

บทที่ 2

ทฤษฎีและแนวคิดที่เกี่ยวข้อง

1.1 กูเกิลแอนดรอยด์ (Google Android)

กูเกิลแอนดรอยด์ (Google Android) คือ ระบบปฏิบัติการ ที่เป็นซอฟต์แวร์แพลตฟอร์มบนมือถือสร้างขึ้นมาจากระบบปฏิบัติการลินุกซ์(Powered by the Linux kernel) พัฒนาขึ้นมาโดยกูเกิล กูเกิลแอนดรอยด์นั้นได้เปิดให้นักพัฒนาสามารถเข้ามาจัดการเขียนโค้ดต่างๆได้ด้วยภาษาจาวา และเขียนควบคุมอุปกรณ์ต่างๆผ่านทางจาวาไลบรารีที่ทางกูเกิลพัฒนาขึ้น โดยเฉพาะ(Google-developed Java libraries) โปรแกรมต่างๆ ที่รันบนกูเกิลแอนดรอยด์สามารถเขียนได้ด้วยภาษาซี(C) และภาษาอื่นส่วนการพัฒนาผ่านการคอมไพล์ด้วยสถาปัตยกรรมแบบ ARM Native Code(32bit) นั้น ยังไม่ได้รับการสนับสนุนจากทางกูเกิลแต่อย่างใด

กูเกิลแอนดรอยด์ ได้เปิดตัวเป็นครั้งแรกในวันที่ 5 พฤศจิกายน พ.ศ. 2550 โดยทางกูเกิลได้ประกาศก่อตั้ง Open Handset Alliance กลุ่มบริษัทฮาร์ดแวร์, ซอฟต์แวร์ และการสื่อสาร 48 แห่ง และได้ออกมาให้ยืมโฉมตัวจริงกันในช่วงปี 2551 ที่ผ่านมา ลิขสิทธิ์ของกูเกิลแอนดรอยด์นั้นจะอยู่ในลักษณะของฟรีซอฟต์แวร์และโอเพ่นซอร์ซ โดยอยู่ภายใต้สิทธิบัตรของ ครีเอทีฟ คอมมอนส์ แอททริบิว 2.5 ซึ่งทำให้ผู้ใช้นั้นสามารถดาวน์โหลดซอฟต์แวร์ของกูเกิลแอนดรอยด์ไปใช้ได้ฟรี และยังสามารถนำซอฟต์แวร์ที่ได้ไปแชร์แจกต่อได้แต่ไม่อนุญาตให้แก้ไขโดยการนำเอาชื่อผู้เขียนซอฟต์แวร์หรือรายการสิทธิบัตรของซอฟต์แวร์นั้นออกตัวโปรแกรม

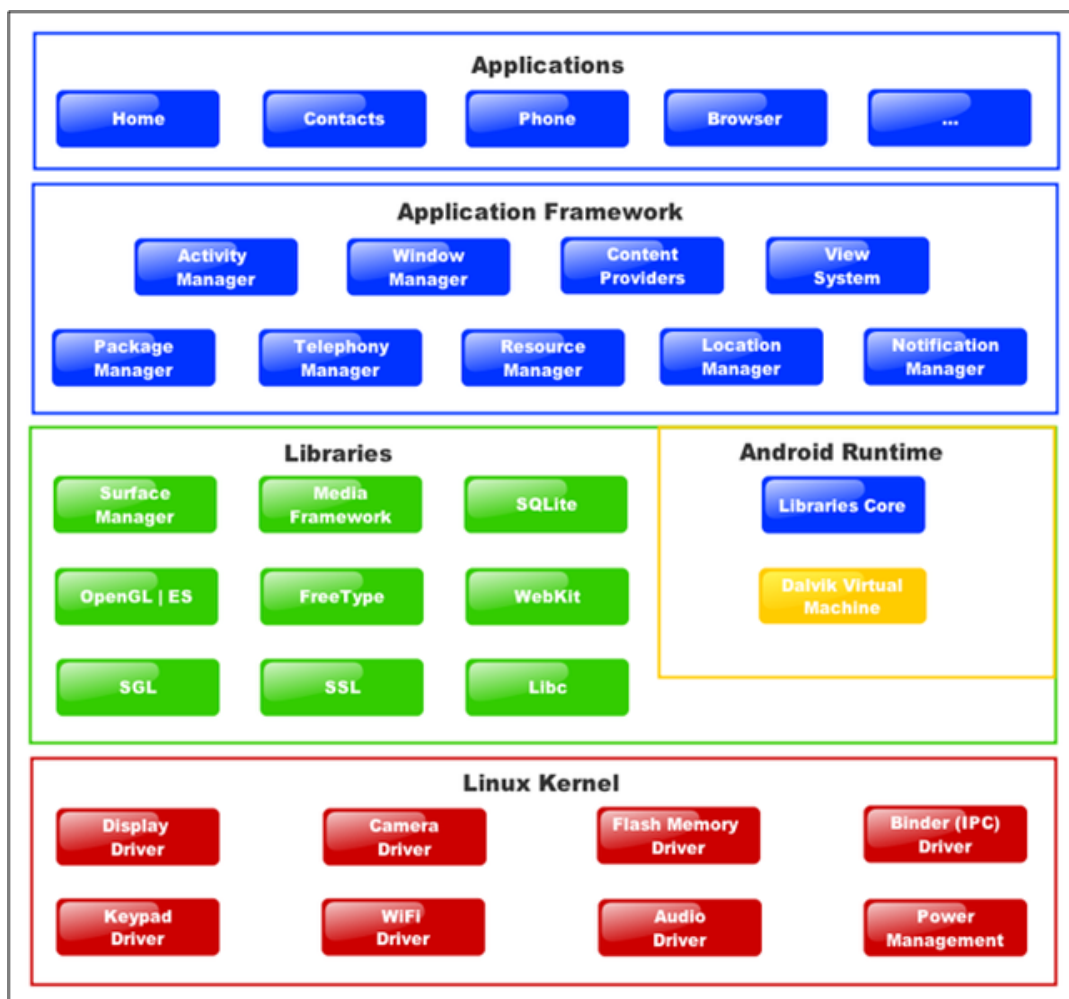
Android คือ ระบบปฏิบัติการ (OS) หรือแพลตฟอร์ม ที่จะใช้ควบคุมการทำงานบนอุปกรณ์อิเล็กทรอนิกส์ต่างๆ สำหรับโทรศัพท์มือถือ และอุปกรณ์พกพา โดยมี กูเกิลอิงก์, ที-โมบาย, เอชทีซี, ควอลคอมม์, โมโตโรลา และบริษัทชั้นนำอีกมากมายร่วมพัฒนาโปรเจกต์ แอนดรอยด์ ผ่านกลุ่มพันธมิตรเครื่องมือถือสื่อสารระบบเปิด (Open Handset Alliance) ซึ่งเป็นกลุ่มพันธมิตรชั้นนำระดับนานาชาติด้านเทคโนโลยีและเครื่องมือถือ สารเคลื่อนที่ ซึ่ง Android ประกอบด้วยระบบปฏิบัติการ ไลบรารี เฟรมเวิร์ค และซอฟต์แวร์อื่นๆ ที่จำเป็น ในการพัฒนา ซึ่งเทียบเท่ากับ Windows Mobile, PalmOS, Symbian, OpenMoko และ Maemo ของโนเกีย โดยใช้องค์ประกอบที่เป็นโอเพ่นซอร์สหลายอย่าง เช่น Linux Kernel, SSL, OpenGL, FreeType, SQLite, WebKit และเขียนไลบรารีเฟรมเวิร์คของตัวเองเพิ่มเติม ซึ่งทั้งหมดจะโอเพ่นซอร์ส ใช้ (Apache License)

ความร่วมมือครั้งนี้มีเป้าหมายในการส่งเสริมนวัตกรรมบนเครื่องมือสื่อสาร เพื่อให้ได้รับประสบการณ์ที่เหนือกว่าแพลตฟอร์มโมบายทั่วไปที่มีอยู่ในปัจจุบัน ทั้งนี้ การนำเสนอมิติใหม่ของแพลตฟอร์มระบบเปิดให้แก่ักพัฒนาจะช่วยช่วยให้กลุ่มคนเหล่านี้ทำงานร่วมกันได้มีประสิทธิภาพยิ่งขึ้น โดย แอนดรอยด์ จะช่วยเร่งและผลักดันบริการระบบสื่อสารรูปแบบใหม่ไปสู่ผู้บริโภคได้อย่างที่ไม่เคยเกิดขึ้นมาก่อน

สถาปัตยกรรมแอนดรอยด์

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

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



รูป 2.1 แสดงสถาปัตยกรรมแอนดรอยด์

ชั้นแอปพลิเคชัน (Application)

ชั้นนี้จะเป็นชั้นที่อยู่บนสุดของโครงสร้างสถาปัตยกรรม Android ซึ่งเป็นส่วนของแอปพลิเคชันที่พัฒนาขึ้นมาใช้งาน เช่น แอปพลิเคชันรับ/ส่งอีเมล, SMS, ปฏิทิน, แผนที่, เว็บเบราว์เซอร์, รายชื่อผู้ติดต่อ เป็นต้น ซึ่งแอปพลิเคชันจะอยู่ในรูปแบบของไฟล์ .apk



รูป 2.2 แสดงตัวอย่างแอปพลิเคชัน

1.1.1 ชั้นแอปพลิเคชันเฟรมเวิร์ค (Application Framework)

ในชั้นนี้จะอนุญาตให้นักพัฒนาสามารถเข้าเรียกใช้งาน โดยผ่าน API (Application Programming Interface) ซึ่ง Android ได้ออกแบบไว้เพื่อลดความซ้ำซ้อนในการใช้งาน application component โดยในชั้นนี้ประกอบด้วยแอปพลิเคชันเฟรมเวิร์คดังนี้

- ViewSystem เป็นส่วนที่ใช้ในการควบคุมการทำงานสำหรับการสร้างแอปพลิเคชัน เช่น lists, grids, textboxes, buttons และ embeddablewebbrowser
- LocationManager เป็นส่วนที่จัดการเกี่ยวกับตำแหน่งของเครื่องอุปกรณ์พกพาเคลื่อนที่
- ContentProvider เป็นส่วนที่ใช้ควบคุมการเข้าถึงข้อมูลที่มีการใช้งานร่วมกัน (Shared data) ระหว่างแอปพลิเคชันที่แตกต่างกัน เช่น ข้อมูลผู้ติดต่อ (Contact)
- ResourceManager เป็นส่วนที่จัดการข้อมูลต่างๆ ที่ไม่ใช่ส่วนของโค้ดโปรแกรม เช่น รูปภาพ, localizedstrings, layout ซึ่งจะอยู่ในไดเรกทอรี res/

- NotificationManager เป็นส่วนที่ควบคุมอีเวนต์ (Event) ต่างๆ ที่แสดงบนแถบสถานะ (Statusbar) เช่น ในกรณีที่ได้รับความหรือสายที่ไม่ได้รับและการแจ้งเตือนอื่นๆ เป็นต้น
- ActivityManager เป็นส่วนควบคุม LifeCycle ของแอปพลิเคชัน

1.1.2 ชั้นไลบรารี (Library)

Android ได้รวบรวมกลุ่มของไลบรารีต่างๆ ที่สำคัญและมีความจำเป็นเอาไว้มากมาย เพื่ออำนวยความสะดวกให้กับนักพัฒนาและง่ายต่อการพัฒนาโปรแกรม โดยตัวอย่างของไลบรารีที่สำคัญเช่น

- **System C library** เป็นกลุ่มของไลบรารีมาตรฐานที่อยู่บนพื้นฐานของภาษา C ไลบรารี (libc) สำหรับ embeddedsystem ที่มีพื้นฐานมาจาก Linux

MediaLibraries เป็นกลุ่มการทำงานมัลติมีเดีย เช่น MPEG4, H.264, MP3, AAC, AMR, JPG, และ PNG

- **SurfaceManager** เป็นกลุ่มการจัดการรูปแบบหน้าจอ การวาดหน้าจอ
- **2D/3D library** เป็นกลุ่มของกราฟิกแบบ 2 มิติ หรือ SGL (ScalableGraphicsLibrary) และแบบ 3 มิติ หรือ OpenGL
- **FreeType** เป็นกลุ่มของบิตแมป (Bitmap) และเวกเตอร์ (Vector) สำหรับการเรนเดอร์ (Render) ภาพ
- **SQLite** เป็นกลุ่มของฐานข้อมูล โดยนักพัฒนาสามารถใช้งานข้อมูลนี้เก็บข้อมูลแอปพลิเคชันต่างๆ ได้
- **BrowserEngine** เป็นกลุ่มของการแสดงผลบนเว็บเบราว์เซอร์ โดยอยู่บนพื้นฐานของ Webkit

1.1.2.1 Android Runtime

ชั้นนี้จะเป็นชั้นย่อยที่ถูกแบ่งมาจากชั้นไลบรารี ซึ่งจะประกอบด้วย 2 ส่วนหลัก คือ DalvikVM (VirtualMachine) ส่วนนี้ถูกเขียนด้วยภาษา Java เพื่อใช้เฉพาะการใช้งานในอุปกรณ์เคลื่อนที่ DalvikVM จะแตกต่างจาก JavaVM (VirtualMachine) คือ DalvikVM จะรันไฟล์ .dex ที่คอมไพล์มาจากไฟล์ .class และ .jar โดยมี tool ที่ชื่อว่า dx ทำหน้าที่ในการบีบอัดคลาส Java ทั้งนี้

ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์เคลื่อนที่มากกว่า .class เพื่อต้องการใช้พลังงานจากแบตเตอรี่อย่างมีประสิทธิภาพสูงสุด

CoreJavaLibrary ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็มีความแตกต่างจากไลบรารีของ JavaSE (JavaStandardEdition) และ JavaME (JavaMobileEdition)

ชั้นลินุกซ์เคอร์เนล (LinuxKernel)

ระบบ Android นั้นถูกสร้างบนพื้นฐานของระบบปฏิบัติการ Linux โดยในชั้นนี้จะมีฟังก์ชันการทำงานหลายๆ ส่วน แต่โดยส่วนมากแล้วจะเกี่ยวข้องกับฮาร์ดแวร์โดยตรง เช่น การจัดการหน่วยความจำ (MemoryManagement) การจัดการโพรเซส (ProcessManagement) การเชื่อมต่อเครือข่าย (Networking) เป็นต้น

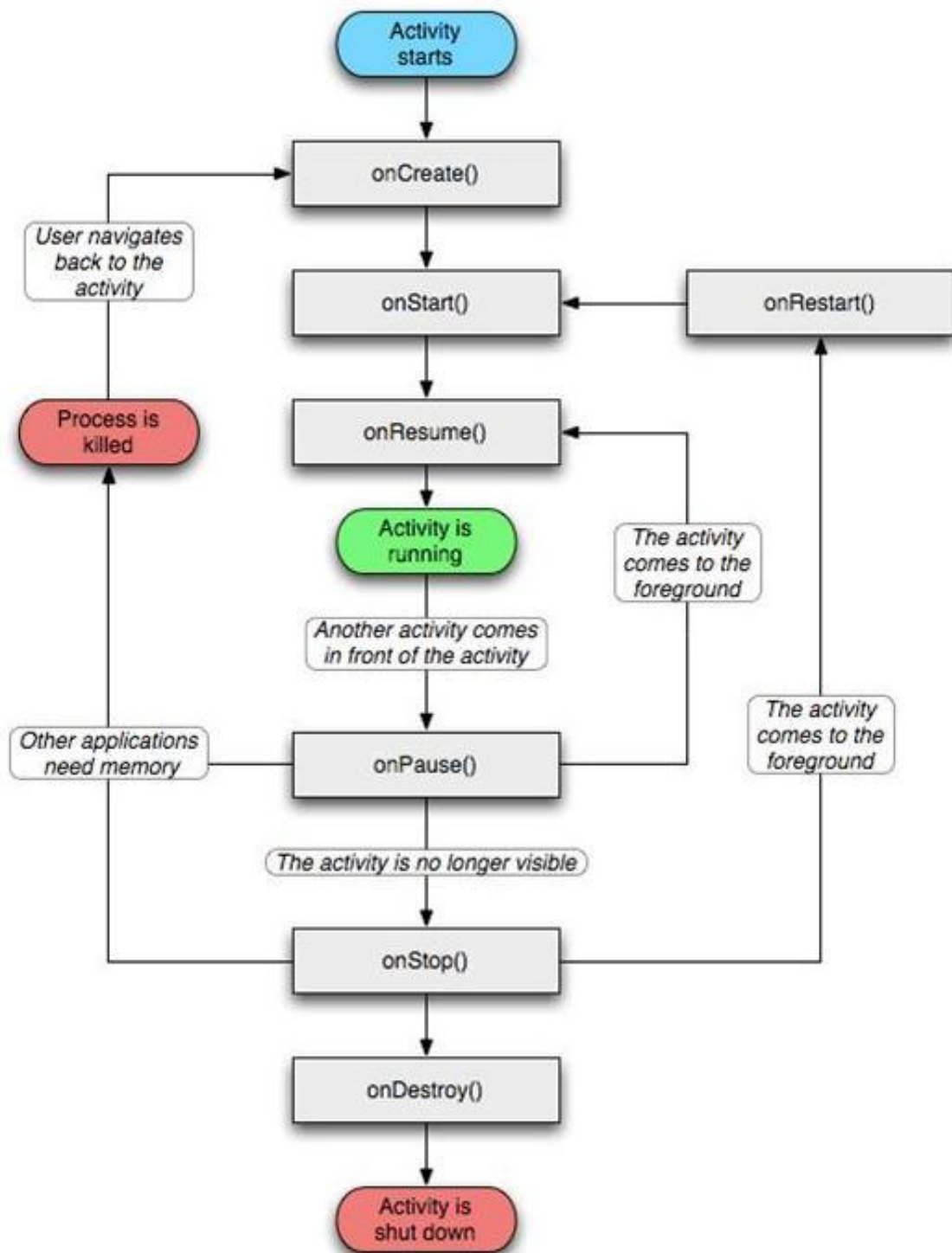
2.3 องค์ประกอบของแอนดรอยด์แอปพลิเคชัน (Android Application Components)

- 2.3.1 Activity (User Interface) คือ สิ่งที่ใช้ในการแสดงผล ออกมาเพื่อให้ผู้ใช้งานได้เห็น และได้ใช้งาน โดยในแต่ละ Application นั้น อาจจะมี activity เดียว หรือ หลายๆ activity ก็ได้ หรืออาจจะไม่มี Activity เลยก็ได้ และสิ่งที่อยู่ใน activity นั้นจะเรียกว่า view ซึ่งก็มีอยู่ในหลายรูปแบบ เช่น buttons, text fields, scroll bars, menu items, check boxes และอื่นๆ
- 2.3.2 Service (Service Provider) คือ สิ่งที่ไม่มีส่วนของการแสดงผล แต่ถูกเรียกว่ารันอยู่ในลักษณะของ background process โดย service นั้นอาจจะมีการทำงานอะไรบ้างอย่าง เช่น ติดต่อรับส่งข้อมูลผ่านเครือข่าย หรือคำนวณค่าต่างๆ แล้วก็ทำการส่งผลลัพธ์นั้นไปแสดงยัง activity ก็ได้
- 2.3.3 Broadcast receiver (Data Provider) คือ ตัวที่ใช้สำหรับคอยรับ และตอบสนองต่อเหตุการณ์ต่างๆ ที่เกิดขึ้น เช่น เมื่อแบตเตอรี่ต่ำ, ผู้ใช้ทำการเปลี่ยนภาษา, มีการโทรออก, มีข้อความเข้า และอื่นๆ ถึงแม้ broadcast receiver จะไม่มีส่วนของการแสดงผล แต่มันก็สามารถที่จะเรียก activity ขึ้นมาแสดงผลให้ผู้ใช้ได้ หรืออาจจะใช้สิ่งที่เรียกว่า Notification Manager ซึ่งจะเป็นตัวที่แจ้งเตือนในรูปแบบของ การสั่น, การแสดงไฟกระพริบที่หน้าจอ หรือการส่งเสียงออกมา โดยจะมี icon แสดงอยู่บน status bar เพื่อให้ผู้ใช้กดเข้าไปเปิดดูเหตุการณ์ที่เกิดขึ้น

- 2.3.4 Content provider (System Event Listener) คือกลุ่มของข้อมูลที่สร้างขึ้นจากแอปพลิเคชัน เพื่อให้แอปพลิเคชันอื่นๆ ได้นำไปใช้ โดยการจับเก็บข้อมูลของ content provider นั้นจะอยู่ในลักษณะของ ไฟล์, ฐานข้อมูล SQLite และอื่นๆ โดยการจับเก็บข้อมูลของ content provider นั้นจะอยู่ในลักษณะของ ไฟล์, ฐานข้อมูล SQLite และอื่นๆ

2.4 Activity Life Cycle

- 2.4.1 Activity มีหน้าที่ในการสร้าง และควบคุมการทำงานของหน้าจอ เพื่อโต้ตอบกับผู้ใช้งาน ซึ่งจะมีได้แค่ Activity เดียวเท่านั้นที่โต้ตอบกับผู้ใช้งานในขณะใดขณะหนึ่ง Activity แต่ละตัวจะมีวงจรชีวิตเป็นของตนเอง โดยแบ่งเป็นสถานะดังนี้
- Running เป็นสถานะที่ Activity กำลังได้ทำงานในขณะนั้น
 - Pause เป็นสถานะที่ Activity ปรากฏอยู่บนหน้าจอ แต่ไม่ได้ทำงานในขณะนั้น
 - Stop เป็นสถานะที่ Activity ไม่ปรากฏบนหน้าจอ
 - Kill เป็นสถานะที่ Activity ถูกทำลายไปแล้วเมื่อ Activity อยู่ในสถานะ Pause หรือ Stop Activity สามารถถูกทำลายได้โดยระบบ



รูปที่ 2.3 แสดงวงจรชีวิตการทำงานของแอนดรอยด์

`onCreate(Bundle)` จะถูกเรียกเมื่อ Activity มีการสร้างตัวเองครั้งแรก เป็น method สำคัญในสร้างหน้าจอกการทำงานต่างๆ หรือกำหนดสถานะต่างๆ ของ control ที่จะใช้งาน

`onRestart()` จะถูกเรียกหลังจากที่ Activity กลับมาจากสถานะ Stop เพื่อเริ่มทำงานใหม่อีกครั้ง

`onStart()` จะถูกเรียกเมื่อ Activity เริ่มแสดงให้ผู้ใช้งานเห็น

`onResume()` จะถูกเรียกเมื่อ Activity สามารถที่จะโต้ตอบกับผู้ใช้งานได้

`onPause()` จะถูกเรียกเมื่อมี Activity อื่นกำลังทำงาน เป็น method สำคัญในการเก็บข้อมูลที่สำคัญ หรือทำการปล่อยหน่วยความจำเพื่อให้ Activity อื่นได้ใช้งาน

`onStop()` จะถูกเรียกเมื่อ Activity ไม่ปรากฏบนหน้าจอ แล้วจะเข้าสู่สถานะ Stop

`onDestroy()` จะถูกเรียกเมื่อ Activity กำลังจะถูกทำลายและยังมี method ที่สำคัญอีก 2 ตัวคือ

`onSaveInstanceState(Bundle)` จะถูกเรียกก่อน `onPause()` เพื่อเก็บสถานะต่างๆ ของ Activity ไว้ใช้เมื่อ Activity นี้ได้กลับมาทำงานอีกครั้ง

`onRestoreInstanceState(Bundle)` – จะถูกเรียกก่อน `onResume()` เพื่อนำสถานะต่างๆ ของ Activity ที่เก็บไว้มาใช้งาน

2.5 Eclipse

Eclipse คือโปรแกรมที่ใช้สำหรับพัฒนาภาษา Java ซึ่งโปรแกรม Eclipse เป็น โปรแกรมหนึ่งที่ใช้ในการพัฒนา Application Sever ใ้ได้อย่างมีประสิทธิภาพ และเนื่องจาก Eclipse เป็นซอฟต์แวร์ Open Source ที่พัฒนาขึ้นเพื่อใช้ โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว

Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งทำให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆจากภายนอก ให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และมืองค์ประกอบที่เรียกว่า Plug-in Development Platform Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse Plug-ins ดังนั้นหากต้องการให้ Eclipse ทำงานใดเพิ่มเติม ก็เพียงแต่พัฒนา Plug-in สำหรับงานนั้นขึ้นมา และนำ Plug-in นั้นมาติดตั้งเพิ่มเติมให้กับ Eclipse ที่มีอยู่เท่านั้น Eclipse Plug-in ที่มี

มาพร้อมกับ Eclipse เมื่อเรา Download มาครั้งแรกก็คือองค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียนและ Debug โปรแกรมภาษา Java

2.6 งานวิจัยที่เกี่ยวข้อง

เนื่องจากปัจจุบันระบบปฏิบัติการ Android มีผู้นิยมใช้เป็นจำนวนมาก และสามารถพัฒนาโปรแกรมได้อย่างอิสระและมีประสิทธิภาพสูงมาก บางครั้งจึงลืมนึกถึงความสะดวกสบายในการใช้งานเป็นหลัก เช่น มีปุ่มฟังก์ชันน้อยมากเกินไป จึงอาจเกิดปัญหาติดขัดในการใช้งานได้ เช่น



รูป 2.4 โปรแกรม DogDressUp

โปรแกรม Dog DressUp คือโปรแกรมการแต่งตัวสุนัข

ข้อดี

- รูปดูเป็นการ์ตูน น่ารัก สีสดใส
- มีไอคอนให้เลือกใช้เยอะ

ข้อเสีย

- คู่มือได้เพียงด้านเดียวไม่สามารถดูมุมมองอื่นได้