited Inquiry Access Code (GIAC)  
GIAC ใช้เพื่อระบุชนิดของการเรียกขอ

ค่าของ *GIAC* มีค่าเท่ากับ *0x9E8B33 (10390323)* ค่านี้บันทึกไว้ใน *Bluetooth Assigned Numbers document*

### - LIAC

public static final int **LIAC**

เป็นการตรวจสอบรหัสการเข้าใช้สำหรับ Limited Dedicated Inquiry Access Code (LIAC)  
LIAC ใช้เพื่อระบุชนิดของการเรียกขอ

ค่าของ *LIAC* มีค่าเท่ากับ *0x9E8B00 (10390272)* ค่านี้บันทึกไว้ใน *Bluetooth Assigned Numbers document*

### - CACHED

public static final int **CACHED**

ใช้ร่วมกับเมธอด `retrieveDevices()` เพื่อที่จะคืนค่าอุปกรณ์ที่เจอในการตรวจสอบก่อนหน้า ถ้าไม่สามารถคืนหาอุปกรณ์ได้ก่อนหน้าเลย ผลที่กลับมาคือ 0

ค่าของ *CACHED* มีค่าเท่ากับ *0x00 (0)*

### - PREKNOWN

public static final int **PREKNOWN**

ใช้ร่วมกับเมธอด `retrieveDevices()` เพื่อที่จะคืนค่าอุปกรณ์ที่ได้กำหนดไว้เป็นอุปกรณ์ประเภท pre-known สำหรับอุปกรณ์ประเภท Pre-known ได้ถูกกำหนดอยู่ใน BCC

ค่าของ *PREKNOWN* มีค่าเท่ากับ *0x01 (1)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.7.2 รายละเอียดของ Method

#### - retrieveDevices

public RemoteDevice[] **retrieveDevices**(int option)

คืนค่าอาร์เรย์ของอุปกรณ์บลูทูธที่ถูพบโดยทั้ง local device ระหว่างร้องขอการตรวจสอบก่อนหน้าหรือเป็นถูกกำหนดเป็นอุปกรณ์ประเภท pre-known ขึ้นอยู่กับอาร์กิวเมนต์

#### ตัวแปร

**option** เป็น CACHED หากอุปกรณ์ที่พบก่อนหน้าควรจะถูกคืนค่ากลับมา เป็น PREKNOWN ถ้าเป็นอุปกรณ์ pre-known

#### Returns

ค่าอาร์เรย์ประกอบไปด้วยอุปกรณ์บลูทูธซึ่งถูกพบโดยก่อนหน้าถ้า option เป็นค่า CACHED ค่าอาร์เรย์ของอุปกรณ์จะเป็น Pre-Known ถ้า option เป็น PREKNOWN และจะเป็น null ถ้าไม่มีอุปกรณ์ใดเลยพบ criteria

#### Throws

java.lang.IllegalArgumentException ถ้า option ไม่เป็น CACHED หรือ PREKNOWN

#### - startInquiry

public boolean **startInquiry**(int accessCode, DiscoveryListener listener)

throws BluetoothStateException

นำอุปกรณ์เข้าไปในโหมดการตรวจสอบ ความยาวของการตรวจสอบนั้นมีผล เมธอดนี้จะค้นหาการตรวจสอบการเข้ารหัสของอุปกรณ์ อุปกรณ์ที่ตอบสนองการตรวจสอบจะคืนค่าไปยังแอพลิเคชันโดยผ่าน deviceDiscovered() ของอินเทอร์เฟส DiscoveryListener ส่วนเมธอด cancelInquiry() จะถูกเรียกเพื่อที่จะหยุดการตรวจสอบ

#### ตัวแปร

**accessCode** เป็นชนิดของการเสร็จสิ้นตรวจสอบ

**listener** เป็นอินเวนต์ listener ที่จะรับการค้นหาอุปกรณ์

#### Returns

true ถ้าหากการตรวจสอบได้เริ่มต้น เป็น false ถ้าหากการตรวจสอบไม่ได้เริ่มขึ้นเพราะ accessCode ไม่ได้สนับสนุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Throws**

java.lang.IllegalArgumentException ถ้ารหัสการเข้าไม่ได้เป็น LIAC,GIAC หรืออยู่ในช่วงของ 0x9E8B00 ถึง 0x9E8B3F

java.lang.NullPointerException ถ้า listener เป็นค่า null

BluetoothStateException ถ้าอุปกรณ์บลูทูธไม่อนุญาตให้การตรวจสอบเริ่มขึ้นเนื่องจากการดำเนินการของอุปกรณ์อื่นอยู่

**- cancelInquiry**

public boolean **cancelInquiry**(DiscoveryListener listener)

ย้ายอุปกรณ์ออกจากโหมดการตรวจสอบ

inquiryCompleted เกิดเมื่อมีการใช้คำสั่ง INQUIRY\_TERMINATED วิธีนี้จะหยุดการใช้ deviceDiscovered() วิธีนี้ยังยับยั้งการคำสั่งหรือการเรียกขอกี่ต่อเมื่อ listener ที่ใช้คือ อันเดียวกับที่เริ่มการใช้คำสั่ง

**ตัวแปร**

listener มีไว้เพื่อเป็นตัวรับการตรวจสอบ

**Returns**

true เมื่อยกเลิกคำสั่งหรือการเรียกขอ false เมื่อ คำสั่งหรือการเรียกขอไม่ได้รับการยกเลิกหรือการเรียกขอไม่ได้เริ่มการใช้ listener

**Throws:**

java.lang.NullPointerException เมื่อ listener มีค่าเป็น null

**- searchService**

public int **searchServices**(int[] attrSet, UUID[] uuidSet, RemoteDevice btDev,

DiscoveryListener discListener) throws BluetoothStateException

ค้นหาบริการบนอุปกรณ์ควบคุมบลูทูธซึ่งมี UUIDs ที่ถูกกำหนดอยู่ใน uuidSet ทั้งหมดเมื่อสิ้นสุดการค้นหาเราจะได้ service ที่ระบุไว้ใน attrSet และ service ประเภทที่ไม่เข้าข่ายการค้นหา ประเภทที่ไม่เข้าข่ายการค้นหาได้แก่ ServiceRecordHandle (0x0000), ServiceClassIDList (0x0001), ServiceRecordState (0x0002), ServiceID (0x0003) และ ProtocolDescriptorList (0x0004) ถ้าค่าของ attrSet เท่ากับ ๐ แสดงว่าผลของการค้นหา device ที่สามารถใช้งานได้ เท่ากับ ๐ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรืออีกนัยหนึ่งคือ ผลที่ได้มีเพียงอุปกรณ์ที่ไม่เข้าข่ายการค้นหา ค่าของ attrSet ไม่จำเป็นต้องเป็นบวก แต่เป็นค่าที่อยู่ในช่วงระหว่าง  $[0 - (2^{16}-1)]$ .

### ตัวแปร

**attrSet** คุณลักษณะที่ได้รับการค้นหาโดย service ที่มี UUIDs ซึ่งได้รับการระบุไว้ใน uuidSet  
**uuidSet** เซ็ตของ UUIDs ที่ถูกค้นหา service ทุกตัวที่ได้รับการค้นหาจะถูกระบุประเภทในหมวดของ UUIDs

**btDev** Bluetooth device ที่อยู่ในระยะใกล้ซึ่งมีไว้ค้นหา service

**discListener** อุปกรณ์ที่ทำหน้าที่รับ service

### Returns

ค่าที่ได้จากการค้นหา จะเป็นบวกเสมอ

### Throws

**BluetoothStateException** เกิดต่อเมื่อค่าของ service ที่ค้นหาได้มีค่าเกินจาก bluetooth.sd.trans.max ระบุไว้ service ที่อยู่ในหมวดของ LocalDevice หรือแม้แต่วิธีทั้งหมดจะไม่สามารถทำงานได้

**java.lang.IllegalArgumentException** เกิดต่อเมื่อ service ที่ได้เป็น service ที่ผิดกฎหมายหรือเกินจากค่าที่ bluetooth.sd.trans.max กำหนดไว้ใน หมวดของ LocalDevice หรือเกิดในกรณีที่ attrSet หรือ uuidSet มีค่ามากกว่า 0 หรือต่อเมื่อ attrSet หรือ uuidSet มีค่าเหมือนกัน

**java.lang.NullPointerException** เกิดต่อเมื่อค่าของ uuidSet, btDev, หรือ discListener เท่ากับ 0 หรือแม้แต่ อีลีเมนต์หนึ่งของ uuidSet มีค่าเท่ากับ 0

### - cancelServiceSearch

public boolean cancelServiceSearch(int transID)

คำสั่งนี้มีหน้าที่เพื่อยกเลิกการค้นหา service รหัสคำสั่งถูกป้อนโดยวิธี searchServices SERVICE\_SEARCH\_TERMINATED จะปรากฏหลังจากคำสั่งถูกป้อน หลังจากคำสั่งถูกป้อนกระบวนการ servicesDiscovered จะหยุดการทำงาน

### ตัวแปร

**transID** เกิดเมื่อการค้นหา service ถูกยกเลิก ซึ่งจะได้รับการตอบกลับโดย searchServices การตอบกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Returns**

True ต่อเมื่อกระบวนการค้นหาถูกยกเลิก false เมื่อ transID ไม่ปรากฏ

**- selectService**

```
public java.lang.String selectService(UUID uuid, int security, boolean master)
```

throws BluetoothStateException

ใช้เพื่อการติดตั้ง service ที่มี uuid ใน ServiceClassIDList ของรายชื่อของ service ที่ได้รับการบันทึกไว้ ผลลัพธ์ของการใช้คำสั่งนี้จะได้เป็นชุดของ service ที่สามารถจะนำไปใช้งานได้ ใน Connector.open เพื่อเปิดใช้ service แต่การเลือก service นั้นขึ้นอยู่กับประเภทของการใช้งาน ขอบเขต

**ตัวแปร**

**uuid** ใช้เพื่อการค้นหาServiceClassIDList

**security** ปรากฏเมื่อป้อนคำสั่ง security service เพื่อเชื่อมต่อไปยัง service ดังต่อไปนี้

ServiceRecord.NOAUTHENTICATE\_NOENCRYPT,

ServiceRecord.AUTHENTICATE\_NOENCRYPT

ServiceRecord.AUTHENTICATE\_ENCRYPT

**master** ปรากฏเมื่อ client เป็นmaster ของการเชื่อมต่อ trueเมื่อ client เป็นmaster แต่เป็น false เมื่อ clientเป็น masterหรือ slave

**Returns**

ผลลัพธ์ที่ได้เป็นชุดของ การเชื่อมโยงไปยัง service ด้วยการ ใช้ UUID ของ uuid ผลลัพธ์เป็น ๐ เมื่อไม่สามารถค้นพบ service ด้วยการ ใช้ UUID ของ uuid ที่บันทึกไว้ใน ServiceClassIDList

**Throws**

BluetoothStateException ปรากฏเมื่อระบบ Bluetooth ไม่ตอบสนองคำสั่ง

java.lang.NullPointerException เมื่อ uuid เท่ากับ ๐

java.lang.IllegalArgumentException เมื่อ security ไม่ใช่

ServiceRecord.NOAUTHENTICATE\_NOENCRYPT,

ServiceRecord.AUTHENTICATE\_NOENCRYPT

ServiceRecord.AUTHENTICATE\_ENCRYPT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.8 Class LocalDevice

`public class LocalDevice extends java.lang.Object`

คลาส LocalDevice กำหนดฟังก์ชันพื้นฐานของการจัดการบลูทูธ การจัดการบลูทูธได้แสดงระดับต่ำสุดของอินเตอร์เฟสที่เป็นไปได้ในบลูทูธสเต็ก มันยังแสดงการเข้าถึงและควบคุมอุปกรณ์บลูทูธ

ตารางที่ 2.23 รายละเอียดในส่วนของเมธอด LocalDevice

Method Summary	
java.lang.String	<b>getBluetoothAddress()</b> ตรวจสอบแอดเดรสบลูทูธของ local device
DeviceClass	<b>getDeviceClass()</b> ตรวจสอบออบเจกต์ DeviceClass ซึ่งเป็นตัวแทนของคลาสการให้บริการ, คลาสอุปกรณ์หลัก, และคลาสอุปกรณ์รองของ local device
int	<b>getDiscoverable()</b> ตรวจสอบโหมดการค้นพบของ local Device
DiscoveryAgent	<b>getDiscoveryAgent()</b> คืนค่า discovery agent สำหรับอุปกรณ์
java.lang.String	<b>getFriendlyName()</b> ตรวจสอบชื่อของ local Device
static LocalDevice	<b>getLocalDevice()</b> ตรวจสอบออบเจกต์ LocalDevice สำหรับอุปกรณ์บลูทูธ local
static java.lang.String	<b>getProperty(java.lang.String property)</b> ตรวจสอบคุณสมบัติของระบบบลูทูธ
ServiceRecord	<b>getRecord(Connection notifier)</b> รับการแจ้งของ Service Record ที่คล้ายกันไปยัง btsp, btl2cap หรือ btgoep

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.23 รายละเอียดในส่วนของเมธอด LocalDevice (ต่อ)

Method Summary	
static boolean	<b>isPowerOn()</b> สอบถามสถานะของอุปกรณ์บลูทูธ
boolean	<b>setDiscoverable(int mode)</b> เซตให้อุปกรณ์เป็น โหมด Discoverable
void	<b>updateRecord(ServiceRecord srvRecord)</b> อัปเดต Service Record ใน SDDB ที่มีตัวแปรคล้ายกันใน ServiceRecord

### 2.3.8.1 รายละเอียดของ Method

#### - getLocalDevice

public static LocalDevice **getLocalDevice()** throws BluetoothStateException

ตรวจสอบออบเจกต์ LocalDevice สำหรับอุปกรณ์บลูทูธ local เมธอดนี้จะ ไม่คืนค่ากลับมาเป็น null

#### Returns

ออบเจกต์ที่เป็นตัวแทนของอุปกรณ์บลูทูธแบบ local

#### Throws

BluetoothStateException ถ้าระบบบลูทูธไม่สารทที่จะเริ่มต้นได้

#### - getDiscoveryAgent

public DiscoveryAgent **getDiscoveryAgent()**

คืนค่า discovery agent สำหรับอุปกรณ์ เมธอดนี้จะ ไม่คืนค่ากลับมาเป็น null

#### Returns

ค่า Discovery Agent สำหรับ local device

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**-getFriendlyName**

```
public java.lang.String getFriendlyName()
```

ตรวจสอบชื่อของ local Device การเรียกชื่อเฉพาะของบลูทูธ “ชื่ออุปกรณ์บลูทูธ” หรือ “ชื่ออย่างไม่เป็นทางการของผู้ใช้”

**Returns**

ชื่อของ local device จะเป็น null ก็ต่อเมื่อไม่สามารถตรวจสอบได้

**- getDeviceClass**

```
public DeviceClass getDeviceClass()
```

ตรวจสอบออบเจกต์ DeviceClass ซึ่งเป็นตัวแทนของคลาสการให้บริการ, คลาสอุปกรณ์หลัก, และคลาสอุปกรณ์รองของ local device จะคืนค่ามาเป็น null หากคลาสการให้บริการ, คลาสอุปกรณ์หลัก หรือคลาสอุปกรณ์รองไม่สามารถตรวจสอบได้

**Returns**

คืนค่ามาเป็น Service classes, major device class และ minor device class ของ local device จะคืนค่ามาเป็น null หากคลาสการให้บริการ, คลาสอุปกรณ์หลัก หรือคลาสอุปกรณ์รองไม่สามารถตรวจสอบได้

**-getProperty**

```
public static java.lang.String getProperty(java.lang.String property)
```

ตรวจสอบคุณสมบัติของระบบบลูทูธ โดยต้องสนับสนุนคุณสมบัติดังต่อไปนี้ แต่สามารถให้ใส่ค่าเพิ่มเข้าไปได้

ตารางที่ 2.24 รายละเอียดของชื่อพรีอเพอติ

Property Name	คำอธิบาย
bluetooth.api.version	เป็นเวอร์ชันของ Java API ซึ่งสนับสนุนเทคโนโลยีไร้สายชนิดบลูทูธ โดยเวอร์ชันนี้จะถูกกำหนดค่าเป็น “1.0”
bluetooth.master.switch	ดูว่า master/Slave นั้นถูกสวิตซ์ว่าใช้ได้หรือไม่ ค่าที่ได้จะเป็น true หรือ false
bluetooth.sd.attr.retrievable.max	เลขจำนวนมากที่สุดของแอตทริบิวต์การให้บริการที่จะได้กลับคืนมาต่อหนึ่ง Service record โดยสตริงจะเป็นเลขฐาน 10
bluetooth.connected.devices.max	เป็นเลขจำนวนที่มากที่สุดของอุปกรณ์การเชื่อมต่อที่สนับสนุน เลขนี้อาจมีค่ามากกว่า 7 ถ้าพัคการควบคุมการเชื่อมต่อมีผล โดยสตริงจะเป็นเลขฐาน 10
bluetooth.l2cap.receiveMTU.max	ขนาดที่ใหญ่ที่สุดของ ReceiveMTU ในหน่วยไบต์ซึ่งสนับสนุนใน L2CAP โดยสตริงจะเป็นเลขฐาน 10
bluetooth.sd.trans.max	เลขจำนวนมากที่สุดของการดำเนินการค้นพบ concurrent service โดยสตริงจะเป็นเลขฐาน 10
bluetooth.connected.inquiry.scan	ได้มีการอนุญาตให้แสกนการตรวจสอบระหว่างการเชื่อมต่อหรือไม่ ค่าที่ได้จะเป็น true หรือ false
bluetooth.connected.page.scan	ได้มีการอนุญาตให้ทำการแสกนเพจการตรวจสอบระหว่างการเชื่อมต่อหรือไม่ ค่าที่ได้จะเป็น true หรือ false เท่านั้น
bluetooth.connected.inquiry	ได้มีการอนุญาตให้มีการตรวจสอบระหว่างการเชื่อมต่อหรือไม่ ค่าที่ได้จะเป็น true หรือ false
bluetooth.connected.page	ได้มีการอนุญาตให้ทำการสแกนเพจระหว่างการเชื่อมต่อหรือไม่ หรือในอีกทางหนึ่ง สามารถที่จะสร้างการเชื่อมต่อในอุปกรณ์ได้หรือไม่หากได้มีการเชื่อมต่อไปแล้วกับอีกอุปกรณ์หนึ่ง ค่าที่ได้จะเป็น true หรือ false

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตัวแปร**

**property** คุณสมบัติที่ได้รับตั้งที่ได้กำหนดลงในคลาส

**Returns**

ค่าของคุณสมบัติที่ได้กำหนดไว้ หรือเป็น null หาก property ไม่ได้ถูกกำหนดไว้

**-getDiscoverable**

```
public int getDiscoverable()
```

ตรวจสอบโหมดการค้นพบของ local Device ซึ่งค่าที่จะได้รับกลับมาจะเป็น DiscoveryAgent.GIAC, DiscoveryAgent.LIAC, DiscoveryAgent.NOT\_DISCOVERABLE, นั้นเป็นค่าในช่วงตั้งแต่ 0x9E8B00 ถึง 0x9E8B3F

**Returns**

โหมด discoverable ที่ปรากฏอยู่ในอุปกรณ์

**- getBluetoothAddress**

```
public java.lang.String getBluetoothAddress()
```

ตรวจสอบแอดเดรสบลูทูธของ local device โดยที่แอดเดรสของบลูทูธจะไม่มีค่าเป็น null แอดเดรสของบลูทูธจะมีความยาวทั้งหมด 12 ตัว ตั้งแต่ 0-9 และ A-F

**Returns**

แอดเดรสบลูทูธของ local device

**- setDiscoverable**

```
public boolean setDiscoverable(int mode) throws BluetoothStateException
```

เซตให้อุปกรณ์เป็นโหมด Discoverable โหมดนี้เป็นเลขใดก็ได้ที่อยู่ในช่วง 0x9E8B00 ถึง 0x9E8B3F ดังที่ได้กำหนดอยู่ในเอกสารการกำหนดค่าให้กับบลูทูธ

**ตัวแปร**

**mode** โหมดที่อุปกรณ์ควรจะเป็น โหมดที่เป็นไปได้ ได้แก่ DiscoveryAgent.GIAC, DiscoveryAgent.LIAC, DiscoveryAgent.NOT\_DISCOVERABLE และค่าใดๆ ก็ได้ที่อยู่ในช่วงตั้งแต่ 0x9E8B00 ถึง 0x9E8B3F

**Returns**

true หากการร้องขอสำเร็จ false หากการร้องขอเกิดความผิดพลาดเพราะ BCC ปฏิเสธการร้องขอ false หากระบบบลูทูธไม่สนับสนุนโหมดการเข้าถึงตามที่ได้กำหนดใน mode

**throws**

java.lang.IllegalArgumentException หากโหมดไม่ได้เป็น DiscoveryAgent.GIAC, DiscoveryAgent.LIAC, DiscoveryAgent.NOT\_DISCOVERABLE และค่าใดๆ ก็ได้ที่อยู่ในช่วงตั้งแต่ 0x9E8B00 ถึง 0x9E8B3F

BluetoothStateException หากระบบบลูทูธอยู่ในสถานะที่ไม่อนุญาตให้โหมด Discoverable สามารถเป็นสถานะได้

**- getRecord**

public ServiceRecord **getRecord**(Connection notifier)

รับการแจ้งของ Service Record ที่คล้ายกันไปยัง btsp, bt2cap หรือ btgoep ในกรณีของการให้บริการ run-before-connect นี้ service record จะคืนค่าโดย getRecord() จะถูกสร้างโดยการเรียกเข้าไปยัง Connector.open() ซึ่งเป็นตัวสร้าง notifier

**ตัวแปร**

**notifier** การเชื่อมต่อที่รอถูกข้ายเพื่อที่จะเชื่อมต่อการบริการบลูทูธ

**Returns**

ServiceRecord ที่ทำงานร่วมกับ notifier

**Throws**

java.lang.IllegalArgumentException ถ้าเปิดการ notifier หรือ notifier ไม่ทำให้มีอินเตอร์เฟสตามที่จกกล่าวถึงต่อไปนี้มีผล นั่นคือ javax.microedition.io.StreamConnectionNotifier,

javax.bluetooth.L2CapConnectionNotifier หรือ javax.obex.SessionNotifier

java.lang.NullPointerException ถ้า noitifier มีค่าเป็น null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - **updateRecord**

public void **updateRecord**(ServiceRecord srvRecord)

อัปเดต Service Record ใน SDDB ที่มีตัวแปรคล้ายกันใน ServiceRecord การอัปเดตจะเป็นไปได้ถ้า sryRecord ได้รับมาโดยการไ้เมธอด getRecord() ส่วน Service Record ใน SDDB จะถูกแก้ไขเพื่อที่จะมีแอทริบิวต์การให้บริการเหมือนกันกับค่าปริมาณที่เท่ากับ sryRecord

### ตัวแปร

**sryRecord** เป็นปริมาณใหม่ที่ใช้สำหรับ service record ใน SDDB

### Throws

java.lang.NullPointerException หากค่า sryRecord มีค่าเป็น null

java.lang.IllegalArgumentException หากโครงสร้างของ sryRecord ได้มี mandatory service attributes ใดๆ หายไป หรือตั้งใจจะเปลี่ยนค่าที่ถูกกำหนดเอาไว้แล้ว

ServiceRegistrationException ถ้า local SDDB ไม่สามารถที่จะอัปเดตได้อย่างสมบูรณ์เนื่องจากพื้นที่ดิสก์หรือฐานข้อมูลมีปัญหา เป็นต้น

### - **isPowerOn**

public static boolean **isPowerOn**()

สอบถามสถานะของอุปกรณ์บลูทูธ

### Returns

True หากอุปกรณ์บลูทูธเปิดสัญญาณ สำหรับกรณีอื่นๆ นอกเหนือจากนี้เป็น false

### 2.3.9 Class RemoteDevice

`public class RemoteDevice extends java.lang.Object`

คลาส RemoteDevice เป็นตัวแทนของการควบคุมอุปกรณ์บลูทูธ แสดงข้อมูลพื้นฐานเกี่ยวกับการควบคุมอุปกรณ์รวมถึงแอดเดรสของอุปกรณ์บลูทูธและชื่ออย่างไม่เป็นทางการของตัวมันเอง

ตารางที่ 2.25 รายละเอียดของคอนสตรัคเตอร์ RemoteDevice

Constructor Summary	
protected	<b>RemoteDevice</b> (java.lang.String address) สร้างฐานของอุปกรณ์บลูทูธขึ้นกับแอดเดรสของมัน

ตารางที่ 2.26 รายละเอียดของเมธอด RemoteDevice

Method Summary	
boolean	<b>authenticate</b> () ตรวจสอบสิทธิ์กับ RemoteDevice() นี้
boolean	<b>authorize</b> (Connection conn) ระบุไว้หาก RemoteDevice นี้ควรที่จะอนุญาตให้เข้าไปยัง local service แสดงโดย Connection
boolean	<b>encrypt</b> (Connection conn, boolean on) สำหรับเปิด/ปิด การเข้ารหัสสำหรับการเชื่อมต่อที่ยังมีอยู่
boolean	<b>equals</b> (java.lang.Object obj) ระบุไว้หาก RemoteDeviceS ทั้งสองตัวมีค่าเท่ากัน
java.lang.String	<b>getBluetoothAddress</b> () รับค่าแอดเดรสบลูทูธของอุปกรณ์
java.lang.String	<b>getFriendlyName</b> (boolean alwaysAsk) คืนค่าชื่อของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.26 รายละเอียดของเมธอด RemoteDevice (ต่อ)

Method Summary	
boolean	<b>isEncrypted()</b> ระบุไว้หากข้อมูลได้แลกเปลี่ยนกับ RemoteDevice นี้ ได้มีการเข้ารหัส
static RemoteDevice	<b>getRemoteDevice(Connection conn)</b> รับค่าอุปกรณ์บลูทูธซึ่งเป็นส่วนสุดท้ายของการเชื่อมต่อโปรไฟล์ Bluetooth Serial Port, การเชื่อมต่อแบบ L2CAP หรือ OBEX ในการเชื่อมต่อแบบ RFCOMM
int	<b>hashCode()</b> คำนวณ hash code สำหรับออบเจกต์นี้
boolean	<b>isAuthenticated()</b> ระบุไว้หาก RemoteDevice นี้ได้ถูกตรวจสอบสิทธิ์แล้ว
boolean	<b>isAuthorized(Connection conn)</b> ระบุไว้หาก RemoteDevice ได้ถูกกำหนดสิทธิ์ก่อนหน้าโดย BCC ของ local device เพื่อที่จะแลกเปลี่ยนข้อมูลขึ้นกับการให้บริการร่วมกับการเชื่อมต่อ
boolean	<b>isTrustedDevice()</b> ระบุไว้หากอุปกรณ์ที่เชื่อถือได้นี้เป็นไปตามข้อกำหนดของ BCC

### 2.3.9.1 รายละเอียดของ Constructor

#### RemoteDevice

**protected RemoteDevice(java.lang.String address)**

สร้างฐานของอุปกรณ์บลูทูธขึ้นกับแอดเดรสของมัน โดยแอดเดรสของบลูทูธต้องเป็นเลขฐาน 16 ซึ่งมีความยาว 12 ตัว โดยมีตั้งแต่ 0-9 และ A-F และไม่เป็น "0x" ในสตริงของมัน

#### ตัวแปร

**address** เป็นแอดเดรสของอุปกรณ์บลูทูธ เป็นเลขฐาน 16 ซึ่งมีความยาว 12 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Throws**

java.lang.NullPointerException หากค่าแอดเดรสมีค่าเป็น null

java.lang.IllegalArgumentException หาก address เป็นแอดเดรสของ local device หรือไม่ใช่แอดเดรสบลูทูธที่ถูกต้อง

**2.3.9.2 รายละเอียดของ Method****- isTrustedDevice**

public boolean isTrustedDevice()

ระบุไว้หากอุปกรณ์ที่เชื่อมต่อได้นี้เป็นไปตามข้อกำหนดของ BCC

**Returns**

true หากอุปกรณ์นั้นเป็นอุปกรณ์ที่เชื่อมต่อได้ สำหรับกรณีอื่นๆ จะเป็น false

**- getFriendlyName**

public java.lang.String getFriendlyName(boolean alwaysAsk) throws java.io.IOException

คืนค่าชื่อของอุปกรณ์ การเรียกชื่อเฉพาะของบลูทูธนี้จะเป็น “ชื่อของอุปกรณ์บลูทูธ” หรือ “ชื่ออย่างไม่เป็นทางการของผู้ใช้” เมธอดนี้จะติดต่อเพียง remote device หากไม่รู้จักชื่อมาก่อน หรือ alwaysask มีค่าเป็น true

**ตัวแปร**

**alwaysAsk** ถ้าเป็น true อุปกรณ์จะติดต่อด้วยชื่อนั้น นอกเหนือจากนั้น ถ้าหากที่มีอยู่เป็นชื่อที่รู้จักแล้ว ชื่อนั้นก็จะถูกคืนค่ากลับมาโดยไม่ต้องติดต่อกับ remote device อีก

**Returns**

ชื่อของอุปกรณ์ หรือ null ถ้าระบบบลูทูธนั้นไม่ได้สนับสนุนส่วนนี้ หาก local device สามารถจะติดต่อ remote device ผลที่ได้จะไม่มีทางมีค่าเป็น null ถ้า remote device ไม่มีชื่อจะได้สตริงที่มีค่าว่างเปล่ากลับมา

**Throws**

java.io.IOException หาก remote device ไม่สามารถที่จะติดต่อกับได้หรือ remote device ไม่สามารถแสดงชื่อของมันได้

### - `getBluetoothAddress`

`public final java.lang.String getBluetoothAddress()`

รับค่าแอดเดรสบลูทูธของอุปกรณ์ แอดเดรสของบลูทูธมีค่าความยาว 12 ตัวอักษร ประกอบด้วย 0-9 และ A-F เมธอดนี้จะไม่มีทางมีค่าเป็น null

#### Returns

แอดเดรสบลูทูธของ local device

### - `equals`

`public boolean equals(java.lang.Object obj)`

ระบุไว้หาก RemoteDeviceS ทั้งสองตัวมีค่าเท่ากัน โดยอุปกรณ์ทั้งสองนี้จะมีค่าเท่ากันได้ หากมีค่าแอดเดรสของอุปกรณ์บลูทูธเดียวกัน

#### Overrides

equal อยู่ในคลาส java.lang.Object

#### ตัวแปร

obj เป็นออบเจกต์ที่ใช้เปรียบเทียบ

#### Returns

true หากทั้งคู่มีแอดเดรสของบลูทูธเดียวกัน false หากอุปกรณ์ทั้งสองมีแอดเดรสไม่ตรงกัน หาก obj มีค่าเป็น null หาก obj ไม่เป็น RemoteDevice

### - `hashCode`

`public int hashCode()`

คำนวณ hash code สำหรับออบเจกต์นี้ เมธอดนี้จะคืนค่าเดียวกันเมื่อถูกเรียกหลายๆครั้ง ในออบเจกต์เดียวกัน

#### Overrides

hashCode อยู่ในคลาส java.lang.Object

#### Returns

จะคืนค่า hash code สำหรับออบเจกต์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - getRemoteDevice

public static RemoteDevice **getRemoteDevice**(Connection conn)

รับค่าอุปกรณ์บลูทูธซึ่งเป็นส่วนสุดท้ายของการเชื่อมต่อ โปรไฟล์ Bluetooth Serial Port, การเชื่อมต่อแบบ L2CAP หรือ OBEX ในการเชื่อมต่อแบบ RFCOMM เมธอดนี้จะไม่คืนค่าเป็น null

#### ตัวแปร

**conn** เป็นสิ่งที่ remote Bluetooth device ต้องการ อาทิเช่น การเชื่อมต่อ Bluetooth Serial Port, การเชื่อมต่อแบบ L2CAP หรือ OBEX ในการเชื่อมต่อแบบ RFCOMM

#### Returns

Remote device ที่ร่วมอยู่ในการเชื่อมต่อ

#### Throws

java.lang.IllegalArgumentException หาก conn ไม่ได้เป็นการเชื่อมต่อ Bluetooth Serial Port, การเชื่อมต่อแบบ L2CAP หรือ OBEX ในการเชื่อมต่อแบบ RFCOMM หรือ conn เป็น L2CAPConnectionNotifier, StreamConnectionNotifier, หรือ SessionNotifier

java.io.IOException หากเปิดการเชื่อมต่อ

java.lang.NullPointerException หาก conn มีค่าเป็น null

### - authenticate

public boolean **authenticate**() throws java.io.IOException

สำหรับตรวจสอบสิทธิ์กับ RemoteDevice() การตรวจสอบสิทธิ์นั้นมีความหมายว่าระบุลักษณะของ remote device การตรวจสอบสิทธิ์รวมไปถึงการเรียกชื่อของ device-to-device และแผนการการตอบสนองซึ่งต้องการคีย์ลับเฉพาะเฉพาะทั่วไปจำนวน 128 บิต ที่ได้รับมาจาก PIN code ซึ่งใช้ร่วมกันระหว่างอุปกรณ์สองอุปกรณ์ หาก PIN code ด้านใดก็ตามไม่เหมือนกัน การตรวจสอบสิทธิ์ก็จะเกิดความผิดพลาดขึ้น และเมธอดก็จะคืนค่าเป็น false และเมธอดก็จะคืนค่าเป็น false อีกด้วยหากการตรวจสอบสิทธิ์ไม่ประสบความสำเร็จกับการติดตั้งค่ารักษาความปลอดภัยของ local device ที่สร้างโดย BCC หากแต่ไม่สนับสนุนการตรวจสอบสิทธิ์เลย

หาก RemoteDevice ก่อนหน้าได้ถูกตรวจสอบสิทธิ์ไปแล้ว ดังนั้นเมธอดนี้จะคืนค่ากลับมาเป็น true โดยไม่ต้อง ตรวจสอบสิทธิ์ของ RemoteDevice ใหม่อีกครั้ง

**Returns**

True หากการตรวจสอบสิทธิ์สำเร็จ นอกเหนือจากนี้จะมีค่าเป็น false

**Throws**

java.io.IOException หากไม่มีการเปิดการเชื่อมต่อระหว่าง local device และ RemoteDevice นี้

**- authorize**

public boolean **authorize**(Connection conn) throws java.io.IOException

ระบุไว้หาก RemoteDevice นี้ควรที่จะอนุญาตให้เข้าไปยัง local service แสดงโดย Connection เมชอด authorize() นี้ยังจะตรวจสอบด้วยว่า RemoteDevice นี้สามารถที่จะพิสูจน์ความถูกต้องระหว่างการตรวจสอบสิทธิ์ หาก RemoteDevice ได้ถูกกำหนดสิทธิ์สำหรับ conn ก่อนหน้าแล้วเมชอดนี้ก็จะคืนค่า true โดยไม่ต้องทำการกำหนดสิทธิ์ใหม่อีกครั้ง

**ตัวแปร**

conn การเชื่อมต่อที่มี RemoteDevice ใช้เพื่อจะเข้าไปยัง local service

**Returns**

true หาก RemoteDevice นั้นตรวจสอบสิทธิ์และกำหนดสิทธิ์ได้ นอกเหนือจากนี้จะเป็น false หากการตรวจสอบสิทธิ์และกำหนดสิทธิ์เกิดล้มเหลวขึ้นมา

**Throws**

java.lang.IllegalArgumentException หาก conn ไม่เชื่อมต่อไปยัง RemoteDevice หรือ local device ในตอนเริ่มต้นของการเชื่อมต่อ

java.io.IOException หาก conn ได้มีการปิดตัวลง

**- encrypt**

public boolean **encrypt**(Connection conn,boolean on) throws java.io.IOException

สำหรับเปิด/ปิด การเข้ารหัสสำหรับการเชื่อมต่อที่ยังมีอยู่ ในกรณีนี้เมื่อตัวแปร on มีค่าเป็น true เมชอดนี้จะตรวจสอบสิทธิ์ RemoteDevice เป็นอย่างแรกหากยังไม่เคยได้ตรวจสอบสิทธิ์มาก่อน หลังจากนั้นก็จะเปิดการเข้ารหัส หากการเชื่อมต่อนั้นได้ถูกเข้ารหัสมาก่อนหน้าแล้วเมชอดนี้ก็จะคืนค่ามาเป็น true นอกเหนือจากนี้ตัวแปร on จะเป็น true ได้อีกดังต่อไปนี้

- เมชอดนี้ได้สำเร็จในการเปิดการเข้ารหัสสำหรับการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมธอดนี้เปิดการเข้ารหัสไม่สำเร็จและคืนค่า `false` จะเกิดขึ้นได้เพราะว่าสแต็กไม่สนับสนุนการเข้ารหัสหรือเพราะว่าการเข้ารหัสไม่ตรงกับการติดตั้งการรักษาความปลอดภัยของผู้ใช้สำหรับอุปกรณ์นั้น

ในกรณีที่ตัวแปร `on` คืนค่ามาเป็น `false` เป็นไปได้ดังต่อไปนี้

- การเข้ารหัสได้ถูกปิดในการเชื่อมต่อและได้คืนค่ามาเป็น `true`
- การเข้ารหัสได้ถูกทิ้งไปสำหรับการเชื่อมต่อและคืนค่า `false` กลับมา

## ตัวแปร

**conn** การเชื่อมต่อซึ่งต้องการสำหรับการเปลี่ยนการเข้ารหัส

**on** `true` เพื่อที่จะเปิดการเข้ารหัส `false` เพื่อจะปิดการเข้ารหัส

## Returns

`True` หากการเปลี่ยนสำเร็จ นอกนั้นจะเป็น `false` หากเกิดข้อผิดพลาดขึ้นมา

## Throws

`java.io.IOException` หาก `conn` ได้มีการปิดตัวลง

`java.lang.IllegalArgumentException` หาก `conn` ไม่คิดต่อไปยัง `RemoteDevice` หาก `conn` ถูกสร้างขึ้นโดยฝั่งลูกข่ายของการติดต่อโดยใช้แบบแผนอื่นนอกเหนือจาก `btspdp`, `btl2cap` หรือ `btgoep` หาก `conn` การแจ้งเตือนถูกใช้โดยแม่ข่ายเพื่อรอการเชื่อมต่อกับลูกข่าย

## -isAuthenticated

`public boolean isAuthenticated()`

ระบุไว้หาก `RemoteDevice` นี้ได้ถูกตรวจสอบสิทธิ์แล้ว อุปกรณ์อาจได้ถูกตรวจสอบสิทธิ์โดยแอปพลิเคชันหรือแอปพลิเคชันอื่น การตรวจสอบสิทธิ์ได้นำมาใช้กับ ACL link ระหว่างอุปกรณ์และการเชื่อมต่อที่ไม่ได้อยู่ในการเชื่อมต่อแบบ L2CAP, RFCOMM, or OBEX ดังนั้น หาก `authenticate()` ได้แสดงเมื่อการติดต่อสื่อสารแบบ L2CAP ได้ใช้กับอุปกรณ์ A ดังนั้น `isAuthenticated()` อาจคืนค่า `true` เมื่อถูกทดสอบเป็นเสมือนส่วนของการสร้างการเชื่อมต่อแบบ RFCOMM ไปยังอุปกรณ์ A

## Returns

`true` หาก `RemoteDevice` นี้ได้ถูกตรวจสอบสิทธิ์แล้วก่อนหน้านี้ เป็น `false` เมื่อยังไม่ได้ถูกตรวจสอบสิทธิ์ หรือไม่ได้เปิดการเชื่อมต่อระหว่าง local device และ `RemoteDevice`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- isAuthorized**

public boolean **isAuthenticated()**

ระบุไว้หาก RemoteDevice ได้ถูกกำหนดสิทธิ์ก่อนหน้าโดย BCC ของ local device เพื่อที่จะแลกเปลี่ยนข้อมูลขึ้นกับการให้บริการร่วมกับการเชื่อมต่อ ทั้งถูกข่ายและแม่ข่ายสามารถเรียกเมธอดนี้ได้ อย่างไรก็ตามสำหรับลูกขายนั้นเมธอดนี้จะคืนค่า false สำหรับค่าตามกฎทั้งหมดของอาร์กิวเมนต์ conn

**ตัวแปร**

**conn** การเชื่อมต่อที่ RemoteDevice นี้ได้ใช้เพื่อที่จะเข้าไปยังแม่ข่ายหรือการบริการที่ให้บริการ

**Returns**

true หาก conn เป็นการเชื่อมต่อฝั่งแม่ข่ายและ RemoteDevice ได้ถูกกำหนดสิทธิ์ false หาก conn เป็นการเชื่อมต่อฝั่งแม่ข่ายหรือเป็นการเชื่อมต่อฝั่งแม่ข่ายที่ไม่ได้ถูกกำหนดสิทธิ์

**Throws**

java.lang.IllegalArgumentException หาก conn ไม่เป็นการเชื่อมต่อไปยัง RemoteDevice ถ้า conn ไม่ได้ถูกสร้าง โดยใช้แบบแผนดังต่อไปนี้ btsp, bt212cap หรือ btgoep หรือหาก conn มีการแจ้งถูกใช้โดยแม่ข่ายเพื่อรอการเชื่อมต่อจากลูกข่าย

java.io.IOException หาก conn ได้ถูกปิดตัวลง

**- isEncrypted**

public boolean **isEncrypted()**

ระบุไว้หากข้อมูลได้แลกเปลี่ยนกับ RemoteDevice นี้ ได้มีการเข้ารหัส การเข้ารหัสอาจมีการเปิดก่อนหน้าโดยแอปพลิเคชันใดๆ ก็ตาม การเข้ารหัสถูกนำมาใช้กับ ACL link ระหว่างอุปกรณ์และการเชื่อมต่อที่ไม่ได้อยู่ในการเชื่อมต่อแบบ L2CAP, RFCOMM, or OBEX ดังนั้นหาก encrypt() ได้ถูกใช้กับตัวแปร on เซตค่าเป็น true เมื่อการเชื่อมต่อแบบ L2CAP ได้ถูกสร้างขึ้นมายังอุปกรณ์ A ดังนั้น isEncrypt() ก็จะคืนค่ามาเป็น true เมื่อถูกทดสอบเสมือนเป็นส่วนของการสร้างการเชื่อมต่อแบบ RFCOMM บนอุปกรณ์ A

**Returns**

true หากข้อมูลที่ดึงเปลี่ยนกับ RemoteDevice ถูกเข้ารหัส false หากไม่ได้ถูกเข้ารหัส หรือไม่มีการเปิดการเชื่อมต่อระหว่าง local device และ RemoteDevice

**2.3.10 Class UUID**

```
public class UUID extends java.lang.Object
```

ตารางที่ 2.27 รายละเอียดของคอนสตรัคเตอร์ UUID

Constructor Summary	
	UUID(long uuidValue)
	UUID(java.lang.String uuidValue, Boolean shortUUID)

ตารางที่ 2.28 รายละเอียดของเมธอด UUID

Method Summary	
boolean	<b>equals</b> (java.lang.Object value) แสดงไม่ว่าออบเจกต์ใดก็ตามมีค่าเท่ากับออบเจกต์ที่ระบุนี้
int	<b>hashCode</b> () คืนค่า hash code สำหรับออบเจกต์นี้
java.lang.String	<b>toString</b> () คืนค่าสตริงที่เป็นตัวแทนของออบเจกต์นี้

**2.3.10.1 รายละเอียดของ Construtor****-UUID**

```
public UUID(long uuidValue)
```

**-UUID**

```
public UUID(java.lang.String uuidValue, boolean shortUUID)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.10.2. รายละเอียดของ Method

- toString

```
public java.lang.String toString()
```

คืนค่าสตริงที่เป็นตัวแทนของออบเจกต์นี้ โดยทั่วไปแล้วเมธอด toString นี้จะคืนค่าสตริงที่เป็นตัวแทนข้อความเดิม ผลที่ได้ควรที่จะสั้นและได้ใจความง่ายต่อการอ่าน

#### Overrides

toString ในคลาสของ java.lang.Object

#### Returns

ค่าสตริงที่เป็นตัวแทนของออบเจกต์นี้

- equals

```
public boolean equals(java.lang.Object value)
```

แสดงไม่ว่าออบเจกต์ใดก็ตามมีค่าเท่ากับออบเจกต์ที่ระบุนี้

#### Overrides

equals อยู่ในคลาส java.lang.Object

ตัวแปร

value เป็นออบเจกต์อ้างอิงซึ่งถูกนำมาเปรียบเทียบ

#### Returns

true หากออบเจกต์นี้มีค่าเหมือนกับอาร์กิวเมนต์ obj เป็น false สำหรับกรณีอื่นๆ

- hashCode

```
public int hashCode()
```

คืนค่า hash code สำหรับออบเจกต์นี้ เมธอดได้รับการสนับสนุนสำหรับข้อดีของ hashtables เช่นที่ ได้แสดง โดย java.util.Hashtable

#### Overrides

hashCode อยู่ในคลาสของ java.lang.Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Returns

ค่า hash code สำหรับ Object นี้

### 2.3.11 BluetoothConnectionException

public class **BluetoothConnectionException** extends java.io.IOException

คลาสนี้จะถูกเรียกเมื่อการเชื่อมต่อบลูทูธไม่สำเร็จ field ต่างๆ ในคลาสนี้แสดงสาเหตุของการผิดพลาด

ตารางที่ 2.29 รายละเอียดของฟิลด์ BluetoothConnectionException

Field Summary	
static int	FAILED_NOINFO แสดงว่าการเชื่อมต่อกับเซิร์ฟเวอร์ล้มเหลวอย่างไม่ทราบสาเหตุ
static int	NO_RESOURCES แสดงว่าการเชื่อมต่อล้มเหลวเนื่องจากความบกพร่องของแหล่งที่มา
static int	SECURITY_BLOCK แสดงว่าการเชื่อมต่อล้มเหลวเนื่องจากการตั้งค่าความปลอดภัยของอุปกรณ์ไม่ตรงกับการร้องขอ
static int	TIMEOUT แสดงว่าการเชื่อมต่อล้มเหลวเนื่องจากหมดเวลา
static int	UNACCEPTABLE_PARAMS แสดงว่าการเชื่อมต่อล้มเหลวเนื่องจากโครงสร้างพารามิเตอร์ไม่รองรับกับอุปกรณ์
static int	UNKNOWN_PSM แสดงว่าการเชื่อมต่อกับเซิร์ฟเวอร์ล้มเหลวเนื่องจากไม่มีบริการสำหรับ PSM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.30 รายละเอียดของคอนสตรักเตอร์ BluetoothConnectionException

Constructor Summary	
BluetoothConnectionException(int error)	สร้างออบเจกต์ BluetoothConnectionException ร่วมกับตัวบอกความผิดพลาด
BluetoothConnectionException(int error, java.lang.String msg)	สร้างออบเจกต์ BluetoothConnectionException ร่วมกับตัวบอกความผิดพลาด และข้อความเฉพาะเจาะจง

ตารางที่ 2.31 รายละเอียดของเมธอด BluetoothConnectionException

Method Summary	
int	getStatus() จะได้ค่าสถานะในคอนสตรักเตอร์ซึ่งจะแสดงผลของการผิดพลาด

### 2.3.11.1 รายละเอียดของ Field

#### - UNKNOWN\_PSM

public static final int UNKNOWN\_PSM

แสดงว่าการเชื่อมต่อกับเซิร์ฟเวอร์ล้มเหลวเนื่องจากไม่มีบริการสำหรับ PSM

ค่าสำหรับ UNKNOWN\_PSM มีค่าเท่ากับ 0x0001 (1)

#### - SECURITY\_BLOCK

public static final int SECURITY\_BLOCK

แสดงว่าการเชื่อมต่อล้มเหลวเนื่องจากการตั้งค่าความปลอดภัยของอุปกรณ์ไม่ตรงกับการร้องขอ

ค่าสำหรับ SECURITY\_BLOCK มีค่าเท่ากับ 0x0002 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### - NO\_RESOURCES

public static final int NO\_RESOURCES

แสดงว่าการเชื่อมต่อล้มเหลวเนื่องจากความบกพร่องของแหล่งที่มา

ค่าสำหรับ NO\_RESOURCES มีค่าเท่ากับ 0x0003 (3)

#### - FAILED\_NOINFO

public static final int FAILED\_NOINFO

แสดงว่าการเชื่อมต่อกับเซิร์ฟเวอร์ล้มเหลวโดยไม่ทราบสาเหตุ

ค่าสำหรับ FAILED\_NOINFO มีค่าเท่ากับ 0x0004 (4)

#### - TIMEOUT

public static final int TIMEOUT

แสดงว่าการเชื่อมต่อล้มเหลวเนื่องจากหมดเวลา

ค่าสำหรับ TIMEOUT มีค่าเท่ากับ 0x0005 (5)

#### - UNACCEPTABLE\_PARAMS

public static final int UNACCEPTABLE\_PARAMS

แสดงว่าการเชื่อมต่อล้มเหลวเนื่องจากโครงสร้างพารามิเตอร์ไม่รองรับกับอุปกรณ์

ค่าสำหรับ UNACCEPTABLE\_PARAMS มีค่าเท่ากับ 0x0006 (6)

### 2.3.11.2 รายละเอียดของ Constructor Detail

#### - BluetoothConnectionException

public BluetoothConnectionException(int error)

สร้างออบเจกต์ BluetoothConnectionException ร่วมกับตัวบอกความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตัวแปร**

**error** บอกลักษณะของความผิดพลาด ซึ่งจะมีค่าได้แค่แบบเดียวจากที่ระบุไว้ในคลาสนี้

**Throws**

java.lang.IllegalArgumentException ถ้าค่าที่รับมาไม่ตรงกับค่าที่ระบุไว้ในคลาสนี้

**- BluetoothConnectionException**

```
public BluetoothConnectionException(int error, java.lang.String msg)
```

สร้างออบเจกต์ BluetoothConnectionException ร่วมกับตัวบอกความผิดพลาด และข้อความเฉพาะเจาะจง

**ตัวแปร**

**error** บอกลักษณะของความผิดพลาด ซึ่งจะมีค่าได้แค่แบบเดียวจากที่ระบุไว้ในคลาสนี้

**msg** อธิบายรายละเอียดของความผิดพลาดซึ่งอาจมีค่าเป็น null

**Throws**

java.lang.IllegalArgumentException ถ้าค่าที่รับมาไม่ตรงกับค่าที่ระบุไว้ในคลาสนี้

**2.3.11.3 รายละเอียดของ Method****- getStatus**

```
public int getStatus()
```

จะได้ค่าสถานะในคอนสตรัคเตอร์ซึ่งจะแสดงผลของการผิดพลาด

**Returns**

สาเหตุของความผิดพลาด ซึ่งอาจจะเป็นแบบใดแบบหนึ่งจากที่ระบุไว้ในคลาสนี้

**2.3.12 Class BluetoothStateException**

```
public class BluetoothStateException extends java.io.IOException
```

คลาสนี้จะถูกเรียกใช้เมื่อมีการร้องขอเกิดขึ้นกับระบบบลูทูธซึ่งระบบไม่สามารถรองรับการร้องขอในขณะนั้นได้

## ตารางที่ 2.32 รายละเอียดของคอนสตรัคเตอร์ BluetoothStateException

Constructor Summary
BluetoothStateException() สร้างออบเจกต์ BluetoothStateException โดยไม่มีข้อความบอกรายละเอียด
BluetoothStateException(java.lang.String msg) สร้างออบเจกต์ BluetoothStateException โดยมีข้อความเจาะจงรายละเอียด

### 2.3.12.1 รายละเอียดของ Constructor

#### - BluetoothStateException

public BluetoothStateException()

สร้างออบเจกต์ BluetoothStateException โดยไม่มีข้อความบอกรายละเอียด

#### - BluetoothStateException

public BluetoothStateException(java.lang.String msg)

สร้างออบเจกต์ BluetoothStateException โดยมีข้อความเจาะจงรายละเอียด

#### ตัวแปร

msg เหตุผลของการเกิดความผิดพลาด

### 2.3.13 Class ServiceRegistrationException

public class ServiceRegistrationException extends java.io.IOException

คลาสนี้จะถูกเรียกเมื่อเกิดความผิดพลาดในการจัดการข้อมูลใน local Service Discovery Database (SDDb) ความผิดพลาดอาจเกิดจากการที่ไม่มีที่สำหรับใส่ข้อมูลใหม่ หรือการแก้ไขข้อมูลอย่างผิดกฎ

### ตารางที่ 2.33 รายละเอียดของคอนสตรัคเตอร์ ServiceRegistrationException

Constructor Summary
ServiceRegistrationException() สร้างออบเจกต์ ServiceRegistrationException โดยไม่มีข้อความบอกรายละเอียด
ServiceRegistrationException(java.lang.String msg) สร้างออบเจกต์ ServiceRegistrationException โดยมีข้อความเจาะจงรายละเอียด

#### 2.3.13.1 รายละเอียดของ Constructor

##### - ServiceRegistrationException

public ServiceRegistrationException()

สร้างออบเจกต์ ServiceRegistrationException โดยไม่มีข้อความบอกรายละเอียด

##### - ServiceRegistrationException

public ServiceRegistrationException(java.lang.String msg)

สร้างออบเจกต์ ServiceRegistrationException โดยมีข้อความเจาะจงรายละเอียด

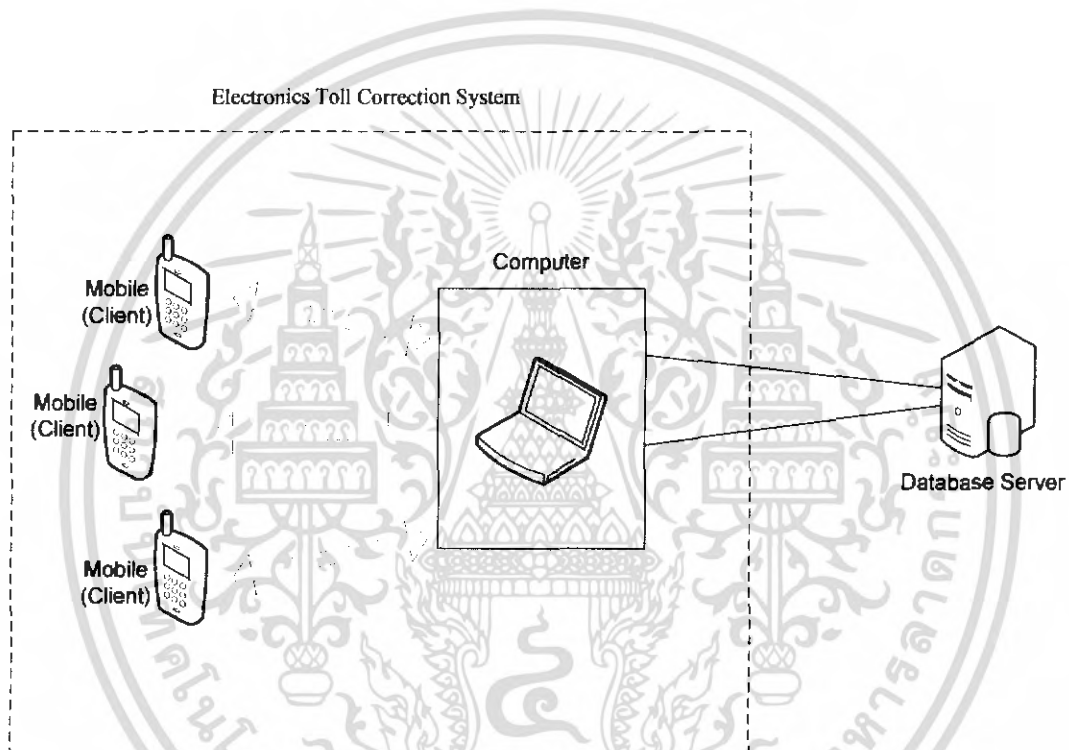
##### ตัวแปร

msg เหตุผลของการเกิดความผิดพลาด

### บทที่ 3

#### การออกแบบและพัฒนา

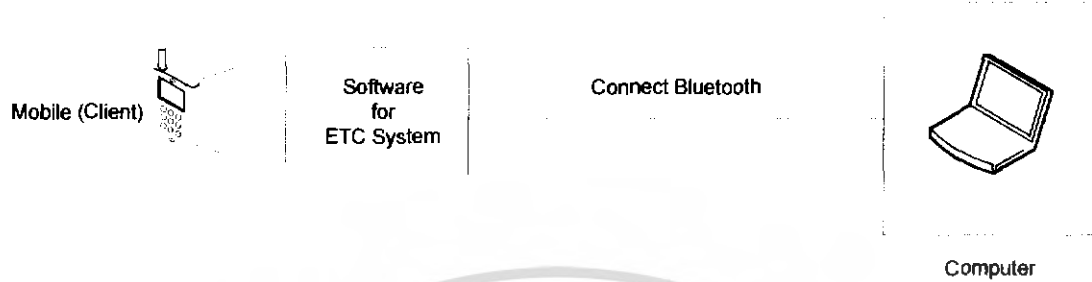
สำหรับการออกแบบและพัฒนาโครงการนี้แบ่งออกเป็น 2 ส่วนด้วยกันคือ ส่วนทางในฝั่งของ Server จะประกอบไปด้วย Database และส่วนทางในฝั่ง Client จะประกอบด้วย Application บนโทรศัพท์เคลื่อนที่



รูปที่ 3.1 ภาพรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1 การออกแบบและพัฒนาทางฝั่ง Client

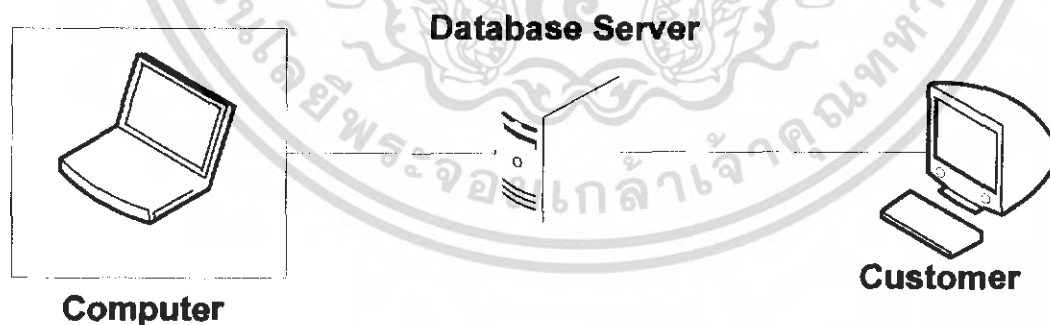


รูปที่ 3.2 การออกแบบและพัฒนาทางฝั่ง Client

ในส่วนนี้จะทำการสร้างซอฟต์แวร์ขึ้นมา เพื่อนำมาพัฒนาใช้บนโทรศัพท์เคลื่อนที่ การทำงานของซอฟต์แวร์ จะทำการขอใช้บริการ โดยใช้เทคโนโลยีบลูทูธเป็นตัวนำส่งข้อมูล โดยจะส่งไปยังคอมพิวเตอร์ เพื่อไปส่งหมายเลขโทรศัพท์ไปยังของบลูทูธไปยัง server เพื่อบันทึกการใช้งาน

### 3.2. การออกแบบและพัฒนาทางด้าน Server

สำหรับในส่วนการออกแบบและพัฒนาทางฝั่ง Server จะประกอบไปด้วย 2 ส่วนคือ



รูปที่ 3.3 การออกแบบและพัฒนาทางฝั่ง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 คอมพิวเตอร์ (Computer)

ในส่วนนี้จะทำการรับข้อมูลจากโทรศัพท์เคลื่อนที่ และทำการบันทึกไปยัง Database Server เพื่อเก็บข้อมูลการใช้ของสมาชิก โดยข้อมูลที่ส่งมาจะเป็น หมายเลขโทรศัพท์ของผู้ใช้

### 3.2.2 Database Server

ในส่วนนี้จะทำการรับข้อมูลต่อมาจากคอมพิวเตอร์เพื่อทำการเก็บข้อมูลที่ส่งลงในฐานข้อมูล โดยข้อมูลที่เก็บนี้เป็นการบันทึกจำนวนครั้งการใช้บริการของผู้ใช้

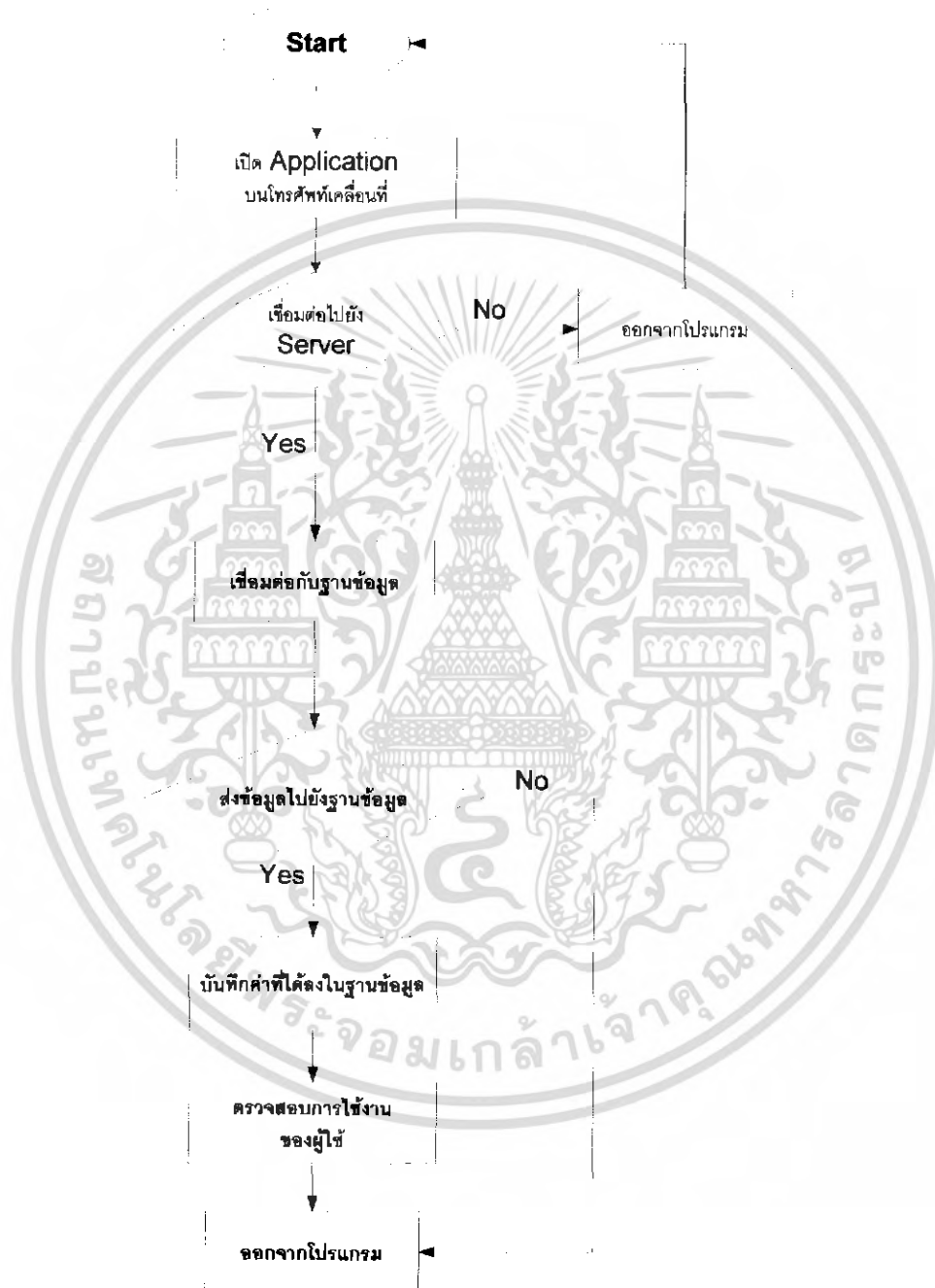
### 3.2.3 Customer

ในส่วนนี้ผู้ใช้สามารถตรวจสอบการใช้งานของตนเองได้ การทำงานของส่วนนี้จะดึงนำเอาข้อมูลที่ถูกบันทึกไว้ในฐานข้อมูลมาใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 โฟลว์ชาร์ต (Flow Chart) การทำงานของระบบชำระค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธ



รูปที่ 3.4 โฟลว์ชาร์ต (Flow Chart) การทำงานของระบบชำระค่าผ่านทาง โดยใช้เทคโนโลยีบลูทูธ

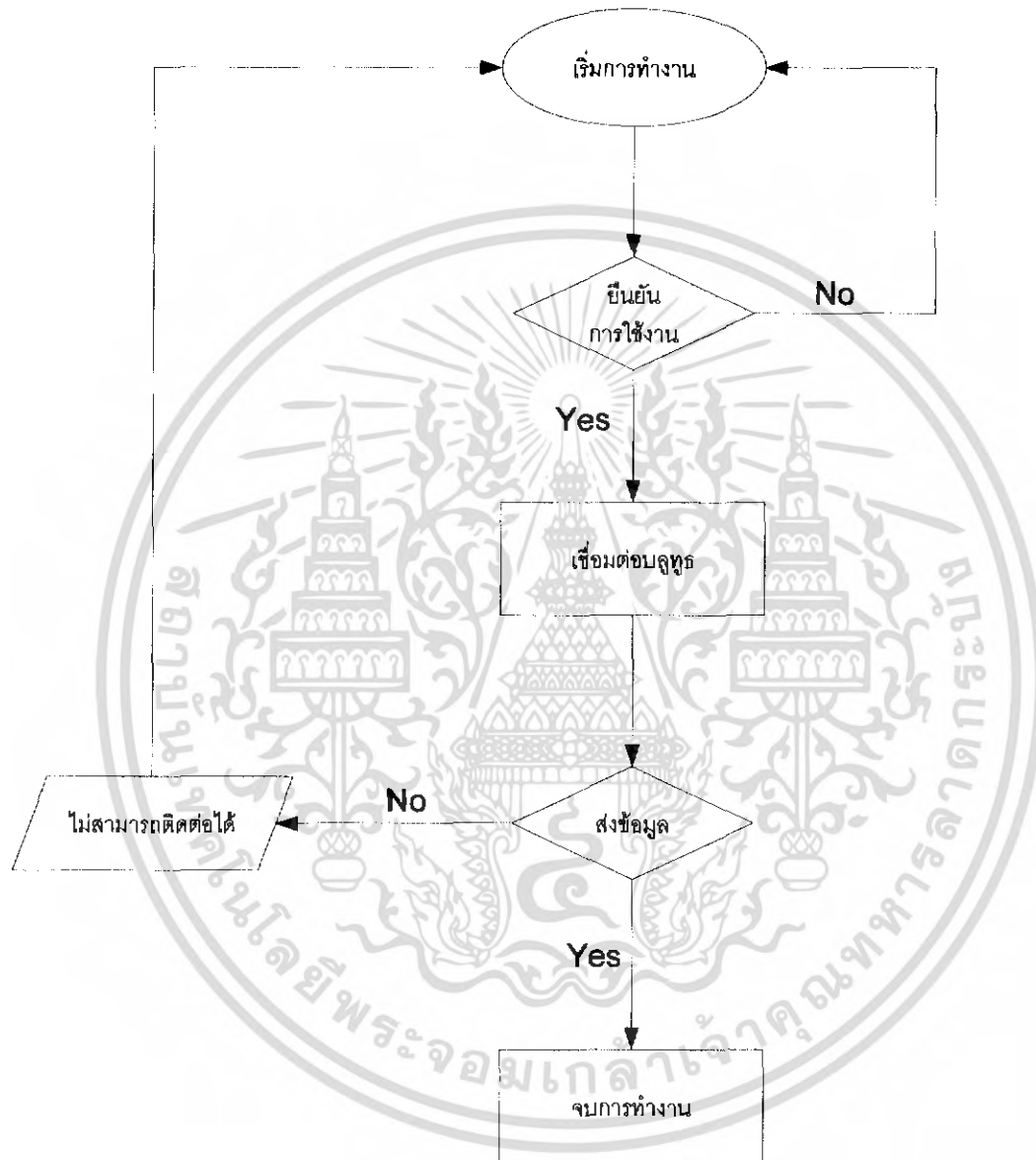
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การทำงานของโฟลว์ชาร์ต (Flow Chart) ของระบบชำระค่าผ่านทางโดยใช้เทคโนโลยีบลูทูธ สามารถอธิบายได้ดังนี้**

- เริ่มต้นด้วยการเปิดแอปพลิเคชันจากโทรศัพท์เคลื่อนที่
- จากนั้นจะเป็นการยืนยันการใช้งาน โดยหากต้องการเชื่อมต่อไปยัง Server ระบบจะทำการเชื่อมต่อไปยัง Server โดยใช้เทคโนโลยีบลูทูธ หากไม่ต้องการเชื่อมต่อไปยัง Server ระบบจะทำการออกจากแอปพลิเคชัน
- ทำการเชื่อมต่อไปยังฐานข้อมูล
- ทำการส่งข้อมูลไปยังฐานข้อมูล หากสามารถติดต่อไปยังฐานข้อมูลได้จะทำการบันทึกค่าลงไปฐานข้อมูล หากไม่สามารถเชื่อมต่อได้จะทำการหยุดการทำงานและออกจากแอปพลิเคชัน
- เมื่อบันทึกค่าลงในฐานข้อมูลแล้วจะทำการตรวจสอบการใช้งานของผู้ใช้
- หลังจากนั้นระบบจะสิ้นสุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 โฟลว์ชาร์ต (Flow Chart) แสดงการทำงานที่ฝั่ง Client



รูปที่ 3.5 โฟลว์ชาร์ต (Flow Chart) แสดงการทำงานที่ฝั่ง Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การทำงานของโฟลว์ชาร์ต (Flow Chart) ของการทำงานที่ฝั่ง Client สามารถอธิบายได้ดังนี้

- เริ่มต้นด้วยการเข้าโปรแกรมเพื่อยืนยันการใช้งานหากต้องการใช้บริการจะดำเนินการต่อ หากไม่ต้องการจะทำการออกจากโปรแกรม
- ทำการเชื่อมต่อไปยังคอมพิวเตอร์โดยใช้เทคโนโลยีบลูทูธ
- เมื่อเชื่อมต่อได้แล้วจะทำการส่งข้อมูลไปยังคอมพิวเตอร์หากต้องการส่ง ระบบจะดำเนินการต่อ หากไม่ต้องการส่งจะทำการออกจากโปรแกรม
- หลังจากนั้นระบบจะสิ้นสุดการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 โฟลว์ชาร์ต (Flow Chart) การทำงานที่ฝั่ง Server



รูปที่ 3.6 โฟลว์ชาร์ตแสดงการทำงานที่ฝั่ง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การทำงานของโฟลว์ชาร์ต (Flow Chart) ของการทำงานที่ฝั่ง Server สามารถอธิบายได้ดังนี้**

- ทำการรับค่าจากโทรศัพท์เคลื่อนที่
- จากนั้นจะทำการติดต่อไปยังฐานข้อมูล
- จากการติดต่อไปยังฐานข้อมูลนี้ เราจะบันทึกการใช้งานของผู้ใช้ลงในระบบฐานข้อมูล
- ตรวจสอบการใช้งานของผู้ใช้
- หลังจากนั้นระบบจะสิ้นสุดการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 ทดลองสร้าง Mobile Application

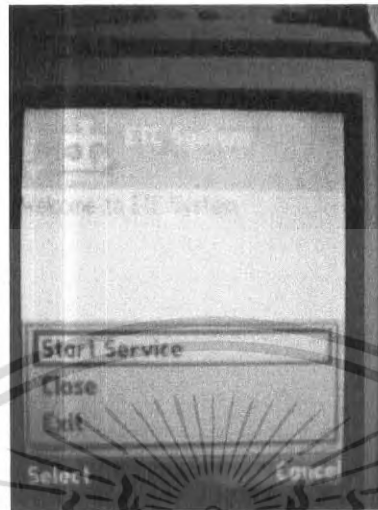
##### 4.1.1 เปิด Application ขึ้นมาเพื่อทำการติดต่อไปยังบริการ



รูปที่ 4.1 หน้าจอโทรศัพท์เคลื่อนที่เมื่อเข้ามาถึงหน้าแรกของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 เมื่อเข้ามายังแอปพลิเคชันแล้วทำการกดปุ่ม Start Service เพื่อทำการเริ่มใช้บริการ



รูปที่ 4.2 กด Start Service เพื่อทำการเริ่มใช้บริการของแอปพลิเคชัน

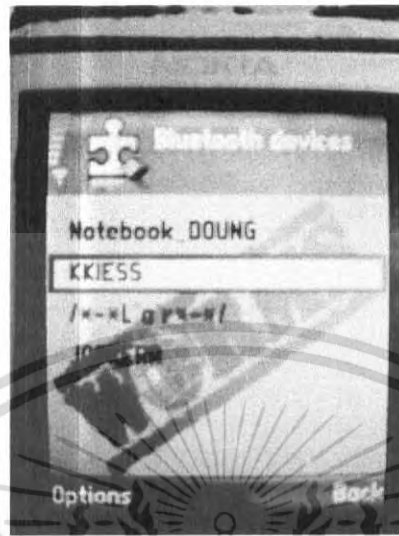
4.1.3 หลังจากเลือก Start Service แล้วโทรศัพท์เคลื่อนที่ จะทำการค้นหาการให้บริการและอุปกรณ์บลูทูธในระยะใกล้เคียง



รูปที่ 4.3 ทำการค้นหาการให้บริการและอุปกรณ์บลูทูธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.4 ทำการเลือกอุปกรณ์หรือการให้บริการบลูทูธจากรายชื่อที่แสดงในโทรศัพท์เคลื่อนที่



รูปที่ 4.4 ทำการแสดงรายชื่อบริการและอุปกรณ์บลูทูธที่ค้นเจอ ทำการเลือกจากรายชื่อที่แสดง

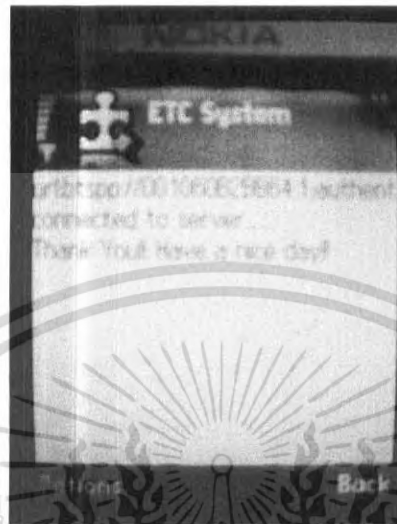
#### 4.1.5 ทำการถามผู้ใช้งานว่าการเปิดบลูทูธจากโทรศัพท์เคลื่อนที่เพื่อเชื่อมต่อไปยังผู้ให้บริการหรือไม่



รูปที่ 4.5 ทำการเชื่อมต่อไปยังผู้ให้บริการโดยใช้เทคโนโลยีบลูทูธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 ทำการแสดงสถานะการติดต่อระหว่างผู้ใช้บริการและให้ผู้บริการและแสดงข้อความ  
 “Thank You!! Have a nice day!!” เพื่อแสดงถึงการสิ้นสุดการเชื่อมต่อ



รูปที่ 4.6 ทำการส่งค่าหมายเลขโทรศัพท์เข้าไปยังเซิร์ฟเวอร์

4.1.7 กดปุ่ม Exit เพื่อทำการออกจากแอปพลิเคชัน



รูปที่ 4.7 กดปุ่ม Exit เพื่อออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.8 ที่หน้าจอฝั่งเซิร์ฟเวอร์จะแสดงการเปิดการเชื่อมต่อระหว่างผู้ให้บริการและผู้ใช้โทรศัพท์เคลื่อนที่ ดังแสดงที่รูป 4.8 และสำหรับในรูปที่ 4.9 เป็นการตอบรับเมื่อมีการติดต่อจากผู้ใช้บริการของเซิร์ฟเวอร์

```
[0]Start Serial Server
[1]Run Serial Client
[2]Exit
3
[0]Start Serial Server
[1]Run Serial Client
[2]Exit
server url: htsp://localhost:102030405060708090a0b0c0d0e0f010;name=SPPServerExample
[WARNING] You are using BlueCove Connector [WARNING]
Print Service Record (# of element: 5)
Print Service Record URL htsp://070341802852:1;authenticate=false;encrypt=false
DataElement[ServiceAvailability] 255
DataElement[ProtocolDescriptorList] 48 (# of element: 2)
  DataElement[ProtocolDescriptorList] 48 (# of element: 1)
    DataElement[ProtocolDescriptorList] L2CAP
    DataElement[ProtocolDescriptorList] 48 (# of element: 2)
      DataElement[ProtocolDescriptorList] RFCOMM
      DataElement[ProtocolDescriptorList] 1
DataElement[ServiceName] SPPServerExample
DataElement[ServiceClassIDList] 48 (# of element: 1)
  DataElement[ServiceClassIDList] 102030405060708090a0b0c0d0e0f010
DataElement[ServiceRecordHandle] 65568
local service waiting for client connection...
```

รูปที่ 4.8 หน้าจอของเซิร์ฟเวอร์เมื่อทำการรับค่าจากโทรศัพท์เคลื่อนที่

```
DataElement[ServiceAvailability] 255
DataElement[ProtocolDescriptorList] 48 (# of element: 2)
  DataElement[ProtocolDescriptorList] 48 (# of element: 1)
    DataElement[ProtocolDescriptorList] L2CAP
    DataElement[ProtocolDescriptorList] 48 (# of element: 2)
      DataElement[ProtocolDescriptorList] RFCOMM
      DataElement[ProtocolDescriptorList] 1
DataElement[ServiceName] SPPServerExample
DataElement[ServiceClassIDList] 48 (# of element: 1)
  DataElement[ServiceClassIDList] 102030405060708090a0b0c0d0e0f010
DataElement[ServiceRecordHandle] 65568
local service waiting for client connection...
accepted a client connection, reading it...
read in data '0059100600' from client
echo '0059100600' back to client
local service waiting for client connection...
```

รูปที่ 4.9 เป็นการรับข้อมูลจากผู้ใช้บริการเมื่อมีผู้ใช้ติดต่อขอให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2. ทดลองสร้างตารางแสดงฐานข้อมูลของระบบ ETC

4.2.1 สำหรับที่การทำงานที่ระบบ ETC ผู้ใช้บริการสามารถตรวจสอบการใช้งานของแต่ละบุคคลได้ โดยได้มีการจัดทำฐานข้อมูลของผู้ใช้บริการไว้ โดยเก็บรายละเอียดต่างๆ ของผู้บริบริการดังแสดงในรูปที่ 4.10

	CusID	Name	Mobile	IDNo	Address	Telephone	Date	Time	Station	QTY
<input type="checkbox"/>	45012210	สิริณี สัมแดง สงวน	0855545124	1- 3690018- 57-1	480/93 ซ. เพชรศรี 18 ก. เพชรศรี รามารั กรมท่า ...	022535355	2007-02-01	00:00:16	Rama 9	1
<input type="checkbox"/>	46012205	สิริญา ปิยญา ไว	0859188600	1-1012- 00024-88- 6	1293/31 ซ.วชิร รรมชาติ 57 แขวงบางจาก โรตารีไฮวง ก...	027461628	2007-02-04	00:00:17	Rama9	2
<input type="checkbox"/>	45012214	ลาภรณ์ โกลิอิน	086154321	1-455323- 32-2	ชั้น 9 ห้องRobot	023452311	2007-02-01	00:00:14	Rama 9	1
<input type="checkbox"/>	45012204	วิมลญา วิมลภา นาง	0863144769	1-345932- 21-1	12/3 บ. 2 ส.โบ เมือง ล.เมือง จ. ชัยภูมิ	021234988	2007-02-01	00:00:12	Rama9	1

รูปที่ 4.10 ตารางระบบฐานข้อมูลของระบบ ETC

4.2.2 สามารถค้นหาข้อมูลการใช้งานของผู้ใช้บริการได้โดยใส่หมายเลขโทรศัพท์เคลื่อนที่สำหรับการค้นหาข้อมูล หลังจากนั้นจะทำการแสดงข้อมูลการใช้งานของผู้ใช้บริการบนหน้าจอหลัก

CusID	Name	Mobile	Date	Time	Station	QTY
▶ 46012205	สิริญา ปิยญาไว	0859188600	2/4/2007	17:00:00	Rama9	2

ค้นหาจากหมายเลขโทรศัพท์ 0859188600

ค้นหา      ข้อมูลเก่า

รูปที่ 4.11 หน้าหลักของฐานข้อมูลการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 สำหรับหน้าข้อมูลผู้ใช้บริการ ผู้ให้บริการสามารถทำการค้นหาโดยพิมพ์หมายเลขสมาชิกและกดปุ่มค้นหา หลังจากนั้นจะแสดงข้อมูลต่างๆ ของลูกค้า โดยทางผู้ให้บริการสามารถเพิ่มเติม แก้ไข หรือลบข้อมูลต่างๆ ได้ และกดปุ่มบันทึกเพื่อบันทึกข้อมูลที่ได้เมื่อมีการเปลี่ยนแปลง ดังแสดงในรูปที่ 4.13 หรือคลิกยกเลิกเมื่อไม่ต้องการเปลี่ยนแปลงข้อมูลใดๆ

Customer

ข้อมูลลูกค้า

ค้นหา

หมายเลขสมาชิก 46012210

ชื่อ-นามสกุล ตรีชัย ชื่นแฉะสรณ

เบอร์โทรศัพท์มือถือ 0866645124

เบอร์โทรศัพท์บ้าน 022535356

หมายเลขบัญชีธนาคาร 1-3690018-57-1

ที่อยู่ 480/93 ซ.เพชรบุรี 18 ต.เพชรบุรี อําเภอศรีกรุง เขตฯ 10400

เพิ่ม แก้ไข ยกเลิก ลบ บันทึก

รูปที่ 4.12 หน้าจอการเรียกดูข้อมูลลูกค้า

ข้อมูลลูกค้า

ค้นหา

หมายเลขสมาชิก 46012210

ชื่อ-นามสกุล ตรีชัย ชื่นแฉะสรณ

เบอร์โทรศัพท์มือถือ 0866645124

เบอร์โทรศัพท์บ้าน 022535356

หมายเลขบัญชีธนาคาร 1-3690018-57-1

ที่อยู่ 480/93 ซ.เพชรบุรี 18 ต.เพชรบุรี อําเภอศรีกรุง เขตฯ 10400

บันทึก

เพิ่ม แก้ไข ยกเลิก ลบ บันทึก

รูปที่ 4.13 หน้าจอเมื่อมีการทำการเพิ่มหรือแก้ไขข้อมูล แล้วทำการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดลองและปัญหาที่พบ

ในส่วนนี้จะกล่าวถึงข้อสรุป, ปัญหาที่พบในการทดลอง และแนวทางแก้ไขปัญหาที่เกิดขึ้นในการทดลอง

#### 5.1 สรุปผลการทดลอง

ในการทดลองนี้ได้ศึกษาเกี่ยวกับเกี่ยวกับเทคโนโลยีบล็อกเชน ซึ่งเป็นเทคโนโลยีใหม่ที่มาพร้อมกับอุปกรณ์สื่อสาร โดยในการทดลองนี้ได้ใช้โทรศัพท์เคลื่อนที่ที่ได้นำเทคโนโลยีบล็อกเชนเข้ามาด้วย โดยจะนำข้อได้เปรียบของเทคโนโลยีบล็อกเชนมาใช้ศึกษาและทดลอง

จากการทดลองนี้สามารถสรุปได้ว่า โครงการนี้จะทำการชำระค่าผ่านทางโดยใช้เทคโนโลยีบล็อกเชนในโทรศัพท์เคลื่อนที่เป็นตัวส่งข้อมูลไปยังเครื่องคอมพิวเตอร์ โดยในส่วนของโปรแกรมอาจจะยังไม่ได้มีความละเอียดมากนัก เพราะอาจต้องมีการแก้ไขในภายหลัง แต่สามารถเข้าใจถึงรายละเอียดในส่วนของการ Application ที่ต้องการ และสามารถกำหนดความสามารถหลักๆ ได้ นอกจากนั้นแล้วในการส่งข้อมูลไปยังเซิร์ฟเวอร์ได้ทำการศึกษาการจัดทำระบบฐานข้อมูลเพื่อใช้สำหรับเก็บข้อมูลของผู้ใช้ได้โดยในส่วนนี้ และยังได้มีการทำอินเตอร์เฟซสำหรับตรวจสอบการใช้งาน และแสดงข้อมูลต่างๆ ของผู้ใช้บริการ โดยการจัดทำโครงการนี้ก็เพื่อเป็นอีกทางเลือกหนึ่งสำหรับการชำระค่าผ่านทาง ซึ่งอาจจะยังไม่สมบูรณ์มากนัก แต่สามารถนำไปพัฒนาต่อไปในภายหน้าได้

#### 5.2 ปัญหาที่พบในการทดลอง

5.1.1 ในส่วนของโปรแกรม User Interface ในตอนแรกไม่สามารถติดต่อไปแต่ละหน้าได้ เพราะเนื่องจากไม่เข้าใจในโปรแกรมบางส่วน

5.1.2 ในตอนแรกไม่สามารถประมวลผลทั้งบนโทรศัพท์ได้ โดยบางหน้าของโปรแกรมไม่สามารถแสดงผลได้

5.1.3 ในการเชื่อมต่อบล็อกเชน มีปัญหาในเรื่องไม่สามารถประมวลผลได้ เนื่องจากไม่เข้าใจในส่วนของโปรแกรมหรืออาจพิมพ์ โปรแกรมผิดพลาด

5.1.4 สำหรับการเขียนโปรแกรมในโทรศัพท์เคลื่อนที่โดยการใช้ Java ไม่สามารถหาเอกสารอ้างอิงที่อธิบายเฉพาะส่วนของเทคโนโลยีบล็อกเชนได้เท่าที่ควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.5 ในปัจจุบัน อุปกรณ์ลูทอซยังไม่มีคุณสมบัติ บางครั้งจึงสามารถติดต่อได้และในบางครั้งก็ไม่สามารถติดต่อได้

5.1.6 สำหรับโทรศัพท์เคลื่อนที่บางรุ่นไม่รองรับ CLDC 1.1 จึงไม่สามารถนำโปรแกรมที่พัฒนาขึ้นไปใช้ได้

5.1.7 ในการประมวลผลโปรแกรมไม่สามารถทำบนอิมูเลเตอร์ได้ เนื่องจากไม่สามารถเห็นผลที่เกิดขึ้นได้ จึงต้องทำการประมวลลงบนโทรศัพท์เคลื่อนที่ แล้วดูผลที่เกิดขึ้นจริง

### 5.3 แนวทางการแก้ไขปัญหา

5.2.1 ลองแก้ไขในส่วนของโปรแกรมอยู่หลายครั้ง เพราะบางครั้งอาจพิมพ์คำสั่งไม่ถูกต้องหรืออาจพิมพ์คำสั่งไม่ครบ และลองประมวลผลใหม่อยู่เรื่อยๆ

5.2.2 ศึกษารายละเอียดคำสั่งในแต่ละส่วนให้มากขึ้น

5.2.3 หาเอกสารอ้างอิงที่มีเนื้อหาเกี่ยวกับเรื่องนี้ให้ได้มากที่สุด

## บรรณานุกรม

- [1] กาญจนา ตันวิสุทธิ. 2545. เขียนเกมและโปรแกรมบนมือถือ J2ME. กรุงเทพฯ : อินโฟเพรส.
- [2] สัตยuth์ สว่างวรรณ. 2547. เครือข่ายคอมพิวเตอร์ COMPUTER NETWORK. พิมพ์ครั้งที่ 1. กรุงเทพฯ : เพียร์สัน เอ็ดดูเคชั่น อินโคไรน่า จำกัด.
- [3] “Java APIs for Bluetooth” [Online]. Available : <http://www.java.sun.com>
- [4] “Sun Java Wireless Toolkit 2.5” [Online]. Available : <http://www.java.sun.com>
- [5] “Bluetooth” [Online]. Available : <http://www.bluetooth.com>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้