

## บทที่ 3

### ระเบียบวิธีวิจัย

#### 3.1 แนวทางการวิจัยและพัฒนา

สารนิพนธ์นี้มีวัตถุประสงค์เพื่อต้องการออกแบบและพัฒนาแอปพลิเคชัน การกรองข้อความสั้น บนโทรศัพท์เคลื่อนที่ ที่รองรับระบบปฏิบัติการแอนดรอยด์ ให้สามารถกรองข้อความสั้นเฉพาะหมายเลขที่มีอยู่ในสมุดโทรศัพท์ได้เพื่อลดจำนวนข้อความสั้นที่ไม่ต้องการ โดยแบ่งขั้นตอนการทำงานออกเป็น 3 ส่วน ดังนี้

3.1.1 ศึกษาข้อมูลทั่วไป ประกอบด้วย หลักการทำงานของระบบปฏิบัติการแอนดรอยด์ วิธีการพัฒนาแอปพลิเคชันบนแอนดรอยด์ เครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชัน และวิธีการเชื่อมต่อและตรวจสอบระหว่างการทำงานของแอปพลิเคชันการกรองข้อความสั้นกับระบบข้อความสั้นบนแอนดรอยด์

3.1.2 ออกแบบแอปพลิเคชันสำหรับการกรองข้อความสั้นบนโทรศัพท์เคลื่อนที่แอนดรอยด์ ในการออกแบบแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์นั้นจะต้องใช้เครื่องมือเพื่อช่วยในการพัฒนา ทั้งด้านของการเชื่อมต่อกันระหว่าง แอปพลิเคชันกับโทรศัพท์เคลื่อนที่แอนดรอยด์ ระบบข้อความสั้นกับแอปพลิเคชันการกรองข้อความ และการจัดการกับข้อความสั้นที่ได้กรองไว้

3.1.3 พัฒนาแอปพลิเคชันการกรองข้อความสั้น ตามที่ได้ออกแบบไว้ และแก้ไขปรับปรุงแอปพลิเคชันให้สามารถทำงานได้ตามวัตถุประสงค์ที่ได้วางไว้ โดยทดสอบกับอิมูเลเตอร์และบนโทรศัพท์เคลื่อนที่จริง

#### 3.2 เครื่องมือที่ใช้ในงานวิจัย

3.2.1 โทรศัพท์เคลื่อนที่ที่รองรับ ระบบปฏิบัติการแอนดรอยด์ จำนวน 1 เครื่อง ได้แก่ Samsung Galaxy S IIระบบ WCDMA/HSPA 850/900/1900/2100 MHz.

โครงข่าย 3G WCDMA/HSPA+

ความเร็วอินเทอร์เน็ต ดาว์นโหลด 21 Mbps อัปโหลด 5.76 Mbps

หน้าจอ 4.3 นิ้ว WVGA Super AMOLED Plus

ความละเอียดหน้าจอ 480x800 พิกเซล 16 ล้านสี

หน่วยประมวลผล 1.2 กิกะเฮิร์ตซ์ คู่อัตตอร์  
 หน่วยความจำภายใน 16 กิกะไบต์  
 กล้องถ่ายรูป 8 ล้านพิกเซล พร้อมระบบบอโต้โฟกัส  
 บลูทูธ V3.0 + A2DP  
 Wi-Fi 802.11 b/g/n  
 ระบบปฏิบัติการ Android 2.3 Gingerbread  
 ข้อความ SMS และ MMS  
 อีเมล POP3/IMAP/SMTP  
 จาวา MIDP emulator  
 แบตเตอรี่ ลิเทียมไอออน 1650 มิลลิแอมป์

### 3.2.2 เครื่องคอมพิวเตอร์ส่วนบุคคล จำนวน 1 เครื่อง มีคุณสมบัติดังนี้

CPU Intel Core i5-2410M (2.30 GHz, 3 MB L3 Cache)  
 RAM 4 GB DDR3  
 HARD DISK 640 GB 5400 RPM  
 Graphic Chip nVidia GeForce GT 520M (1 GB GDDR3)  
 DVD +/-RW Drive  
 USB 2.0  
 Display 14.1 inch WXGA (1366x768) LED  
 MS Windows 7

### 3.2.3 ซอฟต์แวร์สำหรับการพัฒนา

Java Development Kit (JDK)  
 โปรแกรม Eclipse  
 Android SDK  
 Android Development Tools Plugin

## 3.3 แผนการดำเนินงาน

ในการออกแบบและพัฒนาระบบกรองข้อความสั้นเฉพาะเลขหมายนั้น มีแผนการดำเนินงานดังตารางที่ 3.1

ตารางที่ 3.1 แผนการดำเนินงาน

ลักษณะงาน	ตารางเวลาของโครงการ (เดือนที่)							
	1	2	3	4	5	6	7	8
1. ศึกษาการทำงานของระบบปฏิบัติการแอนดรอยด์								
2. ศึกษาวิธีการและเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์								
3. ออกแบบแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์								
4. พัฒนาวิธีการกรองข้อความสั้นที่ใช้พัฒนาแอปพลิเคชัน								
5. ทดสอบแอปพลิเคชัน								
6. แก้ไขข้อผิดพลาดและปรับปรุงให้สมบูรณ์								

### 3.4 ขั้นตอนการดำเนินงานวิจัย

#### 3.4.1 รวบรวมข้อมูลและศึกษาการทำงานของระบบปฏิบัติการแอนดรอยด์

1. ประเภทของชุดซอฟต์แวร์ เนื่องจากแอนดรอยด์นั้นเปิดให้นักพัฒนาเข้าไปดูรหัสต้นฉบับได้ ทำให้มีผู้พัฒนาจากหลายฝ่ายนำเอารหัสต้นฉบับมาปรับแต่ง และสร้างแอนดรอยด์ในแบบฉบับของตนเองขึ้น จึงแบ่งประเภทของแอนดรอยด์ออกได้เป็น 3 ประเภท (จักรชัย โสอินทร์ และ พงษ์ศธร จันทร์ช้อย, 2554) ดังต่อไปนี้คือ

1.1) Android Open Source Project (AOSP) เป็นแอนดรอยด์ประเภทแรกที่ถูกเปิดให้สามารถนำ “ต้นฉบับแบบเปิด” ไปติดตั้งและใช้งานในอุปกรณ์ต่าง ๆ ได้โดยไม่ต้องเสียค่าใช้จ่าย

1.2) Open Handset Mobile (OHM) เป็นแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่มบริษัทผู้ผลิตอุปกรณ์พกพา ที่เข้าร่วมกับกูเกิลในนาม Open Handset Alliances (OHA) ซึ่งบริษัทเหล่านี้จะพัฒนาแอนดรอยด์ในแบบฉบับของตนเองออกมา โดยรูปร่างหน้าตาการแสดงผล และฟังก์ชันการใช้งาน จะมีความเป็นเอกลักษณ์ และมีลิขสิทธิ์เป็นของตน พร้อมได้รับสิทธิ์ในการมีบริการเสริมต่าง ๆ จากกูเกิล ที่เรียกว่า Google Mobile Service (GMS) ซึ่งเป็นบริการเสริมที่ทำให้แอนดรอยด์มีประสิทธิภาพ เป็นไปตามจุดประสงค์ของแอนดรอยด์ แต่การจะได้มาซึ่ง GMS นั้น ผู้ผลิตจะต้องทำการทดสอบระบบ และขออนุญาตกับทางกูเกิลก่อน จึงจะนำเครื่องออกสู่ตลาดได้

1.3) Cooking หรือ Customize เป็นแอนดรอยด์ที่นักพัฒนานำเอารหัสต้นฉบับจากแหล่งต่าง ๆ มาปรับแต่ง ในแบบฉบับของตนเอง โดยจะต้องทำการปลดล็อคสิทธิ์การใช้งานอุปกรณ์ หรือ Unlock เครื่องก่อน จึงจะสามารถติดตั้งได้ โดยแอนดรอยด์ประเภทนี้ถือเป็นประเภทที่มีความสามารถมากที่สุด เท่าที่อุปกรณ์เครื่องนั้น ๆ จะรองรับได้ เนื่องจากได้รับการปรับแต่งให้เข้ากับอุปกรณ์นั้น ๆ จากผู้ใช้งานจริง

2. สิทธิ์ในการใช้งานระบบ เช่นเดียวกับระบบปฏิบัติการทั่วไป ที่มีการจำกัดการใช้งานและการเข้าถึงส่วนต่าง ๆ ภายในระบบ เพื่อความปลอดภัยของระบบ และ ผู้ใช้งาน อุปกรณ์ที่ติดตั้งระบบแอนดรอยด์จึงมีการจำกัดสิทธิ์ไว้ (เว้นแต่ได้ทำการปลดล็อคสิทธิ์ หรือ root เครื่องแล้ว) สามารถแบ่งสิทธิ์ของผู้ใช้ในการเข้าถึงระบบได้ดังต่อไปนี้

2.1) สิทธิ์ root สิทธิ์การใช้ใช้งานระดับราก ซึ่งถือว่าเป็นรากฐานของระบบ จึงมีความสามารถในการเข้าถึงทุก ๆ ส่วนของระบบ

2.2) สิทธิ์ ADB (Android Develop Bridge) นักพัฒนาสามารถเข้าถึงส่วนต่าง ๆ ของระบบได้ผ่านสิทธิ์นี้

2.3) Application & System สิทธิ์ของโปรแกรมในการเข้าถึงระบบ และสิทธิ์ของระบบในการเข้าถึงอุปกรณ์โดยสิทธิ์เหล่านี้ ตัวระบบจะเป็นตัวจัดการมอบและถอนสิทธิ์ ตามเงื่อนไขที่กำหนดซึ่งจะถูกแบ่งย่อยออกเป็นหลายหัวข้อ

### ตารางที่ 3.2 การอนุญาตให้เข้าถึงข้อมูลของข้อความสั้น (SMS)

การอนุญาตที่เกี่ยวกับโทรศัพท์	วัตถุประสงค์
android.permission.RECEIVE_SMS	อนุญาตให้แอปพลิเคชันสามารถตรวจสอบข้อความสั้นที่เข้ามาได้
android.permission.READ_SMS	อนุญาตให้แอปพลิเคชันสามารถอ่านข้อความสั้นที่เข้ามาได้
android.permission.SEND_SMS	อนุญาตให้แอปพลิเคชันสามารถส่งข้อความสั้นได้
android.permission.WRITE_SMS	อนุญาตให้สามารถเขียนข้อความสั้นลงบนแอปพลิเคชันที่สร้างขึ้นได้

ที่มา: Android Developer [online] : เข้าถึง 15 ต.ค. 2554. จาก

<http://developer.android.com/reference/android/Manifest.permission.html>

2.4) End-user ผู้ใช้งานขั้นสุดท้าย ที่ใช้การเข้าถึงส่วนต่าง ๆ ของระบบผ่านช่องทางสิทธิ์ที่โปรแกรมได้รับอีกที โดยจะถูกจำกัดไม่ให้เข้าถึงในส่วนที่เป็นอันตรายต่อแกนระบบและอุปกรณ์

3.4.2 ศึกษาเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ (จักรชัย โสอินทร์ และ พงษ์ศธร จันทร์ช้อย, 2554)

1. Java Development Kit (JDK) เป็นชุดคำสั่งในการพัฒนาโปรแกรมด้วยภาษาจาวา โดยชุดพัฒนาโปรแกรม JDK ประกอบด้วย 3 รุ่นดังนี้

1.1) Java SE (Standard Edition) สำหรับพัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์ทั่วไป

1.2) Java ME (Micro Edition) สำหรับพัฒนาโปรแกรมบนอุปกรณ์พกพา เช่น โทรศัพท์มือถือ หรือ พีดีเอ ส่วนมากใช้สำหรับเขียนโปรแกรมเกมส์

1.3) Java EE (Enterprise Edition) สำหรับพัฒนาโปรแกรมในองค์กรใหญ่ ๆ หรือมีขอบเขตของโครงการที่กว้างมาก ชุดพัฒนาโปรแกรม JDK นั้น เป็นมาตรฐานที่ถูกกำหนดโดยบริษัท ซัน ไมโครซิสเต็มส์ และมีผู้พัฒนาชุดโปรแกรม JDK ออกมามากมาย เช่น ซัน ไมโครซิสเต็มส์ ไอบีเอ็ม และ บีอีเอ ซิสเต็มส์

2. โปรแกรมที่ใช้สำหรับเขียนโปรแกรมภาษาจาวา (Java)(Eclipse) เป็นเครื่องมือที่ช่วยสนับสนุนในการพัฒนาซอฟต์แวร์ โดยเฉพาะภาษาจาวา และที่สำคัญ eclipse เป็นซอฟต์แวร์แบบเปิด (open source) ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งให้บริการขั้นพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่าง ๆ จากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และมีส่วนประกอบที่เรียกว่า plug-in development (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse plug-ins ดังนั้นหากต้องการให้ Eclipse ทำงานใดเพิ่มเติม ก็เพียงแค่พัฒนา plug-in สำหรับงานนั้นขึ้นมา และนำ plug-in นั้นมาติดตั้งเพิ่มเติมให้กับ Eclipse ที่มีอยู่เท่านั้น Eclipse plug-in ที่มาพร้อมกัน เมื่อมีการดาวน์โหลดมาครั้งแรกก็คือองค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียนและ debug โปรแกรมภาษาจาวา

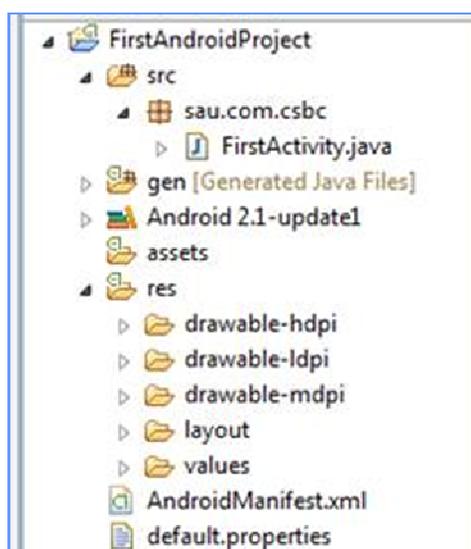
3. Android Development Tool (ADT) โปรแกรมเสริมที่ทำให้พัฒนาแอนดรอยด์ บน Eclipse ได้ ทำหน้าที่เป็นตัวเชื่อมระหว่าง Android SDK และ Eclipse ให้สามารถทำงานเสมือนเป็นเครื่องมือขึ้นเดียวกัน หากใช้ Eclipse แล้วไม่เอา ADT ไปด้วยก็เหมือนกับไม่ได้ใช้ความสามารถ

ของ Eclipse เลย โดยสามารถดาวน์โหลดได้จาก [http://developer.android.com/sdk/adt\\_download.html](http://developer.android.com/sdk/adt_download.html)

4. Android SDK ย่อมาจาก Android Software Development Kit ซึ่งเป็นชุดโปรแกรมที่ ถูกพัฒนาออกมาเพื่อแจกจ่ายให้นักพัฒนาแอปพลิเคชัน หรือผู้สนใจไปดาวน์โหลดเพื่อนำมาใช้ กัน โดยไม่มีค่าใช้จ่าย ซึ่งในชุด SDK นั้นจะมีโปรแกรมและไลบรารีต่าง ๆ ที่จำเป็นต่อการพัฒนา แอปพลิเคชันบนแอนดรอยด์ อย่างเช่น Emulator ซึ่งทำให้ผู้ใช้สามารถสร้างแอปพลิเคชันและนำมา ทดลองรันบนอีมูเลเตอร์ก่อน โดยมีสภาพแวดล้อมเหมือนมือถือที่รันอยู่บนระบบปฏิบัติการแอนดรอยด์จริง ๆ

### 3.4.3 ออกแบบและพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

#### 1. โครงสร้างพื้นฐานของแอนดรอยด์ (android project structure)



ภาพที่ 3.1 โครงสร้างพื้นฐานของโปรแกรมนักพัฒนา

1.1) src (Source Code) เป็นส่วนของซอร์สโค้ดที่สร้างขึ้นและเขียนขึ้นเป็นไฟล์ .java โดยจะมีทั้งชื่อคลาส แอตทริบิวต์ และเมธอด

1.2) gen (Generated Java Files) เป็นส่วนที่ถูกสร้างขึ้นโดยอัตโนมัติ เช่นไฟล์ R.java ซึ่งภายในประกอบด้วย text และ UI Element โดยถูกนำไปยังโปรเจกต์ผ่าน android plug-in ซึ่งไฟล์นี้จะเป็นเหมือนตัวชี้ (pointer) ไปยัง drawable, layout, values directories

1.3) res (Resource) จะเป็นส่วนของการแสดงผลของไฟล์อื่น ๆ ที่นำมาเป็นส่วนประกอบ ร่วมกับโค้ดที่เขียนขึ้น โดยแอนดรอยด์จะมีการแบ่งไฟล์ตามชนิดของ resource นั้น โดยที่เมื่อมีการสร้างโปรเจกต์ขึ้นมาใหม่จะมี directory ย่อย ๆ อีก 5 directory คือ

1.4) drawable-hdpi/drawable-ldpi/drawable-mdpi ซึ่งใช้สำหรับเก็บภาพ

1.5) layout ใช้สำหรับจัดวาง view ต่าง ๆ

1.6) values ใช้สำหรับจัดการค่าต่าง ๆ ที่นำไปใช้ในแอปพลิเคชัน ทั้งนี้ยังสามารถสร้าง directory ต่าง ๆ ขึ้นเองเพื่อเก็บข้อมูลอื่น ๆ ที่จะใช้ในแอปพลิเคชันได้ด้วย

1.7) AndroidManifest.xml เป็นโครงสร้างของ xml ไฟล์ ซึ่งใน xml นั้นจะเป็นการกำหนดคุณสมบัติ และการตั้งค่าต่าง ๆ ของแอปพลิเคชัน เช่น ชื่อของแอปพลิเคชัน เวอร์ชันของโค้ดการกำหนดสิทธิ์ ต่าง ๆ ในการเข้าถึงแอปพลิเคชัน

2. องค์ประกอบของแอนดรอยด์แอปพลิเคชัน (Android Application Components) (พร้อมเลิส หล่อวีจิตร, 2555)

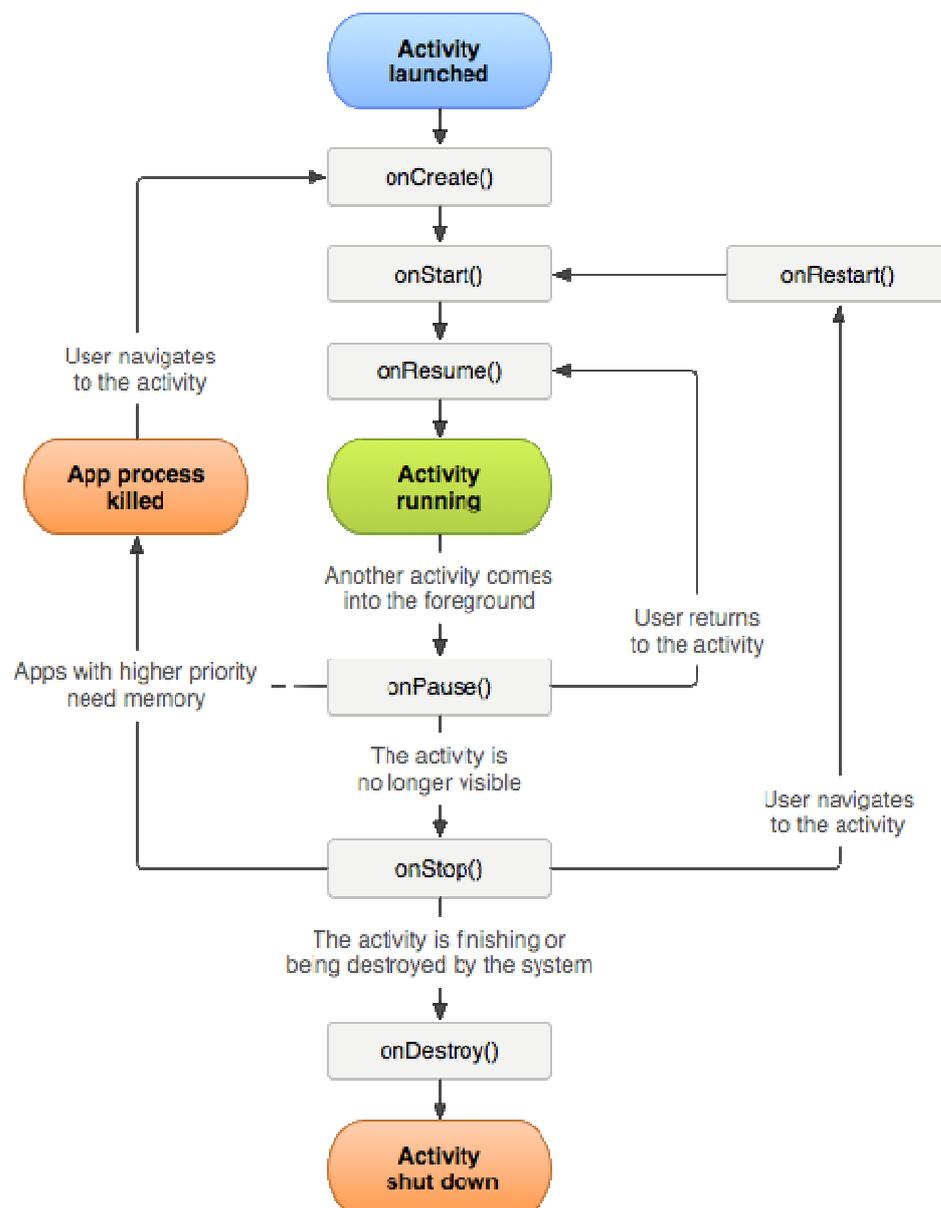
2.1) Activity (User Interface) คือสิ่งที่ใช้ในการแสดงผล เพื่อให้ผู้ใช้งานได้เห็นและได้ใช้งาน โดยในแต่ละแอปพลิเคชันนั้น อาจมี activity เดียว หรือหลาย activity ก็ได้ และสิ่งที่อยู่ใน activity นั้นจะเรียกว่า view ซึ่งก็มีอยู่ในหลาย ๆ รูปแบบ เช่น buttons, text, fields, scroll bars, menu items, check boxes และอื่น ๆ อีกมากมาย

2.2) Service (Service Provider) คือสิ่งที่ไม่มีส่วนของการแสดงผล แต่ถูกเรียกให้รันในลักษณะของ background process โดย service นั้น อาจมีการทำอะไรบางอย่าง เช่น ติดต่อรับส่งข้อมูลผ่านเครือข่าย หรือคำนวณค่าต่าง ๆ แล้วก็ส่งผลลัพธ์นั้นไปแสดงยัง activity ก็ได้

2.3) Broadcast receiver (Data Provider) คือตัวที่ใช้สำหรับคอยรับ และตอบสนองต่อเหตุการณ์ต่าง ๆ ที่เกิดขึ้น เช่น เมื่อแบตเตอรี่ต่ำ ผู้ใช้ทำการเปลี่ยนภาษา มีการโทรออก มีข้อความเข้า และอื่น ๆ ถึงแม้ broadcast receiver จะไม่มีส่วนของการแสดงผล แต่ก็สามารถเรียก activity ขึ้นมาแสดงผลให้ผู้ใช้งานได้ หรืออาจจะใช้สิ่งที่เรียกว่า Notification Manager ซึ่งจะเป็นตัวที่แจ้งเตือนในรูปแบบของการสั่น การแสดงไฟกระพริบที่หน้าจอ หรือการส่งเสียงออกมาจากลำโพง โดยจะมีไอคอน แสดงอยู่บน status bar เพื่อให้ผู้ใช้เข้าไปกดเพื่อดูเหตุการณ์ที่เกิดขึ้น

2.4) Content Provider (System Event Listener) คือกลุ่มของข้อมูลที่สร้างขึ้นจากแอปพลิเคชัน เพื่อให้แอปพลิเคชันอื่น ๆ ได้นำไปใช้ โดยการจัดเก็บข้อมูลของ content provider นั้นจะอยู่ในลักษณะของไฟล์ ฐานข้อมูล SQLite และอื่น ๆ ตัวอย่างแอปพลิเคชันที่ใช้งาน Content Provider ที่เห็นเด่นชัดที่สุดคือโปรแกรม Contacts ที่แสดงรายชื่อใน contact นั้นเอง

3. Activity Life Cycle มีหน้าที่ในการสร้างและควบคุมการทำงานของหน้าจอ เพื่อโต้ตอบกับผู้ใช้งาน ซึ่งจะมีแค่ activity เดียวเท่านั้นที่โต้ตอบกับผู้ใช้งานในขณะใดขณะหนึ่ง โดยแต่ละ activity แต่ละตัวจะมีวงจรชีวิตเป็นของตนเอง โดยแบ่งเป็นสถานะดังนี้



ภาพที่ 3.2 สถานะของ Activity

ที่มา: Activity Lifecycle [online] : เข้าถึง 5 ต.ค. 2554. จาก

<http://developer.android.com/reference/android/app/Activity.html>

- 3.1) Running เป็นสถานะที่ activity กำลังทำงานในขณะนั้น
- 3.2) Pause เป็นสถานะที่ activity ปรากฏอยู่บนหน้าจอแต่ไม่ได้ทำงานในขณะนั้น
- 3.3) เป็นสถานะที่ activity ไม่ปรากฏบนหน้าจอ
- 3.4) Kill เป็นสถานะที่ activity ถูกทำลายไปแล้วเมื่อ activity อยู่ในสถานะ pause หรือ stop activity สามารถ ถูกทำลายได้โดยระบบ
- 3.5) onCreate(Bundle) จะถูกเรียกเมื่อ activity มีการสร้างตัวเองครั้งแรก เป็นเมธอดสำคัญในการสร้างหน้าจอการทำงานต่าง ๆ
- 3.6) onStart() จะถูกเรียกหลังจากที่ activity กลับมาจากสถานะ stop เพื่อเริ่มทำงานใหม่อีกครั้ง
- 3.7) onStart() จะถูกเรียกเมื่อ activity เริ่มแสดงให้ผู้ใช้งานเห็น
- 3.8) onResume() จะถูกเรียกเมื่อ activity สามารถโต้ตอบกับผู้ใช้งานได้
- 3.9) onPause() จะถูกเรียกเมื่อมี activity อื่นกำลังทำงานเป็นเมธอด ในการเก็บข้อมูลที่สำคัญหรือทำการปล่อยหน่วยความจำเพื่อให้ activity อื่น ได้ใช้งาน
- 3.10) onStop() จะถูกเรียกเมื่อ activity ไม่ปรากฏบนหน้าจอ แล้วจะเข้าสู่สถานะ stop
- 3.11) onDestroy() จะถูกเรียกเมื่อ activity กำลังจะถูกทำลาย
- 3.12) onSaveInstanceState(Bundle) จะถูกเรียกก่อน onPause() เพื่อเก็บสถานะต่าง ๆ ของ activity ไว้ใช้เมื่อ activity นี้ได้กลับมาทำงานอีกครั้ง
- 3.13) onRestoreInstanceState(Bundle) จะถูกเรียกก่อน onResume() เพื่อนำสถานะต่าง ๆ ของ activity ที่เก็บไว้มาใช้งาน