

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบบริหารจัดการปัญหาด้วยแอนดรอยด์

ISSUE MANAGEMENT USING ANDROID



T131402

โดย

ณัฐทิศา สินชัย

NUTTITA SINCHAI

อาจารย์ที่ปรึกษา

รศ.ดร. พรฤดี เนติโสภาค

เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

826328

2555

131402

2 ส.ย. 2557

b. 12609638
i.

รายงานนี้เป็นส่วนหนึ่งของวิชาการศึกษาระดับ 2

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISSUE MANAGEMENT USING ANDROID



**A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS OF THE COURSE**

INDEPENDENT STUDY 2

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2013

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	ระบบบริหารจัดการปัญหาด้วยแอนดรอยด์
นักศึกษา	นางสาวณัฐชิตา สีนชัย
รหัสนักศึกษา	54660783
ปริญญา	วิทยาศาสตร์มหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	การจัดการเทคโนโลยีสารสนเทศ
ปีการศึกษา	2555
อาจารย์ที่ปรึกษา	รศ.ดร. พรฤดี เนติโสภาคกุล

บทคัดย่อ

การออกแบบและพัฒนาระบบบริหารจัดการปัญหาด้วยแอนดรอยด์ มีวัตถุประสงค์ เพื่อเพิ่มประสิทธิภาพในการบริหารงานบริการหลังการขาย เกิดความเข้าใจที่ตรงกันระหว่างผู้ใช้นักวิเคราะห์ และนักพัฒนา อีกทั้งมุ่งเน้นอำนวยความสะดวกต่อผู้ใช้งานระบบให้สามารถทำงานได้ทุกที่ผ่านอุปกรณ์พกพาประเภทโทรศัพท์มือถือหรือแท็บเล็ต

สำหรับแนวทางในการศึกษา เริ่มจากการวิเคราะห์ปัญหาและข้อจำกัดที่เกิดขึ้นจากระบบงานปัจจุบัน วิเคราะห์ความต้องการของผู้ใช้งาน รวมทั้งศึกษาความเป็นไปได้ในการพัฒนาระบบใหม่ ซึ่งเป็นการนำเทคโนโลยีโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่หรือแท็บเล็ต และเทคโนโลยีโปรแกรมประยุกต์เชิงเว็บ มาใช้งานร่วมกับฐานข้อมูล โดยระบบนี้จะประกอบด้วย การมอบหมายงานให้แก่โปรแกรมเมอร์ และทีมบริการหลังการขาย การแจ้งเตือนผู้ประสานงาน โครงการเมื่อถึงกำหนดติดตามความคืบหน้า การค้นหาและศึกษาความขัดข้องของระบบที่เคยเกิดขึ้น การบันทึกวิธีแก้ไขและอัปเดตสถานะความขัดข้องของระบบ อันจะส่งผลให้เกิดประโยชน์ในการลดขั้นตอนการทำงานและลดระยะเวลาในการทำงาน เพิ่มความสะดวกรวดเร็วในการติดตามความคืบหน้า การแก้ไขปัญหาของระบบหลังการขายและสามารถปฏิบัติงานตรงตามความต้องการของลูกค้า นำมาซึ่งความพึงพอใจแก่ลูกค้า และภาพลักษณ์ที่ดีต่อองค์กร

Title	Issue Management By Android
Student	Miss Nuttita Sinchai
Student ID.	54660783
Degree	Master of Science
Program	Information Technology
Major	Information Technology Management
Academic Year	2012
Advisor	Assoc. Prof. Dr. Ponrudee Netisopakul

ABSTRACT

The purpose of designing and developing an issue management system using Android was to increase the efficiency of after-sale service management. It helped create mutual understanding between users, analysts, and developers, and also focused on facilitating system users to use it everywhere via portable devices such as mobile phone or tablet.

The methodology for this study began with the analysis of problems and limitations of the current system, the analysis of user's needs, and the study of new system development feasibility. This new system brought mobile or tablet application program technology along with web-based application technology to use with a database. This system consisted of work assignment to the programmer and after-sale service team, notification to the project coordinator when progress monitoring is scheduled, search and identification of system deflection that used to occur, and record of solution and system deflection status update. The benefits of this system were reduction of working process and time, convenient and fast monitor of work progress, after-sale service problem solving, and ability to meet customer's needs which lead to customer's satisfaction and positive image of an organization.

กิตติกรรมประกาศ

โครงการนี้สำเร็จได้ด้วยความกรุณาจากท่านอาจารย์ที่ปรึกษา รศ.ดร. พรฤดี เนติโสภากุล ที่ได้สละเวลาให้คำปรึกษาและชี้แนะแนวทางการดำเนินโครงการ ตลอดจนให้ข้อเสนอแนะ และข้อคิดเห็นที่เป็นประโยชน์จนทำให้โครงการสำเร็จลุล่วงโดยสมบูรณ์ ขอขอบพระคุณ ท่านอาจารย์เป็นอย่างสูงที่ให้ความอนุเคราะห์

ขอขอบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาให้แก่ข้าพเจ้าตลอดระยะเวลาของการศึกษา

ขอขอบพระคุณบริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด (มหาชน) อาคารชินวัตรทาวเวอร์ 3 ชั้น 6 ซึ่งได้เอื้อเฟื้อสถานที่และทรัพยากรต่างๆ ตลอดระยะเวลาในการจัดทำโครงการ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดา และบุคคลในครอบครัวของข้าพเจ้า ที่เป็นกำลังใจ และสนับสนุนให้ข้าพเจ้าดำเนินโครงการนี้ได้สำเร็จลุล่วง รวมถึงขอขอบคุณ มิตรภาพอันเกิดจากเพื่อนๆ ทุกคน ที่เป็นแรงผลักดันในการดำเนินโครงการในครั้งนี้

ณัฐทิศา สิ้นชัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนของการดำเนินการ.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ความหมายของแอนดรอยด์.....	5
2.2 ประเภทของระบบปฏิบัติการแอนดรอยด์.....	6
2.3 คุณสมบัติและความสามารถของแอนดรอยด์.....	6
2.4 สถาปัตยกรรมของแอนดรอยด์.....	7
2.4.1 ลินูกซ์ เคอร์เนล (Linux Kernel).....	8
2.4.2 ไบปรารี (Native Libraries).....	9
2.4.3 แอนดรอยด์รันไทม์ (Android Runtime).....	9
2.4.4 แอปพลิเคชัน เฟรมเวิร์ก (Application Framework).....	9
2.4.5 แอปพลิเคชัน (Applications).....	10
2.5 การเตรียมระบบที่ใช้พัฒนาแอนดรอยด์แอปพลิเคชัน.....	11
2.5.1 การติดตั้ง Java Development Kit (JDK).....	11
2.5.2 การติดตั้ง Eclipse.....	14
2.5.3 การติดตั้ง Android SDK.....	15
2.5.4 การติดตั้งปลั๊กอินสำหรับ Eclipse (ADT Plug-in).....	18
2.5.5 การระบุตำแหน่งของ Android SDK ให้กับ ADT.....	19

สารบัญ (ต่อ)

	หน้า
2.5.6 การอัปเดต ADT Plug-in.....	20
2.6 การสร้าง AVD (Android Virtual Device).....	21
2.7 โครงสร้างโปรเจ็กต์แอนดรอยด์.....	24
2.8 องค์ประกอบของแอนดรอยด์แอปพลิเคชัน.....	25
2.8.1 แอคทีวิตี (Activity).....	25
2.8.2 เซอร์วิส (Service).....	30
2.8.3 บรอดคาสต์รีซีฟเวอร์ (Broadcast receiver).....	33
2.8.4 อินเทนต์ (Intent).....	33
2.8.5 Content provider.....	34
2.8.6 การแจ้งเตือน (Notifications).....	34
บทที่ 3 การศึกษาและการวิเคราะห์ระบบปัจจุบัน.....	35
3.1 ภาพรวมขององค์กร.....	35
3.2 การดำเนินงานปัจจุบัน.....	36
3.2.1 การประสานงานด้านการขาย.....	36
3.2.2 การประสานงาน โครงการด้านการบริการหลังการขาย.....	39
3.3 วิธีการเก็บรวบรวมข้อมูลความต้องการของผู้ใช้.....	43
3.4 ปัญหาและข้อจำกัดของการดำเนินงานปัจจุบัน.....	43
3.5 แนวทางการแก้ไขปัญหา.....	44
บทที่ 4 การวิเคราะห์และออกแบบระบบใหม่.....	45
4.1 การวิเคราะห์ความต้องการของระบบใหม่.....	45
4.1.1 ความต้องการเชิงหน้าที่การทำงาน (Functional Requirement).....	45
4.1.2 ความต้องการเชิงคุณลักษณะ (Non Functional Requirement).....	46
4.2 การออกแบบระบบด้วยยูเอ็มแอล.....	46
4.2.1 ยูสเคสไดอะแกรม.....	46
4.2.2 คลาสไดอะแกรม.....	67
4.2.3 ซีควเอนซ์ไดอะแกรม.....	67
4.3 การออกแบบฐานข้อมูล.....	71
4.3.1 แบบจำลองความสัมพันธ์ระหว่างเอนทิตี.....	71

สารบัญ (ต่อ)

	หน้า
4.3.2 พจนานุกรมข้อมูล (Data Dictionary).....	73
บทที่ 5 บทสรุป.....	78
5.1 สรุปผลการศึกษา.....	78
5.2 ข้อจำกัดและข้อเสนอแนะ.....	79
บรรณานุกรม.....	80
ประวัติผู้เขียน.....	81



สารบัญตาราง

ตารางที่	หน้า
4.1	รายละเอียดของยูสเคส Login (บนอุปกรณ์พกพา).....49
4.2	รายละเอียดของยูสเคส Assign Job.....50
4.3	รายละเอียดของยูสเคส Select Programmer or Service.....51
4.4	รายละเอียดของยูสเคส Set Issue And Deadline.....52
4.5	รายละเอียดของยูสเคส Attach.....53
4.6	รายละเอียดของยูสเคส Search Issue.....54
4.7	รายละเอียดของยูสเคส View Issue History.....55
4.8	รายละเอียดของยูสเคส Login (บนเว็บแอปพลิเคชัน).....58
4.9	รายละเอียดของยูสเคส Track Issue Status.....59
4.10	รายละเอียดของยูสเคส Software Issue.....60
4.11	รายละเอียดของยูสเคส Hardware Issue.....61
4.12	รายละเอียดของยูสเคส Search Issue.....62
4.13	รายละเอียดของยูสเคส View Issue History.....63
4.14	รายละเอียดของยูสเคส Update Status and Solution Issue.....64
4.15	รายละเอียดของยูสเคส Report.....65
4.16	พจนานุกรมข้อมูลเอนทิตีตำแหน่งของพนักงาน.....73
4.17	พจนานุกรมข้อมูลเอนทิตีพนักงาน.....74
4.18	พจนานุกรมข้อมูลเอนทิตีความซับซ้อนของระบบด้านซอฟต์แวร์.....74
4.19	พจนานุกรมข้อมูลเอนทิตีวิธีแก้ไขความซับซ้อนของระบบด้านซอฟต์แวร์.....75
4.20	พจนานุกรมข้อมูลเอนทิตีความซับซ้อนของระบบด้านฮาร์ดแวร์.....76
4.21	พจนานุกรมข้อมูลเอนทิตีวิธีแก้ไขความซับซ้อนของระบบด้านฮาร์ดแวร์.....77

สารบัญรูป

รูปที่		หน้า
2.1	สถาปัตยกรรมของแอนดรอยด์.....	8
2.2	แสดงวิธีการดาวน์โหลดหลังจากพิมพ์เว็บ.....	11
2.3	แสดงวิธีการค้นหา Java จากกูเกิล.....	12
2.4	แสดงขั้นตอนที่ 1 ของการทดสอบว่าเครื่องมี JDK หรือไม่.....	12
2.5	แสดงขั้นตอนที่ 2 ของการทดสอบว่าเครื่องมี JDK หรือไม่.....	13
2.6	แสดงผลของของการทดสอบว่าเครื่องมี JDK หรือไม่.....	13
2.7	แสดงวิธีดาวน์โหลด Eclipse.....	14
2.8	แสดงไฟล์ต่างๆของโปรแกรม Eclipse.....	15
2.9	แสดงวิธีดาวน์โหลด SDK Starter Package.....	16
2.10	แสดงผลการตรวจสอบว่ามี JDK อยู่ในระบบหรือไม่.....	16
2.11	แสดงวินโดว์ Android SDK Manager.....	17
2.12	แสดงวินโดว์ Install.....	18
2.13	แสดงไดอะล็อกบ็อกซ์ Add Repository.....	19
2.14	แสดงวินโดว์ Preferences.....	20
2.15	แสดง ADT ที่มีการอัปเดตเวอร์ชันใหม่.....	21
2.16	แสดงไอคอนที่ใช้เปิด Android Virtual Device Manager.....	22
2.17	แสดงวิธีการสร้าง ADV.....	22
2.18	แสดงการกรอกรายละเอียด AVD ที่ต้องการ.....	23
2.19	โครงสร้างโปรเจกต์แอนดรอยด์.....	24
2.20	โค้ด โครงร่างของแอกทิวิตี.....	26
2.21	แอนดรอยด์แอกทิวิตีสแต็ก.....	26
2.22	แสดงวงจรชีวิตของแอกทิวิตี.....	28
2.23	แสดงลักษณะการเรียกใช้คอมโพเนนต์เซอร์วิสของแอปพลิเคชัน Music.....	30
2.24	แสดงวงจรชีวิตของเซอร์วิสแบบ Unbound Service.....	31
2.25	วงจรชีวิตของ Bound Service.....	32
3.1	ภาพรวมโครงสร้างกลุ่มงานขายและบริการหลังการขาย.....	35
3.2	แอกทิวิตีไดอะแกรมแสดงการประสานงาน โครงการด้านการขาย.....	38

VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.3	แอกทิวิตีไดอะแกรมแสดงการประสานงาน โครงการด้านการบริการหลังการขายด้านซอฟต์แวร์.....40
3.4	แอกทิวิตีไดอะแกรมแสดงการประสานงาน โครงการด้านการบริการหลังการขายด้านฮาร์ดแวร์.....42
4.1	ยูสเคสไดอะแกรมระบบบริการจัดการปัญหาบนอุปกรณ์พกพา.....47
4.2	แสดงหน้า Login บนอุปกรณ์พกพา และแสดงความขัดข้องของระบบที่ยังไม่ได้แก้ไขและวันสิ้นสุดการแก้ไข.....49
4.3	แสดงหน้าหลักของแอปพลิเคชัน.....50
4.4	แสดงหน้าจอสำหรับเลือกพนักงานผู้รับผิดชอบแก้ไขระบบ.....51
4.5	แสดงหน้าจอสำหรับกำหนดวันสิ้นสุดและส่งความขัดข้องของระบบผ่านทางอีเมล.....53
4.6	แสดงหน้าจอรายการให้เลือกแนบไฟล์.....54
4.7	แสดงหน้าจอสำหรับค้นหาความขัดข้องของระบบ.....55
4.8	แสดงหน้าจอที่แสดงรายละเอียดความขัดข้องของระบบ.....56
4.9	ยูสเคสไดอะแกรมระบบบริการจัดการปัญหาบนเว็บแอปพลิเคชัน.....57
4.10	แสดงหน้า Login บนเว็บแอปพลิเคชัน.....59
4.11	แสดงหน้าจอหลักของเว็บแอปพลิเคชัน.....60
4.12	แสดงเมนูสำหรับเลือกประเภทความขัดข้องด้านซอฟต์แวร์.....61
4.13	แสดงเมนูสำหรับเลือกประเภทความขัดข้องด้านฮาร์ดแวร์.....62
4.14	แสดงหน้าจอสำหรับค้นหาความขัดข้องของระบบ.....63
4.15	แสดงหน้าจอสำหรับแสดงสถานะความขัดข้อง วิธีแก้ไข และผู้ทำการแก้ไข.....64
4.16	แสดงหน้าจอสำหรับบันทึกวิธีการแก้ไขความขัดข้อง พร้อมทั้งแก้ไขสถานะ.....65
4.17	แสดงหน้าจอสำหรับเรียกดูรายงานสรุปความขัดข้อง.....66
4.18	คลาสไดอะแกรมของระบบบริหารจัดการปัญหาด้วยแอนดรอยด์.....67
4.19	ซีเควนซ์ไดอะแกรม Assign Job.....68
4.20	ซีเควนซ์ไดอะแกรม Update Status and Solution Issue.....69
4.21	ซีเควนซ์ไดอะแกรม Search Issue.....70
4.22	ซีเควนซ์ไดอะแกรม Report.....71
4.23	อืออาร์ไดอะแกรมระบบบริหารจัดการปัญหาด้วยแอนดรอยด์.....72

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบัน การประสานงานในโครงการพัฒนาระบบ มีความสำคัญและเป็นกระบวนการที่มีความจำเป็นมากต่อการพัฒนาระบบ เนื่องจากการเก็บรวบรวมความต้องการ การทดสอบและปรับปรุงแก้ไขระบบ ล้วนต้องอาศัยการประสานงานระหว่างผู้ใช้ นักวิเคราะห์ และนักพัฒนา ซึ่งกระบวนการดังกล่าว เกิดความล่าช้าและผิดพลาดจากการเข้าใจไม่ตรงกันบ่อยครั้ง เนื่องจากการประสานงาน ส่วนใหญ่จะเกิดจากการที่ผู้ใช้ประสานงานกับผู้ประสานงาน โครงการซึ่งเป็นนักวิเคราะห์ และนักวิเคราะห์จะต้องแจ้งให้โปรแกรมเมอร์ทราบถึงความต้องการหรือปัญหาที่เกิดขึ้นอีกครั้ง ซึ่งการประสานงานทางโทรศัพท์ อีเมล หรือการกลับไปพูดคุยกับโปรแกรมเมอร์หลังประชุมเสร็จสิ้น ทำให้เกิดการได้รับข้อมูลที่ไม่ครบถ้วนและเกิดความล่าช้าในการตรวจสอบ

ในปัจจุบันมีผู้ใช้งานแอปพลิเคชันผ่านอุปกรณ์พกพาประเภท โทรศัพท์มือถือหรือแท็บเล็ต เพิ่มขึ้นเป็นจำนวนมาก และเป็นสิ่งที่สามารถแจ้งเตือนให้ผู้ใช้งานรับทราบปัญหาได้อย่างรวดเร็ว อีกทั้งยังสนับสนุนการส่งภาพ เสียง หรือข้อความไปได้พร้อมกัน

ดังนั้น เพื่อแก้ไขปัญหาค่าเงินงานปัจจุบันดังที่ได้กล่าวมาแล้ว จึงจำเป็นต้องพัฒนาระบบสารสนเทศใหม่ที่มีประสิทธิภาพ และสอดคล้องกับงานด้านการประสานงาน โครงการ ซึ่งจะช่วยอำนวยความสะดวก รวดเร็วและเพิ่มประสิทธิภาพในการดำเนินงานได้ดียิ่งขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

การพัฒนาระบบสารสนเทศเพื่อเป็นเครื่องมือช่วยในการจัดการการแจ้งปัญหาหรือแจ้งความต้องการของลูกค้า และเพื่อสนับสนุนระบบงานประสานงานและงานบริการหลังการขาย โดยมีวัตถุประสงค์ดังนี้

- 1 เพื่อนำระบบสารสนเทศมาประยุกต์ใช้เป็นเครื่องมือช่วยในการบริหารจัดการการแจ้งปัญหาหรือแจ้งความต้องการของลูกค้า
- 2 เพื่อให้เกิดความเข้าใจที่ตรงกันระหว่างผู้ใช้ นักวิเคราะห์ และนักพัฒนา
- 3 เพื่อลดขั้นตอนการทำงานระหว่างการไปตรวจสอบความขัดข้องของระบบและการมอบหมายงานให้สั้นลงและรวดเร็วมากขึ้น

- 4 เพื่อให้ผู้ประสานงานสามารถตรวจสอบบุคคลที่รับผิดชอบในการแก้ไขความขัดข้องของระบบและติดตามความคืบหน้าได้ง่าย
- 5 เพื่อเป็นศูนย์กลางในการจัดเก็บองค์ความรู้ให้กับองค์กร

1.3 ขอบเขตของโครงการงาน

ขอบเขตในการศึกษาและจัดทำระบบสารสนเทศในครั้งนี้ เป็นการศึกษา วิเคราะห์ ออกแบบ และพัฒนาระบบ เพื่อสนับสนุนระบบงานประสานงาน โครงการและงานบริการหลังการขายเป็นหลัก ทั้งการแจ้งปัญหาที่เกิดขึ้นจากระบบหลังการขายหรือการรับความต้องการของลูกค้าหรือผู้ใช้ระบบ พัฒนาเป็นแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ โดยแบ่งขอบเขตความสามารถของระบบออกเป็น 2 ส่วนดังนี้

แอปพลิเคชันบนโทรศัพท์เคลื่อนที่หรืออุปกรณ์พกพา

- 1) ผู้ประสานงาน โครงการสามารถใช้งานแอปพลิเคชันบนโทรศัพท์เคลื่อนที่หรืออุปกรณ์พกพาที่ใช้ระบบปฏิบัติการแอนดรอยด์
- 2) สามารถเชื่อมต่อผ่านระบบอินเทอร์เน็ต ให้ผู้ประสานงาน โครงการสามารถส่งอีเมลในการมอบหมายงานให้แก่โปรแกรมเมอร์ และทีมบริการหลังการขายที่รับผิดชอบสถานที่ติดตั้งระบบ
- 3) ผู้ประสานงาน โครงการสามารถส่งรายละเอียดเมื่อระบบเกิดความขัดข้องทั้งด้านฮาร์ดแวร์ และซอฟต์แวร์ พร้อมทั้งระบุชื่อผู้ทำการแก้ไขระบบผ่านอีเมล
- 4) ผู้ประสานงาน โครงการสามารถแนบรูปภาพ เสียง ข้อความ ที่ได้จากการตรวจสอบความขัดข้องนอกสถานที่ให้แก่โปรแกรมเมอร์ และทีมบริการหลังการขายที่รับผิดชอบสถานที่ติดตั้งระบบ
- 5) สามารถแจ้งเตือนโปรแกรมเมอร์หรือทีมบริการหลังการขาย(เซอร์วิส) ที่ทำการแก้ไขระบบได้ผ่านทางอีเมล
- 6) สามารถแจ้งเตือนผู้ประสานงาน โครงการได้ ตามวันที่ผู้ประสานงาน โครงการกำหนดวันที่แจ้งเตือน
- 7) ผู้ประสานงาน โครงการสามารถค้นหาและศึกษาความขัดข้องของระบบที่เคยเกิดขึ้นผ่านทางโทรศัพท์เคลื่อนที่หรืออุปกรณ์พกพาอื่นๆ

แอปพลิเคชันบนเว็บไซต์

- 1) โปรแกรมเมอร์หรือทีมบริการหลังการขาย(เซอร์วิส) สามารถอัพเดทสถานะความขัดข้องของระบบว่าแก้ไขเสร็จสมบูรณ์หรือไม่ และบันทึกวิธีการแก้ไขความขัดข้องผ่านทางเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) โปรแกรมเมอร์หรือทีมบริการหลังการขาย(เซอร์วิส) สามารถค้นหาและศึกษาความขัดข้องของระบบที่เคยเกิดขึ้นผ่านทางเว็บไซต์
- 3) กรรมการผู้จัดการ สามารถตรวจสอบจำนวนความขัดข้องของระบบที่เคยเกิดขึ้น ในรูปแบบรายงาน

1.4 ขั้นตอนของการดำเนินงาน

สำหรับขั้นตอนการดำเนินงานการออกแบบและพัฒนาระบบสารสนเทศเพื่อสนับสนุนงานประสานงานโครงการและงานบริการหลังการขาย มีขั้นตอนการดำเนินงาน ดังนี้

- 1 ศึกษารูปแบบการทำงานของจัดการการแจ้งปัญหาในรูปแบบเดิม การสังเกตการณ์สัมภาษณ์ผู้ที่เกี่ยวข้อง เพื่อทราบถึงขั้นตอนและวิธีการดำเนินการ รวมถึงปัญหาที่เกิดจากการดำเนินการในปัจจุบัน
- 2 รวบรวมปัญหาที่เกิดขึ้นจากระบบเดิม และสรุปเป็นความต้องการของระบบใหม่ โดยวิเคราะห์ให้ครอบคลุมทุกกรณีที่เกิดขึ้น เพื่อนำมาวิเคราะห์ขอบเขตของระบบใหม่
- 3 ศึกษาความเป็นไปได้ของการพัฒนาระบบใหม่ ทั้งในแง่ของความเป็นไปได้ทางเทคนิคและความเป็นไปได้ในการปฏิบัติงาน
- 4 วิเคราะห์และออกแบบระบบงานใหม่ โดยใช้หลักการพัฒนาระบบสารสนเทศแนวคิดเชิงวัตถุ และยูเอ็มแอด เป็นเครื่องมือในการออกแบบระบบงาน ซึ่งประกอบด้วย ยูสเคสไดอะแกรม คลาสไดอะแกรม ซีควเอนซ์ไดอะแกรม และออกแบบฐานข้อมูลโดยใช้ออร์ไดอะแกรม เพื่อแสดงความสัมพันธ์ของตารางต่างๆของฐานข้อมูล
- 5 ศึกษาเทคโนโลยีที่เกี่ยวข้อง ภาษาคอมพิวเตอร์ และเครื่องมือที่ใช้ในการพัฒนาระบบ
- 6 วิเคราะห์และออกแบบส่วนติดต่อกับผู้ใช้ ลงมือพัฒนาระบบตามที่ออกแบบ และทดสอบระบบ

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

การพัฒนาระบบสารสนเทศเพื่อสนับสนุนงานประสานงานโครงการและงานบริการหลังการขาย คาดว่าจะได้รับประโยชน์ ดังนี้

- 1 ส่งเสริมภาพลักษณ์ที่ดีแก่บริษัท ที่สามารถแก้ไขปัญหาของระบบหลังการขายและสามารถปฏิบัติงานตรงตามความต้องการของลูกค้าได้อย่างรวดเร็ว
- 2 การพัฒนาทรัพยากรบุคคล ช่วยเพิ่มทักษะการวิเคราะห์ออกแบบ การพัฒนาระบบสารสนเทศในการจัดการข้อมูล และทักษะทางด้าน โปรแกรมมิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3 ผู้ประสานงานโครงการสามารถทำงานได้อย่างสะดวกมากขึ้น ลดขั้นตอนการทำงาน ส่งผลให้เวลาในการทำงานลดลง และสามารถติดตามความคืบหน้าของงานได้รวดเร็วมากขึ้น
- 4 การทำงานเป็นทีม ลดความเข้าใจที่ไม่ตรงกัน และลดความผิดพลาดในการประสานงานระหว่างผู้ใช้ ผู้ประสานงานโครงการ (นักวิเคราะห์) ผู้พัฒนา (โปรแกรมเมอร์)
- 5 การศึกษาความขัดข้องด้านซอฟต์แวร์ และด้านฮาร์ดแวร์ที่มีการบันทึกวิธีการแก้ไข ส่งผลให้ผู้ประสานงานโครงการ ผู้พัฒนาและทีมบริการหลังการที่รับมาใหม่ สามารถศึกษาและเป็นแนวทางให้สามารถแก้ไขปัญหาเฉพาะหน้าได้อย่างรวดเร็ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ความหมายของแอนดรอยด์

แอนดรอยด์ (Android) คือ เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อนกันหรือแบบสแต็ก (Stack) ที่รวมเอาระบบปฏิบัติการ มิดเดิลแวร์ และแอปพลิเคชันที่สำคัญเข้าไว้ด้วยกัน เพื่อให้สำหรับทำงานบนอุปกรณ์พกพาเคลื่อนที่โดยเฉพาะ เช่น โทรศัพท์มือถือ และคอมพิวเตอร์แบบแท็บเล็ต (Tablet Computer หรือ Tablet PC) เป็นต้น ซึ่งพัฒนามาจากระบบปฏิบัติการลินุกซ์ (Linux)

การทำงานของแอนดรอยด์มีพื้นฐานอยู่บนระบบลินุกซ์ เคอร์เนล (Linux Kernel) ซึ่งใช้ Android SDK (Software Development Kit) เป็นเครื่องมือสำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ โดยใช้ภาษาจาวาในการพัฒนา

แอนดรอยด์ถูกพัฒนาโดยบริษัท Android Inc. ซึ่งก่อตั้งในปี ค.ศ.2003 โดย Andy Rubin และ Rich Miner ต่อมาในปี ค.ศ. 2005 ถูกซื้อบริษัทดังกล่าว หลังจากนั้นถูกซื้อโดยกลุ่มบริษัททางด้านฮาร์ดแวร์ ซอฟต์แวร์ และการสื่อสาร เช่น Intel, HTC, LG, Texas Instruments เพื่อจัดตั้งองค์กรความร่วมมือที่มีชื่อว่า Open Handset Alliance ขึ้นในปี ค.ศ. 2007 โดยมีจุดประสงค์ในการสร้างแพลตฟอร์ม (Platform) สำหรับอุปกรณ์พกพาที่มีพื้นฐานอยู่บนมาตรฐานเปิด (Open Standard) ซึ่งโปรเจกต์แรกที่กลุ่ม Open Handset Alliance เปิดตัวออกมาก็คือ แอนดรอยด์ ในชื่อโปรเจกต์ว่า The Android Open Source Project

แอนดรอยด์ไม่ใช่สมบัติของบริษัทใดบริษัทหนึ่ง (Non-proprietary) บริษัทผู้ผลิตมือถือหรืออุปกรณ์ที่นำแอนดรอยด์ไปใช้งานจึงไม่มีค่าใช้จ่ายด้านลิขสิทธิ์ นอกจากนี้การที่แอนดรอยด์เป็นแพลตฟอร์มโอเพนซอร์ส (Open Source) ก็ทำให้ผู้ผลิตสามารถปรับแต่งแอนดรอยด์ให้เหมาะกับฮาร์ดแวร์ของตนเอง

สำหรับนักพัฒนาทั่วไปจะสามารถพัฒนาแอปพลิเคชันที่รันบนแอนดรอยด์ได้โดยใช้ภาษาจาวา โดยการเข้าถึงความสามารถต่างๆของแอนดรอยด์จะกระทำผ่าน Java Library ที่ถูกเตรียมไว้ให้ใน Android SDK หรือก็คือชุดพัฒนาซอฟต์แวร์สำหรับแอนดรอยด์ ซึ่งสามารถดาวน์โหลดได้ฟรี

(พร้อมเลิศ หล่อวิจิตร. 2555:11)

2.2 ประเภทของระบบปฏิบัติการ Android

เนื่องจากระบบปฏิบัติการแอนดรอยด์เป็นซอฟต์แวร์แบบเปิด จึงอนุญาตให้นักพัฒนาหรือผู้ที่สนใจสามารถดาวน์โหลดซอร์สโค้ด (Source Code) ได้ ทำให้มีผู้พัฒนาจากหลายๆ ฝ่ายนำซอร์สโค้ดมาปรับแต่งและพัฒนาสร้างแอปพลิเคชันบนระบบแอนดรอยด์ในแบบฉบับของตนเองมากขึ้น โดยสามารถแบ่งประเภทของระบบแอนดรอยด์ออกเป็นกลุ่มๆ ได้ 3 ประเภท ดังต่อไปนี้ (จักรชัย ไสอินทร์ และพงษ์ศร จันทร์ขอย. 2554:3)

- 1 **Android Open Source Project (AOSP)** เป็นระบบแอนดรอยด์ประเภทแรกที่ทางบริษัทกูเกิลให้สามารถนำซอร์สโค้ดไปติดตั้งและใช้งานอุปกรณ์ต่างๆ ได้โดยไม่ต้องเสียค่าใช้จ่าย
- 2 **Open Handset Mobile (OHM)** เป็นแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่ม Open Handset Alliances (OHA) ซึ่งบริษัทเหล่านี้จะพัฒนาระบบแอนดรอยด์ในแบบฉบับของตนเอง โดยมีรูปร่างหน้าตาการแสดงผล และฟังก์ชันการใช้งานที่แตกต่างกัน รวมไปถึงอาจจะมีความเป็นเอกลักษณ์และรูปแบบการใช้งานเป็นของแต่ละบริษัท และโปรแกรมแอนดรอยด์ประเภทนี้ก็จะได้รับสิทธิและบริการเสริมต่างๆ จากกูเกิลที่เรียกว่า **GMS (Google Mobile Service)** ซึ่งเป็นบริการเสริมที่ทำให้ระบบแอนดรอยด์มีประสิทธิภาพมากขึ้น
- 3 **Cooking หรือ Customize** เป็นระบบแอนดรอยด์ที่นักพัฒนานำเอาซอร์สโค้ดจากแหล่งต่างๆ มาปรับแต่งให้อยู่ในรูปแบบฉบับของตนเอง ซึ่งการพัฒนาจะต้องปลดล็อกสิทธิในการใช้งานอุปกรณ์ (Unlock) เสียก่อนจึงจะสามารถติดตั้งได้ ทั้งนี้ระบบแอนดรอยด์ประเภทนี้ถือได้ว่าเป็นประเภทที่มีความสามารถสูงที่สุด เนื่องจากจะได้รับการปรับแต่งขีดความสามารถต่างๆ ให้มีความเข้ากันได้กับอุปกรณ์นั้นๆ

2.3 คุณสมบัติและความสามารถของแอนดรอยด์

คุณสมบัติและความสามารถหลักๆ ของแอนดรอยด์ มีดังนี้ (พร้อมเลิศ หล่อวิจิตร. 2555:12)

- 1 **การเชื่อมต่อ** เทคโนโลยีการเชื่อมต่อที่แอนดรอยด์สนับสนุนประกอบด้วย GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-fi, LTE, NFC และ WiMAX
- 2 **Messaging** สนับสนุน SMS, MMS, Threaded Text Messaging และ Cloud To Device Messaging Framework (C2DM)
- 3 **การจัดเก็บข้อมูล** แอนดรอยด์มี SQLite ซึ่งเป็นฐานข้อมูลเชิงสัมพันธ์ขนาดเล็ก (lightweight) ที่มีประสิทธิภาพสูง สำหรับใช้จัดเก็บข้อมูล
- 4 **เว็บเบราว์เซอร์ (Integrated Browser)** เป็นการผนวก Web Browser เข้าไว้กับแอนดรอยด์ทั้งนี้มีพื้นฐานมาจากซอฟต์แวร์เว็บคิต (Webkit) แอนดรอยด์ติดตั้งมาพร้อมกับโปรแกรมเว็บ-

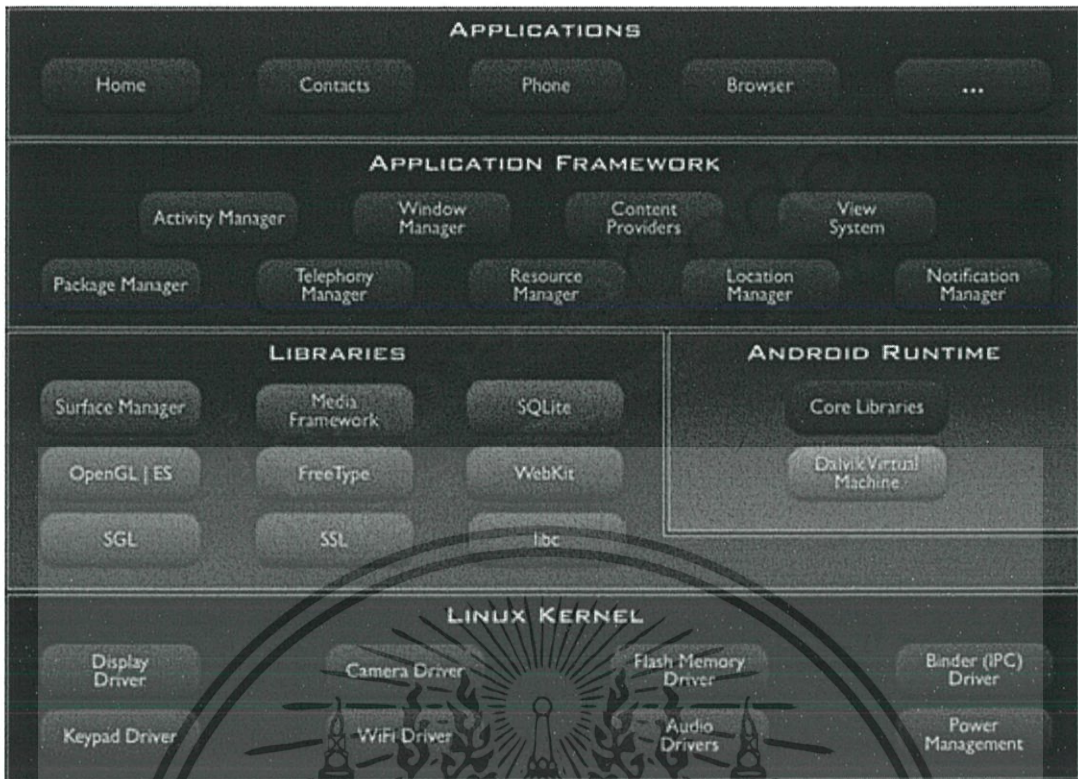
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บราวเซอร์ที่พัฒนาบนเอ็นจิน Webkit และใช้จาวาสคริปต์เอ็นจิน V8 ของเว็บเบราว์เซอร์ Google Chrome
- 5 **มีเดีย (Media)** สนับสนุนเสียง วิดีโอ และรูปภาพในฟอร์แมตยอดนิยมต่างๆ เช่น MPEG4, H.264, MP3, AAC, AMR, JPG และ PNG
 - 6 **สตรีมมิง (Streaming)** สนับสนุน RTP/RTSP streaming และ HTML progressive download
 - 7 **สนับสนุนจาวา** การพัฒนาแอปพลิเคชันบนแอนดรอยด์จะใช้ภาษาจาวา โดยโค้ดจาวาที่คอมไพล์แล้วจะไม่ได้รันใน Java Virtual Machine (JVM) เหมือนจาวาแอปพลิเคชันทั่วไป แต่จะรันใน Dalvik Virtual Machine ซึ่งเป็น VM ที่ถูกพัฒนาขึ้นสำหรับอุปกรณ์พกพาโดยเฉพาะ
 - 8 **มัลติทัช (Multi-touch)** รองรับการใช้นิ้วมือแต่ละหน้าจอเพื่อสั่งงานได้มากกว่า 1 จุดพร้อมกัน
 - 9 **มัลติทาสกิง (Multi-tasking)** คือ ความสามารถในการรันหลายแอปพลิเคชันพร้อมกัน
 - 10 **Tethering (หรือ Mobile Hotspot)** คือ ความสามารถในการแชร์อินเทอร์เน็ตผ่านมือถือหรืออุปกรณ์แอนดรอยด์
 - 11 **สนับสนุนฮาร์ดแวร์เสริมอื่นๆ** เช่น กล้องถ่ายรูป, GPS, Accelerometer และเทอร์โมมิเตอร์ เป็นต้น
 - 12 **Optimized Graphic** เป็นส่วนสนับสนุนการทำงานแบบกราฟิกทั้งในส่ว 2 มิติและ 3 มิติ โดยใช้เครื่องมือ OpenGL
 - 13 **Rich Development Environment** เป็นส่วนสนับสนุนฟังก์ชันต่างๆ ที่ช่วยในการพัฒนาแอปพลิเคชัน เช่น Emulator, Debugging Tool, Memory and Performance Profiling และ Plug-in สำหรับเครื่องมือ Eclipse

2.4 สถาปัตยกรรมของแอนดรอยด์

สถาปัตยกรรมของการออกแบบแอนดรอยด์ (Android Architecture) นั้นถูกแบ่งออกเป็นลำดับชั้นหรือจะเรียก **Layer** โดยที่แต่ละชั้นจะเรียกใช้บริการจากลำดับชั้นที่อยู่ด้านล่างของตัวเอง

การทำความเข้าใจโครงสร้างของระบบปฏิบัติการแอนดรอยด์ ถือว่าเป็นสิ่งสำคัญเพราะถ้า นักพัฒนาโปรแกรม สามารถมองภาพโดยรวมของระบบได้ทั้งหมด จะสามารถเข้าใจถึงกระบวนการทำงานได้ดียิ่งขึ้น และสามารถนำไปช่วยในการออกแบบ โปรแกรมที่ต้องการพัฒนา เพื่อให้เกิดประสิทธิภาพในการทำงาน ดังรูปที่ 2.1 แสดงส่วนประกอบหลักๆ ของระบบปฏิบัติการ แอนดรอยด์



รูปที่ 2.1 สถาปัตยกรรมของแอนดรอยด์

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ จะสังเกตได้ว่า มีการแบ่งออกมาเป็นส่วนๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดจะเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรงซึ่งก็คือส่วนของ (Applications) จากนั้นก็จะลำดับลงมาเป็นองค์ประกอบอื่นๆตามลำดับ และสุดท้ายจะเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทาง Linux Kernel โดยรายละเอียดของโครงสร้างของแอนดรอยด์แต่ละส่วนมีดังนี้ (พร้อมเลิศ หล่อวิจิตร. 2555:13)

2.4.1 ลิ눅ซ์ เคอร์เนล (Linux Kernel)

ส่วนที่เป็นแกนหลักหรือเคอร์เนล (Kernel) ของแอนดรอยด์นั้น คือเคอร์เนลของลินุกซ์ ทำหน้าที่เป็น Hardware Abstraction Layer คือเป็นตัวกลางระหว่างฮาร์ดแวร์กับส่วนของซอฟต์แวร์ที่อยู่ถัดขึ้นไป และทำหน้าที่บริหารจัดการทรัพยากรต่างๆของเครื่อง เช่น การจัดการหน่วยความจำ การจัดการโพรเซส เป็นต้น ผู้ผลิตอุปกรณ์สามารถ “พอร์ต” (port) แอนดรอยด์ให้ไปรันบนฮาร์ดแวร์แบบต่างๆได้ โดยเปลี่ยนแปลงในส่วนของลินุกซ์เคอร์เนลนี้

ผู้ใช้อือถือหรืออุปกรณ์แอนดรอยด์จะไม่เห็นว่ามีลินุกซ์อยู่ในเครื่อง และแอปพลิเคชันที่เราพิจารณาก็จะไม่ได้เรียกไปยังลินุกซ์โดยตรง แต่ในฐานะนักพัฒนาจำเป็นต้องรู้ว่าลินุกซ์อยู่ เนื่องจากโปรแกรมยูนิตีลิตี้บางตัวที่ Android SDK เตรียมมาให้จะติดต่อกับลินุกซ์ในอุปกรณ์แอนดรอยด์ เช่น โปรแกรม adb ที่ช่วยให้เราเรียกใช้คำสั่งเพื่อสำรวจระบบไฟล์ของเครื่อง ดูโพรเซสที่รันอยู่ในขณะนั้น และอื่นๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ไบเบรารี (Native Libraries)

ถัดขึ้นมาจาก Linux Kernel ก็คือส่วนที่เป็น ไบเบรารีของแอนดรอยด์ ซึ่งทั้งหมดเขียนด้วยภาษา C หรือ C++ และถูกคอมไพล์มาสำหรับฮาร์ดแวร์ของอุปกรณ์แต่ละรุ่น โดยแบ่งชุดคำสั่งออกเป็น กลุ่มตามวัตถุประสงค์ของการใช้งานดังนี้

- 1 **Surface Manager** คือ ไบเบรารีจัดการส่วนแสดงผลที่มีความสามารถในการผสมกราฟิกทั้ง 2 มิติและ 3 มิติจากแอปพลิเคชันต่างๆเข้าด้วยกัน ทำให้สามารถสร้างเอฟเฟค เช่น วินโดว์ที่มองเห็นทะลุไปข้างหลังได้ และ Transition ในรูปแบบต่างๆ
- 2 **Media Libraries** คือ ไบเบรารีที่จัดเตรียมบริการในการเล่นและบันทึกเสียง วิดีโอ และรูปภาพในฟอร์แมตต่างๆ เช่น MPEG4, H.264, MP3, AAC, AMR, JPG และ PNG
- 3 **SQLite** คือ Database Engine ที่มีประสิทธิภาพและมีขนาดเล็ก เพื่อให้เราจัดเก็บข้อมูลของแอปพลิเคชันไว้ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ (Relational Database)
- 4 **Webkit** คือ ไบเบรารีที่ใช้แสดงเนื้อหาเว็บเพจ ซึ่งเป็นตัวเดียวกับที่ใช้ใน Google Chrome และ Apple Safari รวมถึงเว็บเบราว์เซอร์ในมือถือ iPhone และมือถือตระกูล S60 ของโนเกียด้วย

2.4.3 แอนดรอยด์รันไทม์ (Android Runtime)

อีกส่วนหนึ่งที่ทำงาบนอยู่บนเคอร์เนลของลินุกซ์ก็คือ Android Runtime ซึ่งประกอบด้วย Core Library สำหรับภาษาจาวา และ Dalvik Virtual Machine ซึ่งอธิบายง่าย ๆ ก็คือ Java Virtual Machine ในรูปแบบของแอนดรอยด์เอง ซึ่งถูกออกแบบมาสำหรับอุปกรณ์ที่ใช้ไฟฟ้าจากแบตเตอรี่ มีหน่วยความจำ (Memory) และมีหน่วยประมวลผลกลาง (CPU) จำกัด ซึ่งการทำงานของ Dalvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงานไปเป็นไฟล์ .DEX ก่อนการทำงาน เหตุผลก็เพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับหน่วยประมวลผลกลางที่มีความเร็วไม่มาก

ในแอนดรอยด์นั้นแต่ละแอปพลิเคชันจะรันอยู่ในโพรเซสของตัวเอง และมี Dalvik VM ของตัวเองอยู่ด้วย ดังนั้นโค้ดของแต่ละแอปพลิเคชันจึงรันอยู่ใน VM ที่แยกออกจากกัน

ส่วนต่อมาก็คือ Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญ ซึ่งเป็น Java Library นั้นส่วนใหญ่จะเหมือนกับใน Java Standard Edition (Java SE) ที่เราใช้พัฒนาจาวาแอปพลิเคชันบนพีซี แต่บางไลเบรารีที่มีใน Java SE จะไม่มีในแอนดรอยด์ และบางไลเบรารีถึงแม้จะมีในแอนดรอยด์แต่ก็มีรูปแบบการใช้งานที่แตกต่างออกไป

2.4.4 แอปพลิเคชัน เฟรมเวิร์ก (Application Framework)

เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากๆ ซึ่งประกอบด้วยคอมโพเนนต์พื้นฐานต่างๆที่ใช้ในการสร้างแอปพลิเคชัน คอมโพเนนต์เหล่านี้จะ

คิดตั้งมากับแอนดรอยด์อยู่แล้ว และเราสามารถแทนที่ด้วยคอมโพเนนต์ที่เราสร้างขึ้นเองได้

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนสำคัญใน Application Framework มีดังนี้

- 1 **Activities Manager** เป็นกลุ่มของชุดคำสั่งที่จัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม(Activity)
- 2 **Content Providers** เป็นกลุ่มของชุดคำสั่ง ที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่น และสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้
- 3 **View System** เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับการจัดการ โครงสร้างของหน้าจอที่แสดงผลในส่วนที่ติดต่อกับผู้ใช้งาน (User Interface) เช่น ปุ่ม เท็กซ์บ็อกซ์ ลิสต์ กริด
- 4 **Telephony Manager** เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลด้านโทรศัพท์ เช่น หมายเลขโทรศัพท์ เป็นต้น
- 5 **Resource Manager** เป็นกลุ่มของชุดคำสั่งในการเข้าถึงข้อมูลใดๆในแอปพลิเคชันที่ไม่ใช่โค้ด เช่น ข้อความ, รูปภาพ และค่าสตริง เป็นต้น
- 6 **Location Manager** เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับตำแหน่งทางภูมิศาสตร์ ที่ระบบปฏิบัติการได้รับค่าจากอุปกรณ์
- 7 **Notification Manager** เป็นกลุ่มของชุดคำสั่งที่จะถูกเรียกใช้เมื่อโปรแกรม ต้องการแสดงผลให้กับผู้ใช้งาน ผ่านทางแถบสถานะ(Status Bar) ของหน้าจอ ควบคุมอีเวนต์ (Event) ต่างๆ เช่น ในกรณีที่ได้รับข้อความและการแจ้งเตือนต่างๆ

2.4.5 แอปพลิเคชัน (Applications)

ส่วนบนสุดของสถาปัตยกรรมแอนดรอยด์ คือ ส่วนของ โปรแกรมหรือแอปพลิเคชันที่มีมาพร้อมระบบปฏิบัติการ (Core Application) เช่น Phone dialer, E-mail, Contacts, Web browser และ Android Market เป็นต้น หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ รวมถึงแอปพลิเคชันที่เราสร้างขึ้นด้วย ทั้งนี้ โปรแกรมในชั้นแอปพลิเคชันนั้นจะอยู่ในรูปแบบของไฟล์ .apk ซึ่งโดยทั่วไปแล้วจะอยู่ในไดเรกทอรี data/app ของโทรศัพท์เคลื่อนที่หรืออุปกรณ์พกพา ผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ดโปรแกรมเอาไว้

2.5 การเตรียมระบบที่ใช้พัฒนาแอนดรอยด์แอปพลิเคชัน

การพัฒนาแอนดรอยด์แอปพลิเคชัน (Android Application) สามารถทำได้บนเครื่องคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการต่างๆดังนี้ (ชัยวุฒิ พรหมบุตร. 2555)

- Windows XP (32 บิต), Windows Vista (32 หรือ 64 บิต) หรือ Windows 7 (32 หรือ 64 บิต)
- Mac OS X 10.5.8 ขึ้นไป
- Linux ที่มี GNU C Library (glibc) 2.7 ขึ้นไป และสำหรับ Ubuntu Linux ให้ใช้เวอร์ชัน 8.04 ขึ้นไป

เมื่อมีเครื่องคอมพิวเตอร์ที่รันบนระบบปฏิบัติการตัวใดตัวหนึ่งข้างต้นแล้ว ต่อไปจะอธิบายขั้นตอนการติดตั้งบน Windows 7 แบบ 32 บิต

2.5.1 การติดตั้ง Java Development Kit (JDK)

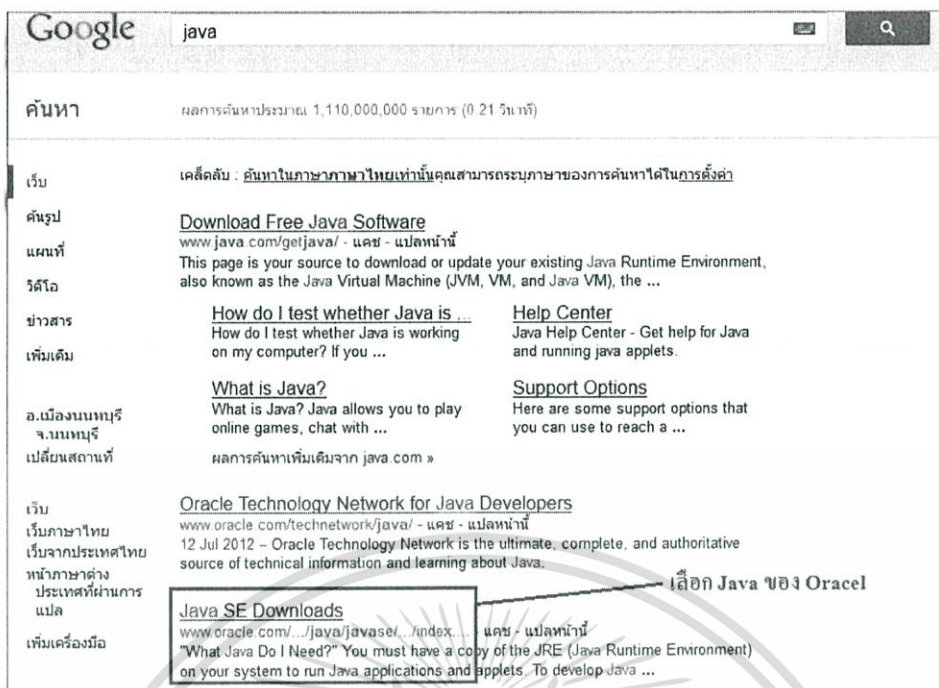
ในการพัฒนาแอปพลิเคชันแอนดรอยด์ จะใช้ภาษาจาวาเป็นหลักในการพัฒนา ฉะนั้นต้องติดตั้ง JDK เพื่อให้คอมพิวเตอร์สามารถคอมไพล์เลอร์ (Compiler) สำหรับคอมไพล์โค้ดจาวาไปเป็นแอปพลิเคชันที่รันได้บนอุปกรณ์แอนดรอยด์ ซึ่งมีขั้นตอนดังนี้

Download JDK

- 1 คำนวณโหลด JDK โดยไปที่ www.oracle.com/technetwork/java/javase/downloads ดังรูปที่ 2.2 หรือค้นหาจากกูเกิลโดยใช้คำค้นหาว่า “java” ดังรูปที่ 2.3

รูปที่ 2.2 แสดงวิธีการดาวน์โหลดหลังจากพิมพ์เว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



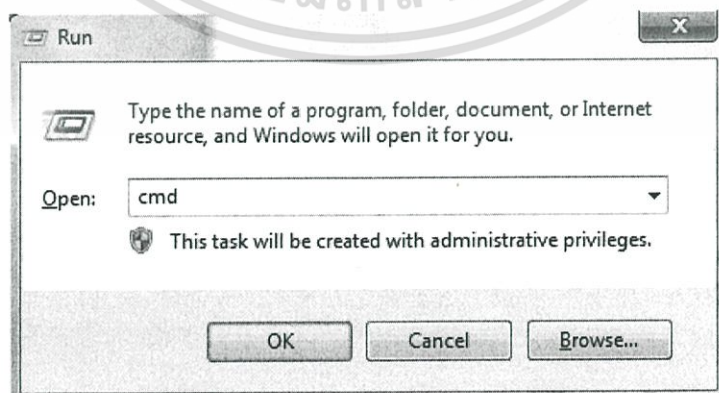
รูปที่ 2.3 แสดงวิธีการค้นหา Java จากกูเกิล

- ดับเบิลคลิกไฟล์ที่ดาวน์โหลดมาเพื่อติดตั้ง JDK ลงในเครื่อง ในที่นี้จะติดตั้ง JDK ไว้ที่โฟลเดอร์ C:\Program Files\Java\jdk1.6.0_32 (jdk เวอร์ชันที่ดาวน์โหลด) ตามค่าดีฟอลต์ที่ตัวติดตั้งระบุมาให้

การทดสอบว่าเครื่องมี JDK หรือไม่

หลังจากที่ได้ทำการติดตั้ง JDK เรียบร้อยแล้ว สามารถตรวจสอบการติดตั้ง JDK ว่าเสร็จหรือสมบูรณ์หรือยัง มีขั้นตอนดังนี้

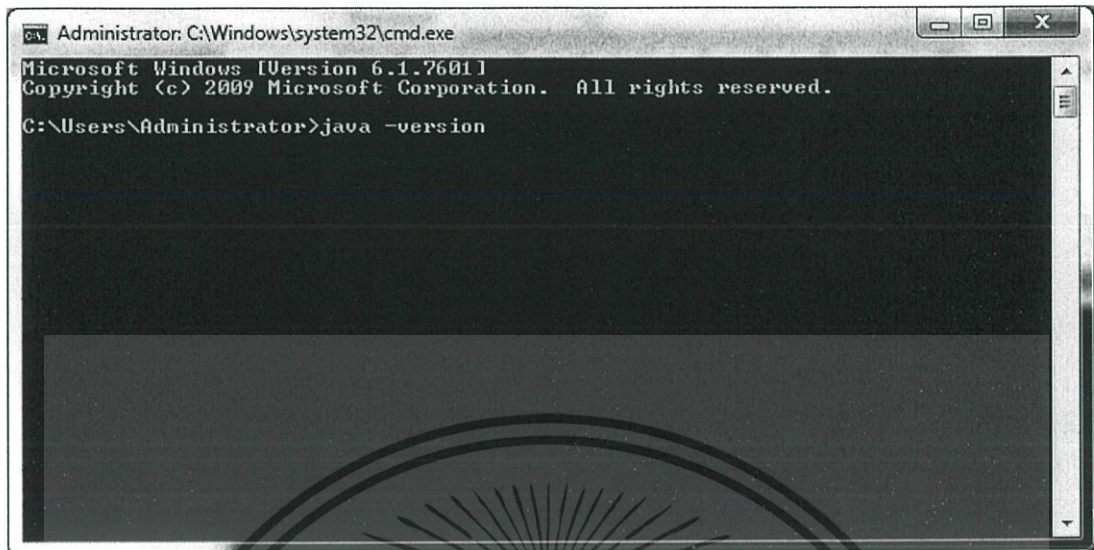
- ไปที่ start > RUN ตรงช่อง Open ให้พิมพ์ cmd แล้วกด Ok ดังรูปที่ 2.4



รูปที่ 2.4 แสดงขั้นตอนที่ 1 ของการทดสอบว่าเครื่องมี JDK หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

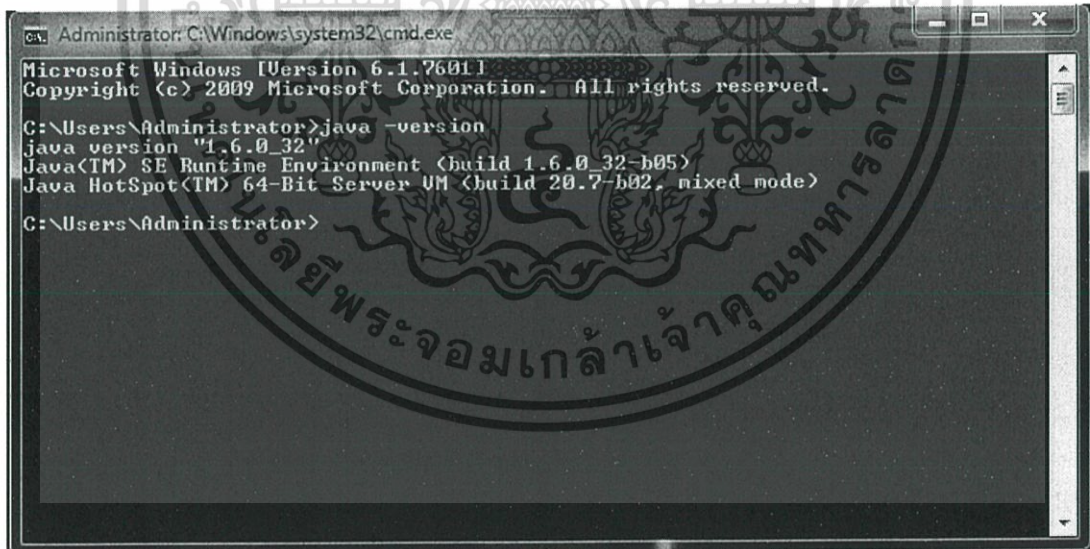
2 พิมพ์ `java -version` ดังรูปที่ 2.5 แล้วกด enter



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>java -version
```

รูปที่ 2.5 แสดงขั้นตอนที่ 2 ของการทดสอบว่าเครื่องมี JDK หรือไม่

ผลที่ได้ บอกได้ว่า จาวาเวอร์ชันอะไร มีตัว Runtime อะไร หากได้ผลดังรูปที่ 2.6 แสดงว่าเราติดตั้ง SDK บนเครื่องคอมพิวเตอร์ของเราสำเร็จ



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>java -version
java version "1.6.0_32"
Java(TM) SE Runtime Environment (build 1.6.0_32-b05)
Java HotSpot(TM) 64-Bit Server VM (build 20.7-b02, mixed mode)
C:\Users\Administrator>
```

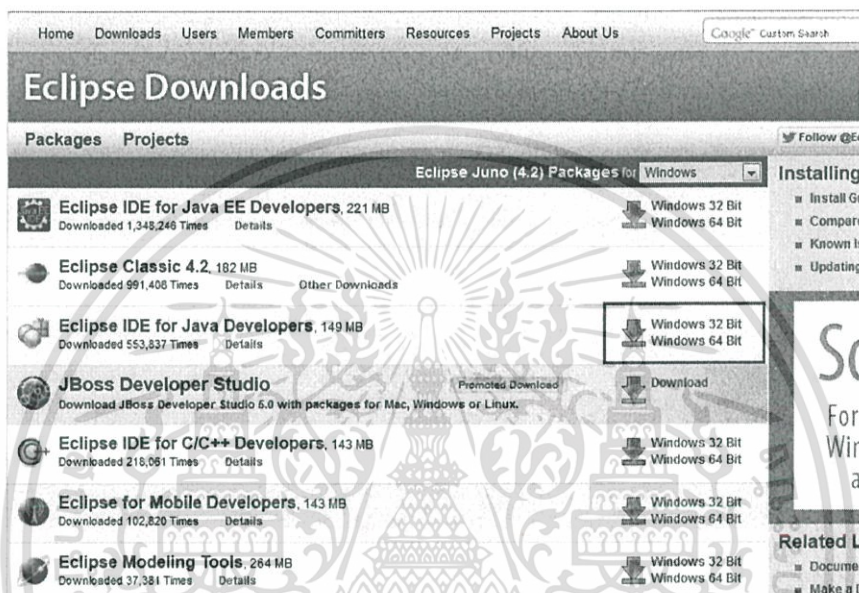
รูปที่ 2.6 แสดงผลของของการทดสอบว่าเครื่องมี JDK หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การติดตั้ง Eclipse

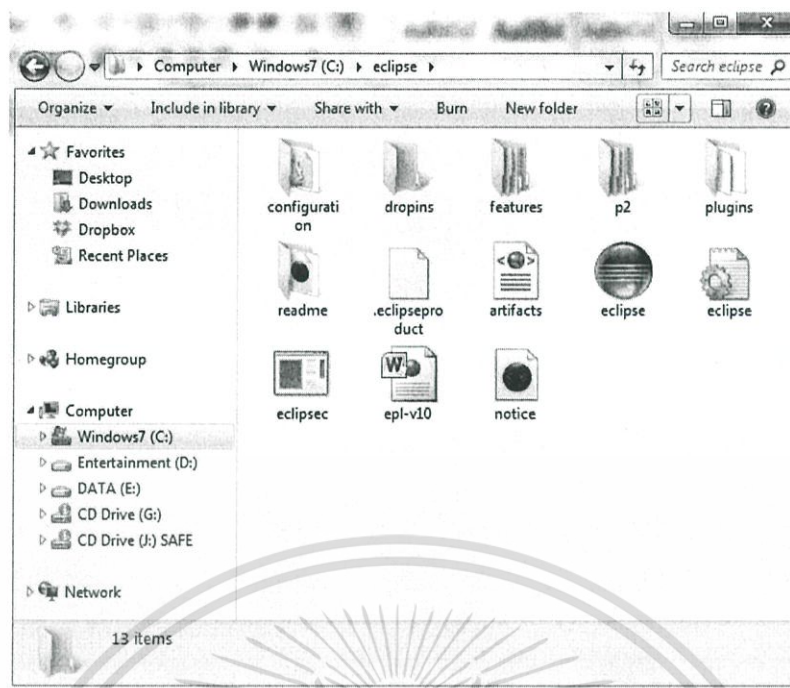
การติดตั้งโปรแกรม Eclipse ซึ่งเป็น IDE (Integrated Development Environment) หรือคือ เอดิเตอร์ (Editor) ที่ใช้ในการเขียนโค้ด ช่วยให้เขียนโค้ดได้ง่ายขึ้น และช่วยตรวจโค้ดให้ถูกต้อง แอนดรอยด์ จะใช้ Eclipse เป็นตัว gen Coding การติดตั้ง Eclipse มีขั้นตอนดังนี้

- 1 ไปที่ www.eclipse.org/downloads แล้วเลือกดาวน์โหลด Eclipse IDE for Java Developers โดยเลือกให้ตรงกับ Platform ดังรูปที่ 2.7



รูปที่ 2.7 แสดงวิธีดาวน์โหลด Eclipse

- 2 หลังจากดาวน์โหลดเสร็จจะได้ไฟล์ .zip ให้แตกไฟล์ .zip ไปที่ C:\ จะทำให้มีโฟลเดอร์ C:\eclipse เพิ่มขึ้นมา ซึ่งไฟล์ต่างๆของโปรแกรม Eclipse จะถูกรวบรวมไว้ในโฟลเดอร์นี้ ดังรูปที่ 2.8



รูปที่ 2.8 แสดงไฟล์ต่างๆของโปรแกรม Eclipse

- 3 การรันโปรแกรม Eclipse นั้นจะต้องดับเบิลคลิกที่ไฟล์ eclipse.exe ดังนั้นเพื่อความสะดวกให้สร้างชอร์ตคัทของไฟล์นี้ไว้บนเดสก์ทอป

2.5.3 การติดตั้ง Android SDK

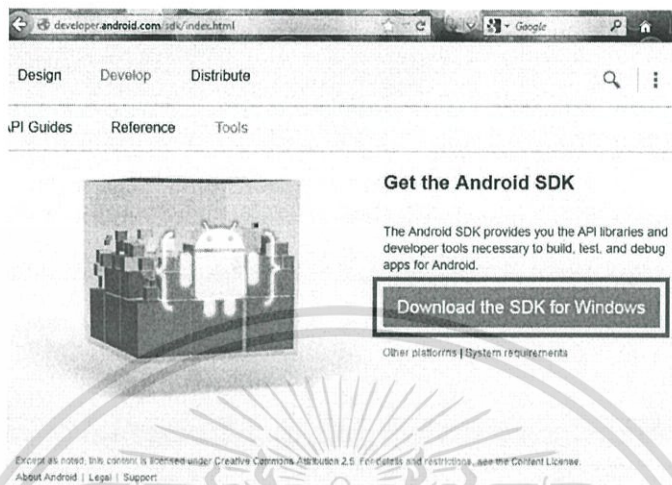
Android SDK หรือ Android Software Development Kit คือชุดพัฒนาซึ่งประกอบด้วยเครื่องมือและเฟรมเวิร์กสำหรับการพัฒนาแอปพลิเคชันแอนดรอยด์ โดย Android SDK แบ่งออกเป็น 2 ส่วน คือ SDK Starter Package กับ SDK Component

- 1 **SDK Starter Package** ประกอบด้วยชุดเครื่องมือหลัก (Core SDK Tools) ที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชันแอนดรอยด์ เครื่องมือเหล่านี้ส่วนใหญ่เป็นคำสั่งที่ต้องเรียกใช้ทางคอมมานด์ไลน์ (Command Line) แต่ถูกเปิดได้พัฒนาปลั๊กอินสำหรับ Eclipse ที่ช่วยให้เราเรียกใช้คำสั่งเหล่านี้ได้อย่างสะดวกจากใน Eclipse
- 2 **SDK Component** ประกอบด้วยเครื่องมือ, ไลบรารีสำหรับการเขียนโค้ด และอื่นๆที่ขึ้นกับเวอร์ชันของแอนดรอยด์ ซึ่งเรียกว่า Android Platform นอกจากนี้ SDK Component ยังรวมถึงคู่มือการใช้งาน ตัวอย่าง และส่วนเสริม (Add-ons) ต่างๆด้วย

การติดตั้ง Android SDK จะต้องติดตั้งส่วนของ SDK Starter Package ก่อน แล้วจึงใช้เครื่องมือใน SDK Starter Package ดาวน์โหลด SDK Component ที่ต้องการใช้งานมาติดตั้งเพิ่มเติม จึงจะได้ Android SDK ที่สมบูรณ์

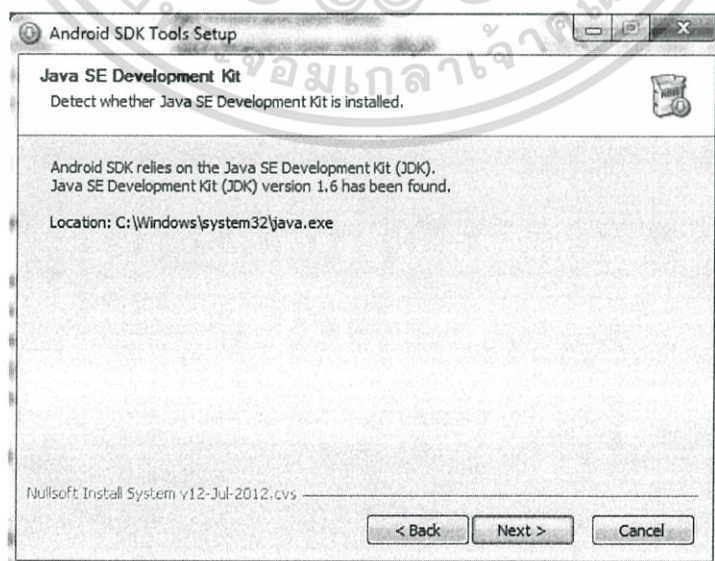
ขั้นตอนการติดตั้ง Android SDK (ทั้ง SDK Starter Package และ SDK Component) มีดังนี้

- 1) ไปที่ developer.android.com/sdk แล้วคลิก Download the SDK for Windows ดังรูปที่ 2.9 เพื่อดาวน์โหลดไฟล์ installer_r20.0.3-windows.exe ซึ่งไฟล์นี้ก็คือ SDK Starter Package



รูปที่ 2.9 แสดงวิธีดาวน์โหลด SDK Starter Package

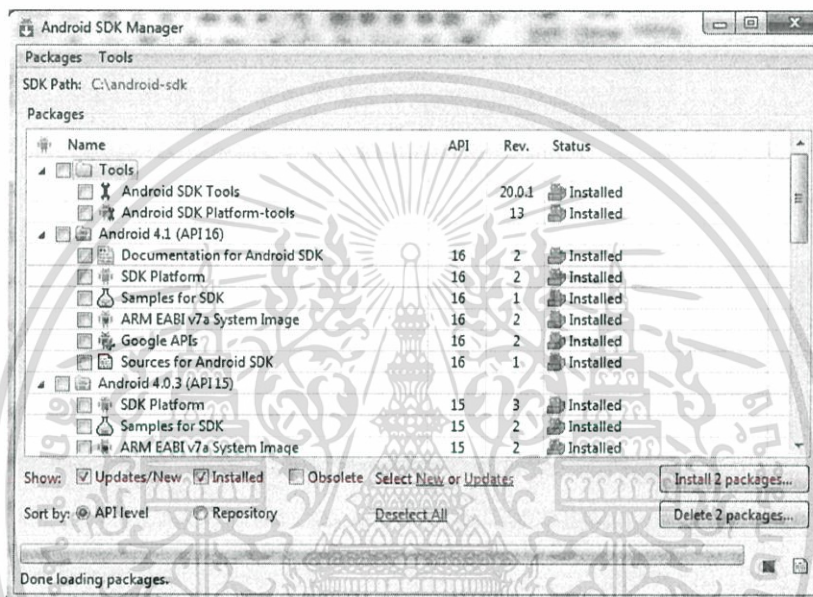
- 2) เมื่อดาวน์โหลดมาแล้ว ให้ดับเบิลคลิกเพื่อเริ่มการติดตั้ง
- 3) ในหน้าจอแรกของการติดตั้งจะมีการตรวจสอบว่ามี JDK อยู่ในระบบหรือไม่ ดังรูปที่ 2.10 เพราะการทำงานของ Android SDK ต้องอาศัย JDK อีกที่หนึ่ง ถ้าหากคุณติดตั้ง JDK แล้วแต่โปรแกรมบอกว่าไม่พบ ก็ให้คลิก Back แล้วคลิก Next ใหม่ก็จะสามารถติดตั้งต่อไปได้ (แต่ถ้าคุณไม่เจอปัญหานี้ก็ให้คลิก Next ไปเรื่อยๆตามปกติ)



รูปที่ 2.10 แสดงผลการตรวจสอบว่ามี JDK อยู่ในระบบหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) ติดตั้งไว้ที่ C:\android-sdk ให้จำตำแหน่งของโฟลเดอร์ที่ติดตั้ง Android SDK ไว้เพราะต้องระบุถึงโฟลเดอร์นี้อีกภายหลัง ตอนที่ติดตั้ง ADT Plug-in ให้กับ Eclipse และตอนที่ต้องการเรียกใช้เครื่องมือ (SDK Tools) ต่างๆทางคอมมานไลน์
- 5) คลิก Install เพื่อติดตั้งไฟล์ต่างๆลงไปที่ C:\android-sdk
- 6) ในหน้าจอสุดท้ายให้เลือก 옵션 Start SDK Manager (ปกติจะถูกเลือกไว้อยู่แล้ว) แล้วคลิก Finish จะปรากฏวินโดว์ Android SDK Manager สำหรับดาวน์โหลดคอมโพเนนต์ (SDK Component) มาติดตั้งเพิ่มเติม ดังรูปที่ 2.11



รูปที่ 2.11 แสดงวินโดว์ Android SDK Manager

- 7) โปรแกรมจะอ่านข้อมูลจากเซิร์ฟเวอร์ แล้วแสดงรายชื่อคอมโพเนนต์ที่สามารถดาวน์โหลดมาติดตั้งเพิ่มได้ ให้เลือก Android SDK Tools และ Android SDK Platform-tools ได้หัวข้อ Tools แล้วให้เลือกหัวข้อเวอร์ชันของแอนดรอยด์ที่เราจะพัฒนาแอปพลิเคชัน แล้วคลิก Install
- 8) ยืนยันการติดตั้ง โดยเลือก Accept All แล้วคลิก Install คอมโพเนนต์ต่างๆที่เลือกจะถูกดาวน์โหลดและติดตั้งลงในเครื่องคอมพิวเตอร์ที่ C:\android-sdk ซึ่งขั้นตอนนี้อาจใช้เวลาหลายนาทีขึ้นอยู่กับจำนวนคอมโพเนนต์ที่เลือกและความเร็วอินเทอร์เน็ต
- 9) เมื่อติดตั้งเสร็จแล้ว วินโดว์ Android SDK Manager จะบอกให้รู้ว่ามีคอมโพเนนต์ใดติดตั้งอยู่ในเครื่องบ้าง โดยให้สังเกตที่คอลัมน์ Status

2.5.4 การติดตั้งปลั๊กอินสำหรับ Eclipse (ADT Plug-in)

เพื่อให้การพัฒนาแอปพลิเคชันแอนดรอยด์ง่ายขึ้น กูเกิลจึงได้สร้าง Android Development ToolKit (ADT) ซึ่งเป็นปลั๊กอิน (Plug-in) สำหรับ Eclipse ขึ้นมา

ADT จะเพิ่มความสามารถและเครื่องมืออำนวยความสะดวกต่างๆที่เกี่ยวกับการพัฒนาแอปพลิเคชันแอนดรอยด์ให้กับ Eclipse เช่น การสร้างโปรเจกแอนดรอยด์ใหม่ การสร้างส่วนติดต่อผู้ใช้ (User Interface) ของแอปพลิเคชัน การดีบั๊กแอปพลิเคชันด้วยเครื่องมือใน Android SDK การสร้างแพ็คเกจไฟล์ (Package File) แล้วส่งไปรันอิมูเลเตอร์หรืออุปกรณ์แอนดรอยด์ เป็นต้น ช่วยให้เราพัฒนาแอปพลิเคชันได้สะดวกขึ้นมาก ซึ่งหากไม่มี ADT เราจะต้องเรียกใช้คำสั่งใน Android SDK เพื่อทำงานเหล่านี้เอง

ขั้นตอนการติดตั้งปลั๊กอิน ADT ให้กับ Eclipse ดังนี้

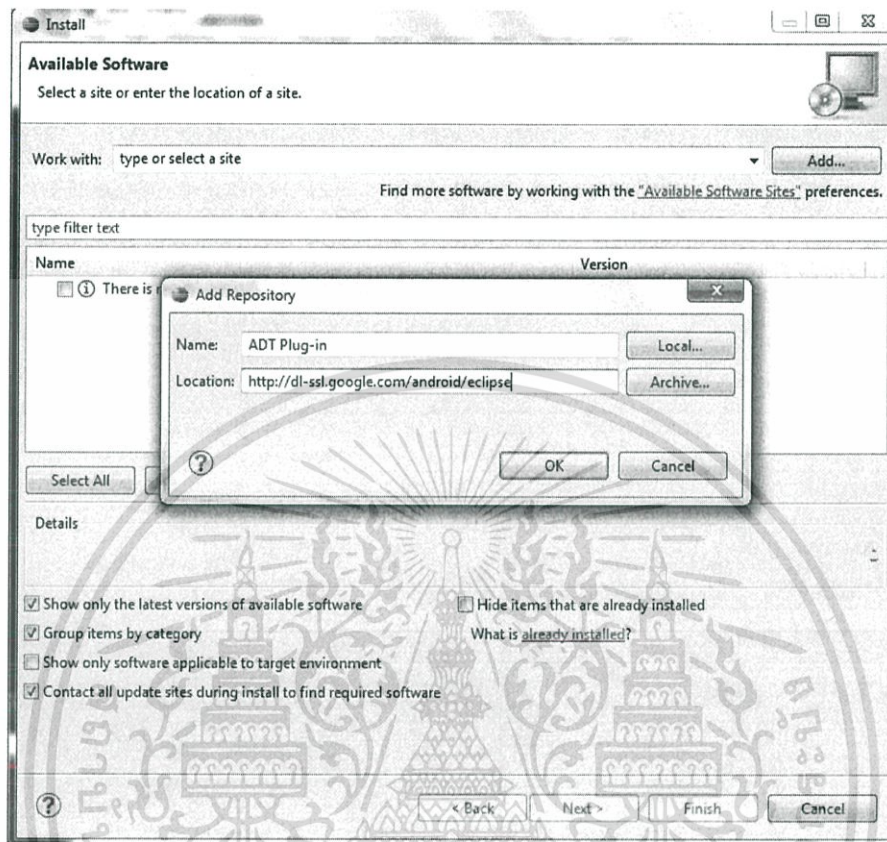
- 1) เปิดโปรแกรม Eclipse ขึ้นมา ถ้าหากมีหน้าจอถามว่าต้องการกำหนดเวิร์กสเปซ (Workspace) เป็นโฟลเดอร์ใด ให้ใช้ค่าดีฟอลต์ที่มีมาให้ แล้วคลิก OK
- เวิร์กสเปซ (Workspace) คือ โฟลเดอร์ที่ Eclipse ใช้จัดเก็บโปรเจกต่างๆที่เราพัฒนาขึ้น
- 2) คลิกเมนู Help > Install New Software จะปรากฏวินโดว์ Install
- 3) คลิกปุ่ม Add ที่มุมบนขวา ดังรูปที่ 2.12



รูปที่ 2.12 แสดงวินโดว์ Install

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) ในไดอะล็อกบ็อกซ์ Add Repository ให้พิมพ์ ADT Plug-in ลงในช่อง Name และพิมพ์ <http://dl-ssl.google.com/android/eclipse/> ลงในช่อง Location แล้วคลิก OK ดังรูปที่ 2.13



รูปที่ 2.13 แสดงไดอะล็อกบ็อกซ์ Add Repository

- 5) โปรแกรมจะติดต่อขอข้อมูลจากเซิร์ฟเวอร์ google.com และสักครู่นึงก็จะมีหัวข้อ Developer Tools เพิ่มขึ้นมาในช่อง Name ให้คลิกเลือกเซ็คบ็อกซ์ข้างหน้าหัวข้อนี้
- 6) คลิก Next ไปเรื่อยๆ โดยในหน้าสุดท้ายจะต้องเลือกอบช้น I accepts the terms of license agreements เพื่อยอมรับข้อตกลงการใช้งาน จึงจะคลิก Finish ได้
- 7) ถ้าหากมีไดอะล็อกบ็อกซ์แจ้งเตือนเรื่องความปลอดภัย และถามว่าต้องการติดตั้งต่อหรือไม่ ให้คลิก OK
- 8) เมื่อติดตั้งเสร็จแล้ว ให้ปิดแล้วเปิด Eclipse ใหม่

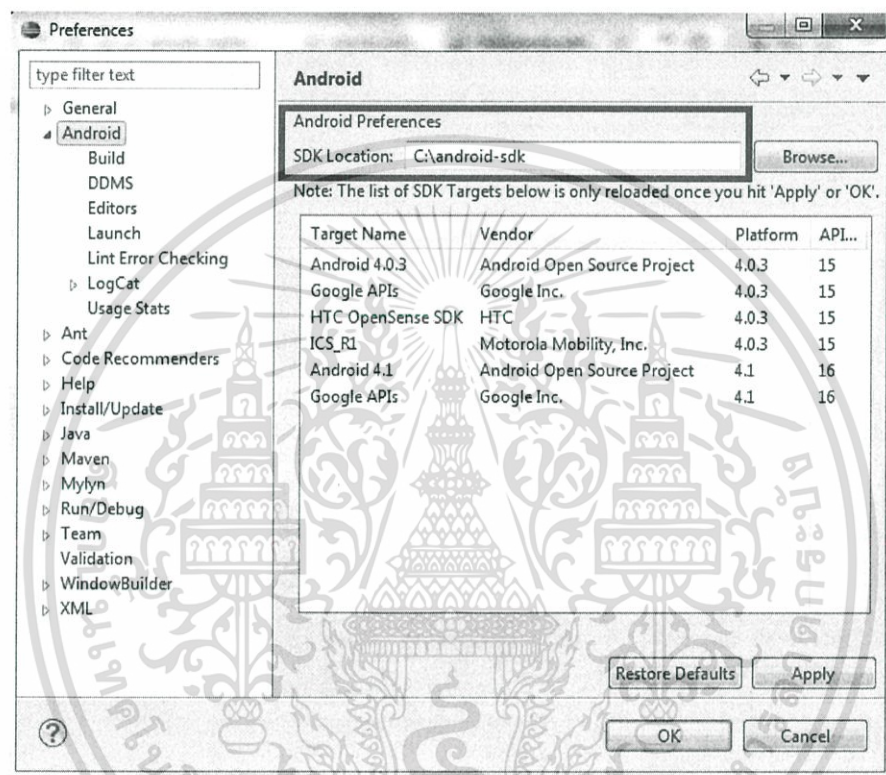
2.5.5 การระบุตำแหน่งของ Android SDK ให้กับ ADT

ต้องบอกให้ ADT รู้ว่า Android SDK ถูกติดตั้งไว้ที่โฟลเดอร์ใด เนื่องจากการทำงานของ ADT จะไปเรียกใช้คำสั่ง (โปรแกรมแบบคอมมานไลน์) ใน Android SDK อีกทีหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติเมื่อปิดแล้วเปิด Eclipse ใหม่ควรจะปรากฏหน้าจอสำหรับตั้งค่า Android SDK ให้เลือก Use existing SDKs แล้วระบุ C:\android-sdk ซึ่งเป็นตำแหน่งที่เราติดตั้ง Android SDK ไว้ลงในช่อง Existing Location แล้วคลิก Finish

หากไม่ปรากฏหน้าจอดังกล่าว ให้เข้าไปกำหนดใน Preferences โดยคลิกเมนู Windows > Preferences เพื่อเปิดวินโดว์ Preferences ขึ้นมา จากนั้นคลิกหัวข้อ Android ทางซ้าย ดังรูปที่ 2.14 พิมพ์ C:\android-sdk ลงในช่อง SDK Location แล้วคลิก OK

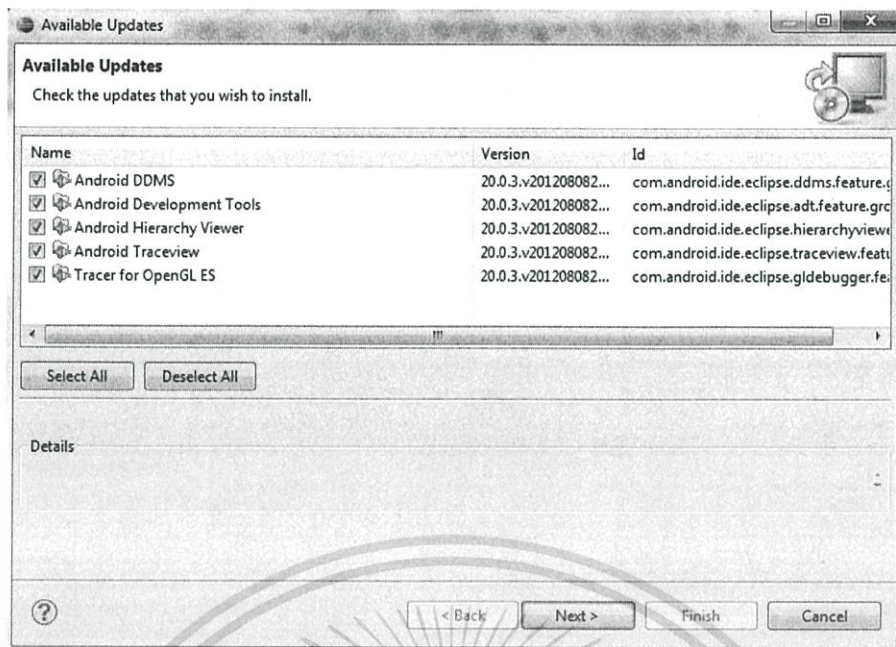


รูปที่ 2.14 แสดงวินโดว์ Preferences

2.5.6 การอัปเดต ADT Plug-in

ADT Plug-in นั้นถูกปรับปรุงอยู่อย่างต่อเนื่อง ทั้งการเพิ่มเติมความสามารถใหม่ๆ และการแก้ไขบั๊กต่างๆ ดังนั้นเมื่อมีเวอร์ชันใหม่ของ ADT Plug-in ออกมา จึงควรอัปเดตทันที มีวิธีการดังนี้

- 1) คลิกเมนู Help > Check for Updates ซึ่งถ้ามีอัปเดตของ ADT จะปรากฏหน้าจอดังรูปที่ 2.15



รูปที่ 2.15 แสดง ADT ที่มีการอัปเดตเวอร์ชันใหม่

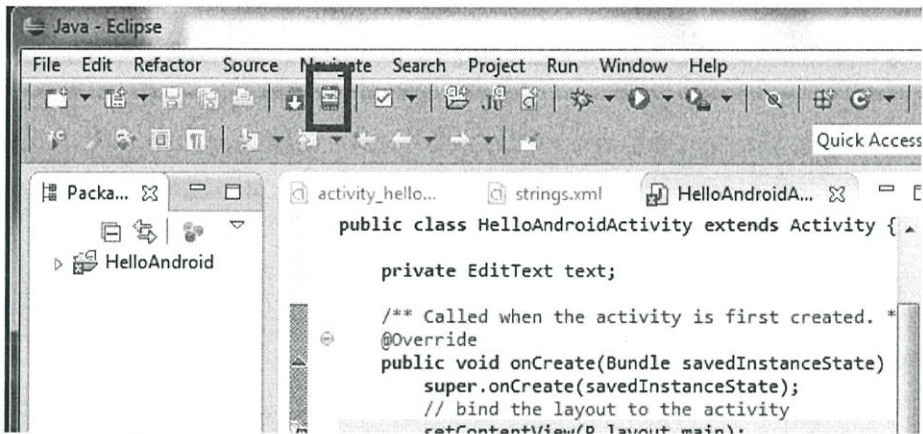
- 2) คลิก Next ไปเรื่อยๆ และคลิก Finish ในหน้าจอสุดท้าย หลังจากนั้นจึงปิดแล้วเปิด Eclipse ใหม่ ก็จะเสร็จสิ้นการอัปเดต ADT

บางครั้งการทำงานของ ADT ที่อัปเดตใหม่อาจผูกกับเครื่องมือรุ่นใหม่ใน Android SDK ดังนั้นต้องทำการอัปเดตแพ็คเกจ SDK Tools โดยใช้ Android SDK Manager ด้วยการเปิด Android SDK Manager ทำได้โดยดับเบิลคลิกไฟล์ SDK Manager.exe ใน C:\android-sdk หรือคลิก Windows > Android SDK Manager ใน Eclipse

2.6 การสร้าง AVD (Android Virtual Device)

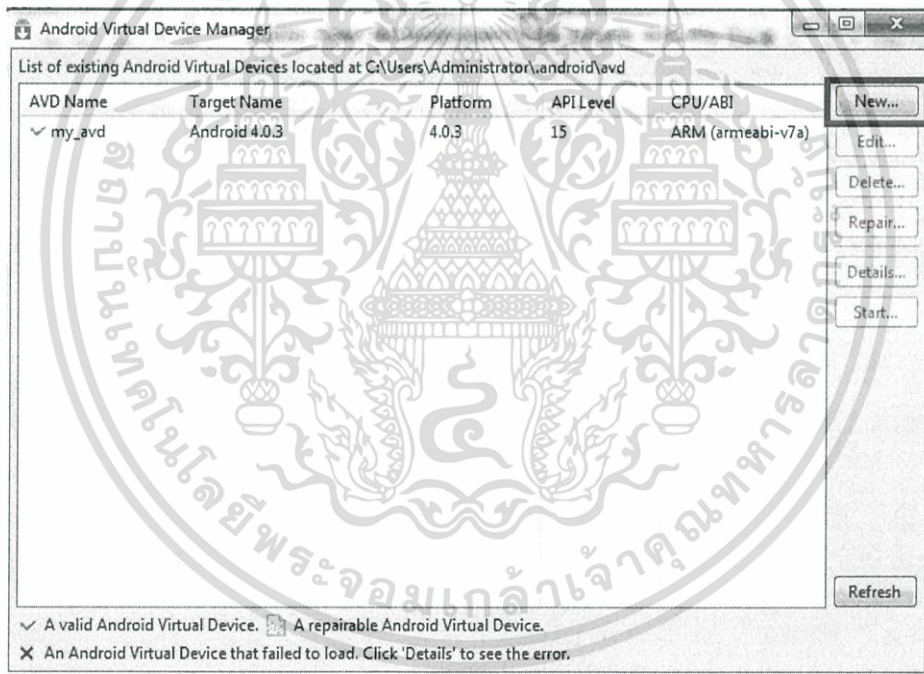
ในขณะที่สร้างโค้ดแอนดรอยด์อยู่นั้น การทดสอบความสมบูรณ์ของตัวโค้ด ส่วนใหญ่จะทำได้ทันทีที่ AVD และเมื่อโค้ดสมบูรณ์ 100% แล้วจึงไปทดสอบบนเครื่องจริง ฉะนั้นการมี AVD หรือ Android Virtual Device ไว้ทดสอบโค้ดบนเครื่องคอมพิวเตอร์ มีขั้นตอนดังนี้ (ชัชวดี พรหมบุตร. 2555)

- 1 เปิด Eclipse แล้วคลิก Android Virtual Device Manager ที่ไอคอน ดังรูปที่ 2.16



รูปที่ 2.16 แสดงไอคอนที่ใช้เปิด Android Virtual Device Manager

- 2 ปราบกวินโดว์ Android Virtual Device Manager กดปุ่ม New ที่อยู่ด้านขวา ดังรูปที่ 2.17 เพื่อสร้าง AVD



รูปที่ 2.17 แสดงวิธีการสร้าง ADV

- 3 กรอกรายละเอียดของ AVD ที่ต้องการ แล้วคลิก Create AVD ดังรูปที่ 2.18 โดยกรอกรายละเอียดดังนี้
- 1) Name ส่วนใหญ่จะชื่อเรียกเวอร์ชันของแอนดรอยด์
 - 2) Target เป็นส่วนที่บอกว่า จะให้ AVD ใช้ระบบปฏิบัติการแอนดรอยด์เวอร์ชันอะไร
 - 3) SD Card ขนาดของหน่วยความจำที่ต้องการเวอร์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) Skin เป็นส่วนของหน้าจอก สามารถเลือกได้ตามใจชอบ

Property Value

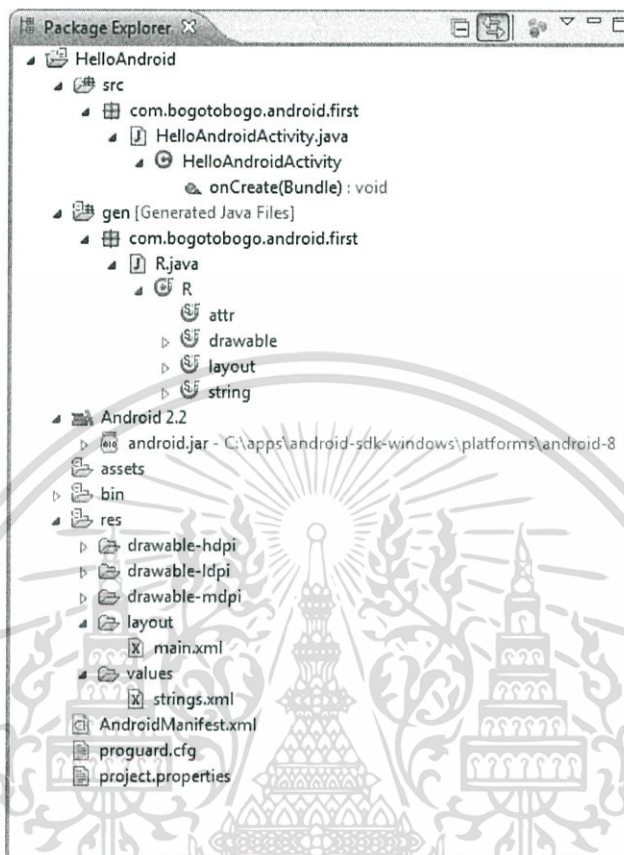
Abstracted LCD density	160
Max VM application hea...	48
Device ram size	512

รูปที่ 2.18 แสดงการกรอกรายละเอียด AVD ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 โครงสร้างโปรเจกแอนดรอยด์

ในการสร้าง Android Project ทุกครั้งจะมีโครงสร้างพื้นฐานของตัวโปรเจก ดังรูปที่ 2.19



รูปที่ 2.19 โครงสร้างโปรเจกแอนดรอยด์

รายละเอียดของโฟลเดอร์และไฟล์ที่สำคัญมีดังนี้ (พร้อมเลข หล่อวีจิตร. 2555:55)

- 1 โฟลเดอร์ **src** เก็บซอร์สไฟล์ (Source Files) ที่เขียนด้วยภาษาจาวา (ไฟล์ .java) ในที่นี้มีอยู่ไฟล์เดียวคือ HelloAndroidActivity.java ซึ่งเป็นแอกทิวิตีหลักและเป็น entry point ของแอปพลิเคชัน (จุดเริ่มต้นการทำงานของแอปพลิเคชัน)
- 2 โฟลเดอร์ **gen** เก็บไฟล์ .java ที่ ADT สร้างให้อัตโนมติ เช่น ไฟล์ R.java ที่อธิบายแล้ว
- 3 โฟลเดอร์ **assets** เป็นโฟลเดอร์ว่างตอนที่โปรเจกถูกสร้างขึ้นมา แอนดรอยด์เตรียมโฟลเดอร์นี้ไว้ให้เราเก็บไฟล์ที่เป็น asset ของแอปพลิเคชัน ซึ่งต่างจาก resource ตรงที่ asset จะไม่ถูกคอมไพล์ แต่จะถูกรวมไว้ใน .apk ตามรูปแบบดั้งเดิมของมัน รวมทั้งชื่อไฟล์ด้วย
- 4 โฟลเดอร์ **bin** คือ output folder ที่เก็บผลลัพธ์การบิลด์ เราจะพบไฟล์ .apk และรีซอร์สที่คอมไพล์แล้วได้ในโฟลเดอร์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5 ไฟล์ `res` เก็บริซอร์สของแอปพลิเคชัน ซึ่งแบ่งออกเป็นไฟล์ย่อยต่างๆตามประเภทของริซอร์ส เช่น `Layout File` จะอยู่ใน `res/layout`, ไฟล์รูปภาพจะอยู่ใน `res/drawable` หรือ `res/drawable-xxx` (สำหรับรูปภาพที่เราเตรียมไว้หลายๆความละเอียด), ค่าสตริงและค่าสีจะอยู่ใน `res/values` เป็นต้น
- 6 ไฟล์ `AndroidManifest.xml` คือ ไฟล์ที่ให้รายละเอียดเกี่ยวกับแอปพลิเคชันว่าประกอบด้วยคอมโพเนนต์ใดบ้าง และคุณสมบัติของคอมโพเนนต์เหล่านั้นเป็นอย่างไร
- 7 ไฟล์ `project.properties` เก็บการตั้งค่าของโปรเจกต์ เช่น `Build Target` เป็นต้น

2.8 องค์ประกอบของแอนดรอยด์แอปพลิเคชัน

คุณลักษณะอย่างหนึ่งของแอนดรอยด์ก็คือ เป็นแอปพลิเคชันที่สามารถใช้เป็นส่วนประกอบของแอปพลิเคชันอื่นๆได้ เช่น ถ้าต้องการสร้างแอปพลิเคชันให้แสดงการเลื่อนของรายการรูปภาพ โดยที่อาจมีแอปพลิเคชันส่วนอื่นที่ได้พัฒนาไว้แล้ว สามารถเรียกใช้แอปพลิเคชันในส่วนที่มีอยู่มาพัฒนาต่อได้ โดยไม่จำเป็นต้องพัฒนาขึ้นมาเอง เพื่อความสะดวกและรวดเร็วในการพัฒนาแอปพลิเคชันซึ่งเรียกแอปพลิเคชันเหล่านั้นว่า **Application Component** ซึ่งองค์ประกอบที่สามารถใช้ในการสร้างแอปพลิเคชัน ได้แก่

2.8.1 แอคทิวิตี (Activity)

Activity (User Interface) คือ คอมโพเนนต์ที่ใช้ในการแสดงผล ออกมาเพื่อให้ผู้ใช้งานได้เห็นและได้ใช้งาน โดยในแต่ละ Application นั้น อาจจะมี activity เดียว หรือ หลายๆ activity ก็ได้ หรือ อาจจะไม่มีการใช้ activity เลยก็ได้ และสิ่งที่อยู่ใน activity นั้นจะเรียกว่า view ซึ่งก็มีอยู่ในหลายรูปแบบ เช่น buttons, text fields, scroll bars, menu items, check boxes และอื่นๆ

แต่ละแอคทิวิตีจะแทนหน้าจอแสดงผลที่แอปพลิเคชันสามารถแสดงส่วนติดต่อผู้ใช้ได้ซึ่งแอคทิวิตีโดยส่วนใหญ่จะครอบคลุมพื้นที่ทั้งหมดของหน้าจอ และสามารถสร้างแอคทิวิตีให้มีลักษณะกึ่งโปร่งใส หรือ โฟลตติ้งได้

1 การสร้างแอคทิวิตี

สามารถสร้างแอคทิวิตีได้โดยสร้างคลาสที่สืบทอดจากคลาส Activity ซึ่งคลาสที่สร้างขึ้นมาจะสามารถกำหนดส่วนติดต่อผู้ใช้และกำหนดฟังก์ชันการทำงานต่างๆ ได้โค้ดโครงร่างพื้นฐานสำหรับแอคทิวิตีใหม่ (Prajak Boonjaratcha. 2554) ดังรูปที่ 2.20

```

package com.android.mobiledep.sample;

import android.app.Activity;
import android.os.Bundle;

public class MyMobiledep extends Activity {

    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

    }

}

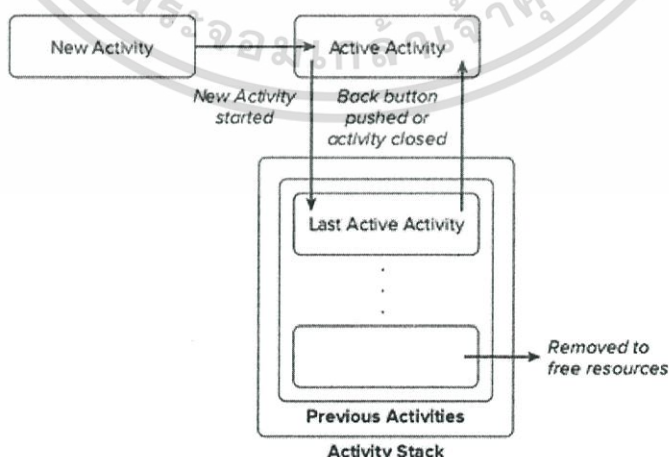
```

รูปที่ 2.20 โค้ดโครงร่างของแอคทิวิตี

จากโค้ดคลาสแอคทิวิตีข้างต้นจะแสดงหน้าจอว่างเปล่าที่รวมถึงส่วนจัดการวินโดว และจากนี้สามารถนำส่วนติดต่อผู้เข้ามาแสดงผลได้

2 แอคทิวิตีสแต็ก (Activity Stack)

แอคทิวิตีในระบบจะถูกจัดการในรูปแบบแอคทิวิตีสแต็ก (Activity stack) เมื่อแอคทิวิตีใหม่เริ่มขึ้นจะวางที่ด้านบนของสแต็กและเป็นแอคทิวิตีที่ทำงานและตอบสนองกับผู้ใช้ได้ ส่วนแอคทิวิตีก่อนหน้าจะถูกซ่อนทับอยู่ด้านล่างในสแต็กและจะไม่แสดงผล เมื่อหน้าที่หน้าจออีกครั้งจนกว่าแอคทิวิตีใหม่ด้านบนจะสิ้นสุดการทำงาน ดังรูปที่ 2.21



รูปที่ 2.21 แอนดรอยด์แอคทิวิตีสแต็ก

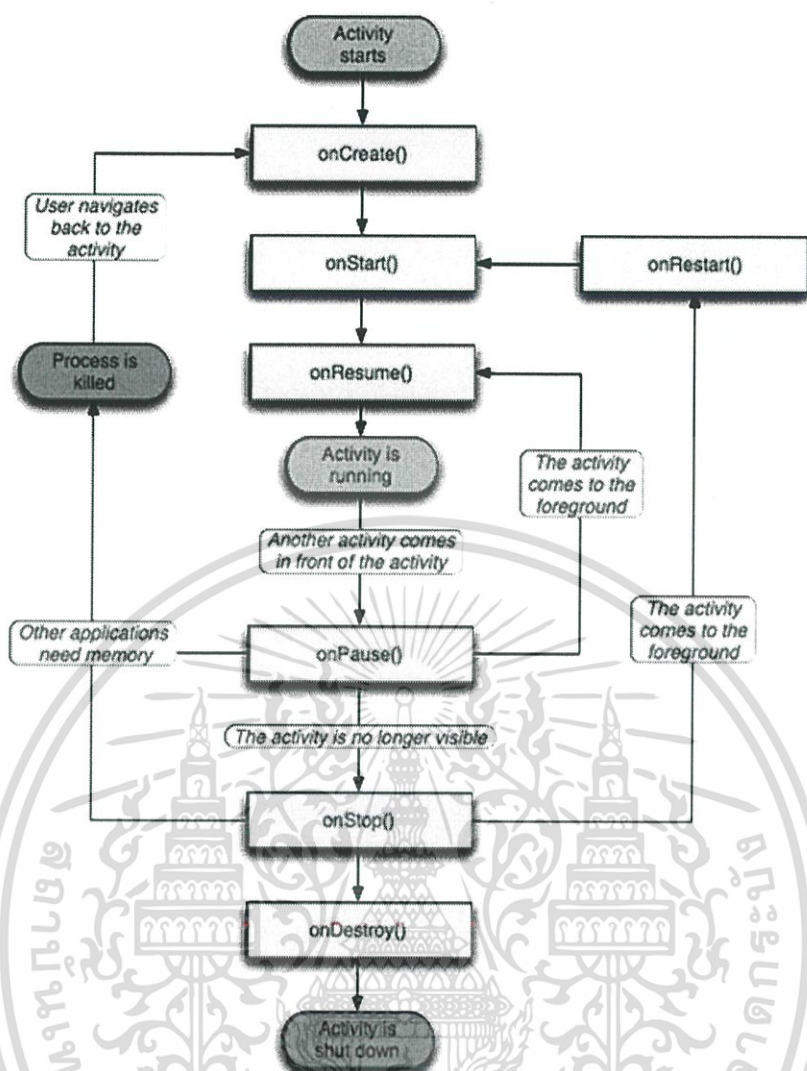
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งแอกทิวิตีจะมีสถานะการทำงานหลักๆ คือ (Prajak Boonjaratcha. 2554)

- 1) ถ้าแอกทิวิตีอยู่ในการทำงานเบื้องหน้าของหน้าจอ (บนสุดของสแต็ก) จะเป็นแอกทิวิตีที่อยู่ในสถานะใช้งานได้ (Active) หรือ กำลังทำงานอยู่ (Running)
- 2) ถ้าแอกทิวิตีไม่ได้โฟกัสแต่ยังคงมองเห็นได้ เนื่องจากแอกทิวิตีใหม่แสดงผลไม่เต็มจอหรือมีลักษณะโปร่งใสถูกโฟกัสอยู่ ด้านบนสุดของแอกทิวิตีเดิม ทำให้แอกทิวิตีเดิมอยู่ในสถานะหยุดการทำงานชั่วคราว (Paused) แอกทิวิตีที่หยุดการทำงานชั่วคราวจะยังคงอยู่แต่จะถูกทำลายได้โดยระบบในกรณีที่หน่วยความจำเหลืออยู่น้อยมาก
- 3) ถ้าแอกทิวิตีถูกบดบังอย่างสมบูรณ์โดยแอกทิวิตีอื่น แอกทิวิตีที่ถูกซ่อนทับจะอยู่ในสถานะหยุด (Stopped) และยังคงรักษาข้อมูลเดิมไว้อยู่ อย่างไรก็ตาม แอกทิวิตีนี้ไม่สามารถให้ผู้ใช้มองเห็นได้และมักจะถูกทำลายโดยระบบเมื่อมีความต้องการหน่วยความจำในที่อื่น
- 4) ถ้าแอกทิวิตีถูกหยุดชั่วคราว (Paused) หรือหยุด (Stopped) ระบบสามารถนำแอกทิวิตีออกจากหน่วยความจำ โดยการร้องขอให้แอกทิวิตีสิ้นสุดการทำงานหรือทำลายโดยระบบ เมื่อมันต้องถูกแสดงผลอีกครั้ง จะต้องเริ่มการทำงานใหม่และคืนค่าสถานะก่อนหน้า

3 วงจรชีวิตของแอกทิวิตี (Android Activity Lifecycle)

แอนดรอยด์ถูกออกแบบมาเพื่อใช้งานกับอุปกรณ์มือถือ ซึ่งจะมีทรัพยากรของระบบที่จำกัด จึงจำเป็นต้องมีกลไกที่คอยจัดการบริหารทรัพยากรเหล่านั้น เรียกว่า วงจรชีวิตของแอนดรอยด์แอกทิวิตี (Android Activity Lifecycle) ซึ่งจะกำหนดสถานะหรืออีเวนต์ที่แอกทิวิตีต้องผ่านตั้งแต่ตอนที่ถูกสร้าง ขึ้นจนกว่าจะเสร็จสิ้นการทำงาน ดังรูปที่ 2.22



รูปที่ 2.22 แสดงวงจรชีวิตของแอคทิวิตี

แต่ละแอคทิวิตีจะตรวจสอบและตอบสนองกับอีเวนต์เหล่านี้ โดยอินสแตนซ์เมธอดที่โอเวอร์ไรด์จากคลาสแอคทิวิตี (จักรชัย โสอินทร์ และพงษ์ศธร จันทร์ขอย, 2554:10) ได้แก่

- 1) **onCreate (Bundle)** จะถูกเรียกเมื่อแอคทิวิตีถูกสร้างขึ้นครั้งแรก ซึ่งปกติจะมีการสร้างส่วนติดต่อผู้ใช้หรือวิว (View) ต่างๆ เปิดไฟล์ข้อมูลถาวร (persistent datafiles) ที่ใช้ในแอคทิวิตี และกำหนดค่าเริ่มต้นของแอคทิวิตี
- 2) **onStart** จะถูกเรียกก่อนที่แอคทิวิตีจะแสดงผลออกมาที่หน้าจอ เมื่อ onStart ทำงานเสร็จถ้าแอคทิวิตีสามารถเป็นโฟร์กราวด์แอคทิวิตีบนหน้าจอ การควบคุมจะถูกส่งต่อไปยัง onResume แต่ถ้าแอคทิวิตีไม่สามารถเป็นโฟร์กราวด์แอคทิวิตีได้ การควบคุมจะถูกส่งต่อไปยังเมธอด onStop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) **onResume** จะถูกเรียกหลังจาก onStart ถ้าแอกทิวิตีเป็น โพรแกรวด์แอกทิวิตี ณ จุดนี้แอกทิวิตีจะทำงานและตอบสนองกับผู้ใช้ นอกจากนี้ onResume ยังถูกเรียกอีกครั้ง ถ้าแอกทิวิตีที่ไม่ได้เป็น โพรแกรวด์กลับมาเป็น โพรแกรวด์แอกทิวิตีอีกครั้ง
- 4) **onPause** จะถูกเรียกเมื่อระบบทำการนำแอกทิวิตีใหม่มาเป็น โพรแกรวด์ ซึ่งแอกทิวิตีเดิมจะไม่สามารถเข้าถึงหน้าจอและจำเป็นต้องหยุดการดำเนินการ ที่ใช้แบตเตอรี่และซีพียูไซเคิลชั่วคราว โดยแอกทิวิตีจำเป็นต้องใช้ประโยชน์จากเมธอดนี้ในการจัดเก็บสถานะหรือค่าต่างๆ ที่จำเป็นจะต้องใช้ในกรณีที่แอกทิวิตีกลับมาเป็น โพรแกรวด์แอกทิวิตีอีกครั้ง ซึ่งแอนดรอยด์ยังไม่รับประกันว่าแอกทิวิตีจะสามารถกลับมาเป็น โพรแกรวด์ได้เสมอไป เนื่องจากถ้าอุปกรณ์มีหน่วยความจำไม่เพียงพอและไม่มีหน่วยความจำเสมือน (Virtual memory) บนดิสก์ที่ขยายต่อได้ ดังนั้นแอกทิวิตีจะถูกทำลายและคืนทรัพยากรให้แก่ระบบ
- 5) **onStop** จะถูกเรียกเมื่อแอกทิวิตีไม่สามารถมองเห็นได้อีก หรือแอกทิวิตีกำลังจะถูกทำลาย ถัดจากนี้เมธอด onRestart จะถูกเรียก ถ้าแอกทิวิตีจะกลับมาแสดงผลและตอบสนองกับผู้ใช้ หรือเมธอด onDestroy จะถูกเรียกเมื่อแอกทิวิตีสิ้นสุดการทำงาน
- 6) **onRestart** จะถูกเรียกเมื่อแอกทิวิตีกำลังจะกลับมาแสดงผลให้ผู้ใช้เห็นอีกครั้ง
- 7) **onDestroy** เมธอดสุดท้ายที่แอกทิวิตีจะทำงานก่อนที่จะถูกทำลายไป ซึ่งอาจเป็นเพราะแอกทิวิตีกำลังจะจบการทำงาน (มีการเรียกเมธอด finish) หรือระบบขอทำลายแอกทิวิตี เพื่อนำหน่วยความจำไปใช้ แต่หากระบบเหลือหน่วยความจำน้อยมากก็อาจไม่เรียกมายังเมธอด onDestroy เลย แต่จะทำลายโปรเซสของแอกทิวิตีไป

โดยทั่วไปช่วงเวลาของแอกทิวิตีมีอยู่ 3 แบบ คือ ช่วงเวลาทั้งหมด, ช่วงเวลาที่แอกทิวิตีแสดงผลออกมาให้เห็น และช่วงเวลาที่แอกทิวิตีอยู่เบื้องหน้า มีรายละเอียดดังนี้

- 1) **ช่วงเวลาทั้งหมด (Entire Lifetime)** เกิดขึ้นระหว่างการเรียกเมธอด onCreate ครั้งแรกไปจนถึงการเรียกเมธอด onDestroy โดยเราจะทำ Initialization และจองทรัพยากรต่างๆ ใน onCreate และปลดปล่อยทรัพยากรใน onDestroy ยกตัวอย่างเช่น ถ้าแอปพลิเคชันของเรามีเซิร์ฟเวอร์สำหรับดาวน์โหลดข้อมูลจากเน็ตเวิร์ค เราจะสร้างเซิร์ฟเวอร์นั้นขึ้นมาใน onCreate และหยุดการทำงานของเซิร์ฟเวอร์ใน onDestroy
- 2) **ช่วงเวลาที่แอกทิวิตีแสดงผลออกมาให้เห็น (Visible Lifetime)** เกิดขึ้นระหว่างการเรียกเมธอด onStart ไปจนถึงการเรียกเมธอด onStop ในช่วงเวลานี้ผู้ใช้จะเห็นแอกทิวิตีบนจอ แม้ว่ามันอาจจะไม่ได้อยู่เบื้องหน้าและโต้ตอบกับผู้ใช้ได้ก็ตาม สองเมธอดนี้คือจุดเริ่มต้น และสิ้นสุดที่เราจะควบคุมทรัพยากรต่างๆ ที่เกี่ยวข้องกับการแสดงผลของแอกทิวิตี
- 3) **ช่วงเวลาที่แอกทิวิตีอยู่เบื้องหน้า (Foreground Lifetime)** เกิดขึ้นระหว่างการเรียกเมธอด onResume ไปจนถึงการเรียกเมธอด onPause ในช่วงเวลานี้แอกทิวิตีจะอยู่เบื้องหน้าและเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โต้ตอบกับผู้ใช้ได้ แอคทิวิตีหนึ่งๆอาจมีการสลับจากเบื้องหน้าไปเบื้องหลังและเบื้องหลังมาเบื้องหน้าบ่อยๆ (onResume > onPause > onResume > onPause ฯลฯ) ดังนั้นโค้ดในเมธอดทั้งสองจึงควรเป็นโค้ดที่ทำงานได้เร็ว

2.8.2 เซอร์วิส (Service)

Service (Service Provider) คือ คอมโพเนนต์ที่ไม่มีส่วนของการแสดงผลหรือส่วนติดต่อกับผู้ใช้ แต่ถูกเรียกมารันอยู่เบื้องหลัง (background process) โดยเซอร์วิสนั้นอาจจะมีการทำอะไรบางอย่าง ได้แก่ การติดต่อรับส่งข้อมูลผ่านเครือข่าย หรือคำนวณค่าต่างๆ แล้วก็ทำการส่งผลลัพธ์นั้นไปแสดงยัง activity ก็ได้ การเปลี่ยนแปลงคอนเทนต์โพรไวเดอร์ เรียกใช้อินเทนต์ หรือเรียกการแจ้งเตือน (Notification) และยังสามารถจัดการอีเวนต์เมื่อแอคทิวิตีของแอปพลิเคชันไม่สามารถมองเห็นหรือกำลังถูกปิดได้ (ลานหน้าบ้าน, 2555)

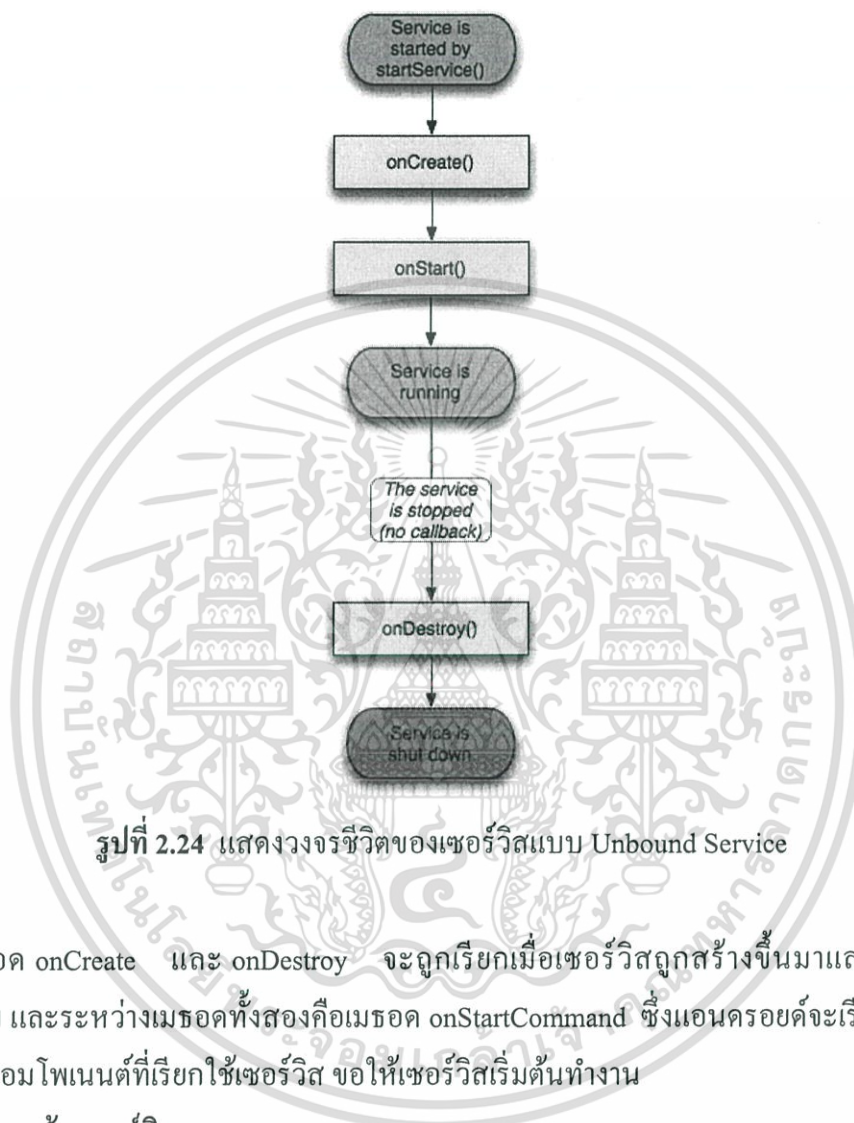
โดยทั่วไปเราจะใช้แอคทิวิตีเป็นตัวสั่งให้เซอร์วิสเริ่มหรือหยุดทำงาน ซึ่งการสั่งให้เซอร์วิสเริ่มหรือหยุดทำงานเท่านั้น เรียกว่า **Unbound Service** ยกตัวอย่างเช่น แอปพลิเคชัน Music ของแอนดรอยด์ ซึ่งมีแอคทิวิตีเป็นหน้าจอสำหรับควบคุมการเล่นเพลง เมื่อเราสั่งเล่นเพลง แอคทิวิตีก็จะไปเรียกเซอร์วิสมาทำหน้าที่นี้ ทำให้เราฟังเพลงได้ต่อเนื่องแม้ว่าเราจะออกจากหน้าจอที่ใช้ควบคุมการเล่นเพลงแล้ว (พร้อมเลิส หล่อวิจิตร, 2555:225) แสดงดังรูปที่ 2.23



รูปที่ 2.23 แสดงลักษณะการเรียกใช้คอมโพเนนต์เซอร์วิสของแอปพลิเคชัน Music

1 วงจรชีวิตของเซอร์วิส (Service Lifecycle)

การสร้างเซอร์วิสจะทำได้โดยสร้างคลาสที่สืบทอดจากคลาสเซอร์วิสของแอนดรอยด์ ซึ่งเซอร์วิสนั้นมีวงจรชีวิตที่เรียบง่ายกว่าแอกทิวิตี (พร้อมเลข ห่อวีจิตร. 2555:226) ดังรูปที่ 2.24



รูปที่ 2.24 แสดงวงจรชีวิตของเซอร์วิสแบบ Unbound Service

เมธอด onCreate และ onDestroy จะถูกเรียกเมื่อเซอร์วิสถูกสร้างขึ้นและถูกทำลายตามลำดับ และระหว่างเมธอดทั้งสองคือเมธอด onStartCommand ซึ่งแอนดรอยด์จะเรียกมายังเมธอดนี้เมื่อคอมโพเนนต์ที่เรียกใช้เซอร์วิส ขอให้เซอร์วิสเริ่มต้นทำงาน

2 การสร้างเซอร์วิส

การสร้างเซอร์วิสสามารถทำได้ดังนี้ (ลานหน้าบ้าน. 2555)

- 1) สร้างคราสจากอินสแตนซ์ Service (ในโปรแกรม Eclipse สามารถทำได้โดยคลิกขวาที่โปรเจกต์ และเลือกเมนู New >> Class และกำหนดค่าของซูเปอร์คราสเป็น android.app.Service)
- 2) ประกาศ Service ในไฟล์ AndroidManifest.xml โดยการเพิ่มคำสั่งนี้ <service android:name=".myService"></service>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

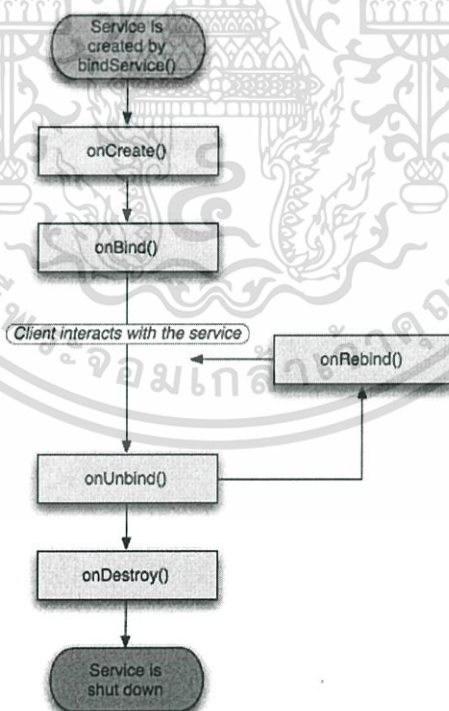
- 3) โอเวอร์ไรด์เมธอด onCreate() และ onDestroy() ซึ่งเมธอดเหล่านี้จะทำงานเมื่อ Service เริ่มหรือหยุดทำงาน
- 4) โอเวอร์ไรด์เมธอด onBind() ซึ่งจะทำงานเมื่อมีการเชื่อมโยงระหว่าง Service กับคอมโพเนนต์
- 5) สั่งให้ Service เริ่มทำงานโดยใช้คำสั่งจากภายนอกมาสั่งการ เพราะ Service ไม่สามารถเริ่มการทำงานด้วยตัวเองได้

3 Bound Service

Bound Service หมายถึงเซอร์วิสที่อนุญาตให้คอมโพเนนต์อื่นๆของแอปพลิเคชัน (โดยทั่วไปคือแอกทิวิตี) ผูกกับมันได้ เพื่อที่จะเรียกใช้ความสามารถของเซอร์วิส (พร้อมเลข หล่อวีจิตร. 2555:232)

ประโยชน์ของ Bound Service คือเราอาจแยกการทำงานบางอย่างไปสร้างเป็นเซอร์วิสที่รันอยู่เบื้องหลัง แล้วจึงเรียกใช้เซอร์วิสนั้นจากคอมโพเนนต์อื่นๆ อีกที นอกจากนี้เราอาจอนุญาตให้แอปพลิเคชันอื่นมาเรียกใช้เซอร์วิสของเราได้ โดยผ่านทาง IPC (Interprocess Communication)

Bound Service จะมีวงจรชีวิตแตกต่างจาก Unbound Service ดังรูปที่ 2.25



รูปที่ 2.25 วงจรชีวิตของ Bound Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ Client ต้องการใช้งาน Bound Service จะต้องเรียกเมธอด Context.bindService เพื่อผูกตัวมันเข้ากับเซอร์วิส ซึ่งแอนดรอยด์จะสร้างออบเจกต์ของเซอร์วิสขึ้น (onCreate จะทำงาน) แล้วเรียกเมธอด onBind ของเซอร์วิสให้อัตโนมัติ เมธอดนี้ต้องส่งคืนออบเจกต์ IBinder กลับออกไป ซึ่ง Client จะเข้าถึงเซอร์วิสได้โดยผ่านทาง IBinder

ความแตกต่างระหว่าง Bound Service กับ Unbound Service คือ สำหรับ Bound Service เราสามารถเรียกใช้ความสามารถต่างๆของเซอร์วิสโดยผ่านทางเมธอดที่เซอร์วิสเตรียมไว้ให้ ในขณะที่ Unbound Service นั้นเราทำได้เพียงสั่งให้เซอร์วิสเริ่มหรือหยุดทำงานเท่านั้น (พร้อมเลข ให้อวีจิตร. 2555:232)

2.8.3 บรอดคาสต์รีซีฟเวอร์ (Broadcast receiver)

Broadcast receiver (Data Provider) คือ คอมโพเนนต์ที่ใช้สำหรับคอยรับ และตอบสนองต่อเหตุการณ์ต่างๆ ที่เกิดขึ้น เช่น เมื่อแบตเตอรี่ต่ำ, ผู้ใช้ทำการเปลี่ยนภาษา, มีการโทรออก, มีข้อความเข้า และอื่นๆถึงแม้ broadcast receiver จะไม่มีส่วนของการแสดงผล แต่มันก็สามารถที่จะเรียก activity ขึ้นมาแสดงผลให้ผู้ใช้ได้ หรืออาจจะใช้สิ่งที่เรียกว่า Notification Manager ซึ่งจะเป็นตัวที่แจ้งเตือนในรูปแบบของการสั่น, การแสดงไฟกระพริบที่หน้าจอ หรือการส่งเสียงออกมา โดยจะมี icon แสดงอยู่บน status bar เพื่อให้ผู้ใช้กดเข้าไปเปิดดูเหตุการณ์ที่เกิดขึ้น

การสร้าง Broadcast Receiver จะทำได้โดยสร้างคลาสที่สืบทอดจากคลาส BroadcastReceiver ของแอนดรอยด์ แล้วโอเวอร์ไรด์เมธอด onReceive เพื่อระบุโค้ดที่เราต้องการให้ทำเมื่อได้รับอินเทนต์เข้ามา ซึ่งออบเจกต์ BroadcastReceiver จะมี Lifetime อยู่เฉพาะในช่วงที่เมธอด onReceive ทำงานเท่านั้น เมื่อใดที่เมธอดนี้ return กลับมา แอนดรอยด์จะถือว่า BroadcastReceiver ทำงานเสร็จสิ้นแล้ว และพร้อมจะทำลายออกจากหน่วยความจำทันที (พร้อมเลข ให้อวีจิตร. 2555:241)

2.8.4 อินเทนต์ (Intent)

แอนดรอยด์มีคอมโพเนนต์หลัก (Core Components) ที่ใช้ในการสร้างแอปพลิเคชันอยู่ 3 ประเภท ได้แก่ แอคทิวิตี, เซอร์วิส และ Broadcast Receiver ซึ่งคอมโพเนนต์เหล่านี้จะเรียก (activate) กันและกันโดยใช้อินเทนต์ นั่นคือ อินเทนต์จะเป็นเสมือน Message ที่คอมโพเนนต์ต่างๆ ส่งถึงกัน โดยคอมโพเนนต์เหล่านี้อาจจะอยู่ในแอปพลิเคชันเดียวกันหรือคนละแอปพลิเคชันกันก็ได้

อินเทนต์ คือ โครงสร้างข้อมูลที่เก็บรายละเอียดของการกระทำที่จะถูกดำเนินการ หรือรายละเอียดของเหตุการณ์บางอย่างที่เกิดขึ้นและถูกประกาศออกไป เราจะใช้อินเทนต์ในกรณีต่างๆ ดังนี้ (พร้อมเลข ให้อวีจิตร. 2555:239)

- 1 ส่งอินเทนต์ให้กับเมธอด Context.startActivity เพื่อรันแอคทิวิตี หรือบอกรันแอคทิวิตีที่รันอยู่แล้วให้ทำอะไรบางอย่าง

- 2 ส่งอินเทนต์ให้กับเมธอด `Context.startService` เพื่อรันเซอร์วิส หรือส่งให้กับเมธอด `Context.bindService` เพื่อผูก Client เข้ากับเซอร์วิส
- 3 ส่งอินเทนต์ให้กับเมธอด `Context.sendBroadcast` เพื่อกระจาย (broadcast) อินเทนต์ออกไป ซึ่งกรณีนี้อินเทนต์จะถูกรับไปดำเนินการโดย Broadcast Receiver ที่ประกาศตัวเองว่าสนใจในอิน-เทนต์ประเภทนั้นๆ การส่งอินเทนต์แบบกระจายออกไปแบบนี้ส่วนใหญ่จะทำโดยระบบแอนดรอยด์เอง เช่น เมื่อมีสายเรียกเข้า แอนดรอยด์จะ broadcast อินเทนต์ `Action_Phone_State_Changed` ออกไป หรือเมื่อแบตเตอรี่ใกล้หมดก็จะ broadcast อิน-เทนต์ `Action_Battery_Low` เป็นต้น

2.8.5 Content provider

Content provider (System Event Listener) คือกลุ่มของข้อมูลที่สร้างขึ้นจากแอปพลิเคชัน เพื่อให้แอปพลิเคชันอื่นๆ ได้นำไปใช้ โดยการจับเก็บข้อมูลของ content provider นั้นจะอยู่ในลักษณะของไฟล์, ฐานข้อมูล SQLite และอื่นๆ ตัวอย่าง Application ที่ใช้งาน Content Provider ที่เห็นชัดเจนที่สุดคือ โปรแกรม Contacts ที่แสดงรายชื่อใน Contacts ของเรา (Developers, 2548)

2.8.6 การแจ้งเตือน (Notifications)

ในแอปพลิเคชันสามารถใช้การแจ้งเตือนเพื่อแจ้งเตือนต่อผู้ใช้ โดยไม่จำเป็นต้องใช้แอกทิวิตี การแจ้งเตือนจะถูกจัดการโดยส่วนจัดการการแจ้งเตือน (Notification Manager) และมีความสามารถในการสร้างไอคอนในแถบสถานะ แสดงข้อมูลเพิ่มเติมในหน้าต่างแถบสถานะแบบขยาย แสดงไฟสว่าง (LED) สั่นเครื่องโทรศัพท์ ส่งเสียงต่างๆ

การใช้งานการแจ้งเตือนเป็นวิธีที่ดีกว่าใช้องค์ประกอบแอปพลิเคชันที่มองไม่เห็น (บรอดคาสต์รีซีฟเวอร์, เซอร์วิส และแอกทิวิตีที่มองไม่เห็น) ในการแจ้งเตือนผู้ใช้เมื่อมีเหตุการณ์ต่างๆ ที่สำคัญและสามารถใช้เพื่อบ่งชี้ถึงเซอร์วิสที่กำลังทำงานอยู่เบื้องหลัง (Developers, 2548)

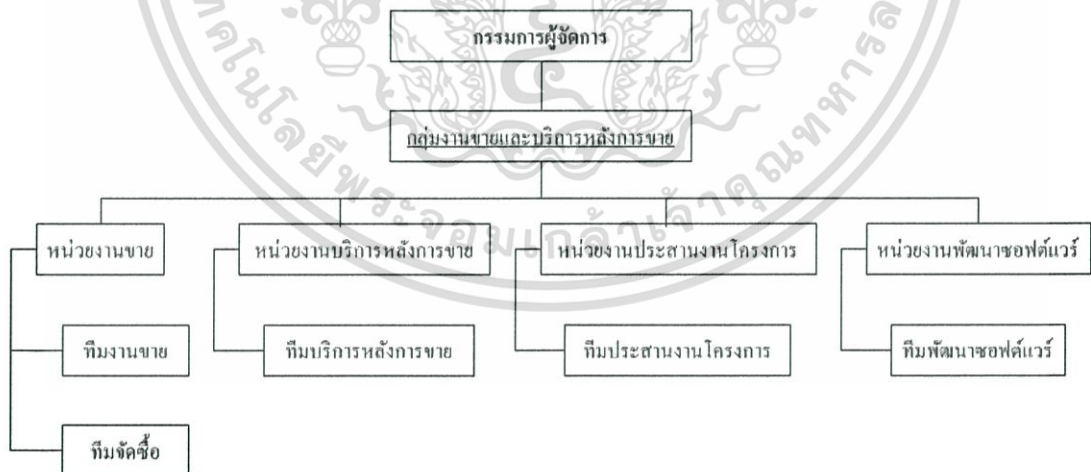
บทที่ 3

การศึกษาและวิเคราะห์ระบบงานปัจจุบัน

การศึกษาและวิเคราะห์ระบบปัจจุบันเป็นขั้นตอนที่สำคัญที่ทำให้ทราบถึงขั้นตอนการทำงานของระบบ ปัญหาที่เกิดจากระบบรวมทั้งความต้องการของระบบงานใหม่ ตลอดจนเข้าใจระบบธุรกิจขององค์กร เพื่อนำมาวิเคราะห์ปัญหาและกำหนดแนวทาง ขอบเขตของการพัฒนาต่อไป

3.1 ภาพรวมขององค์กร

การประสานงานโครงการ เป็นการทำงานหนึ่งในสังกัดกลุ่มงานขายและบริการหลังการขาย ซึ่งมีหน้าที่เป็นตัวกลางในการติดต่อกับลูกค้าและผู้พัฒนาระบบเกี่ยวกับการรับฟังความต้องการของลูกค้า การแก้ไขปัญหาของระบบหลังการขาย และรายงานผลการทำงานให้แก่กรรมการผู้จัดการรับทราบ โดยโครงสร้างการบริหารงานของกลุ่มงานขายและบริการหลังการขาย ประกอบด้วย 4 สายงานที่เกี่ยวข้อง คือ สายงานขาย สายงานบริการหลังการขาย สายงานประสานงานโครงการ สายงานพัฒนาซอฟต์แวร์ ดังรูปที่ 3.1



รูปที่ 3.1 ภาพรวมโครงสร้างกลุ่มงานขายและบริการหลังการขาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับงานประสานงานโครงการ เป็นตัวกลางในการติดต่อระหว่างหน่วยงานภายนอก (ลูกค้า) และหน่วยงานภายในองค์กร โดยแต่ละหน่วยงานจะมีความรับผิดชอบ ดังต่อไปนี้

- 1 หน่วยงานขาย ประกอบด้วย 2 ทีมงานหลัก คือ
 - ทีมงานขาย ทำหน้าที่ติดต่อ และนำเสนอระบบให้กับลูกค้า
 - ทีมงานจัดซื้อ ทำหน้าที่ดูแล จัดการเกี่ยวกับสินค้าประเภทฮาร์ดแวร์ เช่น UPS, หน้าจอคอมพิวเตอร์ เป็นต้น
- 2 หน่วยงานบริการหลังการขาย ประกอบด้วย
 - ทีมงานบริการหลังการขาย (เซอร์วิส) ทำหน้าที่ดูแล แก้ไขและซ่อมบำรุงให้แก่ผู้ใช้ระบบหรือลูกค้า เมื่อระบบเกิดขัดข้อง
- 3 หน่วยงานพัฒนาซอฟต์แวร์ ประกอบด้วย
 - ทีมงานพัฒนาซอฟต์แวร์ ทำหน้าที่พัฒนาและแก้ไขซอฟต์แวร์ของระบบ เมื่อได้รับความต้องการจากลูกค้า และซอฟต์แวร์เกิดขัดข้อง

3.2 การดำเนินงานปัจจุบัน

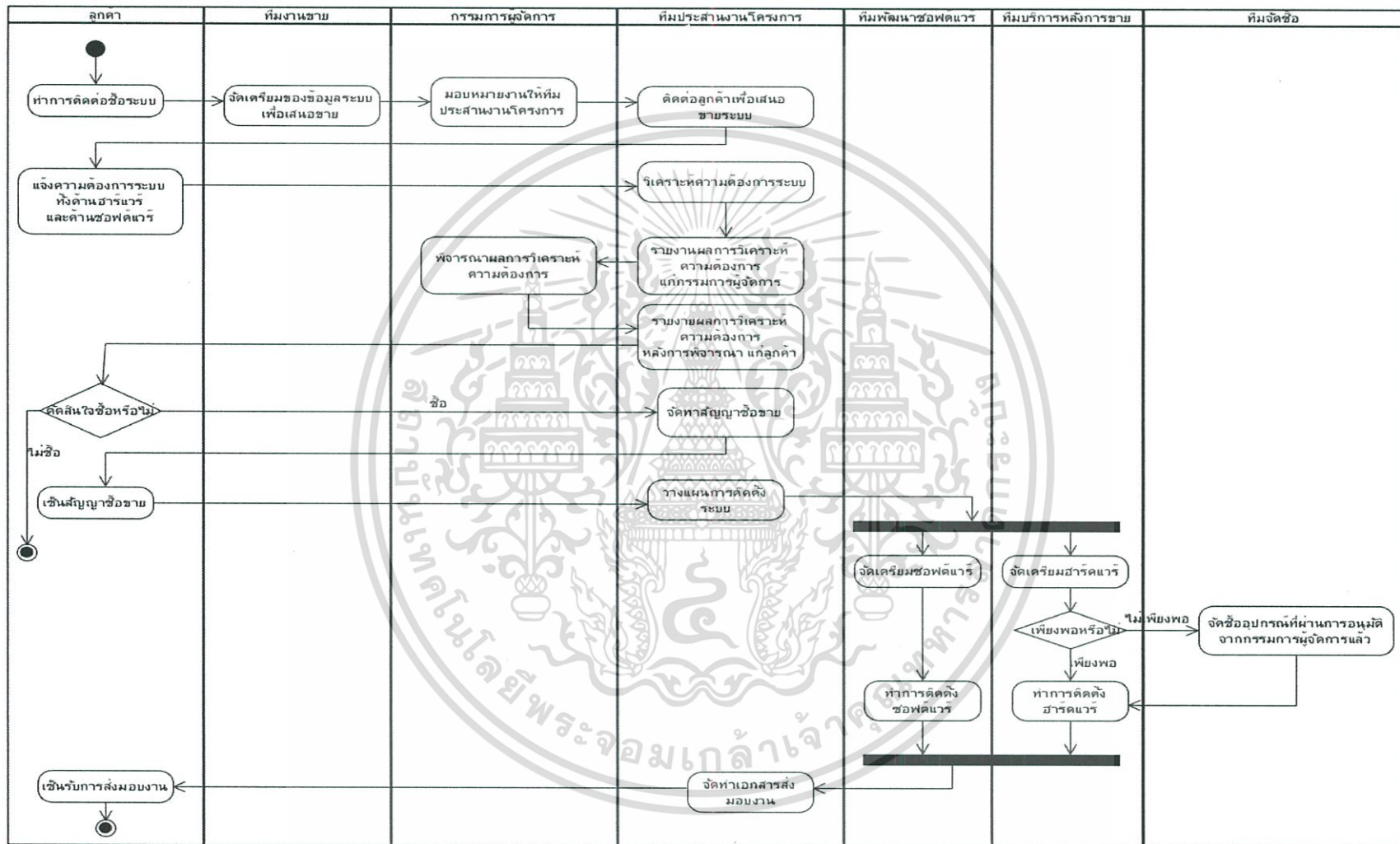
หน่วยงานประสานงานโครงการมีหน้าที่รับผิดชอบการประสานงานโครงการกับหน่วยงานต่างๆ ทั้งหน่วยงานภายในและหน่วยงานภายนอก (ลูกค้า) อาทิ การรับความต้องการด้านซอฟต์แวร์และฮาร์ดแวร์ของลูกค้า เพื่อนำไปวิเคราะห์ว่าตรงตามซอฟต์แวร์และฮาร์ดแวร์ที่มีอยู่หรือไม่ หรือสามารถเพิ่ม แก้ไขและปรับเปลี่ยนซอฟต์แวร์หรือฮาร์ดแวร์ได้หรือไม่ ในระหว่างการเจรจาก่อนการขาย การตรวจสอบและการแก้ไขปัญหาที่เกิดขึ้นหลังการขาย และการมอบหมายงานให้แก่หน่วยงานภายในที่รับผิดชอบโครงการ ซึ่งการประสานงานโครงการสามารถแบ่งได้ 2 ลักษณะ ดังนี้

3.2.1 การประสานงานโครงการด้านการขาย หมายถึง การประสานงานโครงการที่เกี่ยวข้องกับความต้องการของลูกค้าเป็นหลัก โดยมีขั้นตอนการดำเนินงาน ดังนี้

- 1 เมื่อได้รับการติดต่อจากลูกค้า ทีมงานขายจะทำการเตรียมข้อมูลเกี่ยวกับระบบเพื่อใช้ในการเสนอขาย
- 2 หนึ่งในทีมประสานงานโครงการที่ได้รับมอบหมายงานจากกรรมการผู้จัดการให้รับผิดชอบโครงการดังกล่าว ทำการติดต่อกับลูกค้า เพื่อให้ทีมขายแสดงรายละเอียดเกี่ยวกับลักษณะของระบบประกอบการตัดสินใจในการซื้อขาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3 หลังจากการนำเสนอระบบ ลูกค้าทำการแจ้งความต้องการทั้งด้านฮาร์ดแวร์และด้านซอฟต์แวร์ อาทิเช่น ขนาดของป้อม ขนาดของหน้าจอคอมพิวเตอร์ รูปแบบรายงานที่ต้องการจากระบบ เป็นต้น
 - 4 ผู้ประสานงานทำการวิเคราะห์ความต้องการว่าตรงตามซอฟต์แวร์และฮาร์ดแวร์ที่มีอยู่หรือไม่ หรือสามารถเพิ่ม แก้ไขและปรับเปลี่ยนซอฟต์แวร์หรือฮาร์ดแวร์อย่างไร จากนั้นรายงานผลการวิเคราะห์ความต้องการให้แก่กรรมการผู้จัดการพิจารณา ก่อนทำการรายงานผลให้แก่ลูกค้า
 - 5 หลังจากลูกค้าตัดสินใจซื้อระบบ ผู้ประสานงานจัดทำสัญญาซื้อขาย
 - 6 ลูกค้าเซ็นสัญญาซื้อขาย ผู้ประสานงานโครงการทำการวางแผนการเข้าไปติดตั้งระบบ โดยส่วนฮาร์ดแวร์ได้ทำการมอบหมายให้ทีมเซอร์วิสที่ได้รับมอบหมายให้ดูแลรับผิดชอบโครงการ จัดเตรียมอุปกรณ์ฮาร์ดแวร์ให้ตรงตามความต้องการของลูกค้า และส่วนซอฟต์แวร์ได้ทำการมอบหมายให้กับทีมพัฒนาซอฟต์แวร์ที่มีหน้าที่รับผิดชอบโครงการ จัดเตรียมซอฟต์แวร์ให้ตรงตามความต้องการ
 - 7 ในกรณีที่ทีมเซอร์วิสพบว่าอุปกรณ์ไม่เพียงพอ ก็จะทำการติดต่อกับทีมจัดซื้อให้จัดหาอุปกรณ์
 - 8 ทีมตั้งชื่อทำรายการอุปกรณ์ที่จะสั่งซื้อพร้อมทั้งราคา เสนอแก่กรรมการผู้จัดการให้ทำการพิจารณาอนุมัติการสั่งซื้อ
 - 9 ทีมเซอร์วิสและทีมพัฒนาซอฟต์แวร์ทำการติดตั้งระบบให้ตรงตามแผนที่วางไว้
 - 10 ผู้ประสานงานโครงการจัดทำหนังสือส่งมอบงานให้แก่ลูกค้า
- หากลูกค้าเซ็นรับการส่งมอบงานก็จะเป็นการสิ้นสุดขั้นตอนในการประสานงานโครงการด้านการขาย ซึ่งแสดงดังรูปที่ 3.2



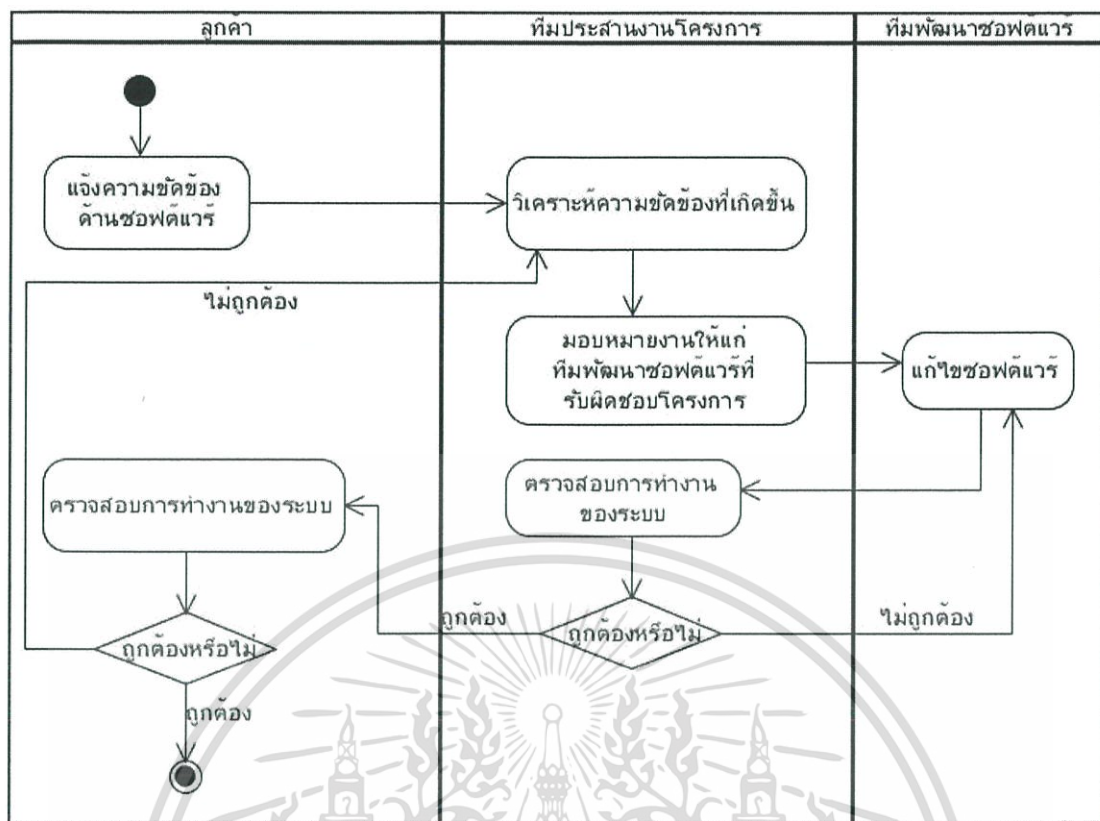
รูปที่ 3.2 แอคทิวิตีไดอะแกรมแสดงการประสานงาน โครงการด้านการขาย

3.2.2 การประสานงานโครงการด้านการบริการหลังการขาย หมายถึง การประสานงานในการดูแลรักษาระบบให้ทำงานได้อย่างปกติ ซึ่งเป็นการประสานงานภายในองค์กรเป็นหลัก การประสานงานโครงการด้านบริการหลังการขายแบ่งเป็น 2 ลักษณะดังนี้

การประสานงานโครงการด้านการบริการหลังการขายด้านซอฟต์แวร์ เป็นการประสานงานระหว่างลูกค้า (ผู้ใช้ระบบ) ผู้ประสานงานโครงการ (นักวิเคราะห์ระบบ) และผู้พัฒนาซอฟต์แวร์ เพื่อให้ระบบซอฟต์แวร์ที่จัดซื้อสามารถทำงานได้อย่างปกติ โดยมีขั้นตอนการดำเนินการ ดังนี้

- 1 เมื่อลูกค้าแจ้งความขัดข้องของระบบด้านซอฟต์แวร์กับผู้ประสานงานโครงการ ผู้ประสานงานโครงการจะทำการวิเคราะห์ความขัดข้องที่เกิดขึ้น หากลูกค้าแจ้งความขัดข้องของระบบมาไม่ชัดเจน ผู้ประสานงานโครงการต้องออกไปปฏิบัติงานนอกสถานที่ เพื่อตรวจสอบความขัดข้องที่ได้รับแจ้งดังกล่าว
- 2 ผู้ประสานงานมอบหมายงานให้แก่ทีมพัฒนาซอฟต์แวร์ที่รับผิดชอบโครงการ เพื่อแก้ไขซอฟต์แวร์ให้ตรงตามการวิเคราะห์ความขัดข้องที่เกิดขึ้น
- 3 ผู้ประสานงานโครงการทำการตรวจสอบระบบทางด้านซอฟต์แวร์ว่าถูกต้อง และใช้งานได้ปกติหรือไม่ หากยังไม่สามารถใช้งานได้ปกติ ก็ทำการแจ้งทีมพัฒนาซอฟต์แวร์ให้แก้ไข
- 4 หลังจากผ่านการตรวจสอบการทำงานจากผู้ประสานงานโครงการแล้ว ผู้ใช้ระบบหรือลูกค้าทำการตรวจสอบการทำงานและข้อมูลที่ได้จากระบบว่าถูกต้องหรือไม่ หากไม่ถูกต้องก็ทำการแจ้งผู้ประสานงานโครงการ เพื่อทำการวิเคราะห์ใหม่

เมื่อลูกค้าหรือผู้ใช้ระบบตรวจสอบการทำงานของระบบว่าทำงานได้ปกติ ได้ข้อมูลที่ถูกต้อง จึงถือว่าการประสานงานการบริการหลังการขายด้านซอฟต์แวร์เสร็จสมบูรณ์ ซึ่งแสดงดังรูปที่ 3.3



รูปที่ 3.3 แอคทิวิตีไดอะแกรมแสดงการประสานงานโครงการด้านการบริการหลังการขาย
ด้านซอฟต์แวร์

การประสานงานโครงการด้านการบริการหลังการขายด้านฮาร์ดแวร์ เป็นการประสานงานระหว่างลูกค้า (ผู้ใช้ระบบ) ผู้ประสานงานโครงการ และทีมบริการหลังการขาย (ทีมเซอร์วิส) เพื่อให้ระบบฮาร์ดแวร์ที่ขัดข้องสามารถทำงานได้อย่างปกติ โดยมีขั้นตอนการดำเนินการ ดังนี้

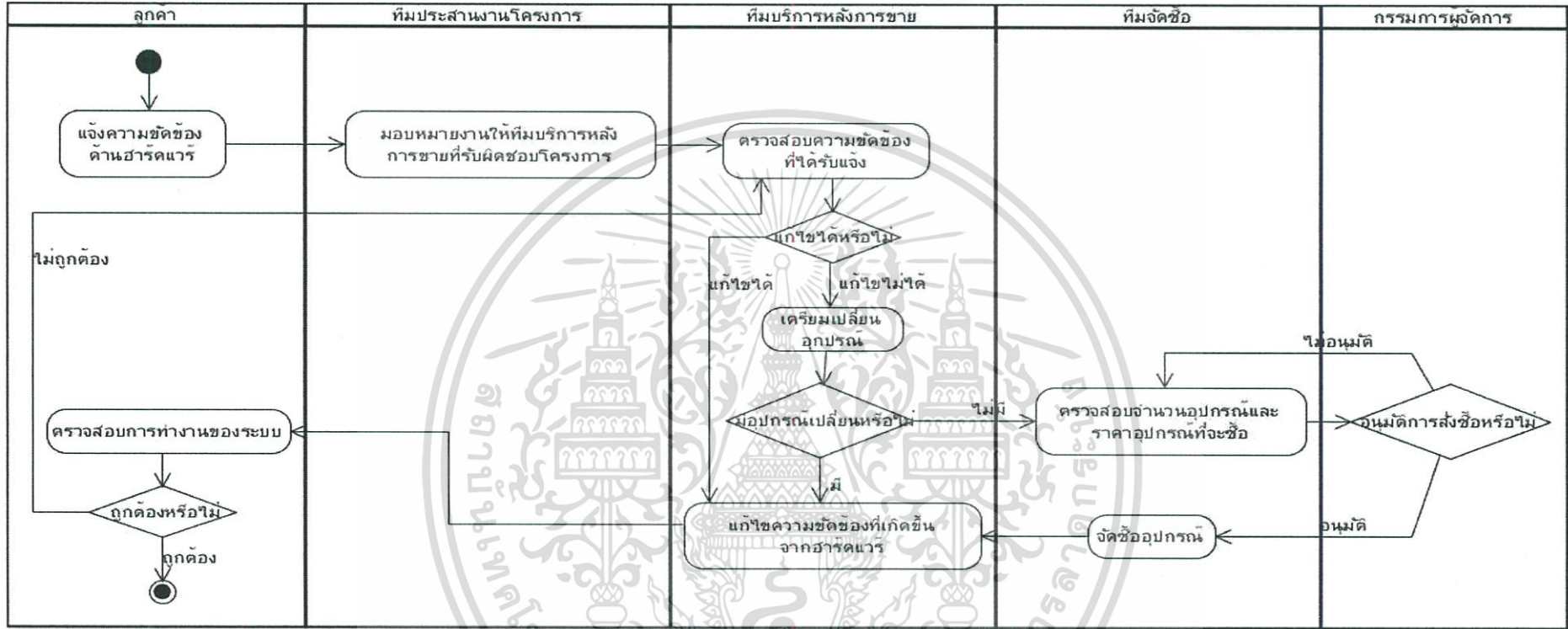
- 1 เมื่อลูกค้าแจ้งความขัดข้องของระบบด้านฮาร์ดแวร์กับผู้ประสานงานโครงการ ผู้ประสานงานโครงการจะทำการมอบหมายงานให้แก่ทีมบริการหลังการขายที่รับผิดชอบโครงการ
- 2 ทีมบริการหลังการขายตรวจสอบและทำการแก้ไขความขัดข้องที่ได้รับแจ้ง หากอุปกรณ์ฮาร์ดแวร์ไม่สามารถแก้ไข ซ่อมแซมได้ ก็ให้ทำการเปลี่ยนแปลงอุปกรณ์ฮาร์ดแวร์ดังกล่าวให้แก่ลูกค้า
- 3 ในกรณีที่อุปกรณ์ฮาร์ดแวร์ที่ทำการเปลี่ยนให้แก่ลูกค้าไม่เพียงพอ ทีมจัดซื้อจัดทำรายการจำนวนอุปกรณ์และราคาอุปกรณ์ที่ต้องจัดซื้อเสนอแก่กรรมการผู้จัดการให้พิจารณาเพื่ออนุมัติการสั่งซื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่มบริการหลังการขายทำการแก้ไข ซ่อมแซม หรือเปลี่ยนแปลงอุปกรณ์ฮาร์ดแวร์ที่
เกิดความขัดข้องเรียบร้อยแล้ว ลูกค้าหรือผู้ใช้ระบบทำการตรวจสอบการทำงานของระบบหลังการ
แก้ไขว่าสามารถทำงานได้ปกติหรือไม่ หากทำงานได้ปกติ ถือว่าการประสานงานการบริการหลัง
การขายด้านฮาร์ดแวร์เสร็จสมบูรณ์ ซึ่งแสดงดังรูปที่ 3.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แลททิวิตีไดอะแกรมแสดงการประสานงานโครงการด้านการบริการหลังการขายด้านฮาร์ดแวร์

3.3 วิธีการเก็บรวบรวมข้อมูลความต้องการของผู้ใช้

การเก็บรวบรวมข้อมูลสำหรับนำไปใช้ในการวิเคราะห์และออกแบบแอปพลิเคชันสนับสนุนระบบบริหารจัดการปัญหา เพื่อให้ได้ข้อมูลที่ครบถ้วนสมบูรณ์ มีวิธีการดำเนินการดังนี้

- 1 การสัมภาษณ์ผู้มีส่วนเกี่ยวข้องกับการประสานงาน โครงการ ซึ่งประกอบด้วย ทีมประสานงาน โครงการ ทีมพัฒนาซอฟต์แวร์ ทีมบริการหลังการขาย หน่วยงานภายนอก (ลูกค้า)
- 2 การสังเกตขั้นตอนการทำงาน แต่ละขั้นตอนเกิดปัญหาใดบ้าง

เมื่อรวบรวมข้อมูลความต้องการของผู้ใช้งานเรียบร้อยแล้ว จะนำข้อมูลที่ได้มาศึกษาวิเคราะห์ถึงปัญหาที่เกิดขึ้นของระบบงานเดิม เพื่อนำมาใช้ปรับปรุงและออกแบบระบบงานใหม่

3.4 ปัญหาและข้อจำกัดของการดำเนินงานปัจจุบัน

จากการศึกษาและวิเคราะห์ถึงขั้นตอนการทำงานของระบบปัจจุบันของหน่วยงานประสานงานโครงการ พบว่ามีปัญหาและข้อจำกัดของการปฏิบัติงานในปัจจุบัน ดังนี้

1 การเกิดความล่าช้าในการตรวจสอบหรือแก้ไขระบบ

หน่วยงานประสานงานโครงการ เมื่อได้รับการแจ้งความขัดข้องทางซอฟต์แวร์จากผู้ใช้ระบบหรือลูกค้า ผู้ประสานงานโครงการซึ่งรับผิดชอบโครงการดังกล่าว ทำหน้าที่วิเคราะห์ความขัดข้องของระบบที่ได้รับการแจ้ง และหากการติดต่อสื่อสารกับผู้ใช้ระบบไม่ชัดเจน ผู้ประสานงานโครงการจะต้องออกไปปฏิบัติงานนอกสถานที่ เพื่อตรวจสอบความขัดข้องที่ผู้ใช้ระบบแจ้ง แล้วจึงนำกลับมามอบหมายงานให้แก่ทีมพัฒนาซอฟต์แวร์หรือทีมบริการหลังการขายที่รับผิดชอบโครงการ แต่หากเกิดความขัดข้องขึ้นพร้อมกันหลายๆสถานที่ หรือมีงานที่จะต้องปฏิบัติงานนอกสถานที่ในวันเดียวกันกับการตรวจสอบความขัดข้อง ส่งผลให้ผู้ประสานงานมอบหมายงานให้แก่ทีมพัฒนาซอฟต์แวร์หรือทีมบริการหลังการขายช้า ทำให้เกิดความล่าช้าในการแก้ไขระบบ

2 ความผิดพลาดจากการเข้าใจไม่ตรงกันหรือการได้รับข้อมูลไม่ครบถ้วน

การติดต่อสื่อสารระหว่างผู้ใช้ระบบกับผู้ประสานงาน ผู้ประสานงานกับทีมพัฒนาซอฟต์แวร์และผู้ประสานงานกับทีมบริการหลังการขาย เกี่ยวกับการรับความต้องการของลูกค้า หรือการแจ้งความขัดข้องของระบบทั้งด้านฮาร์ดแวร์และด้านซอฟต์แวร์ที่เกิดขึ้นหลังการขาย ซึ่งใช้ช่องทาง

การสื่อสาร ได้แก่ โทรศัพท์ อีเมล ส่งผลให้บางครั้งเกิดความเข้าใจไม่ตรงกันเกี่ยวกับความซับซ้อนของระบบและข้อมูลที่ได้รับอาจไม่ครบถ้วน ทำให้การแก้ไขระบบอาจผิดพลาด ไม่ครบถ้วนหรือไม่ตรงตามความต้องการ ตามส่วนที่ผู้ใช้ระบบแจ้ง

3 การไม่มีองค์ความรู้ให้แก่พนักงานใหม่

ในกรณีที่รับพนักงานที่พัฒนาระบบและทีมบริการหลังการขายใหม่ หากมอบหมายให้แก้ไขความซับซ้อนของระบบทั้งด้านซอฟต์แวร์และด้านฮาร์ดแวร์ ส่งผลให้การแก้ไขอาจเกิดความผิดพลาดไม่ตรงจุดที่เกิดความซับซ้อนที่ถูกต้อง และเกิดความล่าช้า ทำให้ผู้ใช้ระบบหรือลูกค้าไม่พอใจ ภาพลักษณ์บริษัทอาจดูแย่ลง

3.5 แนวทางการแก้ไขปัญหา

จากปัญหาที่พบในการดำเนินงานปัจจุบันดังที่ได้กล่าวมาข้างต้นนั้น มีแนวทางการแก้ไขปัญหาด้วยการพัฒนาระบบบริหารจัดการปัญหาให้เป็นการทำงานผ่านอุปกรณ์พกพาที่ใช้ระบบปฏิบัติการแอนดรอยด์ เพื่อเพิ่มความสะดวกในการทำงานและลดขั้นตอนการทำงานให้รวดเร็วขึ้น ซึ่งเป็นการพัฒนาระบบด้วยการจัดเก็บข้อมูลปัญหาที่เกิดขึ้นไว้ในฐานข้อมูลเดียวกัน อาทิ ข้อมูลการมอบหมายงานให้แก่ทีมพัฒนาระบบและทีมบริการหลังการขาย ข้อมูลปัญหาที่เกิดขึ้น รวมถึงวิธีการแก้ปัญหา โดยออกแบบระบบให้ครอบคลุมกระบวนการทำงานในปัจจุบัน เพื่อลดความผิดพลาดจากการทำงานในแบบเดิม เพื่อลดขั้นตอนการทำงาน เพื่อศึกษาปัญหาและวิธีการแก้ไขที่เคยเกิดขึ้น ทำให้แก้ไขระบบได้รวดเร็ว และง่ายต่อการศึกษาของพนักงานใหม่ อีกทั้งยังเพิ่มความสะดวกในการติดตามความคืบหน้าของงานได้ง่าย

บทที่ 4

การวิเคราะห์และออกแบบระบบใหม่

จากปัญหาในการปฏิบัติงานการประสานงานโครงการในปัจจุบัน แนวทางหนึ่งที่สามารถช่วยแก้ปัญหาได้ คือ การนำเทคโนโลยีสารสนเทศเข้ามาใช้งาน ซึ่งในขั้นตอนการวิเคราะห์และออกแบบระบบงานใหม่ มีสิ่งที่ดำเนินการ คือ การศึกษาขั้นตอนการทำงานของระบบเดิมที่เป็นอยู่ในปัจจุบัน การกำหนดแนวทางในการปรับปรุงระบบให้เป็นไปในทิศทางที่ดีขึ้น รวมทั้งการสร้างแบบจำลองกระบวนการและแบบจำลองข้อมูลทำให้ทราบถึงรายละเอียดของระบบ

4.1 การวิเคราะห์ความต้องการของระบบใหม่

จากการศึกษาและวิเคราะห์การทำงานในระบบปัจจุบันพบว่า การประสานงานโครงการต้องการระบบสารสนเทศใหม่ที่จะช่วยสนับสนุนการดำเนินงาน โดยสามารถแบ่งชนิดของความต้องการได้ 2 รูปแบบ ดังนี้

4.1.1 ความต้องการเชิงหน้าที่การทำงาน (Functional Requirement)

- 1 ผู้ประสานงานโครงการสามารถใช้ระบบส่งรายละเอียดความซ้ำซ้อนของระบบและแจกแจงงานให้แก่ทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขาย
- 2 ระบบสามารถแนบเอกสาร ภาพ และเสียง ในการส่งรายละเอียดเพิ่มเติมให้แก่ทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขาย
- 3 ระบบสามารถแจ้งเตือนทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขายถึงความซ้ำซ้อนของระบบที่ต้องทำการแก้ไขผ่านทางอีเมล
- 4 ระบบสามารถแจ้งเตือนผู้ประสานงานโครงการให้ติดตามความคืบหน้างานที่มอบหมายให้ทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขายแก้ไข
- 5 ผู้ประสานงานโครงการสามารถตรวจสอบสถานะของความซ้ำซ้อนของระบบได้
- 6 ทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขายสามารถอัปเดตสถานะของความซ้ำซ้อนของระบบว่าแก้ไขเสร็จเรียบร้อยแล้ว
- 7 ทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขายสามารถบันทึกวิธีการแก้ไขความซ้ำซ้อน

ของระบบลงในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 8 ผู้ประสานงาน โครงการ ทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขายสามารถศึกษาความขัดข้องของระบบและวิธีการแก้ไขความขัดข้องของระบบภายหลังได้

4.1.2 ความต้องการเชิงคุณลักษณะ (Non Functional Requirement)

- 1 ระบบต้องมีลักษณะการทำงานแบบแอปพลิเคชันบนอุปกรณ์พกพาที่ใช้ระบบปฏิบัติการแอนดรอยด์
- 2 ระบบสามารถเชื่อมต่อผ่านระบบอินเทอร์เน็ตได้
- 3 ระบบต้องมีลักษณะการทำงานแบบเว็บแอปพลิเคชัน ที่เชื่อมเข้ากับแอปพลิเคชันบนอุปกรณ์พกพาได้
- 4 ระบบมีการรักษาความปลอดภัยที่ดีในการจัดการข้อมูลและมีการกำหนดสิทธิ์การเข้าถึงข้อมูลตามความเหมาะสมและบทบาทหน้าที่
- 5 ระบบสามารถรองรับการทำงาน ได้ตลอดเวลา
- 6 ระบบสามารถแจ้งเตือนทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขายถึงงานที่ได้รับมอบหมายผ่านทางอีเมล
- 7 ระบบสามารถแจ้งเตือนผู้ประสานงาน โครงการให้ติดตามความคืบหน้างานที่มอบหมายให้ทีมพัฒนาซอฟต์แวร์และทีมบริการหลังการขายแก้ไขผ่านการแจ้งเตือนขอแอปพลิเคชันบนอุปกรณ์พกพา
- 8 ระบบสามารถแสดงความขัดข้องของระบบและวิธีแก้ไขบนแอปพลิเคชันของอุปกรณ์พกพาและบนเว็บแอปพลิเคชัน

4.2 การออกแบบระบบด้วยยูเอ็มแอล

ระบบบริหารจัดการปัญหาด้วยแอนดรอยด์ใช้เทคโนโลยีในการวิเคราะห์และออกแบบระบบตามหลักการวิเคราะห์และออกแบบเชิงวัตถุ โดยใช้ยูเอ็มแอลเป็นเครื่องมือในการจำลองระบบเพื่อใช้สื่อสารระหว่างผู้ที่เกี่ยวข้องทั้งหมดในโครงการให้เข้าใจการพัฒนาระบบได้ง่ายขึ้น โดยแผนภาพที่ใช้ในการอธิบายการทำงานของระบบบริหารจัดการปัญหาด้วยแอนดรอยด์ ประกอบด้วย ยูสเคส ไดอะแกรม ซึ่งมีรายละเอียด ดังนี้

4.2.1 ยูสเคสไดอะแกรม (Use Case Diagram)

ยูสเคสไดอะแกรมแสดงถึงฟังก์ชันหลักและขอบเขตของระบบเพื่อให้เห็นภาพรวมของระบบ

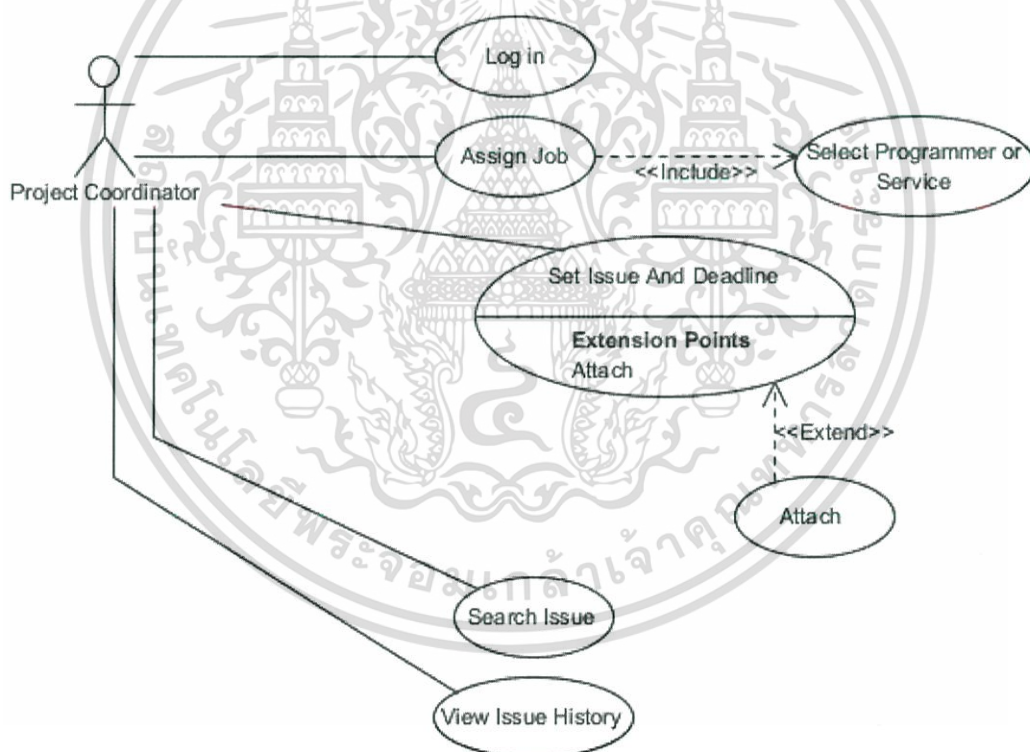
ยูสเคสไดอะแกรมจะแสดงความสัมพันธ์ระหว่างแอกเตอร์กับยูสเคส หรือ ยูสเคสกับยูสเคส และเอกสารเป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการวิเคราะห์กระบวนการของระบบบริหารจัดการปัญหาด้วยแอนดรอยด์ สามารถนำมาสร้าง ยูสเคสไดอะแกรมได้ 2 ไดอะแกรม ดังนี้

ยูสเคสไดอะแกรมระบบบริการจัดการปัญหาบนอุปกรณ์พกพา ดังรูปที่ 4.1

ยูสเคสไดอะแกรมระบบบริการจัดการปัญหาด้วยแอนดรอยด์ มีแอกเตอร์ หรือ ผู้ที่กระทำกับ ระบบบริหารจัดการปัญหาด้วยแอนดรอยด์ ซึ่งแบ่งตามบทบาท ประกอบไปด้วยแอกเตอร์ คือ

- 1 ผู้ประสานงาน โครงการ (Project Coordinator) มีหน้าที่แจกแจงงานสำหรับงานบริการหลังการขาย โดยจะทำการเลือกผู้รับผิดชอบการแก้ไขความขัดข้องของระบบ จากนั้นทำการกำหนดวันสิ้นสุดการแก้ไขและส่งรายละเอียดความขัดข้องของระบบผ่านทางอีเมล อีกทั้งยังสามารถค้นหาความขัดข้องของระบบ เพื่อตรวจสอบสถานะความขัดข้องของระบบหรือศึกษาวิธีการแก้ไขความขัดข้องที่เกิดขึ้นได้



รูปที่ 4.1 ยูสเคสไดอะแกรมระบบบริการจัดการปัญหาบนอุปกรณ์พกพา

ยูสเคส คือ ฟังก์ชันหรือหน้าที่หลักที่ระบบบริหารจัดการปัญหาบนอุปกรณ์พกพาสามารถทำงานได้ ประกอบด้วย 6 ยูสเคส ดังนี้

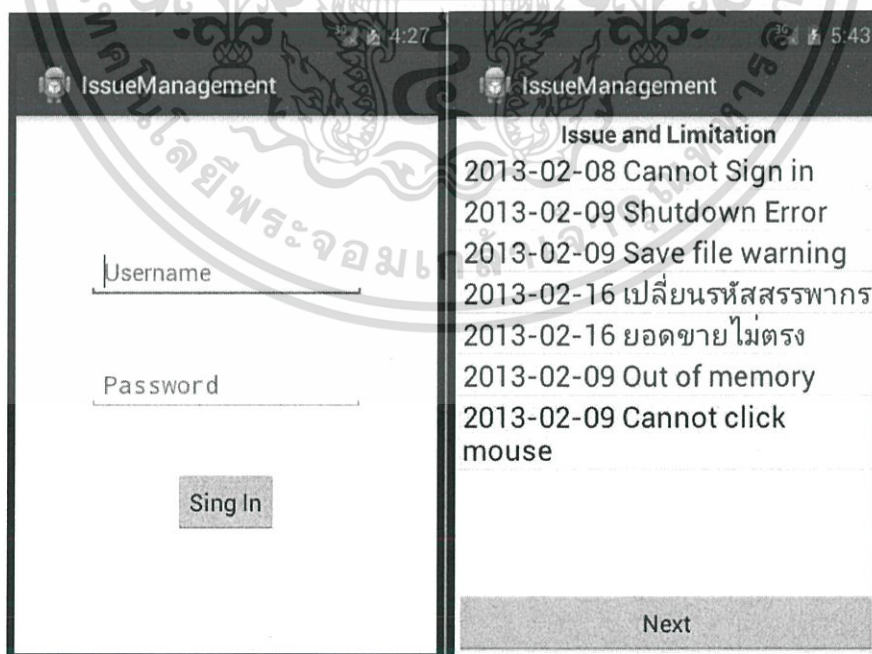
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) Log in คือ ยูสเคสการเข้าสู่ระบบ มีหน้าที่ตรวจสอบชื่อผู้ใช้งานและรหัสผ่านก่อนเข้าใช้ระบบ
- 2) Assign Job คือ ยูสเคสการมอบหมายงาน โดยผู้ประสานงาน โครงการใช้สำหรับมอบหมายงานให้แก่ผู้รับผิดชอบการแก้ไขความขัดข้องของระบบ
- 3) Select Programmer or Service คือ ยูสเคสการเลือกผู้รับผิดชอบการแก้ไขความขัดข้องของระบบ โดยผู้ประสานงาน โครงการทำการเลือกผู้รับผิดชอบการแก้ไขให้ตรงกับความขัดข้องที่เกิดขึ้น เช่น ความขัดข้องที่เกิดจากซอฟต์แวร์ก็ให้เลือกโปรแกรมเมอร์ เป็นต้น
- 4) Set Issue And Deadline คือ ยูสเคสการส่งความขัดข้องของระบบและกำหนดวันสิ้นสุดการแก้ไขระบบ โดยผู้ประสานงาน โครงการกรอกรายละเอียดความขัดข้องของระบบแล้วส่งผ่านทางอีเมลให้แก่ผู้แก้ไขความขัดข้องของระบบที่รับผิดชอบ โครงการผ่านทางอีเมล และสามารถเลือกส่งข้อมูลความขัดข้องของระบบเพิ่มเติม โดยเรียกใช้ยูสเคสการแนบไฟล์ภาพ และยูสเคสการแนบไฟล์เสียง เพื่อให้ผู้แก้ไขความขัดข้องของระบบสามารถเข้าใจถึงความขัดข้องที่เกิดขึ้นได้ชัดเจน พร้อมทั้งกำหนดวันสิ้นสุดการแก้ไขระบบ สำหรับแจ้งเตือนผู้ประสานงานให้ติดตามความคืบหน้า
- 5) Attach คือ ยูสเคสการแนบไฟล์ โดยผู้ประสานงานใช้แนบรายละเอียดเพิ่มเติมเป็นรูปภาพหรือไฟล์เสียง ให้แก่ผู้แก้ไขความขัดข้องของระบบที่รับผิดชอบ โครงการผ่านทางอีเมล
- 6) Search Issue คือ ยูสเคสการค้นหาปัญหาหรือความขัดข้องของระบบ โดยผู้ประสานงานใช้ค้นหาความขัดข้องของระบบที่เคยเกิดขึ้น หรือที่ได้รับแจ้ง
- 7) View Issue History คือ ยูสเคสแสดงรายละเอียดความขัดข้องของระบบที่ได้รับแจ้ง ว่ามีการแก้ไขความขัดข้องเสร็จสมบูรณ์หรือยัง และแสดงความขัดข้องของระบบที่เกิดขึ้นมาแล้วว่าความขัดข้องดังกล่าวมีวิธีแก้ไขอย่างไร

จากส่วนประกอบต่างๆของยูสเคสไดอะแกรมระบบบริการจัดการปัญหาบนอุปกรณ์พกพา ตามที่ได้อธิบายไว้ข้างต้นแล้ว สามารถอธิบายรายละเอียดขั้นตอนการทำงานของแต่ละยูสเคสได้ดังตารางที่ 4.1 ถึง ตารางที่ 4.7

ตารางที่ 4.1 รายละเอียดของยูสเคส Login (บนอุปกรณ์พกพา)

ชื่อยูสเคส	Log in	
แอกเตอร์	Project Coordinator	
รายละเอียดโดยสังเขป	ผู้ประสานงานโครงการ ระบุชื่อผู้ใช้งานและรหัสผ่านบนอุปกรณ์พกพา เพื่อให้ระบบตรวจสอบสิทธิการใช้งาน โดยหากระบุถูกต้องจะแสดงหน้าแรกของระบบ หากไม่ถูกต้องระบบจะฟ้องข้อผิดพลาดเพื่อให้ระบุใหม่	
เงื่อนไขเริ่มต้น	-	
เงื่อนไขภายหลัง	-	
ยูสเคสที่เกี่ยวข้อง	-	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบุชื่อผู้ใช้และรหัสผ่าน และคลิกปุ่ม เข้าสู่ระบบ	2. ระบบรับข้อมูลและตรวจสอบกับฐานข้อมูล 3. ตรวจสอบถูกต้อง ระบบแสดงความขัดข้องของระบบที่ยังไม่ได้แก้ไขและวันสิ้นสุดการแก้ไข
เงื่อนไขทางเลือก	3a กรณีตรวจสอบแล้วไม่ถูกต้อง ระบบแสดงข้อความแจ้งเตือน และกลับสู่หน้าระบุชื่อผู้ใช้งานและรหัสผ่าน	



รูปที่ 4.2 แสดงหน้า Login บนอุปกรณ์พกพา และแสดงความขัดข้องของระบบที่ยังไม่ได้แก้ไขและวันสิ้นสุดการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 รายละเอียดของยูสเคส Assign Job

ชื่อยูสเคส	Assign Job	
แอกเตอร์	Project Coordinator	
รายละเอียดโดยสังเขป	ผู้ประสานงานโครงการใช้สำหรับมอบหมายงานให้แก่ผู้รับผิดชอบ การแก้ไขความขัดข้องของระบบ	
เงื่อนไขเริ่มต้น	-	
เงื่อนไขภายหลัง	-	
ยูสเคสที่เกี่ยวข้อง	-	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้กดปุ่มเลือกมอบหมายงาน	2. ระบบแสดงหน้าจอมอบหมายงาน
เงื่อนไขทางเลือก	-	

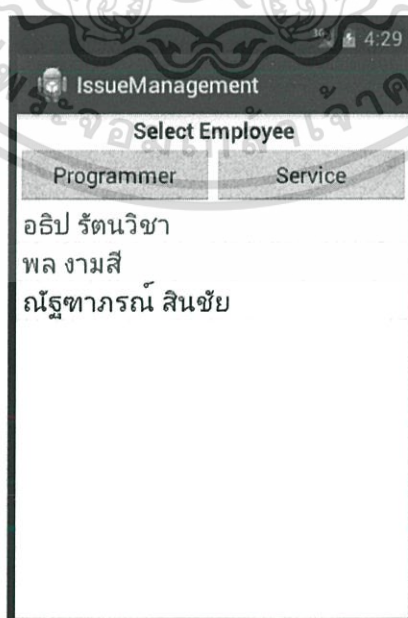


รูปที่ 4.3 แสดงหน้าหลักของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 รายละเอียดของยูสเคส Select Programmer or Service

ชื่อยูสเคส	Select Programmer or Service
แอกเตอร์	Project Coordinator
รายละเอียดโดยสังเขป	ผู้ประสานงานโครงการ เลือกผู้ที่รับผิดชอบในการแก้ไขความขัดข้องของระบบบนแอปพลิเคชันในอุปกรณ์พกพา โดยเลือกผู้รับผิดชอบการแก้ไขให้ตรงกับความขัดข้องที่เกิดขึ้น
เงื่อนไขเริ่มต้น	
เงื่อนไขภายหลัง	
ยูสเคสที่เกี่ยวข้อง	เป็นยูสเคสที่ Include อยู่กับ Assign Job
ขั้นตอนการทำงานหลัก	แอกเตอร์
	ระบบ
	<ol style="list-style-type: none"> 1. ผู้ใช้คลิกปุ่ม Assign Job 2. ระบบแสดงหน้าจอมอบหมายงาน (Select Employees) 3. ผู้ใช้คลิกปุ่มเลือกมอบหมายงานให้โปรแกรมเมอร์หรือเซอร์วิส 4. ระบบแสดงรายชื่อพนักงานตามตำแหน่งที่รับผิดชอบ 5. ผู้ใช้เลือกพนักงานผู้รับผิดชอบแก้ไขความขัดข้องของระบบ 6 ระบบแสดงหน้าจอสำหรับส่งอีเมล
เงื่อนไขทางเลือก	-



รูปที่ 4.4 แสดงหน้าจอสำหรับเลือกพนักงานผู้รับผิดชอบแก้ไขระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 รายละเอียดของยูสเคส Set Issue And Deadline

ชื่อยูสเคส	Set Issue And Deadline	
แอกเตอร์	Project Coordinator	
รายละเอียดโดยสังเขป	<p>ยูสเคสการส่งความขัดข้องของระบบและกำหนดวันสิ้นสุดการแก้ไขระบบ โดยผู้ประสานงานโครงการกรอกรายละเอียดความขัดข้องของระบบแล้วส่งผ่านทางอีเมลให้แก่ผู้แก้ไขความขัดข้องของระบบที่รับผิดชอบโครงการผ่านทางอีเมล และสามารถเลือกส่งข้อมูลความขัดข้องของระบบเพิ่มเติม โดยการแนบไฟล์ภาพ หรือแนบไฟล์เสียง เพื่อให้ผู้แก้ไขความขัดข้องของระบบสามารถเข้าใจถึงความขัดข้องที่เกิดขึ้นได้ชัดเจน พร้อมทั้งกำหนดวันสิ้นสุดการแก้ไขระบบ สำหรับแจ้งเตือนผู้ประสานงานให้ติดตามความคืบหน้า</p>	
เงื่อนไขเริ่มต้น	เลือกผู้ที่รับผิดชอบในการแก้ไขความขัดข้องของระบบ	
เงื่อนไขภายหลัง	ความขัดข้องของระบบและวันสิ้นสุดการแก้ไขถูกบันทึกและถูกส่งไปยังผู้รับผิดชอบ	
ยูสเคสที่เกี่ยวข้อง	ยูสเคส Generalization ของ Attach	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	<p>2. ผู้ใช้กรอกรายละเอียดความขัดข้องของระบบ และกำหนดวันสิ้นสุดการแก้ไข</p> <p>3. ผู้ใช้กดปุ่ม send</p>	<p>1. ระบบแสดงอีเมลผู้รับผิดชอบที่ถูกเลือก</p> <p>4. ระบบบันทึกข้อมูลลงฐานข้อมูล</p> <p>5. ส่งข้อมูลผ่านทางอีเมลให้แก่ผู้รับ</p>
เงื่อนไขทางเลือก	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 แสดงหน้าจอสำหรับกำหนดวันสิ้นสุดและส่งความขัดข้องของระบบผ่านทางอีเมล

ตารางที่ 4.5 รายละเอียดของยูสเคส Attach

ชื่อยูสเคส	Attach	
แอกเตอร์	Project Coordinator	
รายละเอียดโดยสังเขป	ผู้ประสานงาน โครงการเลือกส่งรายละเอียดความขัดข้องของระบบเพิ่มเติม โดยการแนบไฟล์ภาพ หรือไฟล์เสียง	
เงื่อนไขเริ่มต้น	ถูกเรียกใช้จากยูสเคส Set Issue And Deadline	
เงื่อนไขภายหลัง	-	
ยูสเคสที่เกี่ยวข้อง	ยูสเคส Specialization ของ Set Issue And Deadline	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้กดปุ่ม Attach File	2. ระบบแสดงรายการการแนบไฟล์ให้เลือก
	3. ผู้ใช้เลือกแนบไฟล์ที่ต้องการ	4. ระบบแนบไฟล์ภาพที่เลือก 6. ระบบกลับสู่หน้าจอสำหรับส่งอีเมล
เงื่อนไขทางเลือก	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงหน้าจอรายการให้เลือกแนบไฟล์

ตารางที่ 4.6 รายละเอียดของยูสเคส Search Issue

ชื่อยูสเคส	Search Issue	
แอกเตอร์	Project Coordinator	
รายละเอียดโดยสังเขป	ผู้ประสานงาน ใช้ค้นหาความขัดข้องของระบบที่เกิดขึ้นมาแล้ว หรือที่ได้รับแจ้ง	
เงื่อนไขเริ่มต้น	-	
เงื่อนไขภายหลัง	-	
ยูสเคสที่เกี่ยวข้อง	-	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้กดปุ่ม Search Issue 3. ผู้ใช้พิมพ์ความขัดข้องของระบบที่ต้องการค้นหา 5. ผู้ใช้เลือกความขัดข้องของระบบที่ต้องการ	2. ระบบแสดงหน้าจอค้นหาความขัดข้องด้านซอฟต์แวร์และฮาร์ดแวร์ 4. ระบบแสดงรายการความขัดข้องตามที่ผู้ค้นหา 6. ระบบแสดงความพร้อมทำงานต่อเนื่องเพื่อแสดงข้อมูลความขัดข้องที่ถูกเลือก
เงื่อนไขทางเลือก	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

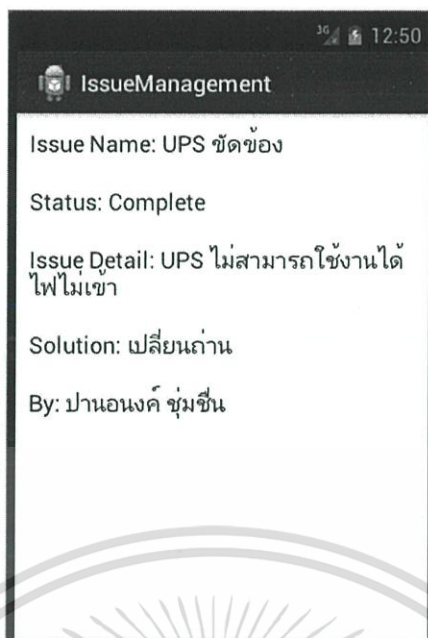


รูปที่ 4.7 แสดงหน้าจอสำหรับค้นหาความขัดข้องของระบบ

ตารางที่ 4.7 รายละเอียดของยูสเคส View Issue History

ชื่อยูสเคส	View Issue History	
แอกเตอร์	Project Coordinator	
รายละเอียดโดยสังเขป	แสดงความขัดข้องของระบบที่ได้รับแจ้ง ว่ามีการแก้ไขความขัดข้องเสร็จสมบูรณ์หรือยัง และแสดงความขัดข้องของระบบที่เกิดขึ้นมาแล้วว่าความขัดข้องดังกล่าวมีวิธีแก้ไขอย่างไร	
เงื่อนไขเริ่มต้น	แสดงรายการความขัดข้องของระบบที่มีให้ผู้ใช้เลือกเพื่อดูรายละเอียด	
เงื่อนไขภายหลัง	-	
ยูสเคสที่เกี่ยวข้อง	-	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้เลือกความขัดข้องของระบบที่ต้องการหลังทำการค้นหา	2. ระบบแสดงรายละเอียดความขัดข้องดังกล่าว
เงื่อนไขทางเลือก	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



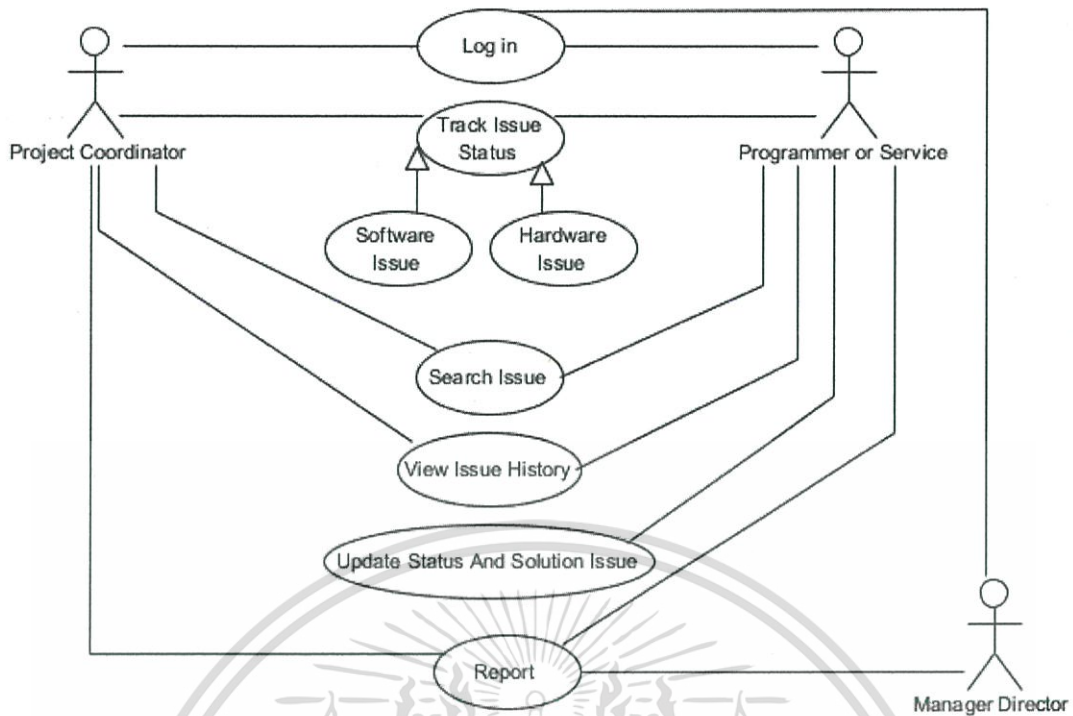
รูปที่ 4.8 แสดงหน้าจอที่แสดงรายละเอียดความขัดข้องของระบบ

ยูสเคสไดอะแกรมระบบบริการจัดการปัญหาบนเว็บแอปพลิเคชัน ดังรูปที่ 4.2

ยูสเคสไดอะแกรมระบบบริการจัดการปัญหาบนเว็บแอปพลิเคชัน มีแอกเตอร์ หรือ ผู้ที่กระทำกับระบบบริหารจัดการปัญหา ซึ่งแบ่งตามบทบาท ประกอบไปด้วยแอกเตอร์ ดังนี้

- 1 ผู้ประสานงาน โครงการ (Project Coordinator) มีหน้าที่ค้นหาปัญหาหรือความขัดข้องของระบบ เพื่อตรวจสอบสถานะความขัดข้องของระบบหรือศึกษาวิธีการแก้ไขความขัดข้องที่เกิดขึ้น
- 2 โปรแกรมเมอร์หรือเซอร์วิส (Programmer or Service) คือ ผู้ที่ได้รับมอบหมายจากหัวหน้าทีมพัฒนาซอฟต์แวร์และหัวหน้าทีมบริการหลังการขายให้ดูแลรับผิดชอบโครงการ ทำหน้าที่อัปเดตสถานะของความขัดข้องของระบบที่ได้รับการมอบหมาย และบันทึกวิธีแก้ไขความขัดข้องของระบบ รวมทั้งการค้นหาเพื่อศึกษาวิธีการแก้ไขความขัดข้องของระบบที่เคยทำการบันทึกไว้
- 3 กรรมการผู้จัดการ (Manager Director) คือ ผู้ตรวจสอบจำนวนความขัดข้องของระบบรายปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ยูสเคสโคดเจอร์ระบบบริการจัดการปัญหาบนเว็บแอปพลิเคชัน

ยูสเคส คือ ฟังก์ชันหรือหน้าที่หลักที่ระบบบริหารจัดการปัญหาสามารถทำงานได้ ประกอบด้วย 4 ยูสเคส ดังนี้

- 1) Log in คือ ยูสเคสการเข้าสู่ระบบ มีหน้าที่ตรวจสอบชื่อผู้ใช้งานและรหัสผ่านก่อนเข้าใช้ระบบผ่านทางเว็บแอปพลิเคชัน และตรวจสอบสิทธิการใช้งานของผู้ใช้ตามสิทธิของแอกเตอร์
- 2) Track Issue Status คือ ยูสเคสการตรวจสอบสถานะความขัดข้องของระบบที่ยังไม่ได้ดำเนินการแก้ไข โดยผู้ประสานงาน ทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) และทีมบริการหลังการขาย (เซอร์วิส) ใช้ตรวจสอบความขัดข้องของระบบที่ยังไม่ได้ดำเนินการแก้ไข
- 3) Software Issue คือ ยูสเคสการตรวจสอบความขัดข้องของระบบด้านซอฟต์แวร์ เป็น Specialization ของยูสเคสการตรวจสอบสถานะความขัดข้องของระบบ โดยผู้ประสานงาน ทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) ใช้ตรวจสอบความขัดข้องของระบบด้านซอฟต์แวร์ที่ยังไม่ได้ดำเนินการแก้ไข
- 4) Hardware Issue คือ ยูสเคสการตรวจสอบความขัดข้องของระบบด้านฮาร์ดแวร์ เป็น Specialization ของยูสเคสการตรวจสอบสถานะความขัดข้องของระบบ โดยผู้ประสานงาน ทีมบริการหลังการขาย (เซอร์วิส) ใช้ตรวจสอบความขัดข้องของระบบด้านฮาร์ดแวร์ที่ยังไม่ได้ดำเนินการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) Search Issue คือ ยูสเคสการค้นหาปัญหาหรือความขัดข้องของระบบ โดยผู้ประสานงาน ทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) และทีมบริการหลังการขาย (เซอร์วิส) ใช้ค้นหาความขัดข้องของระบบที่เคยเกิดขึ้น
- 6) View Issue History คือ ยูสเคสการดูความขัดข้องของระบบที่ได้รับแจ้ง ว่ามีการแก้ไขความขัดข้องเสร็จสมบูรณ์หรือยัง และดูความขัดข้องของระบบที่เกิดขึ้นมาแล้วว่าความขัดข้องดังกล่าวมีวิธีแก้ไขอย่างไร
- 7) Update Status and Solution Issue คือ ยูสเคสการอัปเดตสถานะความขัดข้องของระบบและบันทึกวิธีการแก้ไขความขัดข้อง โดยทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) และทีมบริการหลังการขาย (เซอร์วิส) ใช้อัปเดตสถานะความขัดข้องของระบบหลังการทำการแก้ไขว่าเสร็จสมบูรณ์แล้ว พร้อมทั้งบันทึกวิธีการแก้ไขความขัดข้องดังกล่าว
- 8) Report คือ ยูสเคสสำหรับเรียกดูรายงานสรุปการเกิดความขัดข้องของระบบรายปี โดยกรรมการผู้จัดการ ใช้ตรวจสอบความขัดข้องของระบบที่เกิดขึ้นในปีที่ต้องการ รายงานจะแสดงจำนวนการเกิดความขัดข้องของระบบทั้งทางด้านฮาร์ดแวร์และด้านซอฟต์แวร์ที่เกิดขึ้นในปีที่ต้องการ

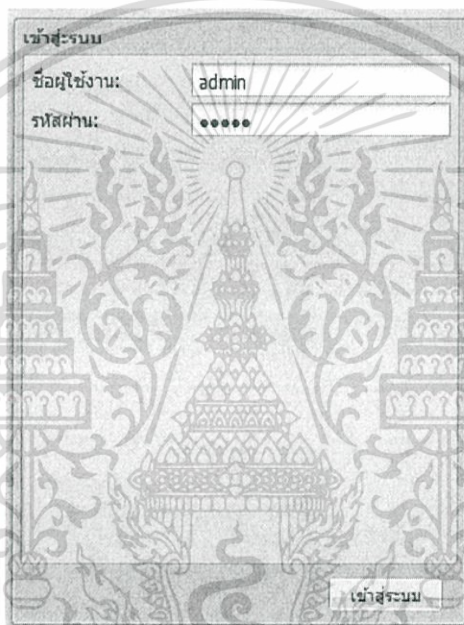
จากส่วนประกอบต่างๆของยูสเคสโคออดิเนตระบบบริการจัดการปัญหาบนเว็บแอปพลิเคชันตามที่ได้อธิบายไว้ข้างต้นแล้ว สามารถอธิบายรายละเอียดขั้นตอนการทำงานของแต่ละยูสเคสได้ดังตารางที่ 4.8 ถึง ตารางที่ 4.15

ตารางที่ 4.8 รายละเอียดของยูสเคส Login (บนเว็บแอปพลิเคชัน)

ชื่อยูสเคส	Login
แอกเตอร์	Project Coordinator, Programmer or Service, Manager Director
รายละเอียดโดยสังเขป	ผู้ประสานงาน โครงการ ทีมพัฒนาระบบ หรือทีมบริการหลังการขาย ระบุชื่อผู้ใช้งานและรหัสผ่านบนเว็บแอปพลิเคชัน เพื่อให้ระบบตรวจสอบสิทธิการใช้งาน โดยหากระบุถูกต้องจะแสดงหน้าแรกของระบบ หากไม่ถูกต้องระบบจะฟ้องข้อผิดพลาดเพื่อให้ระบุใหม่
เงื่อนไขเริ่มต้น	-
เงื่อนไขภายหลัง	-
ยูสเคสที่เกี่ยวข้อง	-

ตารางที่ 4.8 (ต่อ)

ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบุชื่อผู้ใช้และรหัสผ่าน และคลิกปุ่ม เข้าสู่ระบบ	2. ระบบรับข้อมูลและตรวจสอบกับฐานข้อมูล 3 ตรวจสอบถูกต้อง ระบบแสดงหน้าหลักของระบบ
เงื่อนไขทางเลือก	3a กรณีตรวจสอบแล้วไม่ถูกต้อง ระบบแสดงข้อความแจ้งเตือน และกลับสู่หน้าระบุชื่อผู้ใช้งานและรหัสผ่าน	



รูปที่ 4.10 แสดงหน้า Login บนเว็บแอปพลิเคชัน

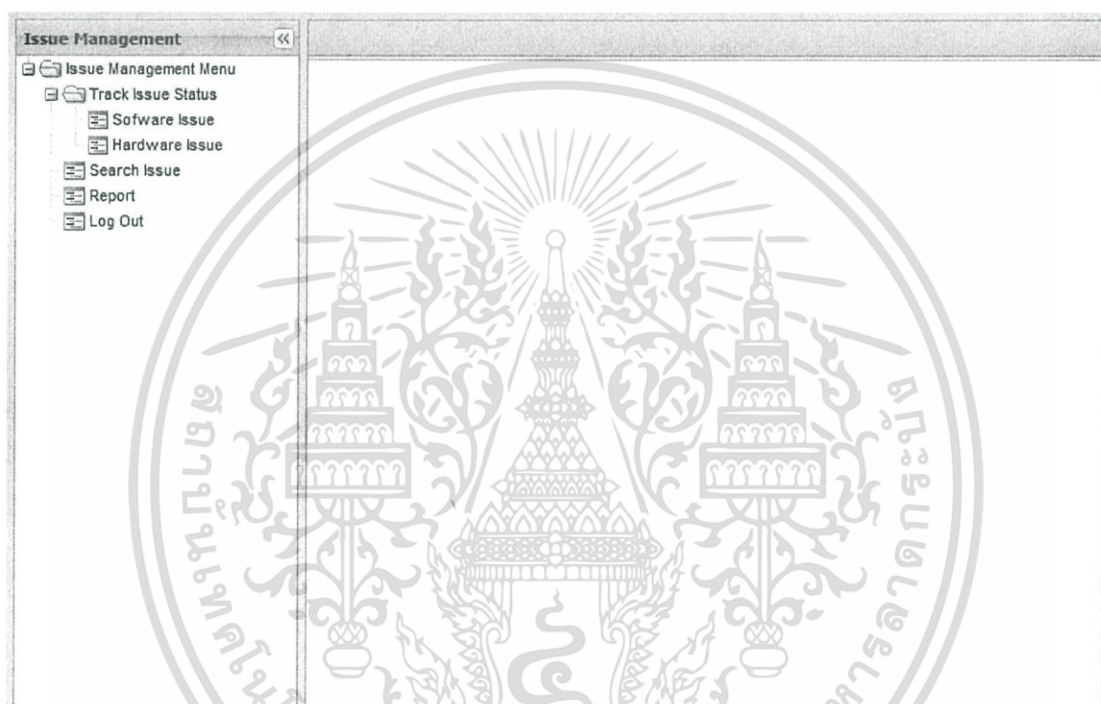
ตารางที่ 4.9 รายละเอียดของยูสเคส Track Issue Status

ชื่อยูสเคส	Track Issue Status
แอกเตอร์	Project Coordinator, Programmer or Service
รายละเอียดโดยสังเขป	ผู้ประสานงาน ทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) และทีมบริการหลังการขาย (เซอร์วิส) ใช้ตรวจสอบสถานะความขัดข้องของระบบที่ยังไม่ได้ดำเนินการแก้ไข
เงื่อนไขเริ่มต้น	เข้าสู่ระบบ
เงื่อนไขภายหลัง	-
ยูสเคสที่เกี่ยวข้อง	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 (ต่อ)

ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้กดเมนู Track Issue Status	2. ระบบแสดงประเภทของความขัดข้อง
	3. ผู้ใช้เลือกประเภทความขัดข้องของระบบที่ต้องการ	4. ระบบแสดงความขัดข้องที่ยังไม่ได้ดำเนินการแก้ไข
เงื่อนไขทางเลือก	-	



รูปที่ 4.11 แสดงหน้าจอหลักของเว็บแอปพลิเคชัน

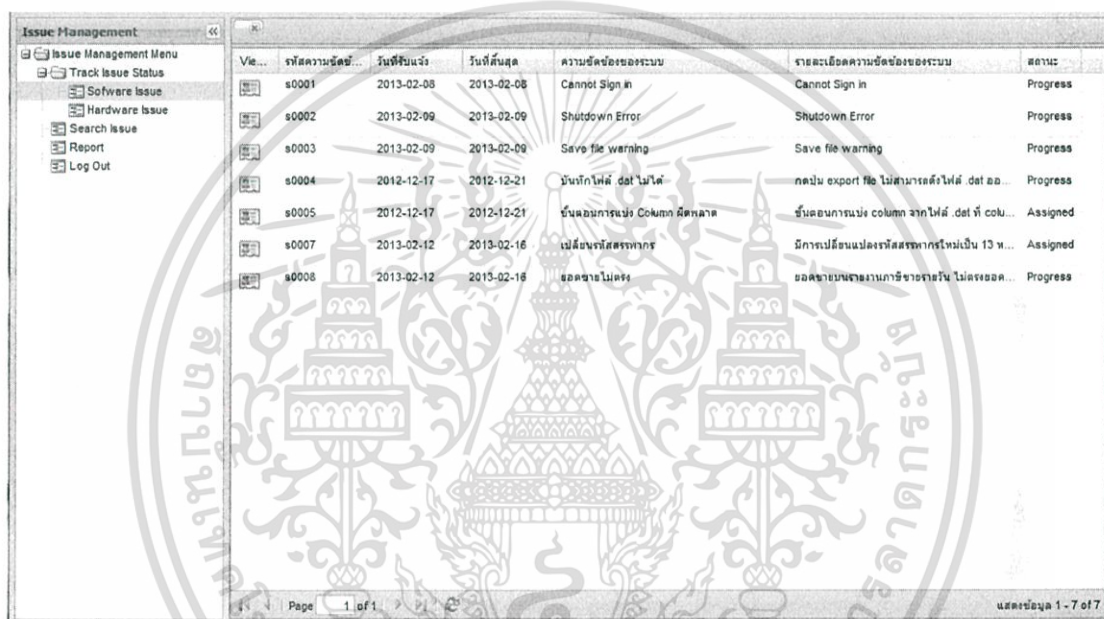
ตารางที่ 4.10 รายละเอียดของยูสเคส Software Issue

ชื่อยูสเคส	Software Issue
แอกเตอร์	Project Coordinator, Programmer
รายละเอียดโดยสังเขป	ผู้ประสานงาน ทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) ใช้ตรวจสอบความขัดข้องของระบบด้านซอฟต์แวร์ที่ยังไม่ได้ดำเนินการแก้ไข
เงื่อนไขเริ่มต้น	ถูกเรียกใช้เมื่อมีการตรวจสอบสถานะความขัดข้องของระบบ
เงื่อนไขภายหลัง	-
ยูสเคสที่เกี่ยวข้อง	-ยูสเคส Specialization ของ Track Issue Status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 (ต่อ)

ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้กดเมนู Track Issue Status 3. ผู้ใช้เลือกประเภทของความขัดข้องด้านซอฟต์แวร์	2. ระบบแสดงประเภทของความขัดข้อง 4. ระบบแสดงความขัดข้องด้านซอฟต์แวร์ที่ยังไม่ได้ดำเนินการแก้ไข
เงื่อนไขทางเลือก	-	



รูปที่ 4.12 แสดงเมนูสำหรับเลือกประเภทความขัดข้องด้านซอฟต์แวร์

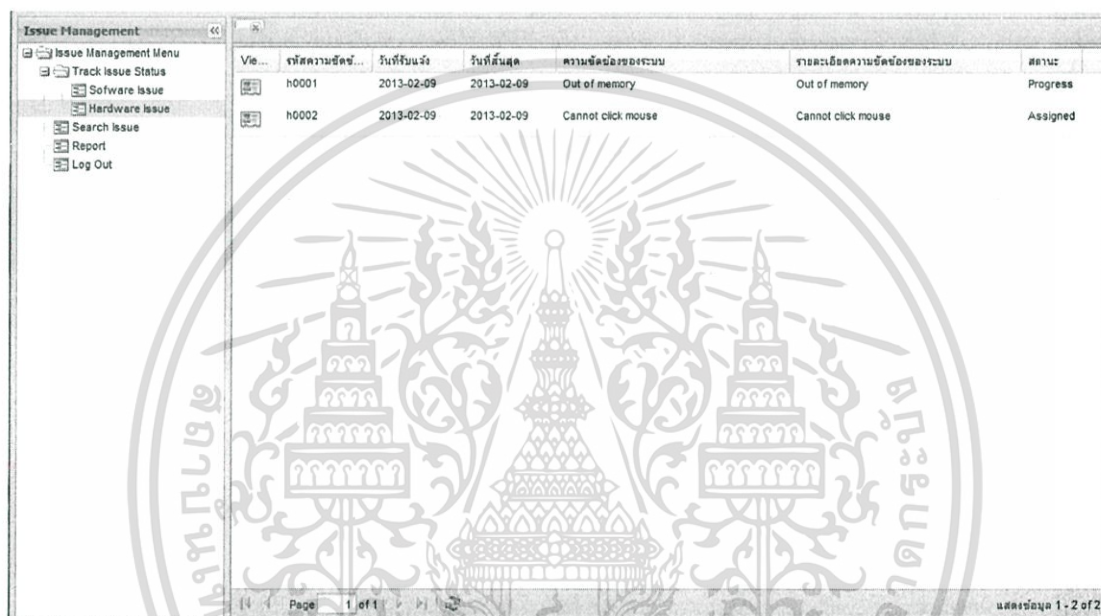
ตารางที่ 4.11 รายละเอียดของยูสเคส Hardware Issue

ชื่อยูสเคส	Hardware Issue
แอกเตอร์	Project Coordinator, Service
รายละเอียดโดยสังเขป	ผู้ประสานงาน ทีมบริการหลังการขาย (เซอร์วิส) ใช้ตรวจสอบความขัดข้องของระบบด้านฮาร์ดแวร์ที่ยังไม่ได้ดำเนินการแก้ไข
เงื่อนไขเริ่มต้น	ถูกเรียกใช้เมื่อมีการตรวจสอบสถานะความขัดข้องของระบบ
เงื่อนไขภายหลัง	-
ยูสเคสที่เกี่ยวข้อง	-ยูสเคส Specialization ของ Track Issue Status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 (ต่อ)

ขั้นตอนการทำงานหลัก	แอกเคอร์	ระบบ
	1. ผู้ใช้กดเมนู Track Issue Status 3. ผู้ใช้เลือกประเภทของความขัดข้องด้านฮาร์ดแวร์	2. ระบบแสดงประเภทของความขัดข้อง 4. ระบบแสดงความขัดข้องด้านฮาร์ดแวร์ที่ยังไม่ได้ดำเนินการแก้ไข
เงื่อนไขทางเลือก	-	



รูปที่ 4.13 แสดงเมนูสำหรับเลือกประเภทความขัดข้องด้านฮาร์ดแวร์

ตารางที่ 4.12 รายละเอียดของยูสเคส Search Issue

ชื่อยูสเคส	Search Issue
แอกเคอร์	Project Coordinator, Programmer or Service
รายละเอียดโดยสังเขป	ผู้ประสานงาน ทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) และทีมบริการหลังการขาย (เซอร์วิส) ใช้ค้นหาความขัดข้องของระบบที่เกิดขึ้นมาแล้วหรือที่ได้รับแจ้ง
เงื่อนไขเริ่มต้น	-
เงื่อนไขภายหลัง	-
ยูสเคสที่เกี่ยวข้อง	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.12 (ต่อ)

ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้คลิกปุ่ม Search Issue	2. ระบบแสดงหน้าจอค้นหาความขัดข้องด้านซอฟต์แวร์และฮาร์ดแวร์
	3. ผู้ใช้พิมพ์ความขัดข้องของระบบที่ต้องการค้นหา	4. ระบบแสดงรายการความขัดข้องตามที่ผู้ใช้ค้นหา
	5. ผู้ใช้เลือกความขัดข้องของระบบที่ต้องการ	6. ระบบแสดงความพร้อมทำงานต่อเนื่องเพื่อแสดงข้อมูลไปยังหน้าจอ View Issue History
เงื่อนไขทางเลือก	-	



รูปที่ 4.14 แสดงหน้าจอสำหรับค้นหาความขัดข้องของระบบ

ตารางที่ 4.13 รายละเอียดของยูสเคส View Issue History

ชื่อยูสเคส	View Issue History
แอกเตอร์	Project Coordinator, Programmer or Service
รายละเอียดโดยสังเขป	ผู้ประสานงาน ทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) และทีมบริการหลังการขาย (เซอร์วิส) ใช้ค้นหาความขัดข้องของระบบที่เกิดขึ้นมาแล้วหรือที่ได้รับแจ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.13 (ต่อ)

เงื่อนไขเริ่มต้น	แสดงรายการความขัดข้องของระบบที่มีให้ผู้ใช้เลือกเพื่อดูรายละเอียด	
เงื่อนไขภายหลัง	-	
ยูสเคสที่เกี่ยวข้อง	-	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้เลือกความขัดข้องของระบบที่ต้องการหลังทำการค้นหา	2. ระบบแสดงสถานะความขัดข้องและวิธีแก้ไขความขัดข้องดังกล่าว
เงื่อนไขทางเลือก	-	



รูปที่ 4.15 แสดงหน้าจอสำหรับแสดงสถานะความขัดข้อง วิธีแก้ไข และผู้ทำการแก้ไข

ตารางที่ 4.14 รายละเอียดของยูสเคส Update Status and Solution Issue

ชื่อยูสเคส	Update Status and Solution Issue
แอกเตอร์	Programmer or Service
รายละเอียด โดยสังเขป	ยูสเคสการอัปเดตสถานะความขัดข้องของระบบและบันทึกวิธีการแก้ไขความขัดข้อง โดยทีมพัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) และทีมบริการหลังการขาย (เซอร์วิส) ใช้อัปเดตสถานะความขัดข้องของระบบหลังการทำการแก้ไขว่าเสร็จสมบูรณ์แล้ว พร้อมทั้งบันทึกวิธีการแก้ไขความขัดข้องดังกล่าว
เงื่อนไขเริ่มต้น	ถูกเรียกใช้หลังจากมีการเลือกความขัดข้องของระบบที่ปรากฏบนหน้าจอ Track Issue Status
เงื่อนไขภายหลัง	-

ตารางที่ 4.14 (ต่อ)

ยูสเคสที่เกี่ยวข้อง	-	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้เลือกความ จัดซื้อของระบบ 3. ผู้ใช้ปรับปรุงสถานะ ความจัดซื้อ กรอก วิธีการแก้ไขความ จัดซื้อ และกดปุ่ม save	2. ระบบแสดงหน้าจอสำหรับบันทึกวิธีการ แก้ไขความจัดซื้อ พร้อมทั้งแก้ไขสถานะ 4. ระบบบันทึกข้อมูลดังกล่าวลงฐานข้อมูล
เงื่อนไขทางเลือก	-	



รูปที่ 4.16 แสดงหน้าจอสำหรับบันทึกวิธีการแก้ไขความจัดซื้อ พร้อมทั้งแก้ไขสถานะ

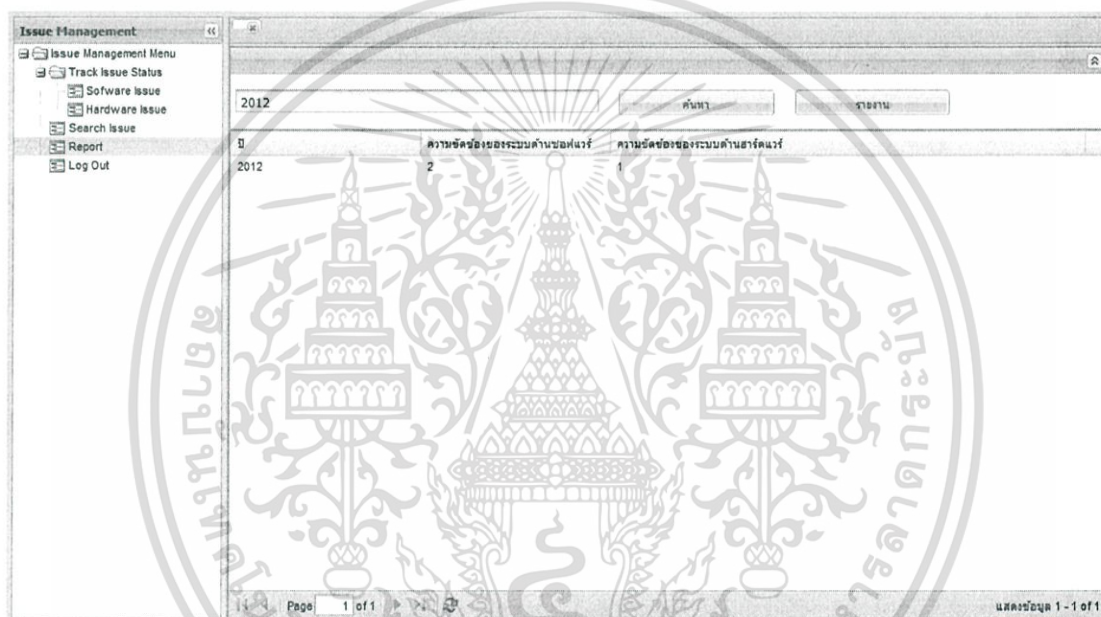
ตารางที่ 4.15 รายละเอียดของยูสเคส Report

ชื่อยูสเคส	Report
แอกเตอร์	Manager Director
รายละเอียดโดยสังเขป	ยูสเคสสำหรับเรียกดูรายงานสรุปการเกิดความจัดซื้อของระบบรายปี โดยกรรมการผู้จัดการ ใช้ตรวจสอบความจัดซื้อของระบบที่เกิดขึ้นในปีที่ต้องการ รายงานจะแสดงจำนวนการเกิดความจัดซื้อของระบบทั้งทางด้านฮาร์ดแวร์และด้านซอฟต์แวร์ที่เกิดขึ้นในปีที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.15 (ต่อ)

เงื่อนไขเริ่มต้น	-	
เงื่อนไขภายหลัง	-	
ยูสเคสที่เกี่ยวข้อง	-	
ขั้นตอนการทำงานหลัก	แอกเตอร์	ระบบ
	1. ผู้ใช้เลือกเมนู Report 2. ผู้ใช้ระบุปีที่ต้องการดูรายงาน	3. ระบบแสดงรายงานตามปีที่ระบุ
เงื่อนไขทางเลือก	-	

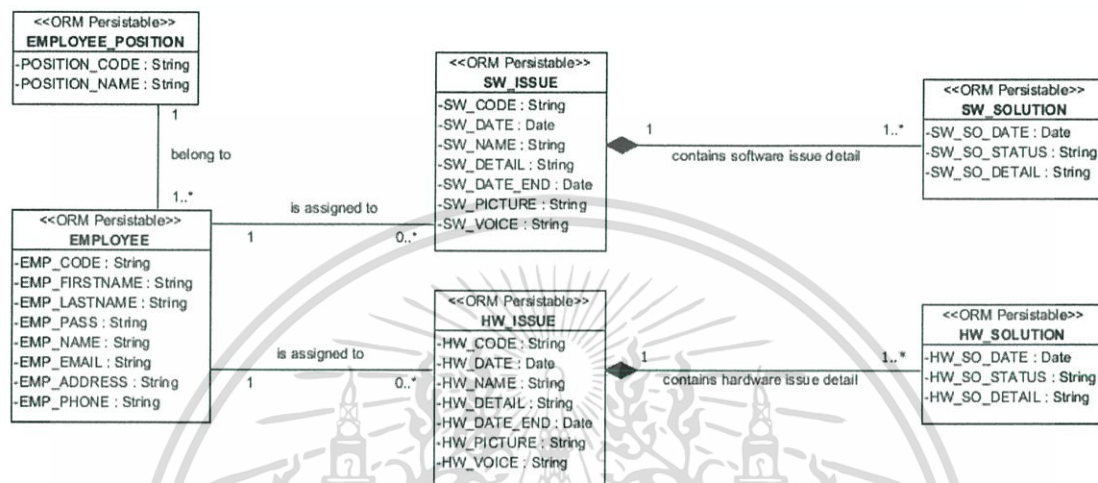


รูปที่ 4.17 แสดงหน้าจอสำหรับเรียกดูรายงานสรุปความขัดข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 คลาสไดอะแกรม (Class Diagram)

คลาสไดอะแกรม แสดงคลาสและความสัมพันธ์ระหว่างคลาส การออกแบบระบบบริหารจัดการปัญหาด้วยแอนครอยด์นั้น สามารถเขียนคลาสไดอะแกรมเพื่อแสดงองค์ประกอบในแต่ละคลาส โดยมีทั้งหมด 6 คลาส แสดงดังรูปที่ 4.2.2



รูปที่ 4.18 คลาสไดอะแกรมของระบบบริหารจัดการปัญหาด้วยแอนครอยด์

ระบบบริหารจัดการปัญหาด้วยแอนครอยด์ประกอบด้วย 6 คลาส ดังต่อไปนี้

- 1 EMPLOYEE_POSITION เป็นคลาสตำแหน่งของพนักงาน
- 2 EMPLOYEE เป็นคลาสพนักงาน
- 3 SW_ISSUE เป็นคลาสความขัดข้องของระบบด้านซอฟต์แวร์
- 4 SW_SOLUTION เป็นคลาสวิธีแก้ไขความขัดข้องของระบบด้านซอฟต์แวร์
- 5 HW_ISSUE เป็นคลาสความขัดข้องของระบบด้านฮาร์ดแวร์
- 6 HW_SOLUTION เป็นคลาสวิธีแก้ไขความขัดข้องของระบบด้านซอฟต์แวร์

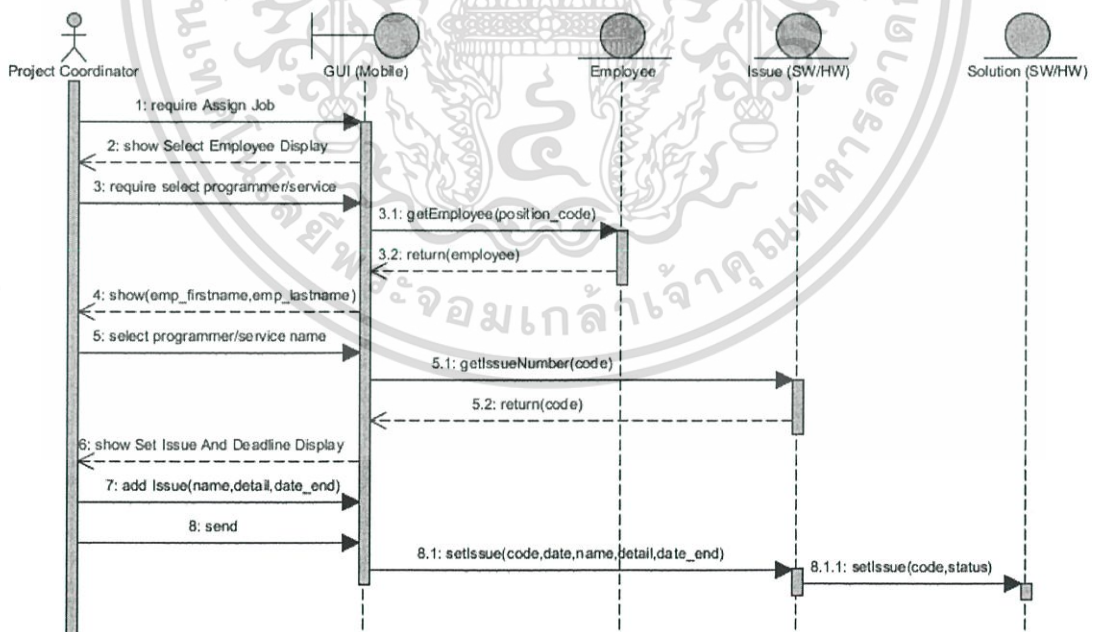
4.2.3 ซีควেনซ์ไดอะแกรม (Sequence Diagram)

การออกแบบระบบสารสนเทศเพื่อสนับสนุนการบริหารงานด้านการบริการหลังการขาย ได้นำซีควেনซ์ไดอะแกรมมาใช้ในการอธิบายลำดับการเกิดเหตุการณ์ และปฏิสัมพันธ์ในการทำงานของระบบระหว่างอ็อบเจกต์ของคลาสต่างๆ ดังที่กล่าวไปข้างต้น โดยสามารถแสดงไดอะแกรมสำหรับยูสเคสที่สำคัญได้ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 ซีควেনซ์ไดอะแกรม Assign Job

การมอบหมายงาน เริ่มจากผู้ประสานงานกลุ่ม Assign Job จากนั้นระบบแสดงหน้าสำหรับเลือกพนักงานผู้รับผิดชอบความขัดข้อง โดยจะแบ่งเป็นโปรแกรมเมอร์ และเซอร์วิส ผู้ประสานงานทำการเลือกโปรแกรมเมอร์หรือเซอร์วิส ระบบจะทำการส่งเมสเสจ `getEmployee` ไปยังอ็อบเจกต์ `Employee` เพื่อแสดงรายชื่อของพนักงานในตำแหน่งที่ทำการเลือกไว้ ผู้ประสานงานทำการเลือกชื่อพนักงานที่ต้องการ จากนั้นระบบจะทำการส่งเมสเสจ `getIssueNumber` ไปยังอ็อบเจกต์ `Issue` (ซึ่งหมายถึงอ็อบเจกต์ `sw_issue` หรือ `hw_issue` ตามประเภทของพนักงานผู้รับผิดชอบ) เพื่อแสดงรหัสความขัดข้องตัวถัดไปจากที่มีอยู่ ระบบแสดงหน้าจอ `Set Issue and Deadline` ซึ่งบนหน้าจอจะแสดงอีเมลของพนักงานที่ผู้ประสานงานทำการเลือก และรหัสความขัดข้องตัวถัดไปจากที่มีอยู่ ผู้ประสานงานพิมพ์ชื่อความขัดข้อง รายละเอียดความขัดข้อง และกำหนดวันสิ้นสุดการแก้ไข จากนั้นกด `send` ระบบจะทำการส่งเมสเสจ `setIssue` ไปยังอ็อบเจกต์ `Issue` (`sw_issue`, `hw_issue`) และ `Solution` (`sw_solution`, `hw_solution`) เพื่อทำการบันทึกรหัสความขัดข้อง วันที่ได้รับแจ้งความขัดข้อง ชื่อความขัดข้อง รายละเอียดความขัดข้อง วันที่สิ้นสุดการแก้ไขลงในตาราง `Issue` (`sw_issue`, `hw_issue`) และบันทึกรหัสความขัดข้อง สถานะความขัดข้องลงในตาราง `Solution` (`sw_solution`, `hw_solution`) ดังรูปที่ 4.19

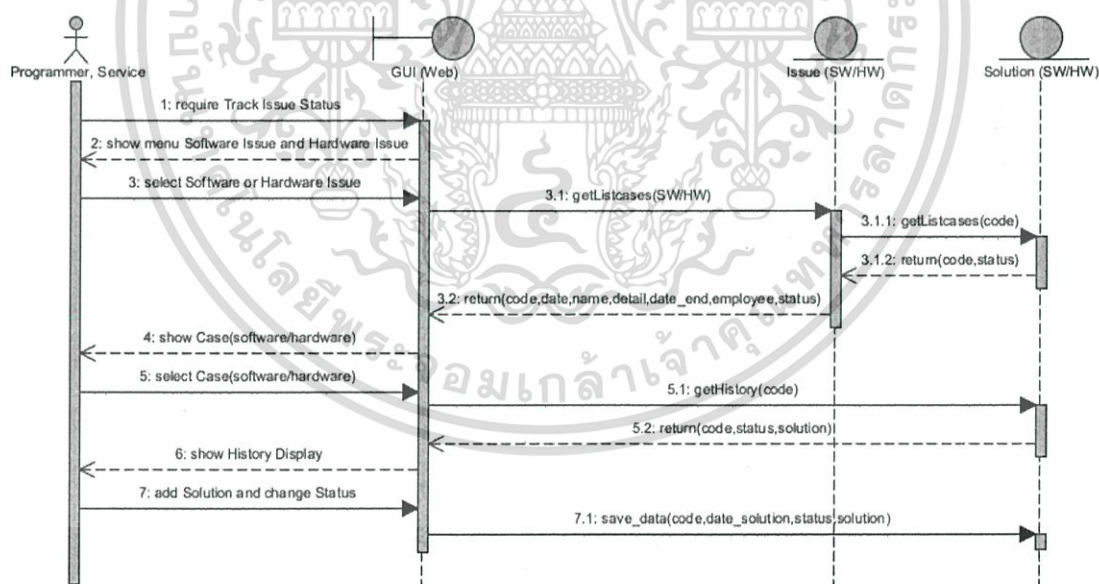


รูปที่ 4.19 ซีควেনซ์ไดอะแกรม Assign Job

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 ซีเควนซ์ไคอะแกรม Update Status and Solution Issue

การอัปเดตสถานะและบันทึกวิธีการแก้ไขความขัดข้อง เริ่มจากโปรแกรมเมอร์หรือเซอร์วิสต้องการดูงานที่ได้รับมอบหมาย โดยเลือกเมนู Track Issue Status ระบบจะแสดงเมนู Software Issue และ Hardware Issue หากผู้ใช้เป็นโปรแกรมเมอร์ก็จะใช้เมนู Software Issue หากผู้ใช้เป็นเซอร์วิสก็จะใช้เมนู Hardware Issue ระบบจะทำการส่งเมสเสจ getListcases ไปยังอ็อบเจกต์ Issue (sw_issue, hw_issue) และ Solution (sw_solution, hw_solution) เพื่อแสดงรหัสความขัดข้อง วันที่ได้รับแจ้ง ชื่อความขัดข้อง รายละเอียดของความขัดข้อง วันที่สิ้นสุดการแก้ไข และสถานะความขัดข้อง โปรแกรมเมอร์หรือเซอร์วิสเลือกความขัดข้องที่ได้รับมอบหมาย ระบบทำการส่งเมสเสจ getHistory พร้อมกับรหัสความขัดข้องที่ทำการเลือก ไปยังอ็อบเจกต์ Solution (sw_solution, hw_solution) เพื่อแสดงชื่อพนักงานที่รับผิดชอบ สถานะและวิธีการแก้ไขของรหัสความขัดข้องนั้น โปรแกรมเมอร์หรือเซอร์วิสกรอกวิธีการแก้ไขความขัดข้อง พร้อมทั้งเปลี่ยนสถานะความขัดข้อง ระบบทำการส่งเมสเสจ save_data ไปยังอ็อบเจกต์ Solution (sw_solution, hw_solution) เพื่อบันทึกวันที่ทำการแก้ไข สถานะความขัดข้องที่เปลี่ยน วิธีการแก้ไขความขัดข้อง ตามรหัสความขัดข้องที่เลือก

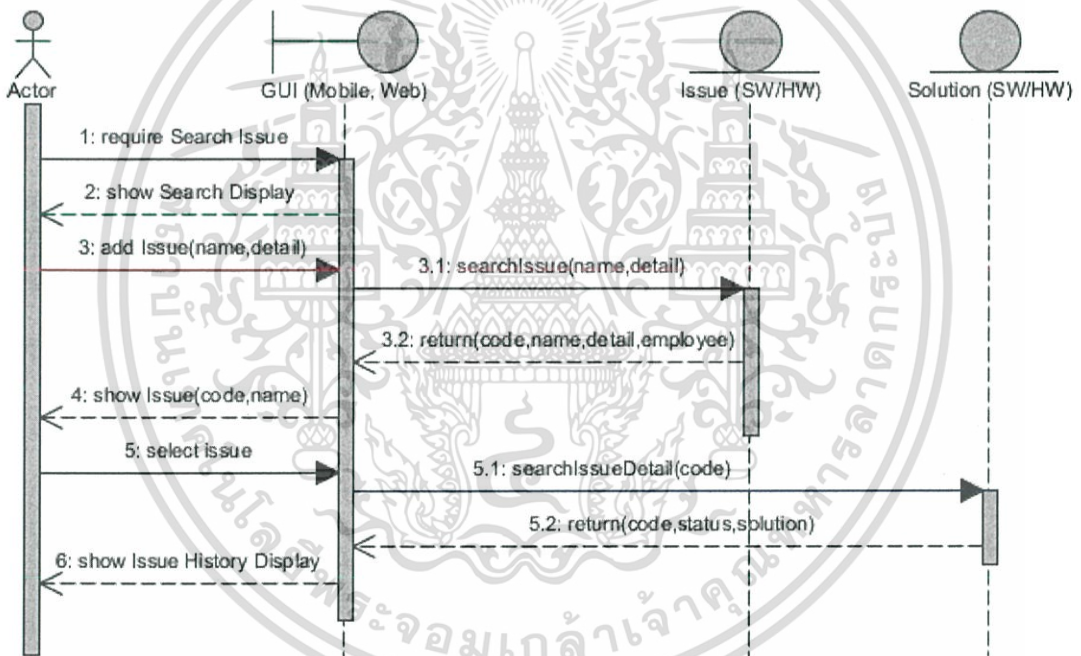


รูปที่ 4.20 ซีเควนซ์ไคอะแกรม Update Status and Solution Issue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 ซีเควนซ์ไคอะแกรม Search Issue

การค้นหาคความขัดข้อง เริ่มจากแอกเตอร์ซึ่งได้แก่ ผู้ประสานงาน โครงการ โปรแกรมเมอร์ เซอร์วิส หรือกรรมการผู้จัดการ ต้องการค้นหาคความขัดข้องของระบบที่ได้รับแจ้งบนอุปกรณ์พกพาประเภท โทรศัพท์เคลื่อนที่ หรือบนเว็บแอปพลิเคชัน ระบบทำการแสดงหน้าจอสำหรับการค้นหา จากนั้นแอกเตอร์ทำการกรอกความขัดข้องของระบบที่ต้องการค้นหา ระบบจะทำการส่งเมสเสจ searchIssue ไปยัง อีอบเจกต์ Issue (sw_issue, hw_issue) เพื่อแสดงคความขัดข้องที่กำลังค้นหา แอกเตอร์เลือกคความขัดข้องที่ต้องการศึกษารายละเอียดหรือดูสถานะ ระบบจะทำการส่งเมสเสจ searchIssueDetail ไปยังอีอบเจกต์ Solution (sw_solution, hw_solution) เพื่อแสดงรายละเอียดคความขัดข้อง สถานะ วิธีแก้ไขคความขัดข้อง และชื่อผู้รับผิดชอบแก้ไข บนหน้าจอ Issue History

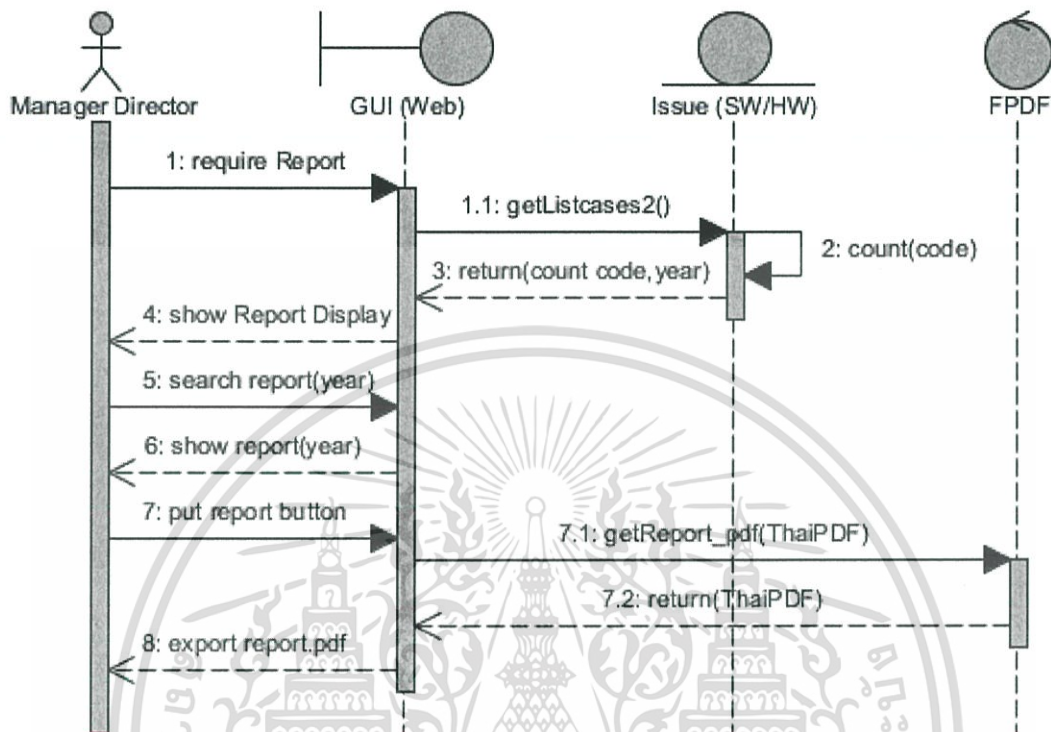


รูปที่ 4.21 ซีเควนซ์ไคอะแกรม Search Issue

4 ซีเควนซ์ไคอะแกรม Report

การตรวจสอบจำนวนคความขัดข้องของระบบบนเว็บแอปพลิเคชัน เริ่มจากกรรมการผู้จัดการเลือกเมนู report ระบบทำการส่งเมสเสจ getListcases2 ไปยังอีอบเจกต์ Issue (sw_issue, hw_issue) เพื่อนับจำนวนคความขัดข้องที่เกิดขึ้นโดยแยกเป็นรายปี แล้วแสดงบนหน้าจอ Report หากกรรมการผู้จัดการต้องการตรวจสอบจำนวนคความขัดข้องเฉพาะปีที่ต้องการ สามารถกรอกปีที่ต้องการเพื่อค้นหา ระบบจะแสดงจำนวนคความขัดข้องตามปีดังกล่าว ในกรณีที่ต้องการ Export รายงานออกมาในรูปแบบไฟล์ pdf เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรรมการผู้จัดการสามารถทำได้โดยกดปุ่มรายงานบนหน้าจอ ระบบจะทำการส่งเมสเสจ getReport_pdf ไปยังอ็อบเจกต์ FPDF เพื่อแสดงรายงานในรูปแบบไฟล์ pdf



รูปที่ 4.22 ซีควเอนซ์ไดอะแกรม Report

4.3 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลสำหรับระบบบริหารจัดการปัญหาด้วยแอนดรอยด์ได้ออกแบบฐานข้อมูลที่เป็นฐานข้อมูลเชิงสัมพันธ์ และนำเสนอผ่านแบบจำลองความสัมพันธ์ระหว่างเอนทิตี เพื่อนำเสนอรายละเอียดทางด้านโครงสร้างของฐานข้อมูล ทั้งในแง่ลักษณะข้อมูลที่จัดเก็บและโครงสร้างข้อมูลไว้ในพจนานุกรมข้อมูล โดยมีรายละเอียดดังต่อไปนี้

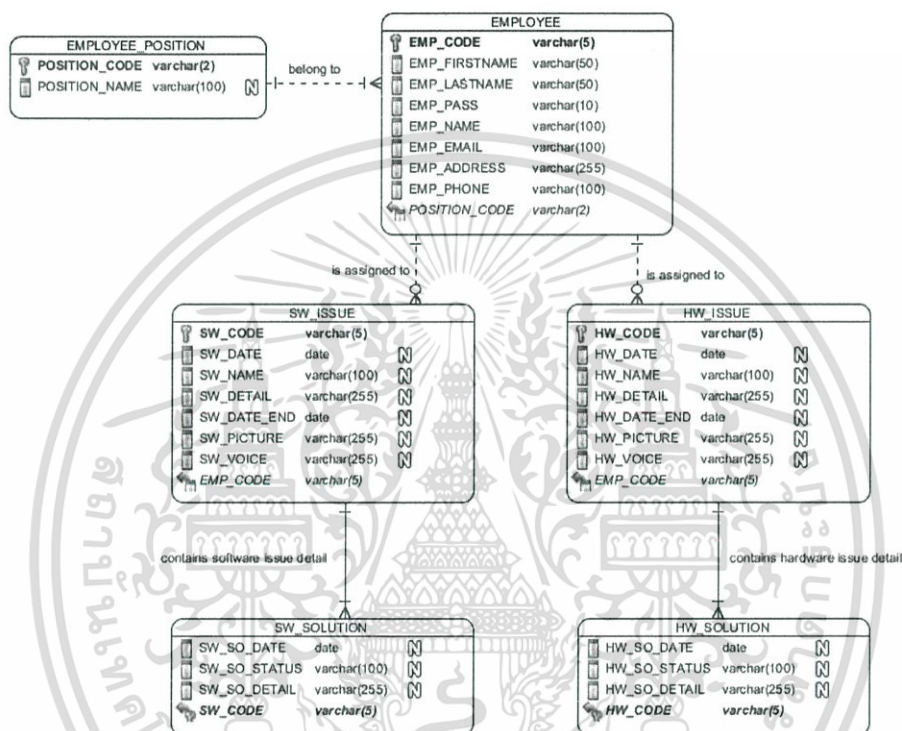
4.3.1 แบบจำลองความสัมพันธ์ระหว่างเอนทิตี

จากการวิเคราะห์และออกแบบการทำงานของระบบบริหารจัดการปัญหาด้วยแอนดรอยด์ และข้อมูลจากคลาสไดอะแกรม ได้นำมาออกแบบฐานข้อมูลเชิงสัมพันธ์โดยใช้แผนภาพความสัมพันธ์ระหว่างเอนทิตี แสดงดังรูปที่ 4.3 ซึ่งประกอบด้วยเอนทิตีสำหรับจัดเก็บข้อมูลจำนวน 6 เอนทิตี ดังนี้

1. EMPLOYEE_POSITION คือ เอนทิตีตำแหน่งของพนักงาน เช่น ผู้ประสานงาน โครงการ โปรแกรมเมอร์ พนักงานเซอร์วิส กรรมการผู้จัดการ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. EMPLOYEE คือ เอนทิตีพนักงานในบริษัท
3. SW_ISSUE คือ เอนทิตีความขัดข้องของระบบด้านซอฟต์แวร์
4. SW_SOLUTION คือ เอนทิตีวิธีแก้ไขความขัดข้องของระบบด้านซอฟต์แวร์
5. HW_ISSUE คือ เอนทิตีความขัดข้องของระบบด้านฮาร์ดแวร์
6. HW_SOLUTION คือ เอนทิตีวิธีแก้ไขความขัดข้องของระบบด้านฮาร์ดแวร์



รูปที่ 4.23 อีอาร์ไดอะแกรมระบบบริหารจัดการปัญหาด้วยแอนดรอยด์

จากแผนภาพอีอาร์ไดอะแกรมข้างต้น สามารถอธิบายความสัมพันธ์ระหว่างเอนทิตีของระบบใหม่ ได้ดังนี้

1. ความสัมพันธ์ระหว่างเอนทิตี EMPLOYEE_POSITION และเอนทิตี EMPLOYEE เป็นความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม โดยพนักงานทุกคนต้องมีตำแหน่งงานที่รับผิดชอบ พนักงานหนึ่งคนจะต้องมีตำแหน่งงานรับผิดชอบหนึ่งตำแหน่ง แต่ในบริษัทอาจจะมีบางตำแหน่งที่มีพนักงานมากกว่าหนึ่งคน

2. ความสัมพันธ์ระหว่างเอนทิตี EMPLOYEE และเอนทิตี SW_ISSUE เป็นความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม โดยความขัดข้องของระบบด้านซอฟต์แวร์จะต้องมีพนักงานรับผิดชอบ ความขัดข้องของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบด้านซอฟต์แวร์ที่เกิดขึ้นหนึ่งรายการจะต้องมีพนักงานรับผิดชอบหนึ่งคน อาจจะมีพนักงานบางคนที่ได้รับผิดชอบความขัดข้องของระบบด้านซอฟต์แวร์มากกว่าหนึ่งรายการ

3. ความสัมพันธ์ระหว่างเอนทิตี SW_ISSUE และเอนทิตี SW_SOLUTION เป็นความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม โดยวิธีแก้ไขความขัดข้องของระบบด้านซอฟต์แวร์เป็นของความขัดข้องของระบบด้านซอฟต์แวร์หนึ่งรายการ ความขัดข้องของระบบด้านซอฟต์แวร์อาจมีวิธีการแก้ไขความขัดข้องของระบบด้านซอฟต์แวร์ได้มากกว่าหนึ่งรายการ

4. ความสัมพันธ์ระหว่างเอนทิตี EMPLOYEE และเอนทิตี HW_ISSUE เป็นความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม โดยความขัดข้องของระบบด้านฮาร์ดแวร์จะต้องมีพนักงานรับผิดชอบ ความขัดข้องของระบบด้านฮาร์ดแวร์ที่เกิดขึ้นหนึ่งรายการจะต้องมีพนักงานรับผิดชอบหนึ่งคน อาจจะมีพนักงานบางคนที่ได้รับผิดชอบความขัดข้องของระบบด้านฮาร์ดแวร์มากกว่าหนึ่งรายการ

5. ความสัมพันธ์ระหว่างเอนทิตี HW_ISSUE และเอนทิตี HW_SOLUTION เป็นความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม โดยวิธีแก้ไขความขัดข้องของระบบด้านฮาร์ดแวร์เป็นของความขัดข้องของระบบด้านฮาร์ดแวร์หนึ่งรายการ ความขัดข้องของระบบด้านฮาร์ดแวร์อาจมีวิธีการแก้ไขความขัดข้องของระบบด้านฮาร์ดแวร์ได้มากกว่าหนึ่งรายการ

4.3.2 พจนานุกรมข้อมูล (Data Dictionary)

พจนานุกรมข้อมูล เป็นที่เก็บเมทาดาทา โดยแสดงรายละเอียดของตารางและความสัมพันธ์ ช่วยให้ ผู้ดูแลฐานข้อมูล ผู้ดูแลระบบ และผู้พัฒนาระบบ ใช้เป็นเครื่องมืออ้างอิงในการพัฒนาระบบ และบำรุงรักษาระบบฐานข้อมูล จากแบบจำลองความสัมพันธ์ระหว่างเอนทิตีสำหรับระบบบริหารจัดการปัญหาด้วยแอนครอยด์ สามารถแสดงรายละเอียดข้อมูลแต่ละเอนทิตี ดังต่อไปนี้

ตารางที่ 4.16 พจนานุกรมข้อมูลเอนทิตีตำแหน่งของพนักงาน

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
POSITION_CODE	รหัสตำแหน่งพนักงาน	VARCHAR(2)	Yes	PK	
POSITION_NAME	ชื่อตำแหน่งพนักงาน	VARCHAR(100)	Yes		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.17 พจนานุกรมข้อมูลเอนทิตีพนักงาน

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
EMP_CODE	รหัสพนักงาน	VARCHAR(5)	Yes	PK	
EMP_FIRSTNAME	ชื่อพนักงาน	VARCHAR(50)	Yes		
EMP_LASTNAME	นามสกุลพนักงาน	VARCHAR(50)	Yes		
EMP_PASS	รหัสผ่านเข้าใช้งานระบบ	VARCHAR(10)	Yes		
EMP_NAME	ชื่อเข้าใช้งานระบบ	VARCHAR(100)	Yes		
EMP_EMAIL	อีเมลพนักงาน	VARCHAR(100)	Yes		
EMP_ADDRESS	ที่อยู่พนักงาน	VARCHAR(255)	Yes		
EMP_PHONE	หมายเลขโทรศัพท์พนักงาน	VARCHAR(100)	Yes		
POSITION_CODE	รหัสตำแหน่งพนักงาน	VARCHAR(2)	Yes	FK	EMPLOYEE_POSITION

ตารางที่ 4.18 พจนานุกรมข้อมูลเอนทิตีความขัดข้องของระบบด้านซอฟต์แวร์

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
SW_CODE	รหัสความขัดข้องของระบบด้านซอฟต์แวร์	VARCHAR(5)	Yes	PK	
SW_DATE	วันที่รับแจ้งความขัดข้องของระบบด้านซอฟต์แวร์	DATE	Yes		
SW_NAME	ชื่อความขัดข้องของระบบด้านซอฟต์แวร์	VARCHAR(100)	Yes		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.18 (ต่อ)

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
SW_DETAIL	รายละเอียดความ จัดซื้อของระบบด้าน ซอฟต์แวร์	VARCHAR(255)	Yes		
SW_DATE_END	วันสิ้นสุดการแก้ไข ความจัดซื้อของ ระบบด้านซอฟต์แวร์	DATE	Yes		
SW_PICTURE	รูปภาพความจัดซื้อ ของระบบด้าน ซอฟต์แวร์	VARCHAR(255)	No		
SW_VOICE	เสียงลูกค้าที่แจ้งความ จัดซื้อของระบบด้าน ซอฟต์แวร์	VARCHAR(255)	No		
EMP_CODE	รหัสพนักงาน	VARCHAR(5)	Yes	FK	EMPLOYEE

ตารางที่ 4.19 พจนานุกรมข้อมูลเอนทิตีวิธีแก้ไขความจัดซื้อของระบบด้านซอฟต์แวร์

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
SW_SO_DATE	วันที่แก้ไขความ จัดซื้อของระบบด้าน ซอฟต์แวร์	DATE	Yes		
SW_SO_STATUS	สถานะการแก้ไขความ จัดซื้อของระบบด้าน ซอฟต์แวร์	VARCHAR(100)	Yes		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.19 (ต่อ)

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
SW_SO_DETAIL	วิธีแก้ไขความขัดข้องของระบบด้านซอฟต์แวร์	VARCHAR(255)	Yes		
SW_CODE	รหัสความขัดข้องของระบบด้านซอฟต์แวร์	VARCHAR(5)	Yes	PK, FK	SW_ISSUE

ตารางที่ 4.20 พจนานุกรมข้อมูลเอนทิตีความขัดข้องของระบบด้านฮาร์ดแวร์

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
HW_CODE	รหัสความขัดข้องของระบบด้านฮาร์ดแวร์	VARCHAR(5)	Yes	PK	
HW_DATE	วันที่รับแจ้งความขัดข้องของระบบด้านฮาร์ดแวร์	DATE	Yes		
HW_NAME	ชื่อความขัดข้องของระบบด้านฮาร์ดแวร์	VARCHAR(100)	Yes		
HW_DETAIL	รายละเอียดความขัดข้องของระบบด้านฮาร์ดแวร์	VARCHAR(255)	Yes		
HW_DATE_END	วันสิ้นสุดการแก้ไขความขัดข้องของระบบด้านฮาร์ดแวร์	DATE	Yes		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.20 (ต่อ)

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
HW_PICTURE	รูปภาพความชัดเจนของระบบด้านฮาร์ดแวร์	VARCHAR(255)	F		
HW_VOICE	เสียงลูกค้ำที่แจ้งความชัดเจนของระบบด้านฮาร์ดแวร์	VARCHAR(255)	F		
EMP_CODE	รหัสพนักงาน	VARCHAR(5)	Y	FK	EMPLOYEE

ตารางที่ 4.21 พจนานุกรมข้อมูลเอนทิตีวิธีแก้ไขความชัดเจนของระบบด้านฮาร์ดแวร์

ชื่อแอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ค่าที่เป็น null ไม่ได้	คีย์	ตารางอ้างอิง
HW_SO_DATE	วันที่แก้ไขความชัดเจนของระบบด้านฮาร์ดแวร์	DATE	Y		
HW_SO_STATUS	สถานะการแก้ไขความชัดเจนของระบบด้านฮาร์ดแวร์	VARCHAR(100)	Y		
HW_SO_DETAIL	วิธีแก้ไขความชัดเจนของระบบด้านฮาร์ดแวร์	VARCHAR(255)	Y		
HW_CODE	รหัสความชัดเจนของระบบด้านฮาร์ดแวร์	VARCHAR(5)	Y	PK, FK	HW_ISSUE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป

5.1 สรุปผลการศึกษา

โครงการศึกษาอิสระการพัฒนาระบบบริหารจัดการปัญหาด้วยแอนดรอยด์เพื่อสนับสนุนงานบริการหลังการขายและเพิ่มความรวดเร็วในการเรียนรู้งาน เป็นการพัฒนาในระบบในรูปแบบของแอปพลิเคชันบนอุปกรณ์พกพาประเภทโทรศัพท์เคลื่อนที่หรือแท็บเล็ต แอปพลิเคชันบนเว็บ ซึ่งได้ออกแบบระบบด้วยแนวคิดเชิงวัตถุ โดยใช้ภาษาจาวาเอ็มแอลเป็นเครื่องมือ ร่วมกับระบบฐานข้อมูลเชิงสัมพันธ์ ที่สร้างด้วยโปรแกรมมายเอสคิวแอล

ขอบเขตของการพัฒนาระบบจะครอบคลุมผู้ใช้งาน 4 กลุ่ม คือ ผู้ประสานงานโครงการ ผู้พัฒนาซอฟต์แวร์ (โปรแกรมเมอร์) ผู้บริการหลังการขาย (เซอร์วิส) และกรรมการผู้จัดการ โดย ส่วนของผู้ประสานงานโครงการ ประกอบด้วย การมอบหมาย การส่งรายรายเอียดความขัดข้องของระบบให้แก่โปรแกรมเมอร์และเซอร์วิส การติดตามความคืบหน้า การค้นหาและศึกษาความขัดข้องของระบบที่เคยเกิดขึ้นผ่านทางโทรศัพท์เคลื่อนที่หรืออุปกรณ์พกพาอื่นๆ ส่วนของผู้พัฒนาซอฟต์แวร์และผู้บริการหลังการขาย ประกอบด้วย การอัปเดตสถานะความขัดข้องของระบบ การบันทึกวิธีการแก้ไขความขัดข้องของระบบ การค้นหาและศึกษาความขัดข้องของระบบที่เกิดขึ้นผ่านทางเว็บไซต์ และ ส่วนของกรรมการผู้จัดการ ประกอบด้วย การตรวจสอบจำนวนความขัดข้องของระบบที่เกิดขึ้นผ่านทางเว็บไซต์ ในรูปแบบรายงาน

นอกจากนี้ข้อมูลพื้นฐานสำคัญของระบบ อันได้แก่ ตำแหน่งของพนักงาน รายชื่อพนักงาน ความขัดข้องของระบบด้านซอฟต์แวร์และด้านฮาร์ดแวร์ รวมถึงวิธีการแก้ไข จะดึงจากฐานข้อมูลกลาง ไปยังแอปพลิเคชันบนอุปกรณ์พกพาและแอปพลิเคชันบนเว็บไซต์ ส่วนชื่อผู้ใช้และรหัสผ่านที่เข้าใช้งานระบบนี้ จะใช้ชุดเดียวกับการเข้าเครื่องคอมพิวเตอร์ของพนักงาน ซึ่งจะช่วยลดการสับสนและลืมรหัสผ่านของผู้ใช้งานได้

สำหรับผลการศึกษาโครงการพัฒนาระบบสารสนเทศนี้ สามารถนำไปศึกษาและพัฒนาเพื่อใช้งานจริงกับบริษัทได้ ซึ่งจะเป็นประโยชน์ในการเพิ่มประสิทธิภาพในการดำเนินงานและอำนวยความสะดวกแก่พนักงาน อีกทั้งยังส่งเสริมภาพลักษณ์ที่ดีแก่บริษัท

5.2 ข้อจำกัดและข้อเสนอแนะ

ระบบบริหารจัดการปัญหาด้วยแอนครอยด์เป็นระบบที่พัฒนาขึ้นสำหรับใช้เป็นกรณีศึกษา และทดลองการใช้งาน เพื่อดูความสอดคล้องกับการปฏิบัติงานจริงให้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งอาจมีความจำเป็นต้องเพิ่มเติมคุณสมบัติบางอย่างเพื่อให้การทำงานมีความเป็นอัตโนมัติมากขึ้น เช่น การตรวจสอบประสิทธิภาพของพนักงาน ความหลากหลายในการออกรายงาน เป็นต้น หรือการพัฒนาต่อให้ระบบสามารถรองรับระบบปฏิบัติการอื่นๆ ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

จักรชัย ไสอินทร์ และพงษ์ศธร จันทร์ข่อย. 2554. **Basic Android App Development**. พิมพ์ครั้งที่ 1. นนทบุรี. บริษัท ไอดีซี พรีเมียร์ จำกัด.

ชัยวุฒิ พรหมบุตร. 2555. **อบรมการพัฒนาโปรแกรมแอนดรอยด์ขั้นพื้นฐาน**. กรุงเทพฯ. สถาบัน EWTC.

พร้อมเลิศ หล่อวิจิตร. 2555. **คู่มือเขียนแอปAndroid สำหรับผู้เริ่มต้น**. พิมพ์ครั้งที่ 1. กรุงเทพฯ. บริษัท โปรวิชั่น จำกัด.

ลานหน้าบ้าน (นามแฝง). **Android Service**. [ออนไลน์]. เข้าถึงได้จาก

:<http://www.laannaabaan.com/android-service/>. (วันที่สืบค้น : 23 สิงหาคม 2555)

Developers. 2548. **App Components**. [ออนไลน์]. เข้าถึงได้จาก :<http://developer.android.com>. (วันที่สืบค้น : 4 กรกฎาคม 2555)

PrajakBoonjaratcha. 2554. **แอนดรอยด์แอคทีวิตี (Activity)**. [ออนไลน์]. เข้าถึงได้จาก :

<http://mobiledep.blogspot.com/2011/09/activity.html>. (วันที่สืบค้น : 4 กรกฎาคม 2555)

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวรัฐชิตา สิ้นชัย
ที่อยู่	8/1 ซอยลาดพร้าว 69 แขวงสะพานสอง เขตวังทองหลาง กรุงเทพฯ 10310
ประวัติการศึกษา	2553 ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ สถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประสบการณ์ทำงาน	ปี 2553 – 2555 ผู้ประสานงานโครงการ บริษัท เอพี พาร์คกิ้ง จำกัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้