



โครงการระบบสารสนเทศ 2

(PROJECT IN INFORMATION SYSTEM II)

การพัฒนาแอปพลิเคชันเพื่อบันทึกการนัดหมายกิจกรรมสำหรับนักศึกษา

จัดทำโดย

- | | | |
|-------------------|-----------------|-------------------------|
| 1. นายพีรพัฒน์ | หนูยัง | รหัสนักศึกษา 5510513011 |
| 2. นางสาวอัญญชิตา | ศิริทนต์ธรรมกุล | รหัสนักศึกษา 5510513013 |

อาจารย์ที่ปรึกษา คร.จันทวรรณ ปิยะวัฒน์

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา 477-403 โครงการระบบสารสนเทศ 2

(PROJECT IN INFORMATION SYSTEM II)

ภาคเรียนที่ 2 ปีการศึกษา 2558

ภาควิชาบริหารธุรกิจ สาขาระบบสารสนเทศ

คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

ชื่อโครงการ	การพัฒนาแอปพลิเคชันเพื่อบันทึกการนัดหมายกิจกรรมสำหรับนักศึกษา		
อาจารย์ที่ปรึกษา	ดร.จันทวรรณ ปิยะวัฒน์		
ผู้รับผิดชอบโครงการ	นายพีรพัฒน์ หนูยัง	รหัสนักศึกษา	5510513011
	นางสาวอัญญชิสิตา ศิริทนุธรรมกุล	รหัสนักศึกษา	5510513013
ระดับการศึกษา	ปริญญาตรีบริหารธุรกิจบัณฑิต		
ภาควิชา	บริหารธุรกิจ สาขาวิชา ระบบสารสนเทศ		
	คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่		
ภาคการศึกษาที่	2		
ปีการศึกษา	2558		

บทคัดย่อ

โครงการการพัฒนาแอปพลิเคชันเพื่อบันทึกการนัดหมายกิจกรรมสำหรับนักศึกษา มีจุดประสงค์ในการจัดทำขึ้นเพื่อนำเสนอการออกแบบและพัฒนาแอปพลิเคชันเพื่อบันทึกการนัดหมายกิจกรรมสำหรับนักศึกษา ซึ่งมีเนื้อหาเกี่ยวกับ แนวคิดที่มาที่ไปของการพัฒนาแอปพลิเคชัน โครงสร้างและความสามารถของระบบ เทคโนโลยีที่ใช้การพัฒนาแอปพลิเคชัน ได้แก่ Android Studio , Active Android, Genymotion, SQLite, SQLite manager, GitHub และ Photoshop รวมถึงแสดง Interface Design และคู่มือการใช้งานของแอปพลิเคชัน

ประโยชน์ที่ได้รับจากการดำเนินงานโครงการนี้ แอปพลิเคชันที่ได้จะเป็นเครื่องมือช่วยอำนวยความสะดวกในการบันทึกกิจกรรมสำหรับนักศึกษาและเป็นเครื่องมือช่วยในการบริหารจัดการเวลา ซึ่งจะทำงานบนระบบปฏิบัติการแอนดรอยด์

กิตติกรรมประกาศ

โครงการฉบับนี้ สามารถสำเร็จลุล่วงได้ เนื่องด้วยความกรุณาจาก ดร.จันทวรรณ ปิยะวัฒน์ และ ผู้ช่วยศาสตราจารย์ ดร.รัชชัช ปิยะวัฒน์ อาจารย์ที่ปรึกษาโครงการ ที่กรุณาเสียสละเวลาอันมีค่า ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการดำเนินโครงการ อีกทั้งยังช่วยแก้ไขปัญหาดัง ๆ ที่เกิดขึ้นระหว่างการดำเนินโครงการโดยตลอดจนสำเร็จไปได้ด้วยดี รวมทั้งจุดประกายแนวคิดใหม่ ๆ และสร้างสรรค์ เพื่อให้สามารถนำไปปรับใช้ในการดำเนินโครงการอีกด้วย ขอขอบพระคุณอาจารย์เป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณบิดา มารดา ที่ส่งเสริมและให้กำลังใจในการศึกษาเล่าเรียน และขอขอบคุณเพื่อนสมาชิกในกลุ่ม ที่ให้การสนับสนุน และความร่วมมือเป็นอย่างดีตลอดการดำเนินโครงการนี้จนประสบความสำเร็จด้วยดี

คณะผู้จัดทำ

สารบัญ

บทคัดย่อ	a
กิตติกรรมประกาศ	b
สารบัญ.....	c
สารบัญรูปภาพ.....	f
สารบัญตาราง.....	h
บทที่ 1	1
บทนำ	1
ที่มาและความสำคัญ	1
ขอบเขตโครงการ	5
วัตถุประสงค์	5
กลุ่มเป้าหมาย.....	6
ผลที่คาดว่าจะได้รับ	6
ขั้นตอนการดำเนินงาน.....	6
บทที่ 2	7
การวิเคราะห์และออกแบบระบบ	7
โครงสร้างฐานข้อมูล(Database Structure)	8
Entity Relationship Diagram (ER)	8
Data Dictionary (SQLITE).....	9
บทที่ 3	11
เทคโนโลยีและเครื่องมือที่ใช้	11

Active android	11
Android Studio	13
JAVA.....	14
Github	15
Adobe Photoshop.....	16
SQLite.....	17
SQLite Manager	18
Genymotion	18
Material Design	19
Icon	19
การเรียกใช้ Icon ในส่วนของ Setting menu	21
การแสดง Date Time Dialog	22
บทที่ 4	24
ผลการดำเนินโครงการ	24
Student Appointment Application.....	24
ความสามารถของแอปพลิเคชัน.....	25
สัญลักษณ์แอปพลิเคชัน(Application Logo).....	26
Student Appointment Application	27
บทที่ 5	31
สรุปผลการดำเนินงาน	31
ข้อดี	31
ข้อจำกัด	32

บรรณานุกรม.....	33
ภาคผนวก ก	34
ภาคผนวก ข	220

สารบัญรูปภาพ

ภาพที่ 1.1. แผนภาพแสดงสถิติผู้ใช้โทรศัพท์ ปี 2557 เทียบกับ จำนวนประชากร	3
ภาพที่ 1.2. แผนภาพแสดงผลจากการศึกษาความพึงพอใจของผู้ใช้แอปพลิเคชันบนสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์	4
ภาพที่ 2.1. แผนภาพแสดงโครงสร้างระบบของแอปพลิเคชัน	7
ภาพที่ 2.2. แผนภาพแสดงรูปแบบโครงสร้างฐานข้อมูล	8
ภาพที่ 3.1. แผนภาพแสดงลักษณะการทำงานของ Active android	11
ภาพที่ 3.2. ภาพแสดงสัญลักษณ์โปรแกรม Android Studio.....	13
ภาพที่ 3.3. ภาพแสดงสัญลักษณ์ Java.....	14
ภาพที่ 3.4. แผนภาพแสดงรูปแบบการใช้งาน GitHub.com	15
ภาพที่ 3.5. แผนภาพแสดงสัญลักษณ์โปรแกรม PhotShop	16
ภาพที่ 3.6. ภาพแสดงตัวอย่างภาพพื้นหลังที่ใช้โปรแกรม photoShop ในการสร้าง	16
ภาพที่ 3.7. ภาพแสดงตัวอย่างวัตถุ 3 มิติ ที่ใช้โปรแกรม photoShop ในการสร้าง	17
ภาพที่ 3.8. ภาพแสดงสัญลักษณ์ฐานข้อมูล SQLite.....	17
ภาพที่ 3.9. ภาพแสดงสัญลักษณ์ SQLite Manager	18
ภาพที่ 3.10. ภาพแสดงสัญลักษณ์โปรแกรม Geny Motion	18
ภาพที่ 3.11. ภาพแสดงแถบ ToolBar ของแอปพลิเคชัน	20
ภาพที่ 3.12. ภาพแสดงแถบ ToolBar ของแอปพลิเคชัน	20
ภาพที่ 3.13. ภาพแสดงแถบ Setting ของแอปพลิเคชัน	21
ภาพที่ 3.14. ภาพแสดง DatePickerDialog	22
ภาพที่ 3.15. ภาพแสดง TimePickerDialog	22
ภาพที่ 4.1 . ภาพแสดงแอปพลิเคชันบนหน้าจอโทรศัพท์มือถือภาพที่	27
ภาพที่ 4.2 . ภาพแสดงหน้าแรกของแอปพลิเคชัน	27
ภาพที่ 4.3 . ภาพแสดงหน้าฟอร์มบันทึกนัดหมาย	27

ภาพที่ 4.4 . ภาพแสดงรายละเอียดฉบับที่กหนดหมาย.....	27
ภาพที่ 4.5 . ภาพแสดงสำหรับการแก้ไขฉบับที่กหนดหมาย	28
ภาพที่ 4.6 . ภาพแสดงแถบการตั้งค่าของแอปพลิเคชัน.....	28
ภาพที่ 4.7 ภาพแสดงฟอร์มการเพิ่มกิจกรรม	28
ภาพที่ 4.8. ภาพแสดงรายชื่อกิจกรรม	28
ภาพที่ 4.9.ภาพแสดงชื่อกิจกรรม	29
ภาพที่ 4.10.ภาพแสดงรายชื่อผู้ติดต่อ	29
ภาพที่ 4.11 . ภาพแสดงรายละเอียดผู้ติดต่อ	29
ภาพที่ 4.12. ภาพแสดงรายชื่อสถานที่กหนดหมาย.....	29
ภาพที่ 4.13 . ภาพแสดงชื่อสถานที่	30
ภาพที่ 4.14 . ภาพแสดงรายงานสรุปสถานะกิจกรรมกหนดหมาย.....	30
ภาพที่ 4.15 . ภาพแสดงหน้าเกี่ยวกับแอปพลิเคชัน.....	30

สารบัญตาราง

ตารางที่ 2.1. แสดงรายละเอียดของ Table Activity	9
ตารางที่ 2.2. แสดงรายละเอียดของ Table Contact_person	9
ตารางที่ 2.3. แสดงรายละเอียดของ Table Location	9
ตารางที่ 2.4. แสดงรายละเอียดของ Table Status	9
ตารางที่ 2.5. แสดงรายละเอียดของ Table Task	10

บทที่ 1

บทนำ

ที่มาและความสำคัญ

กล่าวได้ว่าเวลาเป็นสิ่งที่มีความสำคัญ (Time is money) ผ่านไปแล้วผ่านไปเลย ไม่สามารถย้อนกลับมาได้ คุณค่าของเวลาขึ้นอยู่กับความรู้จักใช้ให้เกิดประโยชน์ ดังนั้นการจะทำให้เวลามีคุณค่าให้เกิดประโยชน์สูงสุดจะต้องอาศัยการบริหารเวลา คือ จะต้องมีการวางแผน มีการจัดการว่าในระยะเวลาหนึ่ง ๆ จะต้องทำอะไรบ้าง เพื่อให้ระยะเวลาในการทำกิจกรรมต่าง ๆ ในแต่ละวัน แต่ละช่วงเวลาเป็นไปอย่างเหมาะสมและมีประสิทธิภาพ มีคำถามเกิดขึ้นในหัวของใครหลาย ๆ คนว่าใน 1 วัน ของแต่ละคนมี 24 ชั่วโมง เท่ากันหรือไม่ ในความเป็นจริงแล้วทุกคนมี 24 ชั่วโมงเท่ากัน แต่ความรู้สึกที่รับรู้บอกว่าเวลาไม่เท่ากัน เพราะ หลาย ๆ ปัจจัย เข้ามาเกี่ยวข้องที่ทำให้การรับรู้มิติเวลาของแต่ละคนไม่เท่ากัน เช่น ปัจจัยด้านอายุ ประสบการณ์ สภาพแวดล้อม และการบริหารจัดการเวลา เป็นต้น มนุษย์ต้องดำเนินชีวิตท่ามกลางเวลาที่กำลังหมุนอย่างไม่หยุดนิ่ง ในแต่ละวันคนเรามีเรื่องราวในชีวิตให้จัดการมากมาย ไม่ว่าจะเป็น การงาน การเรียน การนัดหมายกับบุคคลสำคัญ เป็นต้น ใน 1 ปี มี 365 วัน ใน 1 วัน มี 24 ชั่วโมง ใน 24 ชั่วโมง ของแต่ละวันมีการกิจกรรมมากมายที่ต้องจัดการ มีการนัดหมายที่สามารถเกิดขึ้นได้ตลอดเวลา ทุกโอกาส และทุกสถานที่ หากมีรายการนัดหมายจำนวนมาก ทำให้ยากต่อการจดจำ มันอาจทำให้หลงลืมไปบ้างในบางภารกิจและการพลาดภารกิจไม่ใช่เรื่องตลกคุณอาจจะกลายเป็นคนขาดความน่าเชื่อถือหากลืมนัดสำคัญ

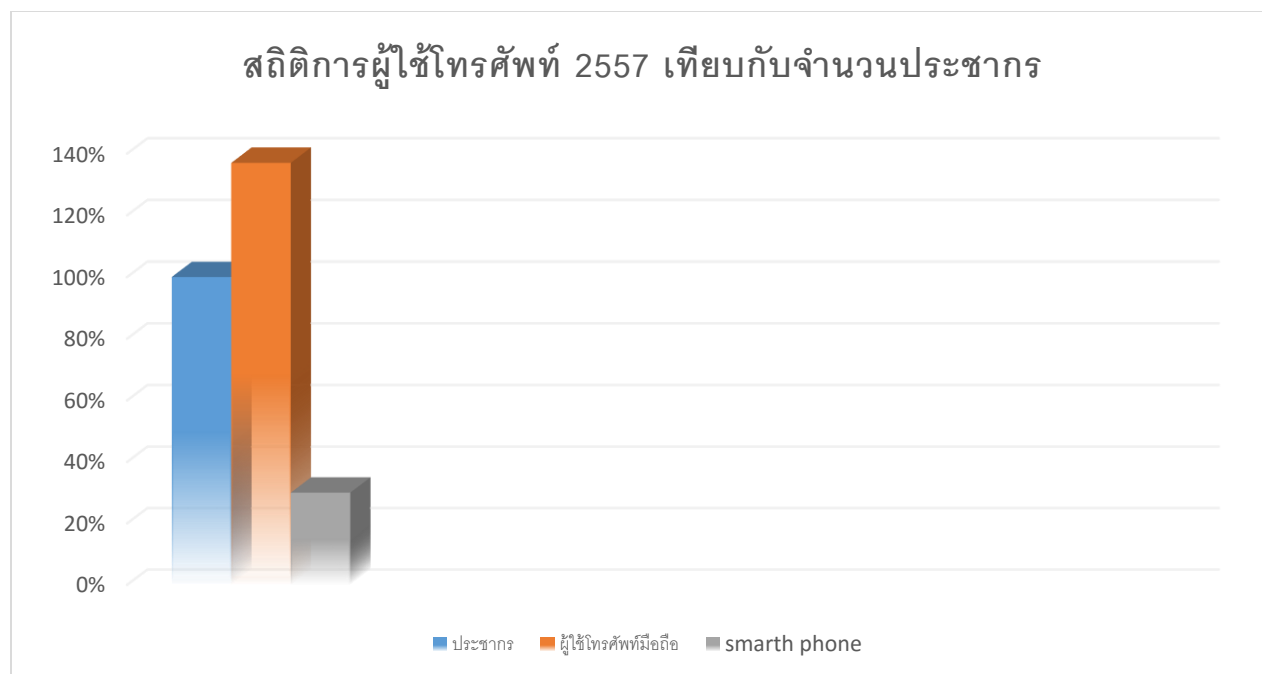
ดร.แพง ชินพงศ์(2554) ได้กล่าวไว้ว่า สาเหตุการหลงลืม มีได้หลายสาเหตุไม่ว่าจะเป็น เรื่องอายุ สุขภาพ ความเครียดและความวิตกกังวลต่าง ๆ ปัญหาที่มีมากมายในชีวิตหรือการมีงานหลายอย่างที่ต้องจัดการ ปัญหาการหลงลืมเกิดขึ้นได้กับทุกคน ส่วนจะมากหรือน้อยแค่ไหนก็เป็นอีกเรื่องหนึ่ง แต่ไม่ว่าอย่างไรก็เป็นเหตุให้รำคาญใจกับตนเองและบุคคลรอบข้าง เช่น การลืมสิ่งที่ต้องทำเป็นประจำ คือ มักจะลืมในสิ่งที่ไม่น่าจะลืมไม่ว่าจะเป็น ลืมว่าต้องจ่ายชำระค่าน้ำ ค่าไฟ ค่าโทรศัพท์ หรือ การลืมนัดสำคัญ การลืมเหตุการณ์ที่สำคัญต่าง ๆ ในชีวิตที่ควรจดจำ เช่น ลืมวันเกิดของตัวเอง หรือ คนในครอบครัว ลืมวันครบรอบวันแต่งงาน วันหยุดหรือวันสำคัญต่างๆ

ดร.แพง ชินพงศ์ ยังได้เสนอวิธีช่วยคนขี้ลืมให้หายลืมได้แบบง่าย ๆ 7 วิธีด้วยกัน ซึ่งหนึ่งใน 7 วิธีนั้นคือการจดบันทึก “คนขี้ลืมควรจะมีความสุขบันทึกทุกคิดว่าไว้เสมอโดยสามารถจดสิ่งต่าง ๆ ที่มีความสำคัญและมีความจำเป็นไว้ในสมุดนี้ เช่น งานที่ต้องทำ การนัดหมายในแต่ละวัน วันสำคัญต่าง ๆ ในชีวิต หากจำอะไรไม่ได้ก็จะได้เปิดดูข้อมูลต่าง ๆ ได้จากสมุดบันทึกนี้”

จะเห็นว่าความสำคัญของการจดบันทึกนอกจากจะช่วยคนขี้ลืมให้หายลืมแล้วยังสามารถช่วยในเรื่องของการบริหารจัดการเวลา การจดบันทึกหรือการเขียน (Write things down) ช่วยในการจดจำข้อมูลที่มากเกินไปโดยการเขียนรายการต่าง ๆ ที่ต้องทำ เพื่อจัดการชีวิตหรือกันลืม เช่นเดียวกับการบันทึกการนัดหมาย หรือกิจกรรมพิเศษ

ต่าง ๆ นวัตกรรม วันสำคัญ การบันทึกสั้น ๆ เพื่อเตือนความจำไว้ ก็จะทำให้เราไม่พลาดสิ่งเหล่านั้น การจดบันทึก รายการการนัดหมายหรือกิจกรรมที่ต้องทำนั้นก็สามารถทำการบันทึกได้หลายรูปแบบ เช่น การบันทึกลงใน กระดาษโน้ต สมุดโน้ตทั่วไป สมุดปฏิทิน หรือแม้กระทั่งบันทึกลงในปฏิทินเลยก็มีให้เห็น ซึ่งอุปกรณ์เหล่านี้ บางอย่างก็สามารถพกพาติดตัวไปได้ แต่บางอย่างก็ไม่สามารถพกพาติดตัว จะเป็นการดีกว่าหากเรามีสิ่งหนึ่งสิ่งใดที่สามารถทดแทนการใช้สมุดจดบันทึกหรืออุปกรณ์ต่าง ๆ สำหรับการนัดหมาย รายการกิจกรรม หรือวันสำคัญในชีวิตที่ควรจดจำ และตอนนี้คงปฏิเสธไม่ได้ว่า สิ่งที่หลายคนพกพาติดตัวอยู่เสมอจนอาจกลายเป็น ปัจจัยที่ 7 ในการดำเนินชีวิตก็ว่าได้ นั่นคือ โทรศัพท์มือถือ โดยเฉพาะโทรศัพท์มือถือแบบสมาร์ตโฟน ซึ่งกำลังเป็นที่ นิยมอย่างแพร่หลายในปัจจุบัน เรียกได้ว่าหายากเข้าออกเป็นสมาร์ตโฟนกันเลยทีเดียว ตั้งแต่ต้นจนกระทั่งเข้า นอน โทรศัพท์มือถือไม่ได้มีไว้แค่ให้เราคุยกันเท่านั้น นับวันยังมีวิวัฒนาการขึ้นเรื่อย ๆ เพื่อทดแทนสิ่งต่าง ๆ หรือ รวมสิ่งต่าง ๆ เข้ามาไว้ที่โทรศัพท์มือถือ เช่น เป็นเว็บเบราว์เซอร์ เพื่อใช้งานอินเทอร์เน็ต กล้องถ่ายภาพ อุปกรณ์ บันทึกข้อมูล เล่นเกมส์ ระบบจัดการข้อมูลส่วนตัว(สมุดบันทึก ปฏิทิน ตารางนัดหมาย) เป็นต้น แล่นั้นยังไม่พอ สมาร์ตโฟนกำลังถูกพัฒนาเพื่อทำหน้าที่เป็น กระเป๋าเงิน กุญแจ กระเป๋าบัตรต่าง ๆ เช่น บัตรเครดิต บัตรเดบิต บัตร เงินสด เพื่อจ่ายชำระค่าอาหาร ค่าเล่าเรียน จ่ายค่ารถแท็กซี่ รถไฟฟ้า หรือใช้เป็นสิทธิพิเศษต่าง ๆ ดังนั้น สมาร์ตโฟน กำลังกลายเป็นเสมือนอุปกรณ์ประจำตัว เพราะสารพัดประโยชน์ อีกทั้งรูปร่างของสมาร์ตโฟนกำลังถูกพัฒนาให้ หูหยาและเบาบางลงเรื่อย ๆ เพื่อความสะดวกในการพกพาและใช้งาน แต่สมาร์ตโฟนจะทำงานไม่ได้หากขาดตัว ขับเคลื่อนที่เรียกว่าระบบปฏิบัติการและมีกำลังเสริมคือแอปพลิเคชัน(Application) ในการสนองความต้องการผู้ใช้

Mobile Application ประกอบด้วยคำสองคำ คือ Mobile กับ Application ซึ่งมีความหมาย คือ Mobile เป็น อุปกรณ์สื่อสารที่ใช้ในการพกพา ซึ่งนอกจากจะใช้งานได้ตามพื้นฐานของโทรศัพท์แล้ว ยังทำงานได้เหมือนกับเครื่อง คอมพิวเตอร์ สำหรับ Application จะหมายถึง ซอฟต์แวร์ที่ใช้เพื่อช่วยการทำงานของผู้ใช้ (User) โดย Application จะต้องมีส่วนที่เรียกว่า ส่วนติดต่อกับผู้ใช้ (User Interface หรือ UI) เพื่อเป็นตัวกลางการใช้งานต่าง ๆ ดังนั้น Mobile Application หมายถึง แอปพลิเคชันที่ช่วยการทำงานของใช้บนอุปกรณ์พกพา เช่น โทรศัพท์มือถือ ซึ่งแอปพลิเคชันจะ ทำงานบนระบบปฏิบัติการ(OS) ที่แตกต่างกันไป ตัวอย่าง ระบบปฏิบัติการ(OS) บนอุปกรณ์พกพาที่เป็นที่นิยมใน ปัจจุบันคือ IOS พัฒนาโดยบริษัท Apple ,Android พัฒนาโดยบริษัท Google ,Windows phone พัฒนาโดยบริษัท Microsoft เป็นต้น แอปพลิเคชันที่ทำงานบนอุปกรณ์พกพา แบ่งเป็น 2 ประเภท คือ 1.แอปพลิเคชันระบบ เป็นส่วน ซอฟต์แวร์ระบบที่รองรับการใช้งานของแอปพลิเคชันหรือโปรแกรมต่าง ๆ 2.แอปพลิเคชันที่ตอบสนองความ ต้องการของกลุ่มผู้ใช้ เนื่องจากผู้มีความต้องการใช้แอปพลิเคชันแตกต่างกัน จึงมีแอปพลิเคชันต่าง ๆ เกิดขึ้น มากมายเพื่อสนองความต้องการของผู้ใช้งาน

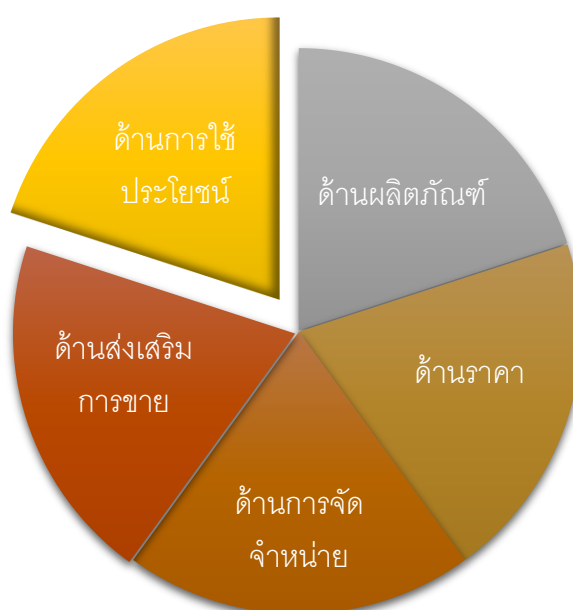


ภาพที่ 1.1. แผนภาพแสดงสถิติผู้ใช้โทรศัพท์ ปี 2557 เทียบกับ จำนวนประชากร

จากสถิติสมาคมโฆษณาดิจิทัล (ประเทศไทย) หรือ DAAT รายงานยอดผู้ใช้โทรศัพท์มือถืออยู่ที่ 94.3 ล้านเครื่อง เพิ่มขึ้น 5.4 ล้านเครื่อง เมื่อเทียบจากไตรมาสสุดท้ายของปี 2556 จะเห็นได้ว่าสัดส่วนของผู้ใช้งานโทรศัพท์มือถือของประเทศไทยเทียบกับจำนวนประชากรของประเทศแล้วนั้นพบว่ายอดผู้ใช้งานโทรศัพท์มือถือของไทยสูงกว่าจำนวนประชากรถึง 138.7% ซึ่งจากสถิติการสำรวจผู้ใช้โทรศัพท์มือถือในประเทศไทยกว่า 94 ล้านเครื่อง พบว่าโดยเฉลี่ยแล้วคนไทยใช้โทรศัพท์ประเภทสมาร์ทโฟน (smart phone) ราว 30% ของมือถือทั้งหมดและคาดว่าจะเพิ่มสูงขึ้นอย่างต่อเนื่อง แนวโน้มการใช้ Mobile Device อย่างสมาร์ทโฟนเพิ่มขึ้นอย่างก้าวกระโดดในช่วงไม่กี่ปีที่ผ่านมา เป็นผลมาจากการพัฒนา Mobile Application และเทคโนโลยีของตัวเครื่องโทรศัพท์จากค่ายผู้ผลิตโทรศัพท์ โดยเฉพาะการพัฒนาต่อยอดแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ต่างๆ ที่แข่งขันกัน เพื่อชิงความเป็นหนึ่งในตลาดด้าน Mobile Application ซึ่งการพัฒนาแอปพลิเคชันแบ่งเป็นการพัฒนาแอปพลิเคชันระบบ (Operation System) และแอปพลิเคชันซอฟต์แวร์ที่ตอบสนองการใช้งานบนอุปกรณ์ และด้วยแอปพลิเคชันที่เพิ่มขึ้นและมีประสิทธิภาพมากขึ้นทำให้ผู้ใช้อุปกรณ์เคลื่อนที่มีแนวโน้มใช้โปรแกรมต่าง ๆ เพื่อตอบสนองกิจกรรมในชีวิตประจำวันได้แก่ ทำธุรกรรมทางการเงิน เชื่อมต่อและสืบค้นข้อมูลบนเครือข่ายอินเทอร์เน็ต ชมภาพยนตร์ ฟังเพลง หรือแม้แต่การเล่นเกม ซึ่งมีทั้งออนไลน์และออฟไลน์ ด้วยอัตราการขยายตัวด้านการใช้งานอุปกรณ์เคลื่อนที่ทำให้บริษัทชั้นนำด้านโทรศัพท์มือถือหลายแห่งหันมาให้ความสำคัญกับการพัฒนาโปรแกรมบนโทรศัพท์มือถือโดยเชื่อว่าจะมีอัตราการดาวน์โหลดเพื่อใช้งานที่เติบโตอย่างเห็นได้ชัด จากงานวิจัยเรื่อง “ก้าวทันลูกค้ายุคใหม่ สร้างรายได้ให้ธุรกิจ” วิทยาลัยการจัดการ มหาวิทยาลัยมหิดล ได้นำเสนอข้อมูลจำนวนการใช้

แอปพลิเคชัน ใน 1 วัน สูงสุดอยู่ที่ 1-3 ชั่วโมงต่อวัน และ พบว่า แอปพลิเคชัน มีความหมายกับชีวิตดังนี้ ร้อยละ 83 คิดว่าทำให้ชีวิตง่ายขึ้น ร้อยละ 62 จะแนะนำ แอปพลิเคชันดีๆ ให้เพื่อน ร้อยละ 55 เปรียบแอปพลิเคชันเสมือนเพื่อนสนิท ร้อยละ 47 ถือว่า แอปพลิเคชัน เป็นส่วนหนึ่งของชีวิต

จากการศึกษาเรื่อง “พฤติกรรมและความพึงพอใจของผู้ใช้แอปพลิเคชัน(Application) บนสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์(Android)” พบว่าผู้ใช้แอปพลิเคชันบนสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์ กลุ่มตัวอย่างส่วนใหญ่เป็นเพศหญิง ร้อยละ 62.0 มีอายุระหว่าง 21-26ปี คิดเป็นร้อยละ 42.7 มีระดับการศึกษาระดับปริญญาตรี คิดเป็นร้อยละ 66.2 ประกอบอาชีพพนักงานเอกชน ลูกจ้าง คิดเป็นร้อยละ 40.2 และมีรายได้เฉลี่ยต่อเดือน 11,001-20,000 บาท คิดเป็นร้อยละ 41.4 พฤติกรรมของผู้ใช้แอปพลิเคชันบนสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์ของกลุ่มตัวอย่าง พบว่ากลุ่มตัวอย่างส่วนใหญ่เลือกใช้สมาร์ตโฟนยี่ห้อ Samsung คิดเป็นร้อยละ 42.5 และมีจำนวนเฉลี่ยในการดาวน์โหลดแอปพลิเคชันในเดือนที่ผ่านมา 1-2 ครั้ง คิดเป็นร้อยละ 43.4 ผลจากการศึกษาความพึงพอใจทั้ง 5 ด้าน ด้านผลิตภัณฑ์ ด้านราคา ด้านการจัดจำหน่าย ด้านการส่งเสริมการขาย ด้านการใช้ประโยชน์



ภาพที่ 1.2. แผนภาพแสดงผลจากการศึกษาความพึงพอใจของผู้ใช้แอปพลิเคชันบนสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์

พบว่ากลุ่มตัวอย่างมีความพึงพอใจในการใช้ประโยชน์โดยภาพรวมอยู่ในระดับมากคิดเป็นค่าเฉลี่ยเท่ากับ 4.08 โดยแอนดรอยด์มีพื้นฐานการทำงานมาจากระบบปฏิบัติการลินุกซ์(Linux) ถูกพัฒนาขึ้นโดยบริษัทที่ชื่อว่าแอนดรอยด์ แล้วถูกนำมาพัฒนาต่อยอดโดย Google พร้อมด้วยความร่วมมือจากบริษัทต่าง ๆ รวมไปถึงผู้ผลิตมือถือมากกว่า 30 ราย สิ่งที่ทำให้แอนดรอยด์ได้รับความสนใจจากบริษัทผลิตมือถือ รวมไปถึงลูกค้า นั่นก็คือเรื่องลิขสิทธิ์นำแอนดรอยด์ไปใช้งานจะขึ้นอยู่กับลักษณะซอฟต์แวร์เสรี หรือสามารถนำแอนดรอยด์ไปใช้งานได้ฟรี อีก

ทั้งยังเปิดโอกาสให้โปรแกรมเมอร์ได้ดาวน์โหลดชุด Software Develop Kit ไปพัฒนาโปรแกรมได้อย่างอิสระ ระบบปฏิบัติการแอนดรอยด์ ถูกพัฒนาให้เหมาะกับการทำงานบนสมาร์ทโฟน ใช้งานง่ายและสะดวกรองรับบริการหรือแอปพลิเคชันมากมายเน้นการควบคุมการทำงานผ่านจอแสดงผล ระบบสัมผัส มีการประมวลผลอย่างรวดเร็วตอบสนองได้ทันใจ รองรับการทำงานหลายโปรแกรมพร้อมกัน หรือMultitaskingระบบปฏิบัติการแอนดรอยด์ เวอร์ชันล่าสุดคือAndroid6.0Marshmallow

ในการพัฒนาแอปพลิเคชันนี้มุ่งเน้นไปยังกลุ่มเป้าหมายที่เป็นนักศึกษาที่มีพฤติกรรมการจดบันทึกเพื่อการนัดหมายหรือกิจกรรมต่าง ๆ ทางการศึกษา เนื่องจากกลุ่มเป้าหมายที่เป็นนักศึกษามีกิจกรรมมากมายที่เกิดขึ้นในชีวิตประจำวันทั้งเรื่องส่วนตัวและเรื่องเรียนซึ่งกิจกรรมต่าง ๆ เหล่านั้นล้วนมีความสำคัญกับตัวนักศึกษาเอง โดยที่นักศึกษาไม่อาจลืมกิจกรรมเหล่านั้นได้ เพราะมันอาจจะส่งผลในด้านลบกับนักศึกษาเองเช่น ลืมเข้าประชุม ลืมลงทะเบียนเรียน ลืมส่งงาน ลืมจ่ายค่าเทอม เป็นต้น การบันทึก หรือ การเขียนจึงเป็นสิ่งที่พื้นฐานที่นักศึกษาเลือกใช้เพื่อเตือนตัวเองป้องกันการลืมและสามารถบริหารจัดการเวลาสำหรับกิจกรรมนัดหมายต่าง ๆ ดังนั้นจึงเล็งเห็นถึงความสำคัญที่จะสร้างเครื่องมือที่ช่วยในการบันทึกการนัดหมายหรือกิจกรรมต่าง ๆ ทางการศึกษา เพื่อเป็นเครื่องมือช่วยกันลืมและเป็นเครื่องมือช่วยในการบริหารจัดการเวลา โดยแอปพลิเคชันนี้จะทำงานบนระบบปฏิบัติการแอนดรอยด์

ขอบเขตโครงการ

1. สามารถบันทึกกิจกรรมนัดหมายลงในวันที่ต้องการได้ โดยจะมีกิจกรรมนัดหมายให้เลือกโดยที่ผู้ใช้ไม่ต้องพิมพ์เอง แต่หากผู้ใช้มีกิจกรรมนัดหมายนอกเหนือจากค่าเริ่มต้นก็สามารถพิมพ์เองได้และสามารถโน้ตสิ่งเกี่ยวกับกิจกรรมนั้นได้
2. สามารถเพิ่มรายการกิจกรรมที่ทำบ่อยเพื่อให้เป็นค่าสำหรับเลือกได้ในการบันทึกครั้งต่อไป
3. สามารถเพิ่มสถานที่นัดหมายที่ใช้บ่อยเพื่อลดเวลาในการสร้างนัดหมายครั้งต่อไป
4. สามารถเรียกดู แก้ไขและลบกิจกรรมนัดหมายที่บันทึกลงไปได้
5. สามารถให้สถานะกิจกรรมและแสดงสรุปสถานะกิจกรรมนัดหมายได้

วัตถุประสงค์

- เพื่อเป็นเครื่องมือสำหรับบันทึกการนัดหมายหรือกิจกรรมต่าง ๆ ทางการศึกษา

กลุ่มเป้าหมาย

- กลุ่มผู้ใช้ที่เป็น นิสิต-นักศึกษาที่มีพฤติกรรมการจดบันทึกเพื่อการนัดหมายหรือกิจกรรมต่าง ๆ ทางการศึกษา

ผลที่คาดว่าจะได้รับ

- แอปพลิเคชันสามารถใช้งานได้จริงและเป็นประโยชน์แก่ผู้ใช้

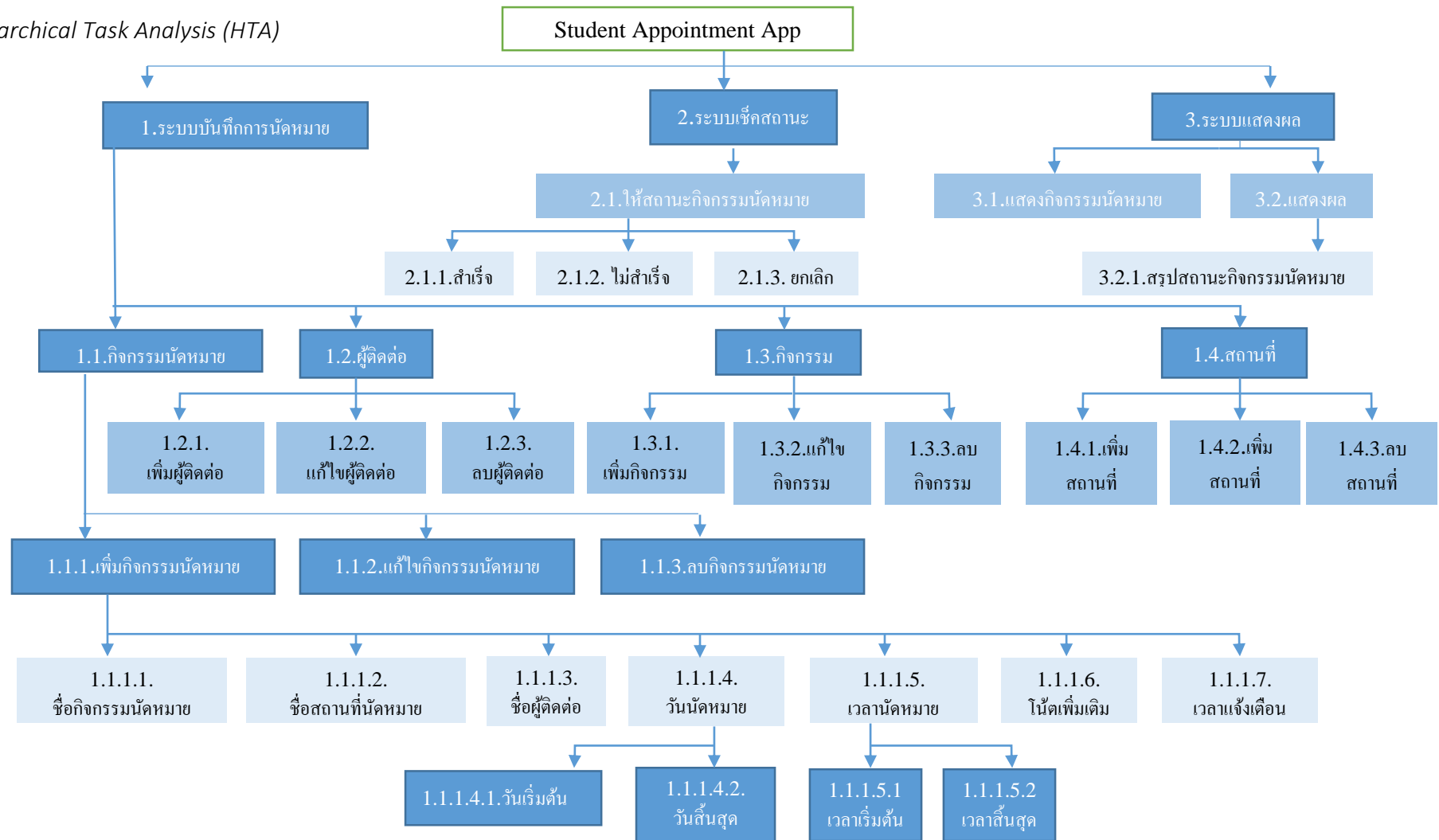
ขั้นตอนการดำเนินงาน

1. Planning
 - 1.1 วางแผนและวางขอบเขตโครงการ
 - 1.2 ศึกษาการใช้เครื่องมือที่เกี่ยวข้องกับการทำแอปพลิเคชัน
 - 1.3 ศึกษาภาษาที่ใช้ในการทำแอปพลิเคชัน
2. Analysis
 - 2.1 วิเคราะห์กระบวนการทำงานของแอปพลิเคชัน
 - 2.2 วิเคราะห์รูปแบบการจัดเก็บข้อมูล
 - 2.3 วิเคราะห์ผู้มีส่วนเกี่ยวข้องกับระบบ
3. Design
 - 3.1 ออกแบบ Interface ของแอปพลิเคชัน
 - 3.2 ออกแบบ Logo แอปพลิเคชัน
4. Coding
 - 4.1 ทำการสร้าง Interface ตามที่ออกแบบไว้
 - 4.2 สร้างแอปพลิเคชัน
5. Documentation
 - 6.1 จัดทำเอกสารประกอบ

บทที่ 2

การวิเคราะห์และออกแบบระบบ

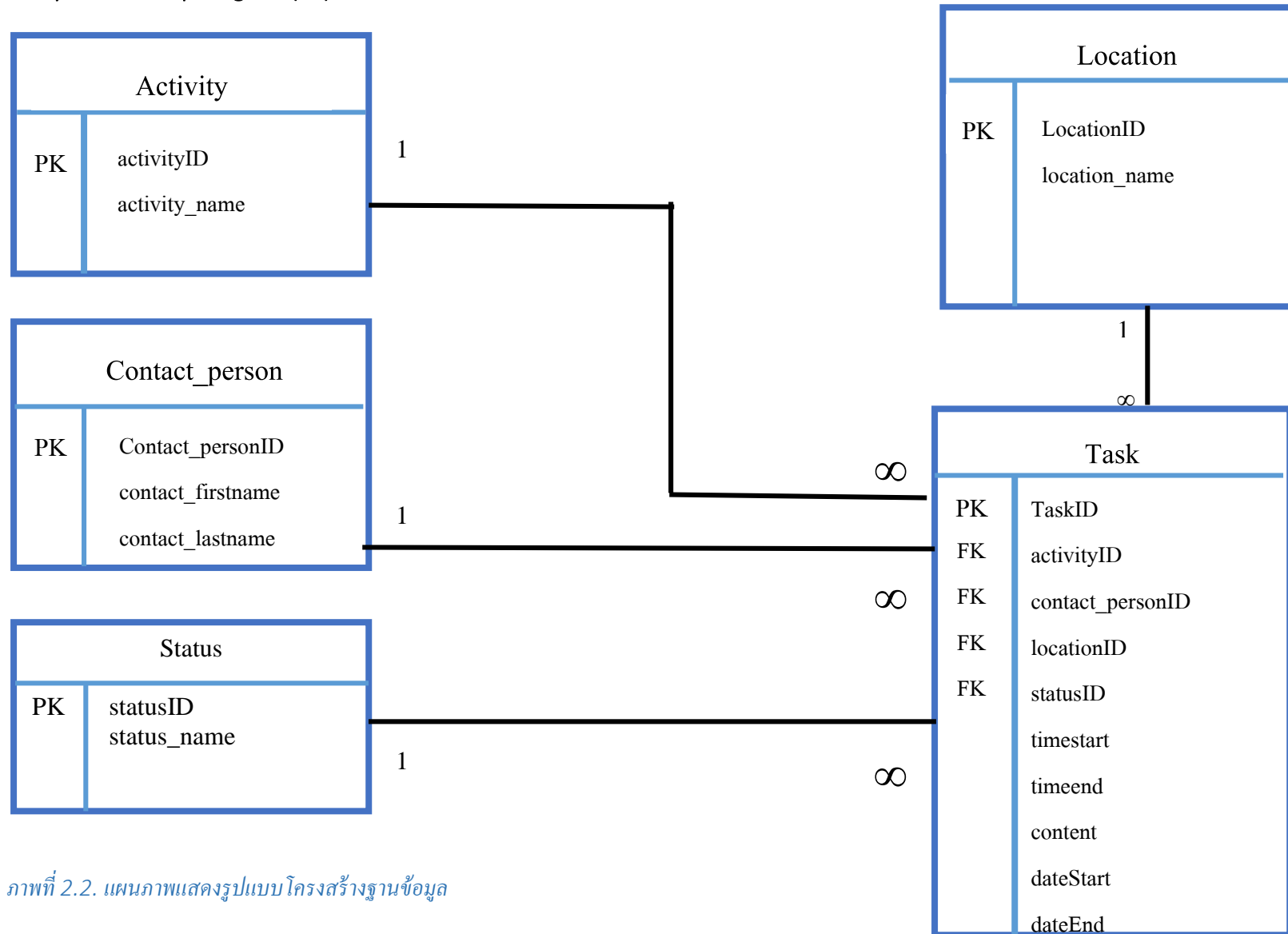
Hierarchical Task Analysis (HTA)



ภาพที่ 2.1. แผนภาพแสดงโครงสร้างระบบของแอปพลิเคชัน

โครงสร้างฐานข้อมูล(Database Structure)

Entity Relationship Diagram (ER)



ภาพที่ 2.2. แผนภาพแสดงรูปแบบโครงสร้างฐานข้อมูล

Data Dictionary (SQLITE)

Table name: Activity			
Field Name		Data type	Value
PK	activityID	Integer	1-999
Activity_name		Text	ชื่อกิจกรรมนัดหมาย(30)

ตารางที่ 2.1. แสดงรายละเอียดของ Table *Activity*

Table name: Contact_person			
Field Name		Data type	Value
PK	Contact_personID	Integer	1-999
contact_firstname		Text	ชื่อผู้ติดต่อ(50)
Contact_lastname		Text	สกุลผู้ติดต่อ(50)

ตารางที่ 2.2. แสดงรายละเอียดของ Table *Contact_person*

Table name: Location			
Field Name		Data type	Value
PK	locationID	Integer	1-999
location_name		Text	ชื่อสถานที่(40)

ตารางที่ 2.3. แสดงรายละเอียดของ Table *Location*

Table name: Status			
Field Name		Data type	Value
PK	statusID	Integer	1-999
status_name		Text	Complete / Incomplete / Cancel

ตารางที่ 2.4. แสดงรายละเอียดของ Table *Status*

Table name: Task			
Field Name		Data type	Value
PK	taskID	Integer	1-999
activityID		Integer	1-999
Contact_personID		Integer	1-999
locationID		Integer	1-999
dateEnd		Text	วันที่สิ้นสุดการนัดหมาย
dateStart		Text	วันที่เริ่มการนัดหมาย
content		Text	รายละเอียด(100)
StatusID		Integer	1-999
Timestart		Text	เวลาเริ่มต้น
Timeend		Text	เวลาสิ้นสุด

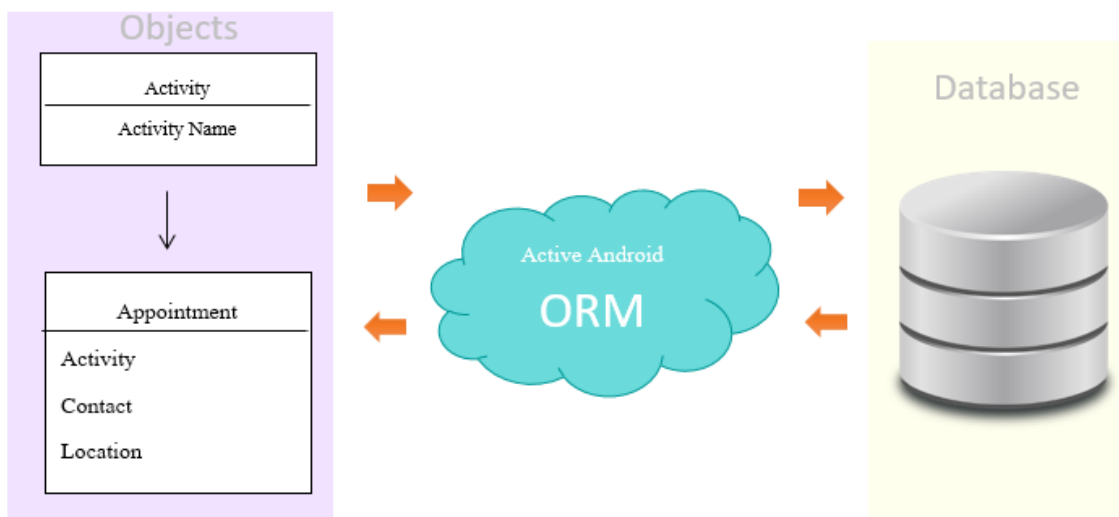
ตารางที่ 2.5. แสดงรายละเอียดของ Table *Task*

บทที่ 3

เทคโนโลยีและเครื่องมือที่ใช้

1. Active android

ลักษณะการทำงานของ Active android (ORM)



ภาพที่ 3.1. แผนภาพแสดงลักษณะการทำงานของ Active android

Active android

Active android คือ ORM (Object Relational Mapping) ที่เข้ามาช่วยให้การติดต่อกับฐานข้อมูลสามารถทำได้ง่าย โดยไม่จำเป็นต้องไปยุ่งเกี่ยวในส่วนของ SQL (SELECT, INSERT, UPDATE, DELETE) เพราะเหล่านี้ active android จะเป็นตัวแปลงภาษาที่เราเขียนไปเป็น SQL หรือ database ให้เราสามารถสร้าง Table โดยเขียนเพียงแค่ JAVA Class โดยใน JAVA Class นี้ก็จะกำหนด attribute ต่าง ๆ ที่ต้องการลงไป จากนั้น active android ก็จะไปแปลงเป็น Table ให้เอง และกำหนด primary key และ foreign key ให้อัตโนมัติ ซึ่งกล่าวได้ว่า active android เป็นเทคโนโลยีที่มีประโยชน์มาก ช่วยให้ผู้พัฒนาแอปพลิเคชันเขียนโค้ดได้ง่ายและเร็วขึ้นเพราะมีคำสั่งไม่มากเหมือนภาษา SQL

ตัวอย่างการนำ active android ไปใช้

เริ่มต้นด้วยการติดตั้ง active android โดยไป setup ที่ android studio ที่ app/build.gradle

```
ใช้คำสั่ง dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.2.1'
    compile 'com.michaelpardo:activeandroid:3.1.0-SNAPSHOT'
}
```

ตัวอย่างโค้ดการสร้าง Table

```
@Table(name = "Location")
public class LocationsTB extends Model {

    @Column(name="locationName")
    public String locationName;

    public static List<LocationsTB> getAll() {
        return new Select().from(LocationsTB.class).execute();
    }

    public void saveWithTimestamp() {
        Date now = new Date();
        updatedAt = now;
        if (createdAt == null)
            createdAt = now;
        save();
    }
}
```

ตัวอย่างโค้ดการสร้างความสัมพันธ์ระหว่างตาราง

```
@Column(name = "title", onUpdate =
Column.ForeignKeyAction.CASCADE, onDelete =
Column.ForeignKeyAction.CASCADE)
public Act title;
```

2. Android Studio



ภาพที่ 3.2. ภาพแสดงสัญลักษณ์โปรแกรม Android Studio

Android Studio เป็นเครื่องมือพัฒนา IDE (Integrated Development Environment) จาก Google ที่ถูกสร้างขึ้นมาเพื่อการพัฒนาแอนดรอยด์แอปพลิเคชัน บนพื้นฐานของแนวคิด IntelliJ คล้าย ๆ กับการทำงานของ Eclipse โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถ Preview ตัวแอปพลิเคชันมุมมองที่แตกต่างกันบน smart phone แต่ละรุ่นแต่ละขนาด สามารถแสดงผลได้ทันทีโดยไม่ต้องทำการรันแอปพลิเคชันบน Emulator

หากใช้ภาษาที่สามารถเข้าใจได้ง่าย กล่าวคือ android studio เป็นแพลตฟอร์มสำหรับสร้างแอปพลิเคชันโดยใช้ JAVA language เพื่อใช้งานบนระบบปฏิบัติการแอนดรอยด์ ซึ่งในการพัฒนาแอปพลิเคชันผู้ใช้ทั่วไปสามารถสร้างแอปพลิเคชันที่มีฟังก์ชันการทำงานไม่ซับซ้อนได้อย่างง่ายดายเพราะ android studio ได้มีเทมเพลตไว้ให้บางส่วนแล้ว และหากผู้พัฒนาต้องการจะเพิ่มการออกแบบ ปุ่มสั่งการต่าง ๆ ก็สามารถ drag and drop ได้ จากเครื่องมือที่ android studio มีมาให้ ปัจจุบัน android studio มีเวอร์ชันล่าสุดอยู่ที่เวอร์ชัน 2.1 ซึ่งถูกปล่อยออกมาเมื่อไม่นานนี้ ประมาณเดือนเมษายน 2559 แต่สำหรับการพัฒนาแอปพลิเคชัน Student appointment ยังคงใช้ android studio 1.5

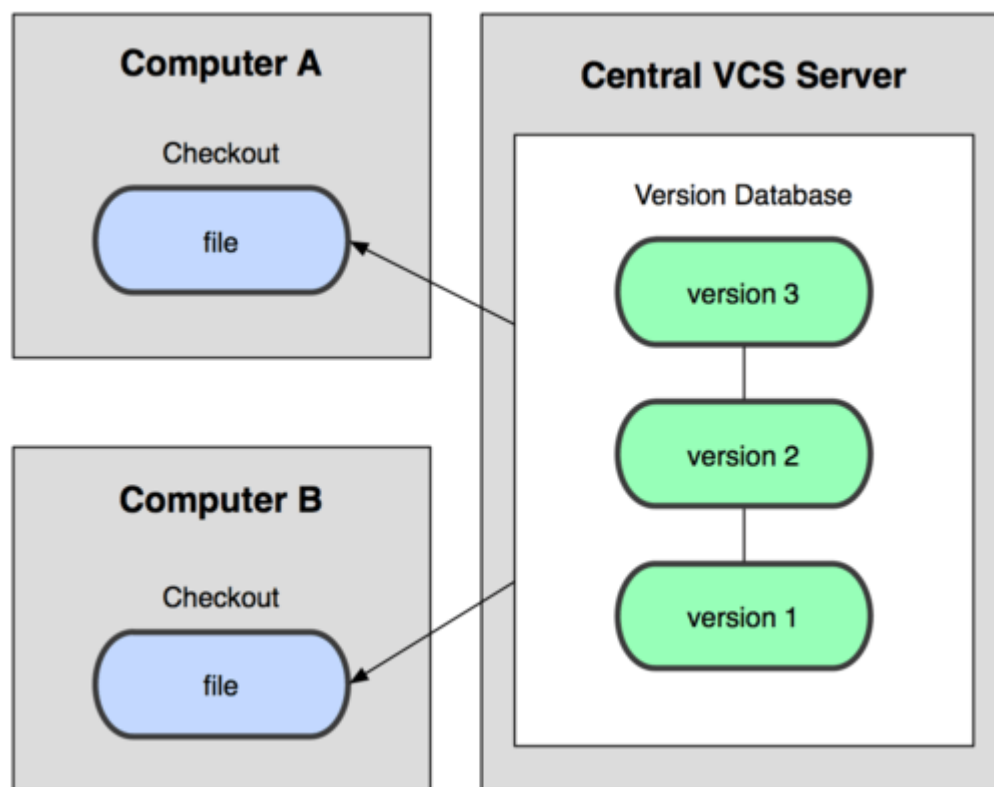
3. JAVA



ภาพที่ 3.3. ภาพแสดงสัญลักษณ์ Java

เป็นภาษาที่ใช้ในการพัฒนาแอปพลิเคชันครั้งนี้นับว่า android studio ภาษา Java เป็นของบริษัท Sun Microsystems ถูกพัฒนาและเริ่มใช้งานประมาณปี พ.ศ. 2539 (ค.ศ. 1996) โดยถูกพัฒนามาแทนที่ภาษา C++ ดังนั้นจึงมีรูปแบบ Syntax คล้าย ๆ กับภาษา C/C++ ในการเขียนภาษา Java จะเป็นภาษาโปรแกรมเชิงวัตถุที่ค่อนข้างจะสมบูรณ์แบบ (Object Oriented Programming) เพราะฉะนั้นการเขียนด้วย Syntax ของภาษา Java นั้น เราจะเขียนในมุมมองของ OOP(Object Oriented Programming) เป็นหลักสำหรับการพัฒนาภาษา Java สามารถพัฒนา Application ได้หลากหลายรูปแบบมาก เช่น Application ที่ทำงานบน Windows Mac , Linux หรือบน Web Application (JSP Java Servlet) และที่กำลังมาแรงสุดในตอนนี้คือ การพัฒนา Application บน Mobile ซึ่งในปัจจุบันสามารถพัฒนาได้บน Android และ BlackBerry ซึ่งปกติแล้วในการพัฒนา Application ด้วยภาษา Java ทั่ว ๆ ไปเราจะใช้รุ่น SE (Standard Edition) ก็จะมี JDK (Java Development Kit) ที่ประกอบไปด้วย compiler และ debugger ของภาษา Java สำหรับนักพัฒนา JRE (Java Runtime Environment) ซึ่งเป็นสิ่งที่รวม library ต่างๆสำหรับการรันโปรแกรมที่พัฒนาด้วย Java ซึ่งถ้าติดตั้ง JDK เพียงตัวเดียวก็จะมี JRE รวมอยู่ด้วย

4. Github



ภาพที่ 3.4. แผนภาพแสดงรูปแบบการใช้งาน GitHub.com

Github.com เป็นเว็บไซต์ที่ให้บริการเหมือน git server เป็น version control ที่ให้บริการฟรีแต่มีเงื่อนไขคือ โปรเจกต์ที่สร้างขึ้นจะต้องเป็นแบบ public เท่านั้น บุคคลทั่วไปสามารถเข้าไปดูโค้ดได้ แต่หากไม่ต้องการให้โปรเจกต์เป็นแบบ public ก็ต้องเสียค่าใช้จ่ายในการใช้บริการเพื่อสร้างโปรเจกต์เป็นแบบ private

สำหรับงานโปรแกรมมิ่งที่มีทีมพัฒนาหลาย ๆ คน github จะมีประโยชน์มากเพราะเป็น server เอาไว้เก็บ repository สามารถเรียกดูได้ว่าไฟล์นั้น ๆ ใครเป็นผู้แก้ไข หรือจะดูว่า ไฟล์นั้น ๆ ถูกเขียนโดยใคร และเป็นแหล่ง backup code เพื่อป้องกันไฟล์สูญหายในกรณีคอมพิวเตอร์มีปัญหา เพราะเราสามารถ clone โปรเจกต์ลงมาจาก github ได้

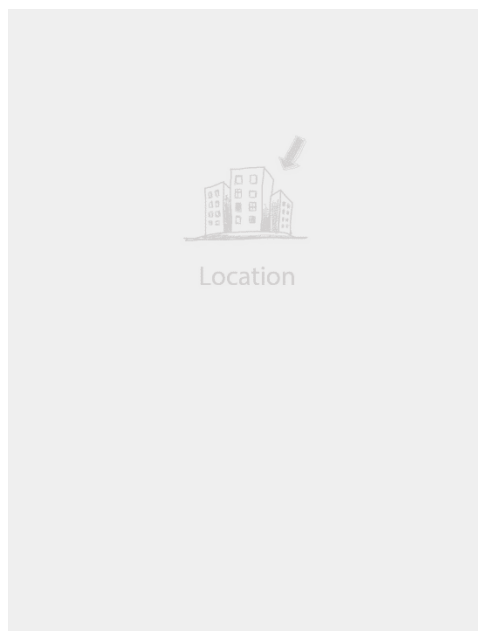
5. Adobe Photoshop



ภาพที่ 3.5. แผนภาพแสดงสัญลักษณ์โปรแกรม PhotShop

Adobe Photoshop เป็นโปรแกรมสร้างและแก้ไขรูปภาพอย่างมืออาชีพ เป็นโปรแกรมที่มีเครื่องมือมากมายเพื่อสนับสนุนการสร้างงานประเภทต่าง ๆ ไม่ว่าจะเป็น สื่อสิ่งพิมพ์ งานวิทัศน์ งานนำเสนอ งานมัลติมีเดีย เป็นต้น สำหรับการพัฒนาแอปพลิเคชัน Student appointment ได้ใช้ Adobe Photoshop เพื่อการออกแบบและสร้าง ภาพพื้นหลัง และ โลโก้ของแอปพลิเคชัน

ตัวอย่างภาพพื้นหลัง(Background) ที่ใช้ Adobe Photoshop ในการสร้าง



ภาพที่ 3.6. ภาพแสดงตัวอย่างภาพพื้นหลังที่ใช้โปรแกรม photoShop ในการสร้าง

ตัวอย่างโลโก้(Logo) ที่ใช้ Adobe Photoshop ในการสร้าง



ภาพที่ 3.7. ภาพแสดงตัวอย่างวัตถุ 3 มิติ ที่ใช้โปรแกรม photoShop ในการสร้าง

6. SQLite



ภาพที่ 3.8. ภาพแสดงสัญลักษณ์ฐานข้อมูล SQLite

SQLite คือ Library ตัวหนึ่งที่มีขนาดเล็กมากที่คอยจัดการ Database ให้กับแอปพลิเคชัน โดยใช้ Syntax SQL เหมือนกับ MySQL Data Type ที่ SQLite support คือ TEXT เหมือนกับ String ใน JAVA, INTEGER เก็บค่าตัวเลข เหมือนกับ long และ REAL เก็บค่าเหมือนกับ double สำหรับ SQLite ใน Android นั้นติดมากับเครื่อง Android ทุกเครื่อง ซึ่งเวลาเขียนแอปเพื่อเชื่อมต่อกับ SQLite ก็ไม่จำเป็นต้องโหลดหรือทำอะไรเพิ่มเพราะว่าทาง Android ได้เตรียม Class ต่างๆ ไว้แล้ว กล่าวคือการเก็บข้อมูลต่างๆ ไว้ภายในโทรศัพท์ ไม่ต้องเชื่อมต่ออินเทอร์เน็ตเพื่อรับหรือส่งข้อมูล

7. SQLite Manager



ภาพที่ 3.9. ภาพแสดงสัญลักษณ์ SQLite Manager

SQLite Manager เป็นเครื่องมือที่มีไว้เพื่ออ่านไฟล์ฐานข้อมูลหรือเข้าถึงไฟล์ .SQL ซึ่ง SQLite Manager เป็น Add-on ของ Browser ยอดนิยมอย่าง Firefox สามารถค้นหาและติดตั้งได้อย่างง่ายดาย

8. Genymotion



ภาพที่ 3.10. ภาพแสดงสัญลักษณ์โปรแกรม Geny Motion

Genymotion เป็น Android Emulator ที่มาพร้อมกับ Android System Image และ AVD สำเร็จรูปมากกว่า 10 ตัว ช่วยให้นักพัฒนาไม่ต้องกังวลเรื่องการตั้งค่า AVD อีกต่อไป ที่สำคัญคือ Genymotion ทำงานเร็วกว่า Emulator ที่มากับ Android SDK มาก Genymotion มีเครื่องมือช่วยนักพัฒนาหลายอย่างทั้งการเชื่อมต่อกับ Webcam, GPS, Screencast และอีกมากมาย Genymotion สามารถดาวน์โหลดมาใช้งานได้ฟรี หรือหากต้องการ feature เสริมอื่น ๆ สามารถเสียค่าใช้จ่ายเพื่อซื้อบริการได้

สำหรับลักษณะการทำงานของ Genymotion นั้น คือการเอา VirtualBox มารัน Android Image โดยมี Genymotion มาครอบการทำงานอีกที จึงทำให้ทำงานได้ลื่นไหล สามารถติดตั้งแอปพลิเคชันด้วยการลากไฟล์ APK มาใส่ในหน้าจอได้เลย ถือว่าเป็น Emulator ตัวหนึ่งที่รองรับทุก Platform ทำงานได้ไหลลื่น และทำขึ้นมาเพื่อใช้สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์โดยเฉพาะ จึงทำให้มีเครื่องมือต่าง ๆ ที่อำนวยความสะดวกแก่ผู้พัฒนา

Material Design

Icon

การเรียกใช้ Icon โดยการ Compile Library ลงใน ไฟล์ Build.gradle(Module:app)

```
dependencies {
```

```
    Compile ('com.mikepenz:materialdrawer:4.6.4@aar') {transitive = true}
    compile 'com.mikepenz:iconics-core:2.5.3@aar'
    compile 'com.mikepenz:material-design-iconic-typeface:2.2.0.1@aar'
}
```

ซึ่งในการใช้งานจะมีการเรียก class MaterialDesignIconic.java มาใช้ ซึ่ง class นี้จะมี icon ที่รวบรวมจาก Google material design เช่น

```
public static enum Icon implements IIcon {
    gmi_save('\uf297'),
    gmi_refresh('\uf1b9'),
    gmi_collection_plus('\uf14e'),
    gmi_collection_speaker('\uf14f'),
    gmi_collection_text('\uf150'),
    gmi_collection_video('\uf151'),
    gmi_compass('\uf152'),
    gmi_cutlery('\uf153'),
    gmi_delete('\uf154'),
    gmi_dialpad('\uf155'),
    gmi_dns('\uf156'),
    gmi_drink('\uf157'),
    gmi_edit('\uf158'),
    gmi_email_open('\uf159'),
    gmi_email('\uf15a'),
    gmi_eye_off('\uf15b'),
    gmi_eye('\uf15c'),
    gmi_eyedropper('\uf15d'),
    gmi_favorite_outline('\uf15e'),
    gmi_favorite('\uf15f'),
    gmi_fullscreen('\uf16d'),
    gmi_functions('\uf16e'),
    gmi_gas_station('\uf16f'),
    gmi_gesture('\uf170'),
    gmi_globe_alt('\uf171'),
    gmi_home('\uf175'),
```

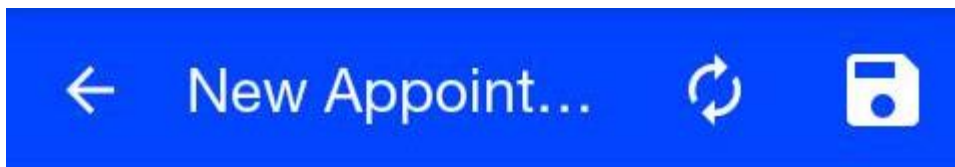
การนำไปใช้เริ่มจาก กำหนดขนาดและการเรียกใช้ของ Icon ไว้ที่ไฟล์ BaseActivity.java

```
protected Drawable buildDrawable(IIcon icon) {
    return new
    IconsDrawable(this).icon(icon).color(Color.WHITE).sizeDp(70).paddingDp(4);
}

protected void addMenuItem(Menu menu, int id, int labelId, Drawable icon) {
    MenuItem menuItem = menu.add(Menu.NONE, id, Menu.NONE, labelId);
    menuItem.setIcon(icon);
    menuItem.setShowAsActionFlags(MenuItem.SHOW_AS_ACTION_IF_ROOM);
}
```

เมื่อต้องการเรียกใช้ Icon ใน menu เพื่อทำงานในฟังก์ชัน refresh และ save ใน FormActivity.java

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, Refresh, R.string.refresh,
        buildDrawable(MaterialDesignIconic.Icon.gmi_refresh));
    addMenuItem(menu, MenuItem_SaveID, R.string.save,
        buildDrawable(MaterialDesignIconic.Icon.gmi_save));
    return true;
}
```



ภาพที่ 3.11. ภาพแสดงแถบ ToolBar ของแอปพลิเคชัน

เมื่อต้องการเรียกใช้ Icon ใน menu เพื่อทำงานในฟังก์ชัน Edit และ Delete ใน ShowActivity.java

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, MenuItem_EditID, R.string.edit,
        buildDrawable(MaterialDesignIconic.Icon.gmi_edit));
    addMenuItem(menu, MenuItem_DeleteID, R.string.delete,
        buildDrawable(MaterialDesignIconic.Icon.gmi_delete));
    return true;
}
```



ภาพที่ 3.12. ภาพแสดงแถบ ToolBar ของแอปพลิเคชัน

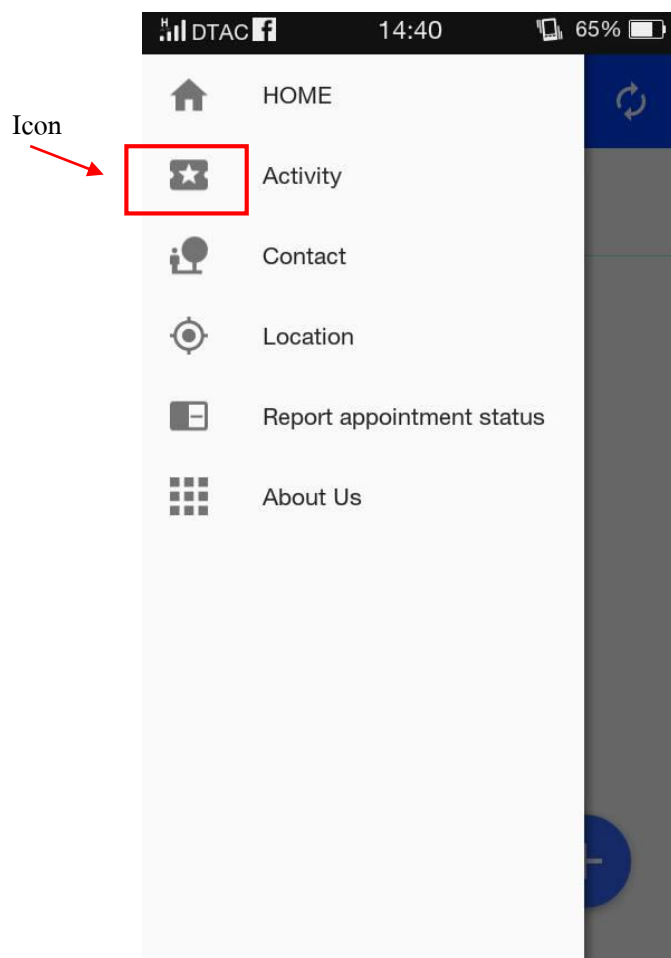
การเรียกใช้ Icon ในส่วนของ Setting menu

สามารถเรียกใช้ Icon ได้โดยการใช้ ตัวแปรที่นำมาจาก class `MaterialDesignIconic.java` มาใช้ได้เลย เช่น

ตัวอย่างการเรียกใช้ Icon ที่มีชื่อว่า `"gmi_local_activity"` ในไฟล์ `BaseActivity.java`

```
ActionBar supportActionBar = getSupportActionBar();
if (supportActionBar != null)
    if (upEnabled) {
        supportActionBar.setDisplayHomeAsUpEnabled(true);
    } else {
        final Intent intent = new Intent(this, ActList.class);
        new DrawerBuilder()
            .withActivity(this)
            .withToolbar(toolbar_main)
            .addDrawerItems(
                new PrimaryDrawerItem()
                    .withName(R.string.setting_Activity)
                    .withIcon(MaterialDesignIconic.Icon.gmi_local_activity)
                    .withSelectable(false)
                    .withIdentifier(1)

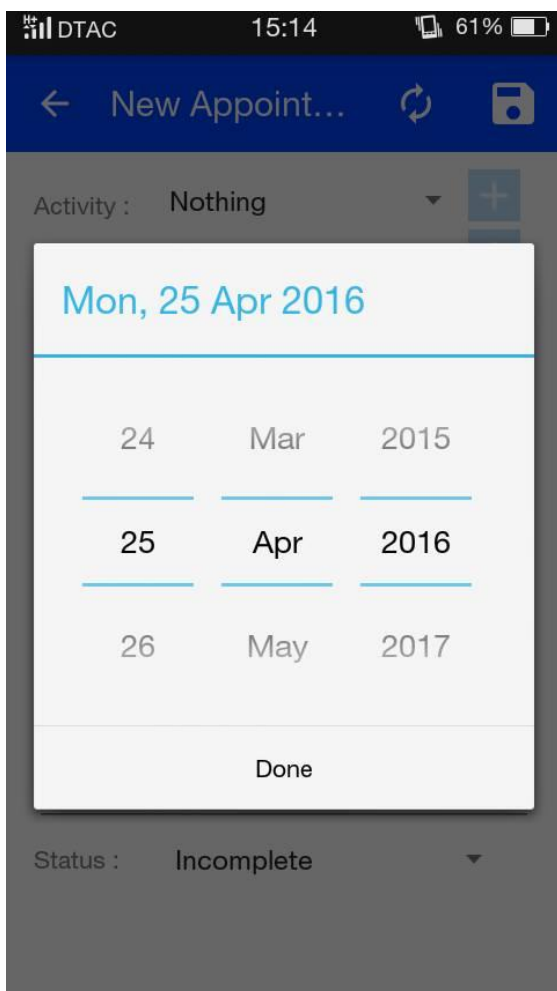
            ).withSelectedItem(-1)
```



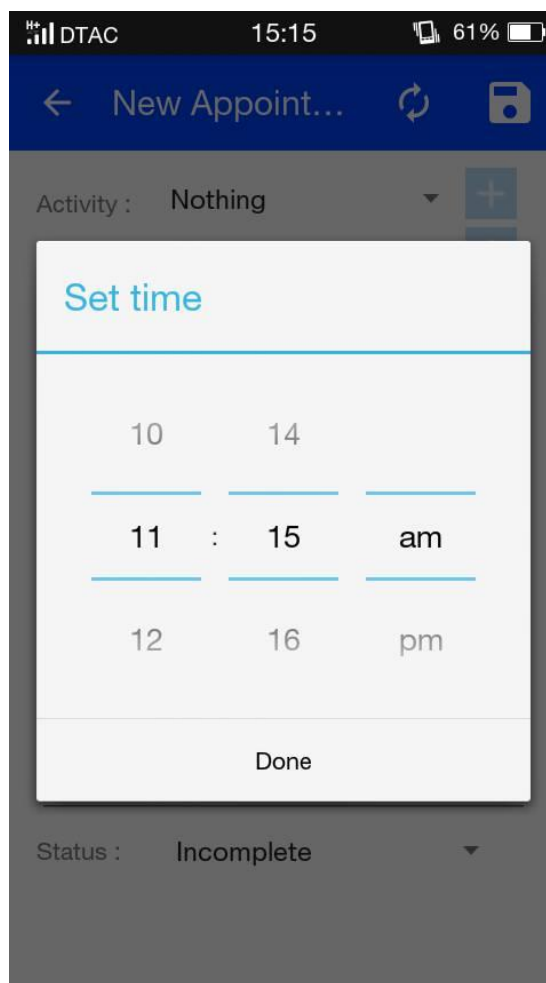
ภาพที่ 3.13. ภาพแสดงแถบ Setting ของแอปพลิเคชัน

การแสดง Date Time Dialog

ในการแสดง Date Time Dialog นั้น ได้มีการเรียกใช้คลาส DatePickerDialog และ TimePickerDialog ซึ่งจะแสดง Dialog ให้ผู้ใช้เลือกป้อนข้อมูล วันที่ และ เวลา ในลักษณะของ AlertDialog หรือกล่องข้อความ ดังภาพ



ภาพที่ 3.14. ภาพแสดง DatePickerDialog



ภาพที่ 3.15. ภาพแสดง TimePickerDialog

ตัวอย่างคำสั่งในการเรียกใช้ Date Time Dialog ในไฟล์ FormActivity.java

```
private DatePickerDialog fromDatePickerDialog;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_form);
    setDrawer(true);
    findViewsById();
    setDateTimeField();

    private void findViewsById() {
        fromDateTxt = (EditText) findViewById(R.id.dateStart);
        fromDateTxt.setInputType(InputType.TYPE_NULL);
        fromDateTxt.requestFocus();
    }

    private void setDateTimeField() {
        fromDateTxt.setOnClickListener(this);

        Calendar newCalendar = Calendar.getInstance();
        mHour = newCalendar.get(Calendar.HOUR_OF_DAY);
        mMinute = newCalendar.get(Calendar.MINUTE);
        fromDatePickerDialog = new DatePickerDialog(this, new
        DatePickerDialog.OnDateSetListener() {

            public void onDateSet(DatePicker view, int year, int monthOfYear, int
            dayOfMonth) {
                Calendar newDate = Calendar.getInstance();
                newDate.set(year, monthOfYear, dayOfMonth);
                fromDateTxt.setText(dateFormatter.format(newDate.getTime()));
            }

            }, newCalendar.get(Calendar.YEAR), newCalendar.get(Calendar.MONTH),
            newCalendar.get(Calendar.DAY_OF_MONTH));
```


บทที่ 4

ผลการดำเนินโครงการ

Student Appointment Application

Student appointment application คือ Native application ที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ ถูกพัฒนาขึ้นบนระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 6.0 สามารถทำงานแบบออฟไลน์ได้และใช้ฐานข้อมูลภายในโทรศัพท์ที่ติดตั้ง (SQLite) Student appointment application เป็นแอปพลิเคชันเพื่อบันทึกการนัดหมายกิจกรรมสำหรับนักศึกษา ซึ่งนักศึกษาเป็นกลุ่มที่มีกิจกรรมด้านการเรียนการศึกษามากมายและ ณ ปัจจุบันนี้ก็ยังไม่มีแอปพลิเคชันที่ถูกพัฒนามาเพื่อบันทึกกิจกรรมนัดหมายสำหรับนักศึกษาโดยตรง จะเห็นได้ว่ามีเพียงแอปพลิเคชันนัดหมายทั่วไปเท่านั้น สำหรับ Student appointment application ได้ถูกออกแบบการทำงานมาเพื่อให้นักศึกษาโดยเฉพาะ มีกิจกรรมนัดหมายมาให้เลือกใช้งาน อาทิเช่น “ส่งการบ้าน ประชุมเชียร์ ดิวหนังสือ นำเสนองาน สอบ” เป็นต้นนอกจากกิจกรรมข้างต้นแล้ว ผู้ใช้ยังสามารถเพิ่มกิจกรรมนัดหมายอื่น ๆ ลงไปในแอปพลิเคชันเพื่อสำหรับเลือกใช้งานในการจะบันทึกกิจกรรมนัดหมาย โดยผู้พัฒนา Student appointment application ได้สร้างความแตกต่างของแอปพลิเคชัน ให้โดดเด่น และคำนึงถึงความสะดวกในการใช้งานของผู้ใช้ โดยที่ผู้ใช้งานแอปพลิเคชันจะต้องใช้เวลาในการบันทึกกิจกรรมนัดหมายน้อยที่สุดและพิมพ์น้อยที่สุด ซึ่งลักษณะพิเศษนี้คือการใส่ค่าต่าง ๆ เพื่อบันทึกเป็นหนึ่งกิจกรรมนัดหมาย จะไม่ใช้การพิมพ์ตัวอักษรลงไป แต่ Student appointment application จะให้บริการในรูปแบบของ dropdown ผู้ใช้งานสามารถเลือกใช้งานได้อย่างรวดเร็ว ไม่ว่าจะเป็นชื่อของกิจกรรมนัดหมาย รายชื่อผู้ติดต่อ สถานที่นัดหมาย ซึ่งเป็นประโยชน์สำหรับการบันทึกกิจกรรมนัดหมายแต่ละครั้งจะใช้เวลาเพียงสั้น ๆ เท่านั้น ไม่เพียงเท่านี้ Student appointment application ยังสามารถให้ผู้ใช้งานสามารถให้สถานะของกิจกรรมนัดหมายต่าง ๆ ได้ หากกิจกรรมนัดหมายได้ยกเลิกไปก็ให้สถานะเป็น Cancel หรือถ้าไม่ได้ทำกิจกรรมนั้น ๆ ตามที่บันทึกไว้ก็ให้สถานะเป็น Incomplete หรือ Complete ในกรณีที่ทำกิจกรรมนั้น ๆ เรียบร้อยแล้ว ซึ่งจากสถานะของกิจกรรมต่าง ๆ เหล่านี้ที่ได้เลือกไว้ ก็สามารถมาดูเป็นสรุปรายงานได้ว่าจำนวนกิจกรรมที่ได้ยกเลิกไป หรือทำสำเร็จ และไม่สำเร็จ มีทั้งหมดกี่กิจกรรม

ความสามารถของแอปพลิเคชัน

- บันทึกนัดหมาย
- แก้ไขนัดหมาย
- ลบนัดหมาย
- ให้สถานะนัดหมาย
- แก้ไขสถานะนัดหมาย
- เพิ่มชื่อกิจกรรมนัดหมายที่ไว้สำหรับเลือกใช้งาน
- แก้ไขชื่อกิจกรรมนัดหมายที่ไว้สำหรับเลือกใช้งาน
- ลบกิจกรรมนัดหมายที่ไว้สำหรับเลือกใช้งาน
- เพิ่มชื่อผู้ติดต่อที่ไว้สำหรับเลือกใช้งาน
- แก้ไขชื่อผู้ติดต่อที่ไว้สำหรับเลือกใช้งาน
- ลบผู้ติดต่อที่ไว้สำหรับเลือกใช้งาน
- เพิ่มสถานที่นัดหมายที่ไว้สำหรับเลือกใช้งาน
- แก้ไขสถานที่นัดหมายที่ไว้สำหรับเลือกใช้งาน
- ลบสถานที่นัดหมายที่ไว้สำหรับเลือกใช้งาน
- ดูรายงานสรุปจำนวนนัดหมายตามสถานะของกิจกรรม

สัญลักษณ์แอปพลิเคชัน(Application Logo)



อักษรย่อ AP

อักษรย่อ AP ที่อยู่บริเวณตรงกลางของโลโก้ มาจากลักษณะแอปพลิเคชันที่เป็น appointment

เส้นที่เชื่อมต่อกัน

เส้นที่เชื่อมต่อกันระหว่างตัว A และตัว P สื่อถึงความสัมพันธ์และความทรงจำ

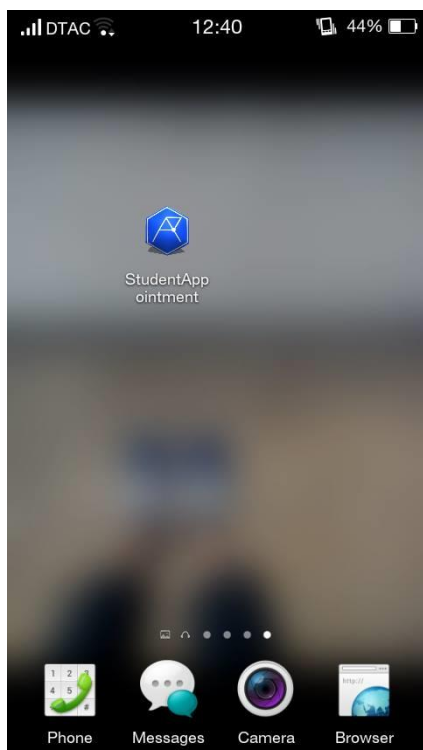
จุดวงกลม

จุดวงกลม คือ สัญลักษณ์จุดนัดหมาย สิ่งที่ต้องทำ

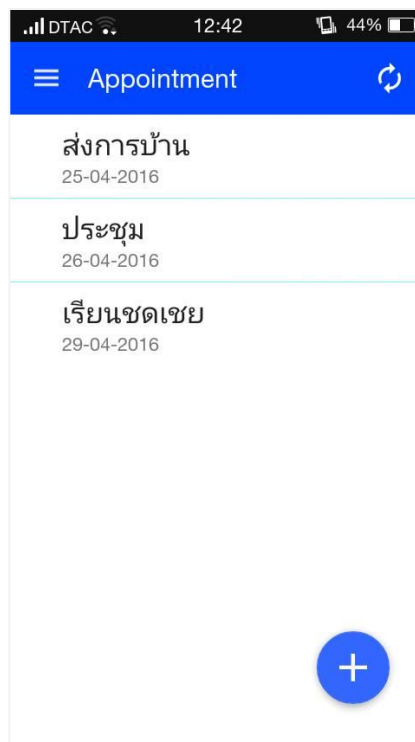
กรอบหกเหลี่ยม

กรอบหกเหลี่ยม สื่อถึงความคิดที่จัดการอย่างเป็นระเบียบและสามารถยืดหยุ่นได้

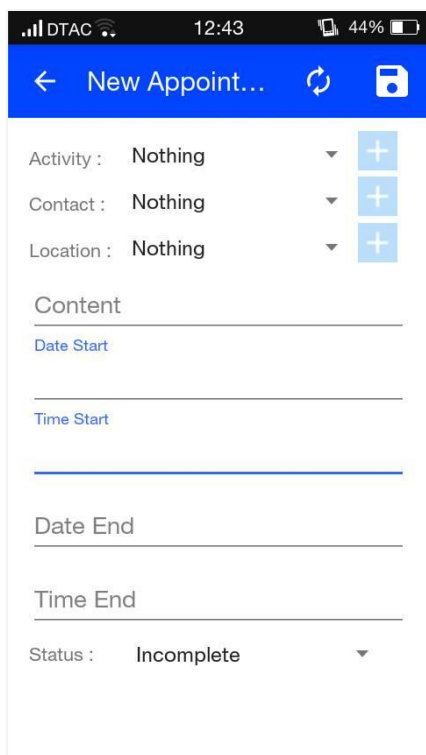
Student Appointment Application



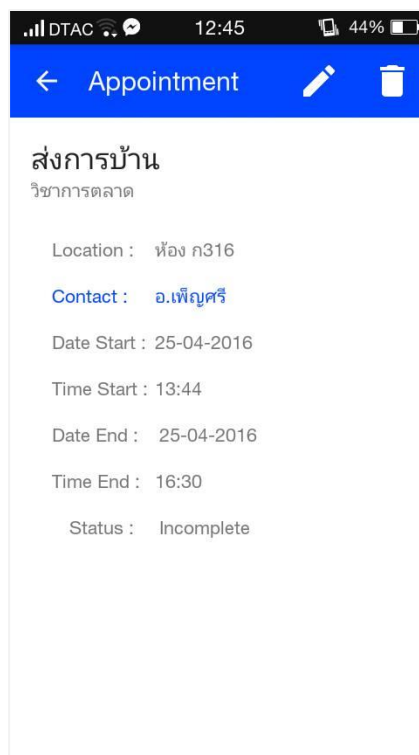
ภาพที่ 4.1 . ภาพแสดงแอปพลิเคชันบนหน้าจอโทรศัพท์มือถือ



ภาพที่ 4.2 . ภาพแสดงหน้าจอแรกของแอปพลิเคชัน



ภาพที่ 4.3 . ภาพแสดงหน้าจอฟอร์มบันทึกนัดหมาย



ภาพที่ 4.4 . ภาพแสดงรายละเอียดบันทึกนัดหมาย

DTAC 12:45 44%

← Edit Task ↻

Activity : ส่งการบ้าน +

Contact : อ.เพ็ญศรี +

Location : ห้อง ก316 +

วิชาการตลาด

Date Start

25-04-2016

Time Start

13:44

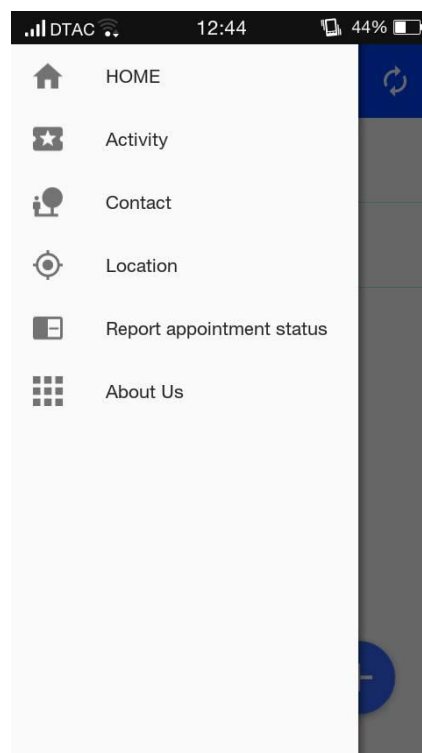
Date End

25-04-2016

Time End

16:30

Status : Incomplete ▾



ภาพที่ 4.5 . ภาพแสดงสำหรับการแก้ไขบันทึกนัดหมาย ภาพที่ 4.6 . ภาพแสดงแถบการตั้งค่าของแอปพลิเคชัน

DTAC 12:48 43%

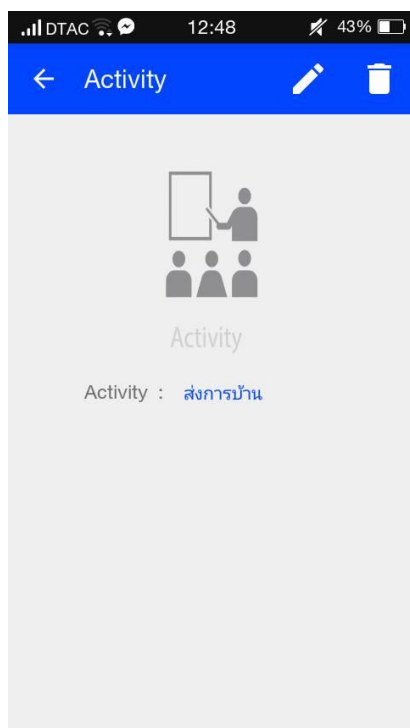
← New activity

ชื่อกิจกรรม



ภาพที่ 4.7 ภาพแสดงฟอร์มการเพิ่มกิจกรรม

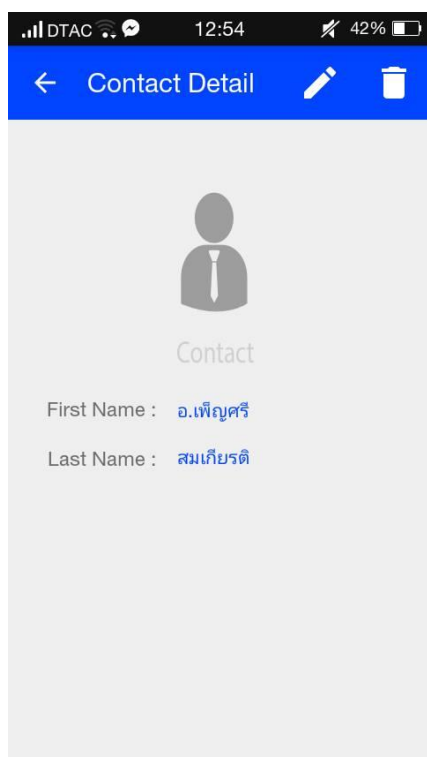
ภาพที่ 4.8.ภาพแสดงรายชื่อกิจกรรม



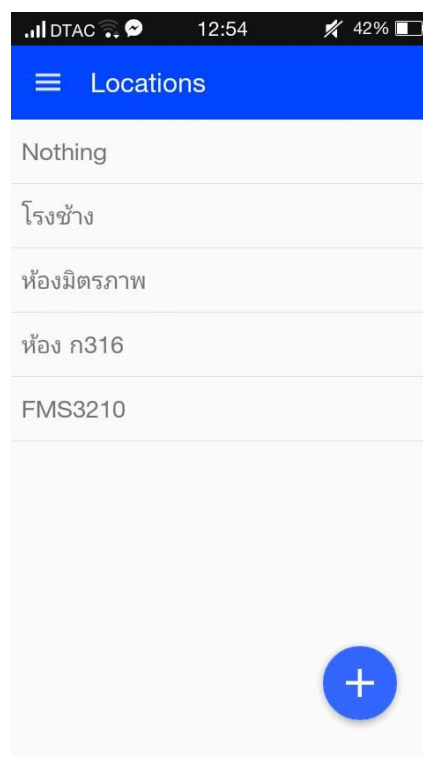
ภาพที่ 4.9.ภาพแสดงชื่อกิจกรรม



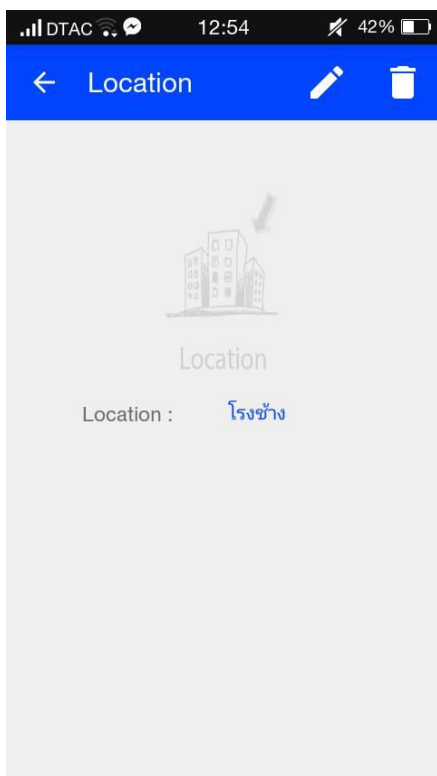
ภาพที่ 4.10.ภาพแสดงรายชื่อผู้ติดต่อ



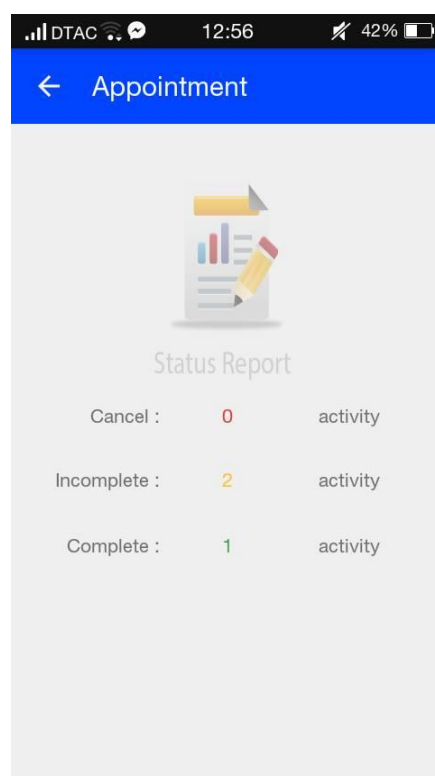
ภาพที่ 4.11 . ภาพแสดงรายละเอียดผู้ติดต่อ



ภาพที่ 4.12. ภาพแสดงรายชื่อสถานที่นัดหมาย



ภาพที่ 4.13 . ภาพแสดงชื่อสถานที่



ภาพที่ 4.14 . ภาพแสดงรายงานสรุปสถานะกิจกรรมนัดหมาย



ภาพที่ 4.15 . ภาพแสดงหน้าเกี่ยวกับแอปพลิเคชัน

บทที่ 5

สรุปผลการดำเนินงาน

จากการได้ดำเนินโครงการจัดทำแอนดรอยด์แอปพลิเคชันภายใต้หัวข้อเรื่องการพัฒนาแอปพลิเคชันเพื่อบันทึกการนัดหมายกิจกรรมสำหรับนักศึกษา ตั้งแต่เริ่มแรกด้วยการศึกษาความเป็นไปได้ในการจัดทำโปรแกรมโดยการ ศึกษาเกี่ยวกับแอนดรอยด์ทั้งหมดไม่ว่าจะเป็นระบบปฏิบัติการแอนดรอยด์ การทำงานของระบบปฏิบัติการแอนดรอยด์ สิ่งที่สำคัญในการพัฒนาแอปพลิเคชันทั้งหมด ศึกษาความต้องการของผู้ใช้งาน และการออกแบบแอปพลิเคชัน เพื่อให้มีความสะดวกต่อการใช้งาน และในขั้นตอนของการเขียนแอปพลิเคชันนั้น ได้ใช้โปรแกรม Android Studio ในการเขียน ซึ่งภาษาหลักในการเขียนนั้นจะเป็นภาษา JAVA และ ภาษา XML หลังจากที่ได้ศึกษาและลงมือปฏิบัติแล้วนั้น การสรุปผลการดำเนินงาน ซึ่งจะกล่าวถึงผลของโครงการ ข้อดี ข้อจำกัด และข้อเสนอแนะที่คาดว่าจะจำเป็นต่อการพัฒนาแอปพลิเคชัน Student Appointment เพื่อให้มีประสิทธิภาพมากยิ่งขึ้น โดยแอปพลิเคชัน Student Appointment นี้ ได้พัฒนาขึ้นเพื่ออำนวยความสะดวกในการบันทึกกิจกรรมนัดหมายของนักเรียน นักศึกษา

5.1. สรุปผลโครงการ

จากผลการดำเนินงานแอนดรอยด์แอปพลิเคชัน Student Appointment มีการดำเนินงานได้ตรงตามเป้าหมายที่ได้คาดหวังไว้คือ เพื่อศึกษาและพัฒนาแอนดรอยด์แอปพลิเคชันให้สามารถนำแอปพลิเคชันที่ได้ทำการพัฒนานี้ไปใช้ประโยชน์ได้จริง โดยในการศึกษาและพัฒนาแอปพลิเคชันนั้น ได้มีการใช้เครื่องมือช่วย คือ Android Studio การที่ใช้เครื่องมือช่วยในที่นี้ เป็นการช่วยให้สามารถทำงานได้สะดวกและมีความยืดหยุ่น โดยภาษาหลักในการเขียนแอปพลิเคชันนั้น ใช้ภาษา Java เป็นหลัก และได้มีการใช้ภาษา Xml ในเรื่องที่เกี่ยวข้องกับการแสดงผล Text ต่าง ๆ รวมไปถึง Image Button และ Background จากวัตถุประสงค์ของการทำโครงการ จึงได้สำเร็จออกมาเป็นแอนดรอยด์แอปพลิเคชัน Student Appointment เพื่อเป็นประโยชน์สำหรับนิสิต นักศึกษา ที่ต้องการเพิ่มความสะดวกในการจดบันทึกกิจกรรมนัดหมายต่าง ๆ

5.2. ข้อดี

1. สะดวกในการพกพา เนื่องจากเป็นแอปพลิเคชันบนมือถือ
2. ช่วยลดเวลาในการบันทึกกิจกรรมนัดหมาย
3. อำนวยความสะดวกในการจดบันทึก โดยสามารถใช้แทนสมุด หรือ กระดาษ ซึ่งช่วยลดปัญหาการเกิด การสูญหายของบันทึก

5.3. ข้อจำกัด

มีข้อจำกัดในการแสดงผล เนื่องจากความแตกต่างของรุ่นและขนาดของจอแสดงผล การแสดงผลจึงมีการผิดเพี้ยนไปบ้าง

5.4. ข้อเสนอแนะ

เพื่อประสิทธิภาพในการใช้งานสูงสุด ควรติดตั้งแอปพลิเคชัน บนระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 6.0 หรือ ระบบปฏิบัติการแอนดรอยด์เวอร์ชันที่ไม่ต่ำกว่า 4.4.2

บรรณานุกรม

การใช้งาน version control.

<https://git-scm.com/book/th/v1/เริ่มต้นใช้งาน-เกี่ยวกับ-version-control> สืบค้น 27 เมษายน 2559

แพ่ง ชินพงศ์. ใครจี๊ด...มาทางนี้มีทางแก้. ASTVผู้จัดการออนไลน์.

<http://www.manager.co.th/Family/ViewNews.aspx?NewsID=9540000013304> 2554

สมาคมโฆษณาติจัต (ประเทศไทย). DAAT เผยข้อมูลตัวเลขผู้ใช้โทรศัพท์มือถือทั่วไทยของไตรมาส 2

ประจำปี 2557. <http://www.daat.in.th/index.php/daat-mobile/#sthash.OJmIF9kn.dpuf>

สืบค้น 8 กันยายน 2558

สุชาดา พลาชัยภิรมย์ศิลป์. 2554. แนวโน้มการใช้โมบายแอปพลิเคชัน . วารสารนักบริหาร ปีที่ 31, ฉบับที่ 4

(ต.ค.-ธ.ค.2554) : 110-115.

วิชา ธีรวิชัยไชยกุล. 2556. พฤติกรรมและความพึงพอใจของผู้ใช้แอปพลิเคชัน (Application)

บนสมาร์ตโฟนระบบปฏิบัติการแอนดรอยด์ (Android). JC Journal. ปีที่ 5, ฉบับที่ 3 (ต.ค./ธ.ค.2556)

: 210-223.

วิทยาลัยการจัดการ มหาวิทยาลัยมหิดล .ก้าวทันลูกค้ายุคใหม่ สร้างรายได้ให้ธุรกิจ,งานวิจัย. 2555

http://inside.cm.mahidol.ac.th/mkt/attachments/531_CMMU%20MK13C%20Mobile%20App.pdf

ศิริกาญจน์ บุญลือ. 2554. ทำความรู้จักกับ ระบบปฏิบัติการ Android .วารสารกรมประชาสัมพันธ์.

ปีที่ 16,ฉบับที่ 171 (ก.พ.2554): 49-51.

ThaiCreate.Com Team. Java คืออะไร.

<http://www.thaicreate.com/java/java-what-is-java.html> 2556

ภาคผนวก ก

ซอร์สโค้ดแอปพลิเคชัน

ภาคผนวก ก

ซอร์สโค้ดแอปพลิเคชัน

ในการเขียนแอนดรอยด์แอปพลิเคชันโดยใช้ Android Studio นั้น จะมีส่วนที่สำคัญและเกี่ยวข้องกันอยู่ 2 ส่วน หลัก ๆ ส่วนไฟล์ R.java จะไม่ขอกล่าวถึง เนื่องจากไฟล์นี้จะประมวลผลให้เราเอง เราไม่ต้องไปยุ่งเกี่ยวหรือปรับปรุงแก้ไขสิ่งใดกับไฟล์นี้ ในส่วนที่จะกล่าวถึงนั้นคือ ส่วนของการสร้างตารางฐานข้อมูล การแสดงผล โดยจะอยู่ในไฟล์ประเภท .xml และส่วนที่เกี่ยวกับการควบคุมการทำงานทั้งหมด โดยจะอยู่ในไฟล์ประเภท .java ยกตัวอย่างเช่น หากภายในหน้า xml มีการแสดงผลที่มีการใช้ปุ่ม (Button) ภายในหน้า xml นั้นจะทำได้เพียงแสดงผลเพียงอย่างเดียว ส่วนการคลิกปุ่มนั้น หากเราต้องการคลิกปุ่มแล้ว แสดงผลต่อไปอย่างไร เราต้องเขียนซอร์สโค้ดสั่งงานภายใต้ไฟล์ .java เท่านั้น และเนื่องจากซอร์สโค้ดในแอปพลิเคชัน หน้าต่าง ๆ ของแอปพลิเคชันมีความคล้ายคลึงกันในการเขียน ผู้จัดทำจึงจะอธิบายเพื่อเป็นแนวทางให้เข้าใจหลักการเขียนเพียงเท่านั้น และจักอธิบายในส่วนของการสร้าง Table โดยใช้ active android เป็นตัวช่วยในการสร้าง Table และ file ต่างๆ ตามที่ต้องการ

Task.java

```
package project.ann.pee.studentappointment;

import android.renderscript.Sampler;
import com.activeandroid.Model;
import com.activeandroid.annotation.Column;
import com.activeandroid.annotation.Table;
import com.activeandroid.query.Select;
import java.sql.Time;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

@Table(name = "Tasks")
public class Task extends Model {

    @Column(name = "title", onUpdate = Column.ForeignKeyAction.CASCADE, onDelete =
Column.ForeignKeyAction.CASCADE)
    public Act title;

    @Column(name = "content")
    public String content;

    @Column(name = "contact", onUpdate = Column.ForeignKeyAction.CASCADE, onDelete =
Column.ForeignKeyAction.CASCADE)
    public ContactTB contact;

    @Column(name = "location", onUpdate = Column.ForeignKeyAction.CASCADE, onDelete =
Column.ForeignKeyAction.CASCADE)

    public LocationsTB location;

    public List<Act> acts() {
        return getMany(Act.class, "activityName");
    }

    public List<ContactTB> contactTBs() {
        return getMany(ContactTB.class, "firstName");
    }

    public List<LocationsTB> locationsTBs() {
        return getMany(LocationsTB.class, "locationName");
    }

    @Column(name = "timeStart")
    public String timeStart;

    @Column(name = "status", onUpdate = Column.ForeignKeyAction.CASCADE, onDelete =
Column.ForeignKeyAction.CASCADE)
    public StatusTB status;

    @Column(name = "timeEnd")
    public String timeEnd;
```

```

@Column(name = "dateStart")
public String dateStart;

@Column(name = "dateEnd")
public String dateEnd;

@Column(name = "dueAt", index = true)
public Date dueAt = null;

@Column(name = "createdAt", index = true)
public Date createdAt = null;

@Column(name = "updatedAt", index = true)
public Date updatedAt = null;

public static List<Task> getAll() {
    return new Select().from(Task.class).orderBy("updatedAt DESC").execute();
}

public void saveWithTimestamp() {
    Date now = new Date();
    updatedAt = now;
    if (createdAt == null)
        createdAt = now;
    save();
}

public static int countIncomplete() {
    return new Select()
        .from(Task.class)
        .where("status = ?", 1)
        .count();
}

public static int countComplete() {
    return new Select()
        .from(Task.class)
        .where("status = ?", 2)
        .count();
}

public static int countCancle() {
    return new Select()
        .from(Task.class)
        .where("status = ?", 3)
        .count();
}
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ Task.java สร้างขึ้นมาเพื่อ เป็นการสร้างตารางเก็บข้อมูล กิจกรรมนัดหมาย โดยมีรายละเอียดดังนี้

package project.ann.pee.studentappointment; คือ ชื่อ package ของแอปพลิเคชัน

```
import android.renderscript.Sampler;
import com.activeandroid.Model;
import com.activeandroid.annotation.Column;
import com.activeandroid.annotation.Table;
import com.activeandroid.query.Select;
import java.sql.Time;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
```

โค้ดข้างต้นคือการ import library มาใช้งาน เพื่อสนับสนุนคำสั่งต่าง ๆ ให้ใช้งานได้อย่างสมบูรณ์ ส่วนถัดมาเริ่มจากใช้คำสั่ง **@Table(name = "Tasks")** เพื่อกำหนดชื่อตาราง และตามด้วย คำสั่ง **public class Task extends Model** เพื่อเรียกใช้ active android ให้ทำงาน

ในส่วนของการสร้างคอลัมน์ (attribute) เราจะใช้คำสั่ง

```
@Column(name = "content")
public String content; เพื่อเก็บข้อมูลเพิ่มเติมของนัดหมาย
```

```
@Column(name = "timeStart")
public String timeStart; เพื่อเก็บข้อมูลเวลานัดหมาย
```

```
@Column(name = "timeEnd")
public String timeEnd; เพื่อเก็บข้อมูลเวลาสิ้นสุดนัดหมาย
```

```
@Column(name = "dateStart")
public String dateStart; เพื่อเก็บข้อมูลวันนัดหมาย
```

```
@Column(name = "dateEnd")
```

```
public String dateEnd;    เพื่อเก็บข้อมูลวันสิ้นสุดนัดหมาย
```

ทำแบบนี้ทุกคอลัมน์ที่ต้องการสร้างและหากต้องการบันทึกข้อมูลในคอลัมน์นั้นเป็น foreign key จะอธิบายในส่วนถัดไป

ในส่วนของการสร้าง

```
@Column(name = "title", onUpdate = Column.ForeignKeyAction.CASCADE, onDelete =
Column.ForeignKeyAction.CASCADE)
```

```
public Act title;
```

คือการเชื่อมความสัมพันธ์ระหว่างตาราง Tasks กับตาราง Act เพื่อรับส่งข้อมูลระหว่างกัน โดยที่ active android จะเป็นตัวกำหนด primary และ foreign key ให้อัตโนมัติ และ

```
public List<Act> acts() {
```

```
    return getMany(Act.class, "activityName");
```

```
} เพื่อเรียกค่าจากตาราง Act มาใช้งาน
```

การสร้าง

```
@Column(name = "contact", onUpdate = Column.ForeignKeyAction.CASCADE, onDelete =
Column.ForeignKeyAction.CASCADE)
```

```
public ContactTB contact;
```

คือการเชื่อมความสัมพันธ์ระหว่างตาราง Tasks กับตาราง Contact (ใช้ชื่อ class ว่า ContactTB) เพื่อรับส่งข้อมูลระหว่างกัน โดยที่ active android จะเป็นตัวกำหนด primary และ foreign key ให้อัตโนมัติ และ

```
public List<ContactTB> contactTBs() {
```

```
    return getMany(ContactTB.class, "firstName");
```

```
}
```

เพื่อเรียกค่าจากตาราง Contact มาใช้งาน

คำสั่ง

```
@Column(name = "location", onUpdate = Column.ForeignKeyAction.CASCADE, onDelete =
Column.ForeignKeyAction.CASCADE)
```

```
public LocationsTB location;
```

คือการเชื่อมความสัมพันธ์ระหว่างตาราง Tasks กับตาราง Location (ใช้ชื่อ class ว่า LocationTB) เพื่อรับส่งข้อมูลระหว่างกัน โดยที่ active android จะเป็นตัวกำหนด primary และ foreign key ให้อัตโนมัติ และ

```
public List<LocationsTB> locationsTBs() {
    return getMany(LocationsTB.class, "locationName");
}
```

เพื่อเรียกค่าจากตาราง Location มาใช้งาน

ส่วนของการเก็บข้อมูลกรณีมีการเพิ่มข้อมูลหรือแก้ไขข้อมูลจะใช้คำสั่ง

```
@Column(name = "dueAt", index = true)
```

```
public Date dueAt = null;
```

```
@Column(name = "createdAt", index = true)
```

```
public Date createdAt = null;
```

```
@Column(name = "updatedAt", index = true)
```

```
public Date updatedAt = null;
```

เพื่อเก็บค่าอ้างอิงการแก้ไขข้อมูลลงในตาราง

ในส่วนของการ stamp เวลาลงไปด้วยเมื่อผู้ใช้นั้นก็กดข้อมูลลงไปในตาราง จะใช้คำสั่ง

```
public void saveWithTimestamp() {
```

```
    Date now = new Date();
```

```
    updatedAt = now;
```

```
    if (createdAt == null)
```

```

        createdAt = now;
    save();
}

```

ในส่วนของการคำนวณค่าภายในตารางเพื่อนำไปแสดงผลเป็นรายงาน จำนวนกิจกรรมนัดหมายตามสถานะของกิจกรรมจะใช้คำสั่ง

```

public static int countIncomplete() {
    return new Select()
        .from(Task.class)
        .where("status = ?", 1)
        .count();
}

```

```

public static int countComplete() {
    return new Select()
        .from(Task.class)
        .where("status = ?", 2)
        .count();
}

```

```

public static int countCancel() {
    return new Select()
        .from(Task.class)
        .where("status = ?", 3)
        .count();
}

```

Act.java

```
package project.ann.pee.studentappointment;
```

```
import com.activeandroid.Model;
```

```
import com.activeandroid.annotation.Column;
```

```
import com.activeandroid.annotation.Table;
```

```
import com.activeandroid.query.Select;
```

```
import java.io.Serializable;
```

```
import java.util.Date;
```

```
import java.util.List;
```

```
@Table(name = "Act")
```

```
public class Act extends Model {
```

```
    @Column(name = "activityName")
```

```
    public String activityName;
```

```
    @Column(name = "dueAt", index = true)
```

```
    public Date dueAt = null;
```

```
    @Column(name = "createdAt", index = true)
```

```
    public Date createdAt = null;
```

```
    @Column(name = "updatedAt", index = true)
```

```
    public Date updatedAt = null;
```

```
    public static List<Act> getAll() {
```

```

        return new Select().from(Act.class).execute();
    }

    public void saveWithTimestamp() {
        Date now = new Date();
        updatedAt = now;
        if (createdAt == null)
            createdAt = now;
        save();
    }
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ Act.java สร้างขึ้นมาเพื่อ เป็นการสร้างตารางเก็บข้อมูล รายชื่อกิจกรรมต่าง ๆ เพื่อให้ผู้ใช้ได้เลือกใช้งาน ลักษณะของคำสั่งต่าง ๆ เป็นเช่นเดียวกับไฟล์ Task.java ในส่วนคำสั่ง

```

public static List<Act> getAll() {
    return new Select().from(Act.class).execute();
}

```

คือการส่งค่าของข้อมูลภายในตารางไปใช้งาน

LocationsTB.java

```

package project.ann.pee.studentappointment;

import com.activeandroid.Model;
import com.activeandroid.annotation.Column;
import com.activeandroid.annotation.Table;
import com.activeandroid.query.Select;
import java.util.Date;
import java.util.List;

@Table(name = "Location")
public class LocationsTB extends Model {

    @Column(name="locationName")
    public String locationName;

    @Column(name = "dueAt", index = true)
    public Date dueAt = null;

    @Column(name = "createdAt", index = true)
    public Date createdAt = null;

    @Column(name = "updatedAt", index = true)
    public Date updatedAt = null;

    public static List<LocationsTB> getAll() {
        return new Select().from(LocationsTB.class).execute();
    }
}

```

```

public void saveWithTimestamp() {
    Date now = new Date();
    updatedAt = now;
    if (createdAt == null)
        createdAt = now;
    save();
}
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ LocationsTB.java สร้างขึ้นมาเพื่อ เป็นการสร้างตารางเพื่อเก็บข้อมูลรายชื่อสถานที่ต่าง ๆ เพื่อให้ผู้ใช้ได้เลือกใช้งาน ลักษณะของคำสั่งต่าง ๆ เป็นเช่นเดียวกับไฟล์

Task.java

ในส่วนคำสั่ง

```

public static List<LocationsTB> getAll() {
    return new Select().from(LocationsTB.class).execute();
}

```

คือการส่งค่าของข้อมูลภายในตารางไปใช้งาน

ContactTB.java

```
package project.ann.pee.studentappointment;

import android.content.Context;
import android.location.Location;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.TextView;
import com.activeandroid.Model;
import com.activeandroid.annotation.Column;
import com.activeandroid.annotation.Table;
import com.activeandroid.query.Select;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

@Table(name = "Contact")
public class ContactTB extends Model {

    @Column(name = "firstName")
    public String firstName;

    @Column(name = "lastName")
    public String lastName;

    @Column(name = "dueAt", index = true)
    public Date dueAt = null;
    @Column(name = "createdAt", index = true)
    public Date createdAt = null;

    @Column(name = "updatedAt", index = true)
    public Date updatedAt = null;

    public static List<ContactTB> getAll() {
        return new Select().from(ContactTB.class).execute();
    }

    public ContactTB() {
        super();
    }

    public List<Task> tasks() {
        return getMany(Task.class, "contact");
    }

    public void saveWithTimestamp() {
        Date now = new Date();
        updatedAt = now;
        if (createdAt == null)
            createdAt = now;
        save();
    }
}
```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ ContactTB.java สร้างขึ้นมาเพื่อ เป็นการสร้างตารางเก็บข้อมูลรายชื่อผู้ติดต่อ เพื่อให้ผู้ใช้ได้เลือกใช้งาน ลักษณะของคำสั่งต่าง ๆ เป็นเช่นเดียวกับไฟล์ Task.java

ในส่วนคำสั่ง

```
public static List<ContactTB> getAll() {
    return new Select().from(ContactTB.class).execute();
}
```

คือการส่งค่าของข้อมูลภายในตารางไปใช้งาน

StatusTB.java

```
package project.ann.pee.studentappointment;
```

```
import com.activeandroid.Model;
import com.activeandroid.annotation.Column;
import com.activeandroid.annotation.Table;
import com.activeandroid.query.Select;
import java.io.Serializable;
import java.util.Date;
import java.util.List;
```

```
@Table(name = "StatusTBs")
```

```
public class StatusTB extends Model {
```

```
    @Column(name = "statusName")
```

```
    public String statusName;
```

```
    @Column(name = "createdAt", index = true)
```

```
    public Date createdAt = null;
```



```

@Column(name = "updatedAt", index = true)

public Date updatedAt = null;

public static List<StatusTB> getAll() {

    return new Select().from(StatusTB.class).execute();

}

public void saveWithTimestamp() {

    Date now = new Date();

    updatedAt = now;

    if (createdAt == null)

        createdAt = now;

    save();

}

}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ StatusTB.java สร้างขึ้นมาเพื่อ เป็นการสร้างตารางเก็บสถานะของกิจกรรม(Incomplete Complete Cancel) เพื่อให้ผู้ใช้ได้เลือกใช้งาน ลักษณะของคำสั่งต่าง ๆ เป็นเช่นเดียวกับไฟล์ Task.java

ในส่วนคำสั่ง

```

public static List<StatusTB> getAll() {

    return new Select().from(StatusTB.class).execute();

```

คือการส่งค่าของข้อมูลภายในตารางไปใช้งาน

Activity_form.xml

DTAC 12:43 64%

← New Appoint... ↻

Activity : Nothing ▼ +

Contact : Nothing ▼ +

Location : Nothing ▼ +

Content

Date Start

Time Start

Date End

Time End

Status : Incomplete ▼

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:background="@color/colorbk"
```

```
tools:context="project.ann.pee.studentappointment.FormActivity">
```

```
<include
```

```
    android:id="@+id/toolbar_main"
```

```
    layout="@layout/toolbar_main" />
```

```
<RelativeLayout
```

```
    android:id="@+id/relativeLayout"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignParentStart="true"
```

```
    android:layout_below="@+id/toolbar_main"
```

```
    android:layout_gravity="center_horizontal"
```

```
    android:orientation="vertical"
```

```
    android:paddingBottom="10dp"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin">
```

```
</RelativeLayout>
```

```
<RelativeLayout
```

```
    android:id="@+id/relativeLayout2"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="100dp"
```

```
    android:layout_below="@+id/relativeLayout"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin">
```

```
<android.support.design.widget.TextInputLayout
```

```
    android:id="@+id/view"
```

```

android:layout_width="170dp"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_marginStart="10dp"
android:layout_toEndOf="@+id/textView2">

```

```
<Spinner
```

```

    android:id="@+id/titleEdit"
    android:layout_width="fill_parent"
    android:layout_height="35dp"
    android:layout_alignParentStart="true"
    android:prompt="@string/spinner_title"
/>

```

```
<Spinner
```

```

    android:id="@+id/spinner"
    android:layout_width="fill_parent"
    android:layout_height="35dp"
    android:layout_alignParentStart="true"
    android:layout_below="@id/titleEdit"
    android:prompt="@string/spinner_title" />

```

```
<Spinner
```

```

    android:id="@+id/locationEdit"
    android:layout_width="fill_parent"
    android:layout_height="35dp"
    android:layout_alignParentStart="true"
    android:layout_below="@id/spinner"

```

```

        android:prompt="@string/spinner_title" />

</android.support.design.widget.TextInputLayout>

```

```

<ImageButton
    android:id="@+id/add_contact"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_centerVertical="true"
    android:layout_marginStart="10dp"
    android:layout_toEndOf="@+id/view"
    android:background="@color/md_blue_100"
    android:src="@drawable/ic_add_white_18dp" />

```

```

<ImageButton
    android:id="@+id/add_location"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="10dp"
    android:layout_toEndOf="@+id/view"
    android:background="@color/md_blue_100"
    android:src="@drawable/ic_add_white_18dp" />

```

```

<ImageButton
    android:id="@+id/add_title"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_marginStart="10dp"

```

```

    android:layout_toEndOf="@+id/view"
    android:background="@color/md_blue_100"
    android:src="@drawable/ic_add_white_18dp" />

```

```
<TextView
```

```

    android:id="@+id/textView2"
    android:layout_width="60dp"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/add_title"
    android:layout_alignParentStart="true"
    android:text="@string/FormAct" />

```

```
<TextView
```

```

    android:layout_width="60dp"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/add_contact"
    android:layout_alignParentStart="true"
    android:text="@string/FormContact" />

```

```
<TextView
```

```

    android:layout_width="65dp"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    android:text="@string/FormLocation" />

```

```
</RelativeLayout>
```

```
<LinearLayout
```

```

    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/relativeLayout2"
    android:layout_gravity="center_horizontal"
    android:orientation="vertical"
    android:paddingBottom="0dp"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="0dp">
    <android.support.design.widget.TextInputLayout
        android:layout_width="269dp"
        android:layout_height="wrap_content">

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    <EditText
        android:id="@+id/contentEdit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"

```

```

        android:hint="@string/task_content"
        android:inputType="textMultiLine" />
</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/dateStart"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:ems="10"
        android:hint="@string/date_start"
        android:inputType="date" />

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/timeStart"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:ems="10"
            android:hint="@string/time_start"
            android:inputType="time" />

        </android.support.design.widget.TextInputLayout>
    </android.support.design.widget.TextInputLayout>

```



```

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/dateEnd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:ems="10"
        android:hint="@string/date_end"
        android:inputType="date" />

</android.support.design.widget.TextInputLayout>

```

```

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/timeEnd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:ems="10"
        android:hint="@string/time_end"
        android:inputType="time" />

</android.support.design.widget.TextInputLayout>

```

```

<RelativeLayout

```

```

    android:id="@+id/spinStatus"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/timeEnd">

```

```

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:text="@string/FormStatus" />

```

```

<Spinner
    android:id="@+id/status"
    android:layout_width="190dp"
    android:layout_height="35dp"
    android:layout_alignParentTop="true"
    android:layout_gravity="center_horizontal"
    android:layout_marginStart="24dp"
    android:layout_toEndOf="@+id/textView"
    android:prompt="@string/spinner_title" />

```

```

</RelativeLayout>

```

```

</LinearLayout>

```

```

</RelativeLayout>

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ activity_form.xml สร้างขึ้นมาเพื่อแสดงผลหน้าบันทึกกิจกรรมนัดหมาย การจะแสดงผลออกมาจะต้องเขียนซอร์สโค้ดภายในแท็ก **<RelativeLayout>** และเราสามารถกำหนดการแสดงผลของหน้าจอได้โดยผู้จัดทำ โดยการกำหนดค่าความกว้างและความยาวของการแสดงผลให้พอดีกับเนื้อหา(Content) โดยใช้ซอร์สโค้ด

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

ในส่วนของเมนูบาร์ด้านบนเป็นการใช้คำสั่งดึงมาจากหน้า toolbar_main.xml(เขียนไว้ไฟล์เดียว เพื่อดึงไปใช้ในหน้าต่าง ๆ ไม่ต้องเขียนซ้ำๆ ทุกหน้าที่ต้องการใช้) คำสั่งคือ

```
<include
```

```
    android:id="@+id/toolbar_main"
```

```
    layout="@layout/toolbar_main" />
```

ส่วนต่อไปนี้เป็นส่วนของ Dropdown จากที่ให้เห็นในหน้าแสดงผล ได้แสดงเป็น dropdown ของชื่อกิจกรรมนัดหมาย ชื่อผู้ติดต่อ ชื่อสถานที่ โดยมีความต้องการให้ผู้ใช้งานสามารถเลือกรายการได้ เขียนโดยใช้คำสั่ง **<Spinner** และปิดด้วย **/Spinner>** ซอร์สโค้ดคือ

```
<Spinner
```

```
    android:id="@+id/titleEdit"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="35dp"
```

```
    android:layout_alignParentStart="true"
```

```
    android:prompt="@string/spinner_title"
```

```
/>
```

```
<Spinner
```

```
    android:id="@+id/spinner"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="35dp"
```

```
    android:layout_alignParentStart="true"
```

```

    android:layout_below="@id/titleEdit"
    android:prompt="@string/spinner_title" />

```

<Spinner

```

    android:id="@+id/locationEdit"
    android:layout_width="fill_parent"
    android:layout_height="35dp"
    android:layout_alignParentStart="true"
    android:layout_below="@id/spinner"
    android:prompt="@string/spinner_title" />

```

ส่วนต่อไปคือปุ่ม สามปุ่ม ที่อยู่ด้านข้าง dropdown ที่มีเพื่อให้ผู้ใช้ได้กดเพิ่มชื่อกิจกรรมนัดหมาย ชื่อผู้ติดต่อ และชื่อสถานที่ เป็นการเขียนซอร์สโค้ดภายใต้แท็ก <ImageButton> และปิดด้วย </ImageButton> ดังตัวอย่างต่อไปนี้

<ImageButton

```

    android:id="@+id/add_contact"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_centerVertical="true"
    android:layout_marginStart="10dp"
    android:layout_toEndOf="@+id/view"
    android:background="@color/md_blue_100"
    android:src="@drawable/ic_add_white_18dp" />

```

<ImageButton

```

    android:id="@+id/add_location"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_alignParentBottom="true"

```

```

android:layout_marginStart="10dp"
android:layout_toEndOf="@+id/view"
android:background="@color/md_blue_100"
android:src="@drawable/ic_add_white_18dp" />

```

<ImageButton

```

android:id="@+id/add_title"
android:layout_width="30dp"
android:layout_height="30dp"
android:layout_marginStart="10dp"
android:layout_toEndOf="@+id/view"
android:background="@color/md_blue_100"
android:src="@drawable/ic_add_white_18dp" />

```

การกำหนดให้รูปของปุ่มเป็นรูปเครื่องหมายบวกเราได้ใช้คำสั่ง

`android:src="@drawable/ic_add_white_18dp"` เพื่ิ่ดรูปเครื่องหมายบวกมาแสดงในปุ่ม

ส่วนต่อไปคือข้อความที่แสดงบอกว่า dropdown นั้นคืออะไร ดังหน้าจอแสดงผลข้างต้นจะมีด้วยกัน

3 text คือ Activity: Contact: และ Location: โดยการใช้การเขียนซอร์สโค้ดภายใต้คำสั่ง `<TextView` และปิดด้วย `/>` ดังนี้

<TextView

```

android:id="@+id/textView2"
android:layout_width="60dp"
android:layout_height="wrap_content"
android:layout_alignBottom="@+id/add_title"
android:layout_alignParentStart="true"
android:text="@string/FormAct" />

```

<TextView

```

    android:layout_width="60dp"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/add_contact"
    android:layout_alignParentStart="true"
    android:text="@string/FormContact" />

```

<TextView

```

    android:layout_width="65dp"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    android:text="@string/FormLocation" />

```

ในส่วนของการแสดงผลเพื่อรับค่าต่าง ๆ ไม่ว่าจะเป็น วันทีนัดหมาย รายละเอียดเพิ่มเติมของการนัดหมาย เวลานัดหมาย วันที่สิ้นสุดของกิจกรรม เวลาสิ้นสุดของกิจกรรม ได้เขียนซอร์สโค้ดภายใต้

<LinearLayout และปิดด้วย /LinearLayout> ซึ่งจะซ่อนอยู่ภายใต้ RelativeLayout ข้างต้นที่ได้กล่าวไปแล้ว โดยจะใช้แท็ก EditText เป็นแท็กในการเขียนเพื่อรับค่า
ดังนี้

<LinearLayout

```

    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/relativeLayout2"
    android:layout_gravity="center_horizontal"
    android:orientation="vertical"
    android:paddingBottom="0dp"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"

```

```
android:paddingTop="0dp">
```

```
<android.support.design.widget.TextInputLayout
```

```
    android:layout_width="269dp"
```

```
    android:layout_height="wrap_content">
```

```
</android.support.design.widget.TextInputLayout>
```

```
<android.support.design.widget.TextInputLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content">
```

```
</android.support.design.widget.TextInputLayout>
```

```
<android.support.design.widget.TextInputLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content">
```

```
<EditText
```

```
    android:id="@+id/contentEdit"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:ems="10"
```

```
    android:hint="@string/task_content"
```

```
    android:inputType="textMultiLine" />
```

```
</android.support.design.widget.TextInputLayout>
```

```
<android.support.design.widget.TextInputLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content">
```

```

<EditText
    android:id="@+id/dateStart"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:ems="10"
    android:hint="@string/date_start"
    android:inputType="date" />

```

```

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```

```

<EditText
    android:id="@+id/timeStart"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:ems="10"
    android:hint="@string/time_start"
    android:inputType="time" />

```

```

</android.support.design.widget.TextInputLayout>

```

```

</android.support.design.widget.TextInputLayout>

```

```

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```


<EditText

```

    android:id="@+id/dateEnd"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:ems="10"
    android:hint="@string/date_end"
    android:inputType="date" />

```

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```

<EditText

```

    android:id="@+id/timeEnd"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:ems="10"
    android:hint="@string/time_end"
    android:inputType="time" />

```

</android.support.design.widget.TextInputLayout>

และปิดท้ายของซอร์สโค้ดหน้า activity_form.xml ด้วย dropdown ของ status เพื่อให้ผู้ใช้ได้เลือกให้สถานะของกิจกรรมนัดหมาย ลักษณะของโค้ดเป็นเช่นเดียวกันกับ dropdown ของ ชื่อกิจกรรมนัดหมาย ชื่อผู้ติดต่อ และชื่อสถานที่ ดังที่ได้กล่าวไปข้างต้น คือการใช้คำสั่ง <Spinner และปิดด้วย /Spinner> และใช้ TextView เพื่อบอกว่า dropdown นั้นคือ Status ซึ่งคำสั่งของ dropdown และ TextView ตรงนี้จะเขียนภายใต้แท็ก <RelativeLayout และปิดด้วย /RelativeLayout> ดังซอร์สโค้ดต่อไปนี้

<RelativeLayout

```

    android:id="@+id/spinStatus"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/timeEnd">

```

<TextView

```

    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:text="@string/FormStatus" />

```

<Spinner

```

    android:id="@+id/status"
    android:layout_width="190dp"
    android:layout_height="35dp"
    android:layout_alignParentTop="true"
    android:layout_gravity="center_horizontal"
    android:layout_marginStart="24dp"
    android:layout_toEndOf="@+id/textView"

    android:prompt="@string/spinner_title" />

```

</RelativeLayout>

สำหรับหน้า activity_form.xml จะถูกเรียกใช้เมื่อผู้ใช้กดเพิ่มกิจกรรมนัดหมายหรือกดแก้ไขนัดหมาย

FormActivtiy.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.renderscript.Sampler;
import android.support.design.widget.FloatingActionButton;
import android.text.InputType;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.DatePicker;
import android.widget.EditText;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import com.mikepenz.materialdrawer.model.ContainerDrawerItem;

```

```

import android.view.View.OnClickListener;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;
import android.widget.AdapterView.OnItemClickListener;

public class FormActivity extends BaseActivity implements OnClickListener {

    private static final int MenuItem_SaveID = 1;
    private static final int Refresh = 2;

    private Task task = null;
    private ContactTB contactTB;
    private EditText fromDateEtxt;
    private EditText toDateEtxt;
    private EditText fromTime;
    private EditText toTime;
    private EditText contentEdit;
    private EditText time_StarEdit;
    private EditText time_EndEdit;

```

```

private EditText date_startEdit;
private EditText date_endEdit;
private Spinner location;
private Spinner spinner;
private Spinner titleEdit;
private Spinner spn;
private int mHour;
private int mMinute;
private ArrayList<ContactTB> contactTBs;
private ArrayList<Act> acts;
private ArrayList<LocationsTB> locats;
private ArrayList<StatusTB> statusTBs;
private DatePickerDialog fromDatePickerDialog;
private DatePickerDialog toDatePickerDialog;
private TimePickerDialog fromTimePickerDialog;
private TimePickerDialog toTimePickerDialog;
private SimpleDateFormat dateFormatter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_form);
    setDrawer(true);
    dateFormatter = new SimpleDateFormat("dd-MM-yyyy", Locale.US);
    findViewById();
    setDateTimeField();
    setView();

    ImageButton add_title = (ImageButton) findViewById(R.id.add_title);

```

```

add_title.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(FormActivity.this, FormActDefault.class));
    }
});

```

```

ImageButton add_contact = (ImageButton) findViewById(R.id.add_contact);
add_contact.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(FormActivity.this, FormContactDefault.class));
    }
});

```

```

ImageButton add_location = (ImageButton) findViewById(R.id.add_location);
add_location.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(FormActivity.this, FormLocationDefault.class));
    }
});

```

```

spn = (Spinner) findViewById(R.id.status);
spinner = (Spinner) findViewById(R.id.spinner);
titleEdit = (Spinner) findViewById(R.id.titleEdit);
location = (Spinner) findViewById(R.id.locationEdit);

```

```

contentEdit = (EditText) findViewById(R.id.contentEdit);
time_StarEdit = (EditText) findViewById(R.id.timeStart);
time_EndEdit = (EditText) findViewById(R.id.timeEnd);
date_startEdit = (EditText) findViewById(R.id.dateStart);
date_endEdit = (EditText) findViewById(R.id.dateEnd);

long id = getIntent().getLongExtra("id", 0);
if (id == 0) {
    setTitle(R.string.new_task);
} else {
    setTitle(R.string.edit_task);
    task = Task.load(Task.class, id);
    contactTB = ContactTB.load(ContactTB.class, id);
    if (task != null) {
        contentEdit.setText(task.content);
        time_StarEdit.setText(task.timeStart);
        time_EndEdit.setText(task.timeEnd);
        date_startEdit.setText(task.dateStart);
        date_endEdit.setText(task.dateEnd);

        int spinnerPosition = task.contact.getId().intValue() - 1;
        spinner.setSelection(spinnerPosition);

        int spinnerPosition4 = task.status.getId().intValue() - 1;
        spn.setSelection(spinnerPosition4);

        int spinnerPosition1 = task.title.getId().intValue() - 1;
        titleEdit.setSelection(spinnerPosition1);
    }
}

```

```

        int spinnerPosition2 = task.location.getId().intValue() - 1;

        location.setSelection(spinnerPosition2);

    } else {

        finish();

    }

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

    addMenuItem(menu, Refresh, R.string.refresh,
buildDrawable(MaterialDesignIconic.Icon.gmi_refresh));

    addMenuItem(menu, MenuItem_SaveID, R.string.save,
buildDrawable(MaterialDesignIconic.Icon.gmi_save));

    return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

        case android.R.id.home:

            if (isEdited()) {

                AlertDialog.Builder alert = new AlertDialog.Builder(this);

                alert.setTitle(android.R.string.dialog_alert_title);

                alert.setMessage(R.string.unsaved_exit_alert);

                alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {

                    @Override

                    public void onClick(DialogInterface dialog, int which) {

                        onBackPressed();

                    }

                }

            }

        }

    }

}

```



```

});

alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {

    @Override

    public void onClick(DialogInterface dialog, int which) {

        }

    });

alert.show();

return true;

}

break;

case MenuItem_SaveID:

    save();

    break;

case Refresh:

    refresh();

    break;

}

return super.onOptionsItemSelected(item);

}

private void refresh() {

    Intent i = getIntent();

    finish();

    startActivity(i);

}

private boolean isEdited() {

    if (task == null)

        return contentEdit.getText().length() > 0 || time_StarEdit.getText().length() > 0 ||

time_EndEdit.getText().length() > 0

        || date_startEdit.getText().length() > 0

```

```

        || date_endEdit.getText().length() > 0;
    else
        return !task.content.equals(contentEdit.getText().toString()) ||
!task.timeStart.equals(time_StarEdit.getText().toString())
        || !task.timeEnd.equals(time_EndEdit.getText().toString())
        || !task.dateStart.equals(date_startEdit.getText().toString())
        || !task.dateEnd.equals(date_endEdit.getText().toString());
    }

```

```

private void findViewsById() {
    fromDateEtxt = (EditText) findViewById(R.id.dateStart);
    fromDateEtxt.setInputType(InputType.TYPE_NULL);
    fromDateEtxt.requestFocus();

    fromTime = (EditText) findViewById(R.id.timeStart);
    fromTime.setInputType(InputType.TYPE_NULL);
    fromTime.requestFocus();

    toDateEtxt = (EditText) findViewById(R.id.dateEnd);
    toDateEtxt.setInputType(InputType.TYPE_NULL);

    toTime = (EditText) findViewById(R.id.timeEnd);
    toTime.setInputType(InputType.TYPE_NULL);
}

```

```

private void setDateTextField() {
    fromDateEtxt.setOnClickListener(this);
    toDateEtxt.setOnClickListener(this);

    fromTime.setOnClickListener(this);
}

```

```

toTime.setOnClickListener(this);

Calendar newCalendar = Calendar.getInstance();

mHour = newCalendar.get(Calendar.HOUR_OF_DAY);
mMinute = newCalendar.get(Calendar.MINUTE);

fromDatePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {

    public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {

        Calendar newDate = Calendar.getInstance();
        newDate.set(year, monthOfYear, dayOfMonth);

        fromDateetxt.setText(dateFormatter.format(newDate.getTime()));

    }

}, newCalendar.get(Calendar.YEAR), newCalendar.get(Calendar.MONTH),
newCalendar.get(Calendar.DAY_OF_MONTH));

fromTimePickerDialog = new TimePickerDialog(this
    , new TimePickerDialog.OnTimeSetListener() {

        @Override

        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

            fromTime.setText(hourOfDay + ":" + minute);

        }

    }, mHour, mMinute, false);

toTimePickerDialog = new TimePickerDialog(this
    , new TimePickerDialog.OnTimeSetListener() {

        @Override

        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

            toTime.setText(hourOfDay + ":" + minute);

        }

    }, mHour, mMinute, false);

```

```

toDatePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {

    public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {

        Calendar newDate = Calendar.getInstance();

        newDate.set(year, monthOfYear, dayOfMonth);

        toDateEtxt.setText(dateFormatter.format(newDate.getTime()));

    }

}, newCalendar.get(Calendar.YEAR), newCalendar.get(Calendar.MONTH),
newCalendar.get(Calendar.DAY_OF_MONTH));

}

private void save() {

    if (date_startEdit.getText().length() > 0) {

        if (task == null)

            task = new Task();

        task.status = (StatusTB) spn.getSelectedItem();

        task.title = (Act) titleEdit.getSelectedItem();

        task.content = contentEdit.getText().toString();

        task.contact = (ContactTB) spinner.getSelectedItem();

        task.timeStart = time_StarEdit.getText().toString();

        task.timeEnd = time_EndEdit.getText().toString();

        task.dateStart = date_startEdit.getText().toString();

        task.dateEnd = date_endEdit.getText().toString();

        task.location = (LocationsTB) location.getSelectedItem();

        task.saveWithTimestamp();

        setResult(Activity.RESULT_OK, new Intent().putExtra("id", task.getId()));

        this.finish();

    } else {

        AlertDialog.Builder alert = new AlertDialog.Builder(this);

        alert.setTitle(android.R.string.dialog_alert_title);

        alert.setMessage(R.string.title_is_required);

```

```

alert.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {

    @Override

    public void onClick(DialogInterface dialog, int which) {

        }

    });

    alert.show();

}

}

@Override

public void onClick(View v) {

    if (v == fromDateTxt) {

        fromDatePickerDialog.show();

    }

    if (v == fromTime) {

        fromTimePickerDialog.show();

    } else if (v == toDateTxt) {

        toDatePickerDialog.show();

    }

    if (v == toTime) {

        toTimePickerDialog.show();

    }

}

private void setView() {

    Spinner sp = (Spinner) findViewById(R.id.spinner);

    sp.setOnItemSelectedListener(new ContactTBAdapter(this, contactTBs));

    contactTBs = new ArrayList<>(ContactTB.getAll());

    Spinner sp1 = (Spinner) findViewById(R.id.titleEdit);

```

```

sp1.setOnItemClickListener(new ActivityAdapter(this, acts));
acts = new ArrayList<>(Act.getAll());

Spinner sp2 = (Spinner) findViewById(R.id.locationEdit);
sp2.setOnItemClickListener(new LocationAdapter(this, locats));
locats = new ArrayList<>(LocationsTB.getAll());

Spinner sp3 = (Spinner) findViewById(R.id.status);
sp3.setOnItemClickListener(new StatusAdapter(this, statusTBs));
statusTBs = new ArrayList<>(StatusTB.getAll());

sp.setAdapter(new ContactTBAdapter(this, contactTBs));
ArrayAdapter<ContactTB> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, contactTBs);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

sp1.setAdapter(new ActivityAdapter(this, acts));
ArrayAdapter<Act> adapter1 = new ArrayAdapter<>(this, android.R.layout.simple_spinner_item, acts);
adapter1.setDropDownViewResource(android.R.layout.simple_spinner_item);

sp2.setAdapter(new LocationAdapter(this, locats));
ArrayAdapter<LocationsTB> adapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item, locats);
adapter2.setDropDownViewResource(android.R.layout.simple_spinner_item);

sp3.setAdapter(new StatusAdapter(this, statusTBs));
ArrayAdapter<StatusTB> adapter3 = new ArrayAdapter<StatusTB>(this,
android.R.layout.simple_spinner_item, statusTBs);

```

```

        adapter3.setDropDownViewResource(android.R.layout.simple_spinner_item);

    }

    private class LocationAdapter extends ArrayAdapter<LocationsTB> implements OnItemSelectedListener
    {

        public LocationAdapter(Context context, ArrayList<LocationsTB> locats) {

            super(context, R.layout.support_simple_spinner_dropdown_item, locats);

        }

        @Override

        public View getDropDownView(int position, View convertView, ViewGroup parent) {

            LocationsTB locationsTB = getItem(position);

            if (convertView == null) {

                convertView =

                LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_dropdown_item, parent, false);

            }

            TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);

            sp1.setText(locationsTB.locationName);

            return convertView;

        }

        @Override

        public View getView(int position, View convertView, ViewGroup parent) {

            LocationsTB locationsTB = getItem(position);

            if (convertView == null) {

                convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_item,

                parent, false);

```

```

    }

    TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
    sp1.setText(locationsTB.locationName);

    return convertView;
}

@Override
public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

}

@Override
public void onNothingSelected(AdapterView<?> parent) {

}
}

private class ActivityAdapter extends ArrayAdapter<Act> implements OnItemSelectedListener {
    public ActivityAdapter(Context context, ArrayList<Act> acts) {
        super(context, R.layout.support_simple_spinner_dropdown_item, acts);
    }

    @Override
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        Act act = getItem(position);
        if (convertView == null) {
            convertView =
                LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_dropdown_item, parent, false);
        }
        TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
        sp1.setText(act.activityName);
        return convertView;
    }
}

```



```

    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        Act act = getItem(position);

        if (convertView == null) {
            convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_item,
parent, false);
        }

        TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
        sp1.setText(act.activityName);

        return convertView;
    }

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
}

class ContactTBAdapter extends ArrayAdapter<ContactTB> implements OnItemSelectedListener {
    public ContactTBAdapter(Context context, ArrayList<ContactTB> contactTBs) {
        super(context, R.layout.support_simple_spinner_dropdown_item, contactTBs);
    }

    @Override
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        ContactTB contactTB = getItem(position);

        if (convertView == null) {
            convertView =

```

```

LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_dropdown_item, parent, false);
    }

    TextView sp = (TextView) convertView.findViewById(android.R.id.text1);
    sp.setText(contactTB.firstName);

    return convertView;
}

@Override

public View getView(int position, View convertView, ViewGroup parent) {
    ContactTB contactTB = getItem(position);

    if (convertView == null) {
        convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_item,
parent, false);
    }

    TextView sp = (TextView) convertView.findViewById(android.R.id.text1);
    sp.setText(contactTB.firstName);

    return convertView;
}

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
}

@Override

public void onNothingSelected(AdapterView<?> parent) {

}
}

private class StatusAdapter extends ArrayAdapter<StatusTB> implements OnItemSelectedListener {
    public StatusAdapter(Context context, ArrayList<StatusTB> statusTBs) {
        super(context, R.layout.support_simple_spinner_dropdown_item, statusTBs);
    }
}

```

@Override

public View getDropDownView(**int** position, View convertView, ViewGroup parent) {

 StatusTB statusTB = getItem(position);

if (convertView == **null**) {

 convertView =

 LayoutInflater.from(getContext()).inflate(android.R.layout.*simple_spinner_dropdown_item*, parent, **false**);

 }

 TextView sp1 = (TextView) convertView.findViewById(android.R.id.*text1*);

 sp1.setText(statusTB.*statusName*);

return convertView;

}

@Override

public View getView(**int** position, View convertView, ViewGroup parent) {

 StatusTB statusTB = getItem(position);

if (convertView == **null**) {

 convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.*simple_spinner_item*,
parent, **false**);

 }

 TextView sp1 = (TextView) convertView.findViewById(android.R.id.*text1*);

 sp1.setText(statusTB.*statusName*);

return convertView;

}

@Override

public void onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {

}

@Override

```

    public void onNothingSelected(AdapterView<?> parent) {
    }
}
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ FormActivity.java สร้างขึ้นมาเพื่อรับค่าจากหน้า
 จอแสดงผลที่มี activity_form.xml ทำหน้าที่แสดงผล เพื่อส่งค่าไปเก็บยังฐานข้อมูล ไม่เพียงเท่านั้นยังทำหน้าที่
 ดึงค่าต่าง ๆ มาแสดง ไม่ว่าจะเป็น icon บนเมนูบาร์หรือจะเป็น นาฬิกาเพื่อให้ผู้ใช้เลือกเวลานัดหมาย หรือจะ
 เป็นวันที่ เพื่อให้ผู้ใช้เลือกวันที่นัดหมายและวันที่สิ้นสุดนัดหมาย

ในส่วนแรก

package project.ann.pee.studentappointment; เป็นชื่อเรียก package ของแอปพลิเคชัน

ส่วนต่อมาก็คือ

```

import android.app.Activity;
import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.renderscript.Sampler;
import android.support.design.widget.FloatingActionButton;
import android.text.InputType;
import android.util.Log;
import android.view.LayoutInflater;

```

```

import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.DatePicker;
import android.widget.EditText;

import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import com.mikepenz.materialdrawer.model.ContainerDrawerItem;

import android.view.View.OnClickListener;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;
import android.widget.AdapterView.OnItemSelectedListener;

```

ส่วนนี้เป็นการ import library ของแอปพลิเคชันมาใช้งาน เพื่อให้คำสั่งต่าง ๆ ภายใน class สามารถทำงานได้อย่างสมบูรณ์

ส่วนถัดมาเป็นการประกาศค่าและตัวแปรต่าง ๆ

```
private static final int MenuItem_SaveID = 1;
private static final int Refresh = 2;
private Task task = null;
private ContactTB contactTB;
private EditText fromDateTxt;
private EditText toDateTxt;
private EditText fromTime;
private EditText toTime;
private EditText contentEdit;
private EditText time_StarEdit;
private EditText time_EndEdit;
private EditText date_startEdit;
private EditText date_endEdit;
private Spinner location;
private Spinner spinner;
private Spinner titleEdit;
private Spinner spn;
private int mHour;
private int mMinute;
private ArrayList<ContactTB> contactTBs;
private ArrayList<Act> acts;
private ArrayList<LocationsTB> locats;
private ArrayList<StatusTB> statusTBs;
private DatePickerDialog fromDatePickerDialog;
private DatePickerDialog toDatePickerDialog;
```

```
private TimePickerDialog fromTimePickerDialog;
```

```
private TimePickerDialog toTimePickerDialog;
```

```
private SimpleDateFormat dateFormatter;
```

และตามด้วยการเรียกใช้หน้าแสดงผลนั้นก็คือหน้า activity_form โดยใช้คำสั่ง

```
setContentView(R.layout.activity_form);
```

ส่วนต่อไปคือการสั่งงานให้ปุ่มสามปุ่มที่มีไว้เพื่อเพิ่มชื่อกิจกรรมนัดหมาย ชื่อผู้ติดต่อ และชื่อสถานที่
นัดหมาย สามารถทำงานได้ตามต้องการ ดังซอร์สโค้ดต่อไปนี้

```
ImageButton add_title = (ImageButton) findViewById(R.id.add_title);
```

```
add_title.setOnClickListener(new OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
        startActivity(new Intent(FormActivity.this, FormActDefault.class));
```

```
    }
```

```
});
```

```
ImageButton add_contact = (ImageButton) findViewById(R.id.add_contact);
```

```
add_contact.setOnClickListener(new OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
        startActivity(new Intent(FormActivity.this, FormContactDefault.class));
```

```
    }
```

```
});
```

```
ImageButton add_location = (ImageButton) findViewById(R.id.add_location);
```

```
add_location.setOnClickListener(new OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
        startActivity(new Intent(FormActivity.this, FormLocationDefault.class));
```

```

    }
});

```

ส่วนต่อมาเพื่อให้การทำงานระหว่างตัวสั่งการและตัวแสดงผลทำงานด้วยกันได้ จึงประกาศว่าตัวแปรที่เราประกาศไว้ข้างต้นสัมพันธ์กับแท็กตัวใดในไฟล์ xml ที่เราได้เขียนไว้ก่อนหน้านี้แล้ว ดังซอร์สโค้ดต่อไปนี้

```

spn = (Spinner) findViewById(R.id.status);

spinner = (Spinner) findViewById(R.id.spinner);

titleEdit = (Spinner) findViewById(R.id.titleEdit);

location = (Spinner) findViewById(R.id.locationEdit);

contentEdit = (EditText) findViewById(R.id.contentEdit);

time_StarEdit = (EditText) findViewById(R.id.timeStart);

time_EndEdit = (EditText) findViewById(R.id.timeEnd);

date_startEdit = (EditText) findViewById(R.id.dateStart);

date_endEdit = (EditText) findViewById(R.id.dateEnd);

```

ส่วนต่อไป เกี่ยวข้องกับการทำงานตอนผู้ใช้กลับมาแก้ไขนัดหมายเพราะฟังก์ชันนี้จะทำงานเมื่อผู้ใช้งานต้องการแก้ไขนัดหมาย ซึ่งเป็นการเรียกข้อมูลนัดหมายที่ผู้ใช้ได้บันทึกไว้ก่อนหน้านี้กลับมาแสดง ซอร์สโค้ดที่ใช้คือ

```

long id = getIntent().getLongExtra("id", 0);

if (id == 0) {

    setTitle(R.string.new_task);

} else {

    setTitle(R.string.edit_task);

    task = Task.load(Task.class, id);

    contactTB = ContactTB.load(ContactTB.class, id);

    if (task != null) {

        contentEdit.setText(task.content);

        time_StarEdit.setText(task.timeStart);

        time_EndEdit.setText(task.timeEnd);

        date_startEdit.setText(task.dateStart);

```



```

    date_endEdit.setText(task.dateEnd);

    int spinnerPosition = task.contact.getId().intValue() - 1;
    spinner.setSelection(spinnerPosition);

    int spinnerPosition4 = task.status.getId().intValue() - 1;
    spn.setSelection(spinnerPosition4);

    int spinnerPosition1 = task.title.getId().intValue() - 1;
    titleEdit.setSelection(spinnerPosition1);
    int spinnerPosition2 = task.location.getId().intValue() - 1;
    location.setSelection(spinnerPosition2);

} else {
    finish();
}
}
}

```

ในส่วนของ icon บนเมนูบาร์ด้านบน สร้างโดยใช้คำสั่ง

`@Override`

```

public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, Refresh, R.string.refresh, buildDrawable(MaterialDesignIconic.Icon.gmi_refresh));
    addMenuItem(menu, MenuItem_SaveID, R.string.save,
buildDrawable(MaterialDesignIconic.Icon.gmi_save));
    return true;
}

```

ในส่วนของการสั่งงานให้ปุ่ม ทำงานสั่งงานโดยใช้คำสั่งต่อไปนี้ ทั้งปุ่มบันทึกและปุ่มรีเฟรช

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            if (isEdited()) {
                AlertDialog.Builder alert = new AlertDialog.Builder(this);
                alert.setTitle(android.R.string.dialog_alert_title);
                alert.setMessage(R.string.unsaved_exit_alert);
                alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        onBackPressed();
                    }
                });
                alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                    }
                });
                alert.show();
                return true;
            }
            break;
        case MenuItem_SaveID:
            save();
            break;
        case Refresh:
            refresh();
            break;
    }
}
```

```

    }

    return super.onOptionsItemSelected(item);
}

```

```

private void refresh() {
    Intent i = getIntent();
    finish();
    startActivity(i);
}

```

ส่วนถัดมาคือคำสั่งให้ส่งค่า(Update) กลับไปยังฐานข้อมูลในกรณีที่ผู้ใช้แก้ไขนัดหมาย ซึ่งจะทำงานตอนผู้ใช้เรียกใช้หน้าแก้ไขนัดหมาย ดังซอร์สโค้ดต่อไปนี้

```

private boolean isEdited() {
    if (task == null)

        return contentEdit.getText().length() > 0 || time_StarEdit.getText().length() > 0 ||
time_EndEdit.getText().length() > 0
        || date_startEdit.getText().length() > 0
        || date_endEdit.getText().length() > 0;
    else
        return !task.content.equals(contentEdit.getText().toString()) ||
!task.timeStart.equals(time_StarEdit.getText().toString())
        || !task.timeEnd.equals(time_EndEdit.getText().toString())
        || !task.dateStart.equals(date_startEdit.getText().toString())
        || !task.dateEnd.equals(date_endEdit.getText().toString());
}

```

ในส่วนนี้คือการสร้าง TimePicker และ DatePicker ทั้งสองส่วนคือการแสดงวันที่และเวลาขึ้นมาให้ผู้ใช้เลือกโดยไม่ต้องพิมพ์เป็นตัวอักษรลงไป ซึ่งใช้คำสั่งดังต่อไปนี้

```

private void findViewsById() {
    fromDateetxt = (EditText) findViewById(R.id.dateStart);
    fromDateetxt.setInputType(InputType.TYPE_NULL);
    fromDateetxt.requestFocus();

    fromTime = (EditText) findViewById(R.id.timeStart);
    fromTime.setInputType(InputType.TYPE_NULL);
    fromTime.requestFocus();
    toDateetxt = (EditText) findViewById(R.id.dateEnd);
    toDateetxt.setInputType(InputType.TYPE_NULL);

    toTime = (EditText) findViewById(R.id.timeEnd);
    toTime.setInputType(InputType.TYPE_NULL);
}

private void setDateTextField() {
    fromDateetxt.setOnClickListener(this);
    toDateetxt.setOnClickListener(this);
    fromTime.setOnClickListener(this);
    toTime.setOnClickListener(this);

    Calendar newCalendar = Calendar.getInstance();
    mHour = newCalendar.get(Calendar.HOUR_OF_DAY);
    mMinute = newCalendar.get(Calendar.MINUTE);

    fromDatePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {
        public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
            Calendar newDate = Calendar.getInstance();
            newDate.set(year, monthOfYear, dayOfMonth);
            fromDateetxt.setText(dateFormatter.format(newDate.getTime()));
        }
    }, newCalendar.get(Calendar.YEAR), newCalendar.get(Calendar.MONTH),

```

```

newCalendar.get(Calendar.DAY_OF_MONTH);

fromTimePickerDialog = new TimePickerDialog(this
    , new TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
            fromTime.setText(hourOfDay + ":" + minute);
        }
    }, mHour, mMinute, false);

toTimePickerDialog = new TimePickerDialog(this
    , new TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
            toTime.setText(hourOfDay + ":" + minute);
        }
    }, mHour, mMinute, false);

toDatePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {
    public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
        Calendar newDate = Calendar.getInstance();
        newDate.set(year, monthOfYear, dayOfMonth);
        toDateEtxt.setText(dateFormatter.format(newDate.getTime()));
    }
}, newCalendar.get(Calendar.YEAR), newCalendar.get(Calendar.MONTH),
newCalendar.get(Calendar.DAY_OF_MONTH));
}

```

และหัวใจหลักของหน้านี้ (FormActivity) คือการส่งค่าไปเก็บที่ฐานข้อมูล โดยใช้คำสั่งดังต่อไปนี้

```

private void save() {
    if (date_startEdit.getText().length() > 0) {
        if (task == null)
            task = new Task();
    }
}

```

```

task.status = (StatusTB) spn.getSelectedItem();
task.title = (Act) titleEdit.getSelectedItem();
task.content = contentEdit.getText().toString();
task.contact = (ContactTB) spinner.getSelectedItem();
task.timeStart = time_StarEdit.getText().toString();
task.timeEnd = time_EndEdit.getText().toString();
task.dateStart = date_startEdit.getText().toString();
task.dateEnd = date_endEdit.getText().toString();
task.location = (LocationsTB) location.getSelectedItem();
task.saveWithTimestamp();

setResult(Activity.RESULT_OK, new Intent().putExtra("id", task.getId()));

this.finish();
} else {

    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.setTitle(android.R.string.dialog_alert_title);
    alert.setMessage(R.string.title_is_required);
    alert.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {

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

        }

    });
    alert.show();
}
}

```

ซึ่งจากการใช้ active android จะทำให้คำสั่งไม่เยอะในการสั่งการให้ส่งค่าไปเก็บยังฐานข้อมูล
ยกตัวอย่าง

task.content = **contentEdit**.getText().toString(); คือการส่งค่าที่เป็นตัวอักษรไปเก็บในฐานข้อมูล
และ **task.title** = (Act) **titleEdit**.getSelectedItem(); คือการรับ IDของค่าที่เลือก จากตาราง Act เพื่อไปเก็บไว้ใน
คอลัมน์ title ของตาราง task (Foreign key)

ส่วนถัดมาเป็นการเรียกให้คำสั่ง DatePicker และ TimePicker ให้ทำงาน เมื่อผู้ใช้กด คำสั่งที่ใช้คือ

`@Override`

```
public void onClick(View v) {
    if (v == fromDateTxt) {
        fromDatePickerDialog.show();
    }
    if (v == fromTime) {
        fromTimePickerDialog.show();
    } else if (v == toDateTxt) {
        toDatePickerDialog.show();
    }
    if (v == toTime) {
        toTimePickerDialog.show();
    }
}
```

ส่วนต่อไปคือการสร้าง dropdown หรือภาษาแอนดรอยด์เรียกว่า Spinner โดยจะทำงานร่วมกันสองส่วนคือ ส่วนสร้าง Spinner และส่วนที่สองคือการสร้าง Adapter เพื่อกำหนดค่าที่เราต้องการให้แสดงใน dropdown ทั้งนี้ขอแสดงโค้ดให้ดูทั้งสี่ Spinner คือ กิจกรรมนัดหมาย ชื่อผู้ติดต่อ ชื่อสถานที่ และ สถานะของนัดหมาย ซึ่งใช้คำสั่ง private void SetView และทั้งสี่ Adapter ที่ควบคุมการแสดงผลข้อมูลของ Spinner ทั้งสี่ ซึ่งใช้การเปิด private class ขึ้นมา ซอร์สโค้ดที่ใช้คือ

```
private void setView() {
    Spinner sp = (Spinner) findViewById(R.id.spinner);
    sp.setOnItemSelectedListener(new ContactTBAdapter(this, contactTBs));
}
```

```

contactTBs = new ArrayList<>(ContactTB.getAll());

Spinner sp1 = (Spinner) findViewById(R.id.titleEdit);
sp1.setOnItemClickListener(new ActivityAdapter(this, acts));
acts = new ArrayList<>(Act.getAll());

Spinner sp2 = (Spinner) findViewById(R.id.locationEdit);
sp2.setOnItemClickListener(new LocationAdapter(this, locats));
locats = new ArrayList<>(LocationsTB.getAll());

Spinner sp3 = (Spinner) findViewById(R.id.status);
sp3.setOnItemClickListener(new StatusAdapter(this, statusTBs));
statusTBs = new ArrayList<>(StatusTB.getAll());

sp.setAdapter(new ContactTBAdapter(this, contactTBs));
ArrayAdapter<ContactTB> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, contactTBs);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

sp1.setAdapter(new ActivityAdapter(this, acts));
ArrayAdapter<Act> adapter1 = new ArrayAdapter<>(this, android.R.layout.simple_spinner_item, acts);
adapter1.setDropDownViewResource(android.R.layout.simple_spinner_item);

sp2.setAdapter(new LocationAdapter(this, locats));
ArrayAdapter<LocationsTB> adapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item, locats);
adapter2.setDropDownViewResource(android.R.layout.simple_spinner_item);

```



```

sp3.setAdapter(new StatusAdapter(this, statusTBs));

ArrayAdapter<StatusTB> adapter3 = new ArrayAdapter<StatusTB>(this,
android.R.layout.simple_spinner_item, statusTBs);

adapter3.setDropDownViewResource(android.R.layout.simple_spinner_item);
}

private class LocationAdapter extends ArrayAdapter<LocationsTB> implements OnItemSelectedListener
{
    public LocationAdapter(Context context, ArrayList<LocationsTB> locats) {
        super(context, R.layout.support_simple_spinner_dropdown_item, locats);
    }

    @Override
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        LocationsTB locationsTB = getItem(position);

        if (convertView == null) {
            convertView =
LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_dropdown_item, parent, false);
        }

        TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
        sp1.setText(locationsTB.locationName);

        return convertView;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LocationsTB locationsTB = getItem(position);

        if (convertView == null) {
            convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_item,
parent, false);

```

```

    }

    TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
    sp1.setText(locationsTB.locationName);

    return convertView;
}

@Override
public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
}
}

private class ActivityAdapter extends ArrayAdapter<Act> implements OnItemSelectedListener {
    public ActivityAdapter(Context context, ArrayList<Act> acts) {
        super(context, R.layout.support_simple_spinner_dropdown_item, acts);
    }

    @Override
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        Act act = getItem(position);
        if (convertView == null) {
            convertView =
                LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_dropdown_item, parent, false);
        }

        TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
        sp1.setText(act.activityName);

        return convertView;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

```

```

        Act act = getItem(position);

        if (convertView == null) {
            convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_item,
parent, false);
        }

        TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
        sp1.setText(act.activityName);

        return convertView;
    }

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
}

class ContactTBAdapter extends ArrayAdapter<ContactTB> implements OnItemSelectedListener {
    public ContactTBAdapter(Context context, ArrayList<ContactTB> contactTBs) {
        super(context, R.layout.support_simple_spinner_dropdown_item, contactTBs);
    }

    @Override
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        ContactTB contactTB = getItem(position);

        if (convertView == null) {
            convertView =
LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_dropdown_item, parent, false);
        }

        TextView sp = (TextView) convertView.findViewById(android.R.id.text1);
        sp.setText(contactTB.firstName);
    }
}

```

```

        return convertView;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ContactTB contactTB = getItem(position);

        if (convertView == null) {
            convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_item,
parent, false);
        }

        TextView sp = (TextView) convertView.findViewById(android.R.id.text1);
        sp.setText(contactTB.firstName);

        return convertView;
    }

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
}

private class StatusAdapter extends ArrayAdapter<StatusTB> implements OnItemSelectedListener {
    public StatusAdapter(Context context, ArrayList<StatusTB> statusTBs) {
        super(context, R.layout.support_simple_spinner_dropdown_item, statusTBs);
    }

    @Override
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        StatusTB statusTB = getItem(position);

```

```

        if (convertView == null) {
            convertView =
LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_dropdown_item, parent, false);
        }
        TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
        sp1.setText(statusTB.statusName);
        return convertView;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        StatusTB statusTB = getItem(position);
        if (convertView == null) {
            convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_spinner_item,
parent, false);
        }
        TextView sp1 = (TextView) convertView.findViewById(android.R.id.text1);
        sp1.setText(statusTB.statusName);
        return convertView;
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
}
}

```

activity_list.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/colorbk"
    tools:context="project.ann.pee.studentappointment.ListActivity">

    <include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />

    <ListView
        android:id="@+id/listView"
        android:scrollbars="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/toolbar_main"
        android:divider="#81F7F3"
        android:dividerHeight="0.5dp"
        android:visibility="gone"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/no_task"
```

```

    android:id="@+id/emptyLabel"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:textColor="@android:color/darker_gray"
    android:visibility="gone"/>

```

```

<android.support.design.widget.FloatingActionButton
    android:id="@+id/newFab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginRight="@dimen/activity_horizontal_margin"
    android:layout_marginBottom="@dimen/activity_vertical_margin"
/>

```

</RelativeLayout>

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ activity_list.xml สร้างขึ้นมาเพื่อแสดงรายการนัดหมายที่ได้บันทึกไว้ โดยแสดงเป็นลิสต์รายการผ่านแท็ก ListView ดังนี้

<ListView

```

    android:id="@+id/listView"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/toolbar_main"
    android:divider="#81F7F3"
    android:dividerHeight="0.5dp"
    android:visibility="gone"/>

```

ในส่วนของ TextView ที่ปรากฏในซอร์สโค้ดของ activity_list.xml จะแสดงขึ้นมาในกรณีที่ยังไม่มีรายการนัดหมายอยู่ในรายการ

<TextView

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/no_task"
    android:id="@+id/emptyLabel"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:textColor="@android:color/darker_gray"
    android:visibility="gone"/>

```

นอกจากนี้ในส่วนของหน้าแสดงลิสต์รายการนัดหมายที่แสดงรายการนัดหมายแล้ว ยังมี ปุ่มเพื่อต้องการให้ผู้ใช้งานกดเพิ่มรายการนัดหมาย ซึ่งในที่นี้จะใช้เป็น Float button โดยใช้คำสั่ง

<android.support.design.widget.FloatingActionButton

```

    android:id="@+id/newFab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginRight="@dimen/activity_horizontal_margin"
    android:layout_marginBottom="@dimen/activity_vertical_margin"
/>

```

ListActivity.java

```
package project.ann.pee.studentappointment;
```

```
import android.app.Activity;
```



```

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.Adapter;
import android.widget.ArrayAdapter;
import android.widget.ImageButton;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.TextView;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import java.util.ArrayList;

```

```

public class ListActivity extends BaseActivity {

```

```

    private static final int NEW_TASK = 1;
    private static final int SHOW_TASK = 2;
    private static final int Refresh = 3;
    private ArrayList<Task> tasks;
    private ListView listView;

```

```

private TextView emptyLabel;

private ScrollView scrollView;

@Override

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_list);
    setDrawer(false);
    setTitle(R.string.tasks);

    listView = (ListView) findViewById(R.id.listView);
    emptyLabel = (TextView) findViewById(R.id.emptyLabel);
    scrollView = (ScrollView) findViewById(R.id.scrollView);

    FloatingActionButton newFab = (FloatingActionButton) findViewById(R.id.newFab);
    newFab.setImageDrawable(buildDrawable(MaterialDesignIconic.Icon.gmi_plus));
    newFab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(ListActivity.this, FormActivity.class);
            startActivityForResult(intent, NEW_TASK);
        }
    });
    setContentView();
}

@Override

public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, Refresh, R.string.refresh,
buildDrawable(MaterialDesignIconic.Icon.gmi_refresh));
}

```

```

        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        switch (Refresh) {

            case Refresh:

                Refresh();

                break;

        }

        return super.onOptionsItemSelected(item);
    }

    private void Refresh() {

        Intent i = getIntent();

        finish();

        startActivity(i);

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        if (resultCode == Activity.RESULT_OK) {

            Bundle extras = data.getExtras();

            switch (requestCode) {

                case NEW_TASK:

                    if (extras != null && extras.getLong("id", 0) > 0)

                        setView();

                    break;

                case SHOW_TASK:

                    if (extras != null && extras.getBoolean("refreshNeeded", false))

```

```

        setView();
        break;
    }
}

}

private void setView() {
    tasks = new ArrayList<>(Task.getAll());
    if (tasks.isEmpty()) {
        listView.setVisibility(View.GONE);
        emptyLabel.setVisibility(View.VISIBLE);
    } else {
        emptyLabel.setVisibility(View.GONE);
        listView.setVisibility(View.VISIBLE);
        listView.setAdapter(new TasksAdapter(this, tasks));
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent intent = new Intent(ListActivity.this, ShowActivity.class);
                intent.putExtra("id", tasks.get(position).getId());
                startActivityForResult(intent, SHOW_TASK);
            }
        });
    }
}

class TasksAdapter extends ArrayAdapter<Task> {

```

```

public TasksAdapter(Context context, ArrayList<Task> acts) {
    super(context, 0, acts);
}

@Override

public View getView(int position, View convertView, ViewGroup parent) {
    Task task = getItem(position);

    if (convertView == null) {
        convertView =
LayoutInflater.from(getContext()).inflate(android.R.layout.simple_expandable_list_item_2, parent, false);
    }

    TextView tv = (TextView) convertView.findViewById(android.R.id.text1);
    tv.setText(task.title.activityName);

    TextView t = (TextView) convertView.findViewById(android.R.id.text2);
    t.setText(task.dateStart);

    return convertView;
}
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ ListActivity.java สร้างขึ้นมาเพื่อสั่งงานให้แอปพลิเคชันแสดงลิสต์รายการนัดหมายที่ได้บันทึกไว้ โดยทำงานร่วมกับหน้าแสดงผล xml คือ activity_list.xml ซึ่งจะขอกล่าวในส่วนที่แตกต่างจากจาวาคลาสก่อนนี้ที่ได้อธิบายไปแล้ว โดยเริ่มจากการสั่งงานให้ปุ่มม(Float button) สามารถทำงานได้ ซอร์สโค้ดที่ใช้คือ

```

FloatingActionButton newFab = (FloatingActionButton) findViewById(R.id.newFab);
newFab.setImageDrawable(buildDrawable(MaterialDesignIconic.Icon.gmi_plus));
newFab.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

```

```

Intent intent = new Intent(ListActivity.this, FormActivity.class);
startActivityForResult(intent, NEW_TASK);
}
});

```

ส่วนต่อไปคือการเรียกข้อมูลจากฐานข้อมูลมาแสดงโดยเขียนโค้ดสองส่วนด้วยกันคือ ส่วนของ `setView` เพื่อติดต่อกับ `ListView` และส่วนที่สองคือส่วนของ `Adapter` โดยเขียนเป็น `private class` ขึ้นมา เพื่อกำหนดเอาเฉพาะข้อมูลที่เราต้องการให้แสดงผล ดังซอร์สโค้ดต่อไปนี้

```

private void setView() {
    tasks = new ArrayList<>(Task.getAll());
    if (tasks.isEmpty()) {
        listView.setVisibility(View.GONE);
        emptyLabel.setVisibility(View.VISIBLE);
    } else {
        emptyLabel.setVisibility(View.GONE);
        listView.setVisibility(View.VISIBLE);
        listView.setAdapter(new TasksAdapter(this, tasks));
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent intent = new Intent(ListActivity.this, ShowActivity.class);
                intent.putExtra("id", tasks.get(position).getId());
                startActivityForResult(intent, SHOW_TASK);
            }
        });
    }
}
}

```

```

class TasksAdapter extends ArrayAdapter<Task> {
    public TasksAdapter(Context context, ArrayList<Task> acts) {
        super(context, 0, acts);
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        Task task = getItem(position);
        if (convertView == null) {
            convertView =
                LayoutInflater.from(getContext()).inflate(android.R.layout.simple_expandable_list_item_2, parent, false);
        }
        TextView tv = (TextView) convertView.findViewById(android.R.id.text1);
        tv.setText(task.title.activityName);

        TextView t = (TextView) convertView.findViewById(android.R.id.text2);
        t.setText(task.dateStart);
        return convertView;
    }
}

```

ซึ่งลักษณะโค้ดจะใกล้เคียงกันอีกกับ 3 ไฟล์ ได้แก่ ActList.java LocationList.java ContactList.java อย่างที่กล่าวไปข้างต้นว่าใช้คำสั่งเหมือนกัน ต่างกันเพียงชื่อคลาสของฐานข้อมูลและชื่อคอลัมน์ในการเรียกข้อมูลมาแสดง ซึ่งจะอยู่ในส่วนของ Adapter

ActList.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import java.util.ArrayList;

public class ActList extends BaseActivity {

    private static final int NEW_TASK = 1;
    private static final int SHOW_TASK = 2;
    private ArrayList<Act> tasks;

    private ListView listView;
    private TextView emptyLabel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```



```

super.onCreate(savedInstanceState);

setContentView(R.layout.act_list);

setDrawer(false);

setTitle(R.string.act);

listView = (ListView) findViewById(R.id.listView);

emptyLabel = (TextView) findViewById(R.id.emptyLabel);

FloatingActionButton newFab = (FloatingActionButton) findViewById(R.id.newFab);

newFab.setImageDrawable(buildDrawable(MaterialDesignIconic.Icon.gmi_plus));

newFab.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        Intent intent = new Intent(ActList.this, FormActDefault.class);

        startActivityForResult(intent, NEW_TASK);

    }

});

setView();

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (resultCode == Activity.RESULT_OK) {

        Bundle extras = data.getExtras();

        switch (requestCode) {

            case NEW_TASK:

                if (extras != null && extras.getLong("id", 0) > 0)

                    setView();

                break;

            case SHOW_TASK:

```

```

        if (extras != null && extras.getBoolean("refreshNeeded", false))
            setView();
        break;
    }
}

private void setView() {
    tasks = new ArrayList<Act>(Act.getAll());
    if (tasks.isEmpty()) {
        listView.setVisibility(View.GONE);
        emptyLabel.setVisibility(View.VISIBLE);
    } else {
        emptyLabel.setVisibility(View.GONE);
        listView.setVisibility(View.VISIBLE);
        listView.setAdapter(new ActAdapter(this, tasks));
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent intent = new Intent(ActList.this, ShowAct.class);
                intent.putExtra("id", tasks.get(position).getId());
                startActivityForResult(intent, SHOW_TASK);
            }
        });
    }
}
}

```

```

class ActAdapter extends ArrayAdapter<Act> {

    public ActAdapter(Context context, ArrayList<Act> acts) {

        super(context, 0, acts);

    }

    @Override

    public View getView(int position, View convertView, ViewGroup parent) {

        Act act = getItem(position);

        if (convertView == null) {

            convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_list_item_1, parent,
false);

        }

        TextView tv = (TextView) convertView.findViewById(android.R.id.text1);

        tv.setText(act.activityName);

        return convertView;

    }

}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ ActList.java สร้างขึ้นมาเพื่อสั่งงานให้แอปพลิเคชันแสดงรายชื่อกิจกรรมนัดหมายโดยทำงานร่วมกันกับหน้าแสดงผลคือ act_list.xml

act_list.xml

```

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical"

    android:background="@color/colorbk"

```

```

tools:context="org.usablelabs.duedo.ListActivity">
<include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />
<ListView
    android:id="@+id/listView"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/toolbar_main"
    android:visibility="gone"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/no_act"
    android:id="@+id/emptyLabel"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:textColor="@android:color/darker_gray"
    android:visibility="gone"/>
<android.support.design.widget.FloatingActionButton
    android:id="@+id/newFab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginRight="@dimen/activity_horizontal_margin"
    android:layout_marginBottom="@dimen/activity_vertical_margin" />
</RelativeLayout>

```

LocationList.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import java.util.ArrayList;

public class LocationList extends BaseActivity {

    private static final int NEW_TASK = 1;
    private static final int SHOW_TASK = 2;
    private ArrayList<LocationsTB> tasks;
    private ListView listView;
    private TextView emptyLabel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);

setContentView(R.layout.location_list);

setDrawer(false);

setTitle(R.string.locations);


listView = (ListView) findViewById(R.id.listView);

emptyLabel = (TextView) findViewById(R.id.emptyLabel);

FloatingActionButton newFab = (FloatingActionButton) findViewById(R.id.newFab);

newFab.setImageDrawable(buildDrawable(MaterialDesignIconic.Icon.gmi_plus));

newFab.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        Intent intent = new Intent(LocationList.this, FormLocationDefault.class);

        startActivityForResult(intent, NEW_TASK);

    }

});

setView();
}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (resultCode == Activity.RESULT_OK) {

        Bundle extras = data.getExtras();

        switch (requestCode) {

            case NEW_TASK:

                if (extras != null && extras.getLong("id", 0) > 0)

                    setView();

                break;

            case SHOW_TASK:

                if (extras != null && extras.getBoolean("refreshNeeded", false))

```

```

        setView();
        break;
    }
}

}

private void setView() {
    tasks = new ArrayList<>(LocationsTB.getAll());
    if (tasks.isEmpty()) {
        listView.setVisibility(View.GONE);
        emptyLabel.setVisibility(View.VISIBLE);
    } else {
        emptyLabel.setVisibility(View.GONE);
        listView.setVisibility(View.VISIBLE);
        listView.setAdapter(new LocationAdapter(this, tasks));
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent intent = new Intent(LocationList.this, ShowLocation.class);
                intent.putExtra("id", tasks.get(position).getId());
                startActivityForResult(intent, SHOW_TASK);
            }
        });
    }
}
}
}

```

```

class LocationAdapter extends ArrayAdapter<LocationsTB> {
    public LocationAdapter(Context context, ArrayList<LocationsTB> acts) {

```

```

    super(context, 0, acts);
}

@Override

public View getView(int position, View convertView, ViewGroup parent) {
    LocationsTB locationsTB = getItem(position);

    if (convertView == null) {
        convertView = LayoutInflater.from(getContext()).inflate(android.R.layout.simple_list_item_1, parent,
false);
    }

    TextView tv = (TextView) convertView.findViewById(android.R.id.text1);
    tv.setText(locationsTB.locationName);

    return convertView;
}
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ LocationList.java สร้างขึ้นมาเพื่อสั่งงานให้แอปพลิเคชัน
แสดงรายชื่อสถานที่ โดยทำงานร่วมกันกับหน้าแสดงผลคือ location_list.xml

location_list.xml

```

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical"

    tools:context="project.ann.pee.studentappointment.ListActivity">

```



```

<include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />
<ListView
    android:id="@+id/listView"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/toolbar_main"
    android:visibility="gone"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/no_location"
    android:id="@+id/emptyLabel"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:textColor="@android:color/darker_gray"
    android:visibility="gone"/>
<android.support.design.widget.FloatingActionButton
    android:id="@+id/newFab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginRight="@dimen/activity_horizontal_margin"
    android:layout_marginBottom="@dimen/activity_vertical_margin" />
</RelativeLayout>

```

ContactList.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import java.util.ArrayList;

public class ContactList extends BaseActivity {

    private static final int NEW_TASK = 1;
    private static final int SHOW_TASK = 2;
    private ArrayList<ContactTB> tasks;
    private ListView listView;
    private TextView emptyLabel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.contact_list);

setDrawer(false);

setTitle(R.string.contact_person);


listView = (ListView) findViewById(R.id.listView);

emptyLabel = (TextView) findViewById(R.id.emptyLabel);

FloatingActionButton newFab = (FloatingActionButton) findViewById(R.id.newFab);

newFab.setImageDrawable(buildDrawable(MaterialDesignIconic.Icon.gmi_plus));

newFab.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        Intent intent = new Intent(ContactList.this, FormContactDefault.class);

        startActivityForResult(intent, NEW_TASK);

    }

});

setView();
}

```

```

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (resultCode == Activity.RESULT_OK) {

        Bundle extras = data.getExtras();

        switch (requestCode) {

            case NEW_TASK:

                if (extras != null && extras.getLong("id", 0) > 0)

                    setView();

                break;

            case SHOW_TASK:

                if (extras != null && extras.getBoolean("refreshNeeded", false))

```

```

        setView();
        break;
    }
}
}

private void setView() {
    tasks = new ArrayList<>(ContactTB.getAll());
    if (tasks.isEmpty()) {
        listView.setVisibility(View.GONE);
        emptyLabel.setVisibility(View.VISIBLE);
    } else {
        emptyLabel.setVisibility(View.GONE);
        listView.setVisibility(View.VISIBLE);
        listView.setAdapter(new ContactAdapter(this, tasks));
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Intent intent = new Intent(ContactList.this, ShowContact.class);
                intent.putExtra("id", tasks.get(position).getId());
                startActivityForResult(intent, SHOW_TASK);
            }
        });
    }
}
}
}

```

```

class ContactAdapter extends ArrayAdapter<ContactTB> {

    public ContactAdapter(Context context, ArrayList<ContactTB> acts) {
        super(context, 0, acts);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ContactTB contactTB = getItem(position);
        if (convertView == null) {
            convertView =
                LayoutInflater.from(getContext()).inflate(android.R.layout.simple_expandable_list_item_2, parent, false);
        }
        TextView tv = (TextView) convertView.findViewById(android.R.id.text1);
        tv.setText(contactTB.firstName);

        TextView tv1 = (TextView) convertView.findViewById(android.R.id.text2);
        tv1.setText(contactTB.lastName);

        return convertView;
    }
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ ContactList.java สร้างขึ้นมาเพื่อสั่งงานให้แอปพลิเคชัน
แสดงรายชื่อผู้ติดต่อ โดยทำงานร่วมกันกับหน้าแสดงผลคือ contact_list.xml

Contact_list.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="project.ann.pee.studentappointment.ListActivity">

    <include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />

    <ListView
        android:id="@+id/listView"
        android:scrollbars="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/toolbar_main"
        android:visibility="gone"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/no_contact"
        android:id="@+id/emptyLabel"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"
        android:textColor="@android:color/darker_gray">
```

```
        android:visibility="gone"/>
<android.support.design.widget.FloatingActionButton
    android:id="@+id/newFab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginRight="@dimen/activity_horizontal_margin"
    android:layout_marginBottom="@dimen/activity_vertical_margin" />

</RelativeLayout>
```

ต่อไปนี้จะเป็นการอธิบายซอร์สโค้ดของการเพิ่มและแก้ไขชื่อกิจกรรมนัดหมาย ชื่อผู้ติดต่อ และชื่อสถานที่ ซึ่งจะกล่าวถึงการเขียนหน้าแสดงผล (xml) และการสั่งการทำงานและรับส่งข้อมูลไปยังฐานข้อมูลโดยจาวาคลาส(java) ลักษณะของซอร์สโค้ดของทั้ง 3 กิจกรรม จะคล้ายคลึงกัน ต่างกันเพียงแค่ text และ ชื่อฐานข้อมูล ดังนั้นจึงขออธิบายเพียงหนึ่งกิจกรรมประกอบไปด้วย FormActDefault.java และหน้าแสดงผล act_form.xml

Act_form.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@color/colorbk"

    tools:context="project.ann.pee.studentappointment.FormActDefault">

    <include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />


    <LinearLayout

        android:orientation="vertical"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:paddingBottom="@dimen/activity_vertical_margin"

        android:paddingLeft="@dimen/activity_horizontal_margin"

        android:paddingRight="@dimen/activity_horizontal_margin"

        android:paddingTop="@dimen/activity_vertical_margin"

        android:layout_below="@+id/toolbar_main"

        android:layout_gravity="center_horizontal">


        <android.support.design.widget.TextInputLayout

            android:layout_width="match_parent"

            android:layout_height="wrap_content">


            <EditText

                android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"

        android:ems="10"

        android:id="@+id/actEdit"

        android:layout_gravity="center_horizontal"

        android:hint="@string/actDefault" />

</android.support.design.widget.TextInputLayout>

</LinearLayout>

</RelativeLayout>

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ act_form.xml สร้างขึ้นมาเพื่อแสดงผลหน้าเพิ่มชื่อกิจกรรม
นัดหมาย โดยทำงานร่วมกันกับ FormActDefault.java ซึ่งใช้ EditText ในการรับข้อมูลจากผู้ใช้และเขียนภายใต้
แท็ก <LinearLayout และปิดด้วย /LinearLayout> ดังซอร์สโค้ดต่อไปนี้

```

<LinearLayout

    android:orientation="vertical"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:paddingBottom="@dimen/activity_vertical_margin"

    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:layout_below="@+id/toolbar_main"

    android:layout_gravity="center_horizontal">

    <android.support.design.widget.TextInputLayout

        android:layout_width="match_parent"

        android:layout_height="wrap_content">

```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:id="@+id/actEdit"
    android:layout_gravity="center_horizontal"
    android:hint="@string/actDefault" />
</android.support.design.widget.TextInputLayout>
```

FormActDefault.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;

public class FormActDefault extends BaseActivity {

    private static final int MenuItem_SaveID = 1;

    private Act act = null;
    private EditText actEdit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.act_form);

        setDrawer(true);

        actEdit = (EditText) findViewById(R.id.actEdit);

        long id = getIntent().getLongExtra("id", 0);

```

```

    if (id == 0) {
        setTitle(R.string.new_activity);
    } else {
        setTitle(R.string.edit_activity);
        act = Act.load(Act.class, id);
        if (act != null) {
            actEdit.setText(act.activityName);

        } else {
            finish();
        }
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, MenuItem_SaveID, R.string.save,
buildDrawable(MaterialDesignIconic.Icon.gmi_save));

    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            if (isEdited()) {
                AlertDialog.Builder alert = new AlertDialog.Builder(this);
                alert.setTitle(android.R.string.dialog_alert_title);
                alert.setMessage(R.string.unsaved_exit_alert);
                alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
                    @Override

```

```

        public void onClick(DialogInterface dialog, int which) {
            onBackPressed();
        }
    });

    alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });

    alert.show();

    return true;
}

break;

case MenuItem_SaveID:
    save();

    break;
}

return super.onOptionsItemSelected(item);
}

private boolean isEdited() {
    if (act == null)

        return actEdit.getText().length() > 0;

    else

        return !act.activityName.equals(actEdit.getText().toString());
}

private void save() {
    if (actEdit.getText().length() > 0) {

        if (act == null)

            act = new Act();
    }
}

```

```

    act.activityName = actEdit.getText().toString();

    act.saveWithTimestamp();

    setResult(Activity.RESULT_OK, new Intent().putExtra("id", act.getId()));

    this.finish();
} else {

    AlertDialog.Builder alert = new AlertDialog.Builder(this);

    alert.setTitle(android.R.string.dialog_alert_title);

    alert.setMessage(R.string.activity_is_required);

    alert.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {

        @Override

        public void onClick(DialogInterface dialog, int which) {

            }

        });

    alert.show();

}

}

}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ FormActDefault.java สร้างขึ้นมาเพื่อให้แอปพลิเคชันสามารถเพิ่มรายชื่อกิจกรรมนัดหมาย โดยทำงานร่วมกันกับหน้าแสดงผล act_form.xml ซึ่งเริ่มจากการ import library เพื่อให้คำสั่งต่าง ๆ ทำงานได้อย่างสมบูรณ์

```

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.support.design.widget.FloatingActionButton;

```

```

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageButton;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;

```

ส่วนต่อมาคือการ ประกาศตัวแปรและเรียกหน้าแสดงผล act_form.xml โดยใช้คำสั่ง

```

private static final int MenuItem_SaveID = 1;

private Act act = null;

private EditText actEdit;

```

@Override

```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.act_form);

    setDrawer(true) ;

```

จากนั้นเป็นการเชื่อมต่อตัวแปรที่ประกาศไว้ให้สัมพันธ์กับ EditText ที่ได้สร้างไว้ในหน้า xml ตามคำสั่งนี้

```

actEdit = (EditText) findViewById(R.id.actEdit);

```

และคำสั่งต่อไปนี่คือคำสั่งที่สั่งให้ icon บนเมนูบาร์สามารถทำงานได้ ตามที่อธิบายไว้ข้างต้นแล้ว

@Override

```

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

        case android.R.id.home:

            if (isEdited()) {

                AlertDialog.Builder alert = new AlertDialog.Builder(this);

                alert.setTitle(android.R.string.dialog_alert_title);

```



```

alert.setMessage(R.string.unsaved_exit_alert);

alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {

    @Override

    public void onClick(DialogInterface dialog, int which) {

        onBackPressed();

    }

});

alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {

    @Override

    public void onClick(DialogInterface dialog, int which) {

    }

});

alert.show();

return true;

}

break;

case MenuItem_SaveID:

    save();

    break;

}

return super.onOptionsItemSelected(item);

}

```

และคำสั่งนี้ก็ทำงานเช่นเดียวกันกับ `FormActivity.java` ตามที่เคยอธิบายไว้ข้างต้น คือจะทำงานตอนผู้ใช้แก้ไขชื่อกิจกรรมนี้ค้หมาย

```

private boolean isEdited() {

    if (act == null)

        return actEdit.getText().length() > 0;

    else

```

```

        return !act.activityName.equals(actEdit.getText().toString());
    }

```

ฟังก์ชันบันทึก save คือการรับค่าจาก EditText ที่ผู้ใช้ป้อนเข้ามา ส่งไปเก็บที่ฐานข้อมูล

```

private void save() {
    if (actEdit.getText().length() > 0) {
        if (act == null)
            act = new Act();
        act.activityName = actEdit.getText().toString();

        act.saveWithTimestamp();
        setResult(Activity.RESULT_OK, new Intent().putExtra("id", act.getId()));
        this.finish();
    } else {
        AlertDialog.Builder alert = new AlertDialog.Builder(this);
        alert.setTitle(android.R.string.dialog_alert_title);
        alert.setMessage(R.string.activity_is_required);
        alert.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        });
        alert.show();
    }
}

```

location.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    tools:context="project.ann.pee.studentappointment.FormActDefault">
```

```
    <include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />
```

```
<LinearLayout
```

```
    android:orientation="vertical"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    android:layout_below="@+id/toolbar_main"
```

```
    android:layout_gravity="center_horizontal">
```

```
    <android.support.design.widget.TextInputLayout
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content">
```

```
    <EditText
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
android:ems="10"  
android:id="@+id/locationEdit"  
android:layout_gravity="center_horizontal"  
android:hint="@string/locationDefault" />
```

```
</android.support.design.widget.TextInputLayout>
```

```
</LinearLayout>
```

```
</RelativeLayout>
```

FormLocationDefault.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.EditText;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;

public class FormLocationDefault extends BaseActivity {

    private LocationsTB locationsTB = null;

    private static final int MenuItem_SaveID = 1;
    private static final int MenuItem_DeleteID = 2;

    private LocationsTB location = null;
    private EditText locationEdit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.location_form);
        setDrawer(true);

        locationEdit = (EditText) findViewById(R.id.locationEdit);
    }

```

```

long id = getIntent().getLongExtra("id", 0);

if (id == 0) {
    setTitle(R.string.new_location);
} else {
    setTitle(R.string.edit_location);

    location = LocationsTB.load(LocationsTB.class, id);

    if (location != null) {
        locationEdit.setText(location.locationName);

    } else {
        finish();
    }
}
}

@Override

public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, MenuItem_SaveID, R.string.save,
buildDrawable(MaterialDesignIconic.Icon.gmi_save));

    return true;
}

@Override

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {

        case android.R.id.home:
            if (isEdited()) {
                AlertDialog.Builder alert = new AlertDialog.Builder(this);
                alert.setTitle(android.R.string.dialog_alert_title);
            }
        }
    }
}

```

```

        alert.setMessage(R.string.unsaved_exit_alert);

        alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {

            @Override

            public void onClick(DialogInterface dialog, int which) {

                onBackPressed();

            }

        });

        alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {

            @Override

            public void onClick(DialogInterface dialog, int which) {

            }

        });

        alert.show();

        return true;

    }

    break;

    case MenuItem_SaveID:

        save();

        break;

    }

    return super.onOptionsItemSelected(item);

}

private boolean isEdited() {

    if (location == null)

        return locationEdit.getText().length() > 0;

    else

        return !location.locationName.equals(locationEdit.getText().toString());

}

```

```

private void save() {
    if (locationEdit.getText().length() > 0) {
        if (location == null)
            location = new LocationsTB();
        location.locationName = locationEdit.getText().toString();

        location.saveWithTimestamp();
        setResult(Activity.RESULT_OK, new Intent().putExtra("id", location.getId()));
        this.finish();
    } else {
        AlertDialog.Builder alert = new AlertDialog.Builder(this);
        alert.setTitle(android.R.string.dialog_alert_title);
        alert.setMessage(R.string.location_is_required);
        alert.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        });
        alert.show();
    }
}
}

```

contact_form.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="project.ann.pee.studentappointment.FormActDefault">

    <include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />

    <LinearLayout

        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:layout_below="@+id/toolbar_main"
        android:layout_gravity="center_horizontal">

        <android.support.design.widget.TextInputLayout

            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText

                android:layout_width="match_parent"
                android:layout_height="wrap_content"
```

```

        android:ems="10"

        android:id="@+id/firstNameEdit"

        android:layout_gravity="center_horizontal"

        android:hint="@string/firsNameDefault" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

    android:layout_width="match_parent"

    android:layout_height="wrap_content">

    <EditText

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:ems="10"

        android:id="@+id/lastNameEdit"

        android:layout_gravity="center_horizontal"

        android:hint="@string/lastNameDefault" />

</android.support.design.widget.TextInputLayout>

</LinearLayout>

</RelativeLayout>

```

FormContactDefault.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;

import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;

public class FormContactDefault extends BaseActivity {

    private static final int MenuItem_SaveID = 1;
    private ContactTB contact = null;
    private EditText firstNameEdit;
    private EditText lastNameEdit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.contact_form);
        setDrawer(true);
    }

```

```

firstNameEdit = (EditText) findViewById(R.id.firstNameEdit);
lastNameEdit = (EditText) findViewById(R.id.lastNameEdit);
long id = getIntent().getLongExtra("id", 0);
if (id == 0) {
    setTitle(R.string.new_contact);
} else {
    setTitle(R.string.edit_contact);
    contact = ContactTB.load(ContactTB.class, id);
    if (contact != null) {
        firstNameEdit.setText(contact.firstName);
        lastNameEdit.setText(contact.lastName);

    } else {
        finish();
    }
}
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, MenuItem_SaveID, R.string.save,
buildDrawable(MaterialDesignIconic.Icon.gmi_save));

    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:

```

```

    if (isEdited()) {
        AlertDialog.Builder alert = new AlertDialog.Builder(this);
        alert.setTitle(android.R.string.dialog_alert_title);
        alert.setMessage(R.string.unsaved_exit_alert);
        alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                onBackPressed();
            }
        });
        alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        });
        alert.show();
        return true;
    }

    break;

    case MenuItem_SaveID:
        save();
        break;
}

return super.onOptionsItemSelected(item);
}

private boolean isEdited() {
    if (contact == null)

        return firstNameEdit.getText().length() > 0 ||

```

```

        lastNameEdit.getText().length() > 0;
    else

        return !contact.firstName.equals(firstNameEdit.getText().toString())
            || !contact.lastName.equals(lastNameEdit.getText().toString());
    }

    private void save() {
        if (firstNameEdit.getText().length() > 0) {
            if (contact == null)
                contact = new ContactTB();

            contact.firstName = firstNameEdit.getText().toString();
            contact.lastName = lastNameEdit.getText().toString();
            contact.saveWithTimestamp();

            setResult(Activity.RESULT_OK, new Intent().putExtra("id", contact.getId()));
            this.finish();
        } else {
            AlertDialog.Builder alert = new AlertDialog.Builder(this);
            alert.setTitle(android.R.string.dialog_alert_title);
            alert.setMessage(R.string.contact_is_required);
            alert.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                }
            });
            alert.show();
        }
    }
}

```

ซอร์สโค้ดที่จะอธิบายต่อไปนี้เป็นารแสดงรายละเอียดของนัดหมาย ไม่ว่าจะเป็นชื่อกิจกรรมนัดหมาย รายละเอียดเพิ่มเติม วันที่นัดหมาย เวลานั้นนัดหมาย ชื่อผู้ติดต่อ สถานที่นัดหมาย วันที่สิ้นสุดนัดหมาย เวลาสิ้นสุด และสถานะของนัดหมาย ซึ่งมีสองส่วนที่ทำงานร่วมกัน คือ จาวาคลาส(java)และหน้าแสดงผล(xml) นั่นก็คือ ShowActivity.java และ activity_show.xml โดยลักษณะของโค้ดจะใกล้เคียงกันกับการแสดงชื่อกิจกรรมนัดหมาย (act_show.xml และ ShowAct.java) ชื่อผู้ติดต่อ(contact_show.xml และ ShowContact.java) ชื่อสถานที่ (location_show.xml และ ShowLocation.java) ซึ่งผู้ใช้ได้เพิ่มไว้ก่อนหน้าเพื่อไว้เลือกใช้งาน ลักษณะของซอร์สโค้ดจะแตกต่างกันเพียงชื่อตัวแปรและชื่อฐานข้อมูล รวมถึงหน้าแสดงผลที่เรียกมาใช้งาน ส่วนคำสั่งต่าง ๆ จะมีลักษณะเหมือนกัน ดังนั้นจึงขออธิบายซอร์สโค้ดเพียง ShowActivity.java และ activity_show.xml

ShowActivity.java

package project.ann.pee.studentappointment;

import android.app.Activity;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.AlertDialog;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.TextView;

import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;

import org.w3c.dom.Text;

public class ShowActivity **extends** BaseActivity {

private Task **task** = null;

private ContactTB **contactTB**;

private LocationsTB **locationsTB**;

private StatusTB **statusTB**;

private TextView **titleView**;

```

private TextView contentView;
private TextView contactView;
private TextView time_startView;
private TextView time_endView;
private TextView date_startView;
private TextView date_endView;
private TextView location;
private TextView statusShow;

private static final int MenuItem_EditID = 1;
private static final int MenuItem_DeleteID = 2;
private static final int EDIT_TASK = 10;

```

@Override

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_show);
    setDrawer(true);
    setTitle(R.string.tasks);

    titleView = (TextView) findViewById(R.id.titleView);
    contentView = (TextView) findViewById(R.id.contentView);
    contactView = (TextView) findViewById(R.id.contactView);
    time_startView = (TextView) findViewById(R.id.timeStart);
    time_endView = (TextView) findViewById(R.id.timeEnd);
    date_startView = (TextView) findViewById(R.id.dateStart);
    date_endView = (TextView) findViewById(R.id.dateEnd);
    location = (TextView) findViewById(R.id.location);
    statusShow = (TextView) findViewById(R.id.statusShow);

```



```

        long id = getIntent().getLongExtra("id", 0);
        setView(id);
    }

    private void setView(long id) {
        if (id > 0)
            task = Task.load(Task.class, id);
        contactTB = ContactTB.load(ContactTB.class, id);
        locationsTB = LocationsTB.load(LocationsTB.class, id);
        statusTB = StatusTB.load(StatusTB.class, id);
        if (task != null) {
            titleView.setText(task.title.activityName);
            contentView.setText(task.content);
            contactView.setText(task.contact.firstName);
            time_startView.setText(task.timeStart);
            time_endView.setText(task.timeEnd);
            date_startView.setText(task.dateStart);
            date_endView.setText(task.dateEnd);
            location.setText(task.location.locationName);
            statusShow.setText(task.status.statusName);

        } else {
            finish();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        addMenuItem(menu, MenuItem_EditID, R.string.edit,
        buildDrawable(MaterialDesignIconic.Icon.gmi_edit));
    }

```

```

        addItem(menu, MenuItem_DeleteID, R.string.delete,
buildDrawable(MaterialDesignIconic.Icon.gmi_delete));

        return true;
    }

```

@Override

```

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case MenuItem_EditID:
            Intent intent = new Intent(this, FormActivity.class);
            intent.putExtra("id", task.getId());
            startActivityForResult(intent, EDIT_TASK);
            break;
        case MenuItem_DeleteID:
            AlertDialog.Builder alert = new AlertDialog.Builder(this);
            alert.setTitle(android.R.string.dialog_alert_title);
            alert.setMessage(R.string.are_you_sure);
            alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    task.delete();
                    setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));
                    finish();
                }
            });
            alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {

```

```

    });
    alert.show();
    break;
}

return super.onOptionsItemSelected(item);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == Activity.RESULT_OK) {
        Bundle extras = data.getExtras();
        switch (requestCode) {
            case EDIT_TASK:
                if (extras != null && extras.getLong("id", 0) > 0) {
                    setView(task.getId());
                    setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));
                }
                break;
            }
        }
    }
}
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ ShowActivity.java สร้างขึ้นมาเพื่อให้แอปพลิเคชันแสดงรายละเอียดของกิจกรรมนัดหมาย โดยทำงานร่วมกับหน้าแสดงผล activity_show.xml ซึ่งในส่วนของจาวา คลาสสามารถอธิบายได้ดังนี้

package project.ann.pee.studentappointment; คือชื่อของ package แอปพลิเคชัน
 ส่วนถัดมาคือการ import library เพื่อให้คำสั่งต่าง ๆ ภายในคลาสสามารถทำงานได้

```

import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;

```

public class ShowActivity **extends** BaseActivity ในส่วนนี้คือการประกาศคลาส และ เรียกใช้คำสั่งต่าง ๆ ที่เราเขียนไว้ใน BaseActivity BaseActivity ถูกเขียนขึ้นมาเพื่อกำหนดการทำงานหลัก ๆ ที่หน้าต่าง ๆ ต้องการใช้งาน โดยไม่ต้องเขียนคำสั่ง ๆ ซ้ำๆ ทุกหน้า

ส่วนต่อไปคือการประกาศตัวแปรและค่าต่าง ๆ

```

private Task task = null;
private ContactTB contactTB;
private LocationsTB locationsTB;
private StatusTB statusTB;
private TextView titleView;
private TextView contentView;
private TextView contactView;
private TextView time_startView;
private TextView time_endView;
private TextView date_startView;
private TextView date_endView;
private TextView location;
private TextView statusShow;
private static final int MenuItem_EditID = 1;

```

```
private static final int MenuItem_DeleteID = 2;
```

```
private static final int EDIT_TASK = 10;
```

ในส่วนนี้คือการเรียกใช้หน้าแสดงผล xml นั่นก็คือหน้า `activity_show` และการสร้างความสัมพันธ์ระหว่างตัวแปรที่เราประกาศไว้ข้างต้นกับ `TextView` ในหน้าแสดงผล เพื่อให้ตัวแปรและคำสั่งรู้จักกัน

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_show);
```

```
    setDrawer(true);
```

```
    setTitle(R.string.tasks);
```

```
    titleView = (TextView) findViewById(R.id.titleView);
```

```
    contentView = (TextView) findViewById(R.id.contentView);
```

```
    contactView = (TextView) findViewById(R.id.contactView);
```

```
    time_startView = (TextView) findViewById(R.id.timeStart);
```

```
    time_endView = (TextView) findViewById(R.id.timeEnd);
```

```
    date_startView = (TextView) findViewById(R.id.dateStart);
```

```
    date_endView = (TextView) findViewById(R.id.dateEnd);
```

```
    location = (TextView) findViewById(R.id.location);
```

```
    statusShow = (TextView) findViewById(R.id.statusShow);
```

```
    long id = getIntent().getLongExtra("id", 0);
```

```
    setView(id);
```

```
}
```

ส่วนต่อไปคือการเรียกข้อมูลจากฐานข้อมูลมาแสดงผลทางหน้าจอ

```
private void setView(long id) {
    if (id > 0)
        task = Task.load(Task.class, id);
        contactTB = ContactTB.load(ContactTB.class, id);
        locationsTB = LocationsTB.load(LocationsTB.class, id);
        statusTB = StatusTB.load(StatusTB.class, id);

    if (task != null) {
        titleView.setText(task.title.activityName);
        contentView.setText(task.content);
        contactView.setText(task.contact.firstName);
        time_startView.setText(task.timeStart);
        time_endView.setText(task.timeEnd);
        date_startView.setText(task.dateStart);
        date_endView.setText(task.dateEnd);
        location.setText(task.location.locationName);
        statusShow.setText(task.status.statusName);
    } else {
        finish();
    }
}
```

จากซอร์สโค้ดสามารถสังเกตได้ว่าการเรียกข้อมูลจากฐานข้อมูลใช้ภาษาการเขียนที่ค่อนข้างง่าย ไม่เยอะ เป็นเพราะการใช้ active android เข้ามาช่วย หากเขียนด้วยภาษา SQL จะทำให้ต้องเขียนเยอะกว่านี้ ลักษณะของซอร์สโค้ดดังกล่าวสามารถอธิบายเพิ่มเติมได้ว่า `contentView.setText(task.content);` เป็นการสั่งการว่าให้ตัวแปร contentView ไปดึงข้อมูลจากตาราง task คอลัมน์ content มาแสดงเป็นตัวอักษร ส่วนการดึงข้อมูลจากคอลัมน์ที่มีการเชื่อมความสัมพันธ์ระหว่างตารางไว้จะใช้ซอร์สดังนี้

`location.setText(task.location.locationName);` ซึ่งก็คือการบอกว่าให้ตัวแปร location ไปดึงข้อมูลจากตาราง task คอลัมน์ location และให้ Foreign key ไปดึงข้อมูลจากคอลัมน์ locationName ในตาราง location มา

แสดงผลเป็นตัวอักษร ซึ่งก่อนหน้านี้เราได้สร้างความสัมพันธ์ระหว่างตารางไว้แล้วในไฟล์ สร้างฐานข้อมูล

ส่วนต่อไปคือการสร้างเมนูบาร์และไอคอน โดยซอร์สโค้ดดังนี้

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, MenuItem_EditID, R.string.edit,
    buildDrawable(MaterialDesignIconic.Icon.gmi_edit));
    addMenuItem(menu, MenuItem_DeleteID, R.string.delete,
    buildDrawable(MaterialDesignIconic.Icon.gmi_delete));
    return true;
}
```

และตามด้วยคำสั่งที่จะทำให้ไอคอนดังกล่าวทำงานได้ ทั้งการสั่งงานให้ลบ และแก้ไข

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case MenuItem_EditID:
            Intent intent = new Intent(this, FormActivity.class);
            intent.putExtra("id", task.getId());
            startActivityForResult(intent, EDIT_TASK);
            break;
        case MenuItem_DeleteID:
            AlertDialog.Builder alert = new AlertDialog.Builder(this);
            alert.setTitle(android.R.string.dialog_alert_title);
            alert.setMessage(R.string.are_you_sure);
            alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    task.delete();
                    setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));
```

```
        finish();
    }
});

alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {

    @Override

    public void onClick(DialogInterface dialog, int which) {

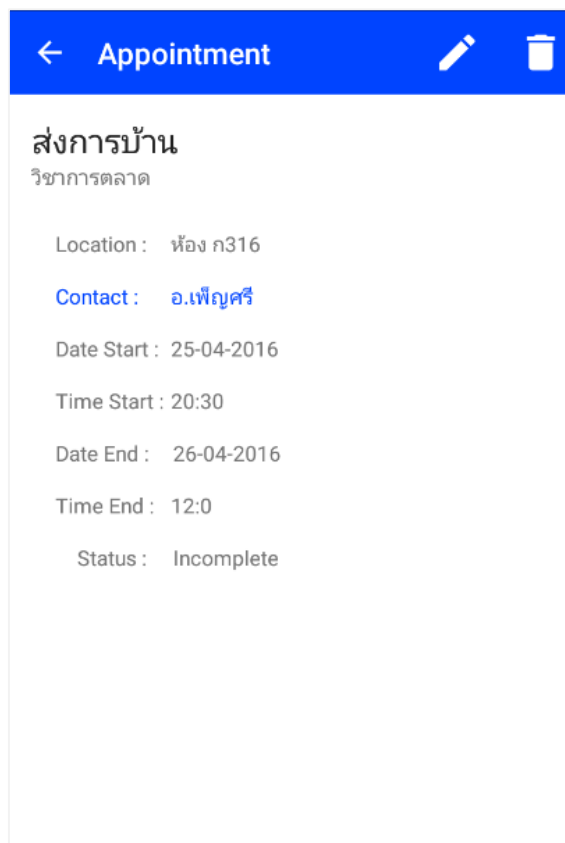
    }

});

alert.show();

break;
}

return super.onOptionsItemSelected(item);
}
```



← Appointment

ส่งการบ้าน
วิชาการตลาด

Location : ห้อง 316

Contact : อ.เพ็ญศรี

Date Start : 25-04-2016

Time Start : 20:30

Date End : 26-04-2016

Time End : 12:00

Status : Incomplete

Activity_show.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorbk"
    android:orientation="vertical"
    tools:context="project.ann.pee.studentappointment.ShowActivity">

    <include
        android:id="@+id/toolbar_main"
```

```
layout="@layout/toolbar_main" />
```

```
<TextView
```

```
    android:id="@+id/titleView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/toolbar_main"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:text="title"
    android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
<ScrollView
```

```
    android:id="@+id/scrollView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/titleView"
    android:layout_centerHorizontal="true">
```

```
<TextView
```

```
    android:id="@+id/contentView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
```

```
    android:text="content" />
```

```
</ScrollView>
```

```
<RelativeLayout
```

```
    android:id="@+id/relativeLayout1"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_below="@+id/titleView"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin">
```

```
<TextView
```

```
    android:id="@+id/location_name"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignParentStart="true"
```

```
    android:layout_below="@+id/titleView"
```

```
    android:layout_marginTop="45dp"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:text="Location :" />
```

```
<TextView
```

```
    android:id="@+id/location"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignBottom="@+id/location_name"
```

```

    android:layout_toEndOf="@+id/location_name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="16dp"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="location " />

```

<TextView

```

    android:id="@+id/contact__name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/location_name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Contact :"
    android:textColor="@color/primary" />

```

<TextView

```

    android:id="@+id/contactView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/dateStart_name"
    android:layout_toEndOf="@+id/location_name"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="contact"

```

```
android:textColor="@color/primary" />
```

```
<TextView
```

```
    android:id="@+id/dateStart_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/contact__name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Date Start :" />
```

```
<TextView
```

```
    android:id="@+id/timeStart_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/dateStart_name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Time Start :" />
```

```
<TextView
```

```
    android:id="@+id/dateEnd_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/timeStart_name"
```

```

    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Date End : " />

```

<TextView

```

    android:id="@+id/timeEnd_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/dateEnd_name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Time End : " />

```

<TextView

```

    android:id="@+id/timeEnd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/timeStart"
    android:layout_below="@+id/dateEnd_name"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="timeEnd" />

```

<TextView

```

    android:id="@+id/timeStart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/dateStart"
    android:layout_below="@+id/dateStart_name"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="timeStart" />

```

<TextView

```

    android:id="@+id/dateStart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/contactView"
    android:layout_alignTop="@+id/dateStart_name"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="dateStart" />

```

<TextView

```

    android:id="@+id/dateEnd"

```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/timeStart_name"
    android:layout_toEndOf="@+id/dateEnd_name"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="dateEnd" />

```

<TextView

```

    android:id="@+id/Status1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Status :"
    android:layout_below="@+id/timeEnd_name"
    android:layout_toStartOf="@+id/timeEnd" />

```

<TextView

```

    android:id="@+id/statusShow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"

```



```

    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="Status"
    android:layout_below="@+id/timeEnd_name"
    android:layout_alignStart="@+id/dateEnd" />

```

```
</RelativeLayout>
```

```
</RelativeLayout>
```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ activity_show.xml สร้างขึ้นมาเพื่อแสดงรายละเอียดของกิจกรรมนัดหมายโดยทำงานร่วมกันกับ ShowActivity.java ซึ่งการแสดงผลของหน้าแสดงรายละเอียดกิจกรรมนัดหมายประกอบด้วยส่วนหลัก ๆ สองส่วนคือ ข้อความที่แสดงว่าข้อมูลที่โชว์คืออะไร และส่วนที่สองคือส่วนที่รับข้อมูลจากจาวาคลาสมาแสดง โดยทั้งสองส่วนจะเขียนภายในคำสั่ง TextView และภายใต้แท็ก

```
<RelativeLayout> </RelativeLayout> ทั้งหมด ดังนี้
```

```
<TextView
```

```

    android:id="@+id/location_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/titleView"
    android:layout_marginTop="45dp"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:text="Location :" />

```

```
<TextView
```

```

    android:id="@+id/location"
    android:layout_width="wrap_content"

```

```

    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/location_name"
    android:layout_toEndOf="@+id/location_name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="16dp"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="location " />

```

<TextView

```

    android:id="@+id/contact__name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/location_name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Contact :"
    android:textColor="@color/primary" />

```

<TextView

```

    android:id="@+id/contactView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/dateStart_name"
    android:layout_toEndOf="@+id/location_name"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"

```

```

    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="contact"
    android:textColor="@color/primary" />

```

<TextView

```

    android:id="@+id/dateStart_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/contact__name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Date Start :" />

```

<TextView

```

    android:id="@+id/timeStart_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/dateStart_name"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Time Start :" />

```

<TextView

```

    android:id="@+id/dateEnd_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

android:layout_alignParentStart="true"
android:layout_below="@+id/timeStart_name"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_horizontal_margin"
android:text="Date End :" />

```

<TextView

```

android:id="@+id/timeEnd_name"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentStart="true"
android:layout_below="@+id/dateEnd_name"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_horizontal_margin"
android:text="Time End :" />

```

<TextView

```

android:id="@+id/timeEnd"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignStart="@+id/timeStart"
android:layout_below="@+id/dateEnd_name"
android:paddingBottom="16dp"
android:paddingLeft="16dp"
android:paddingRight="16dp"
android:paddingTop="@dimen/activity_horizontal_margin"
android:scrollHorizontally="false"
android:scrollbars="vertical"
android:text="timeEnd" />

```

<TextView

```

    android:id="@+id/timeStart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/dateStart"
    android:layout_below="@+id/dateStart_name"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="timeStart" />

```

<TextView

```

    android:id="@+id/dateStart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/contactView"
    android:layout_alignTop="@+id/dateStart_name"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="dateStart" />

```

<TextView

```

    android:id="@+id/dateEnd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/timeStart_name"
    android:layout_toEndOf="@+id/dateEnd_name"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
    android:text="dateEnd" />

```

<TextView

```

    android:id="@+id/Status1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:text="Status :"
    android:layout_below="@+id/timeEnd_name"
    android:layout_toStartOf="@+id/timeEnd" />

```

<TextView

```

    android:id="@+id/statusShow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingBottom="16dp"

```

```
        android:paddingLeft="16dp"
        android:paddingRight="16dp"
        android:paddingTop="@dimen/activity_horizontal_margin"
        android:scrollHorizontally="false"
        android:scrollbars="vertical"
        android:text="Status"
        android:layout_below="@+id/timeEnd_name"
        android:layout_alignStart="@+id/dateEnd" />
</RelativeLayout>
```

ShowContact.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import org.w3c.dom.Text;

public class ShowContact extends BaseActivity {

    private ContactTB    contactTB = null;
    private TextView firstView;
    private TextView lastView;
    private static final int MenuItem_EditID = 1;
    private static final int MenuItem_DeleteID = 2;
    private static final int EDIT_TASK = 10;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.contact_show);
        setDrawer(true);
        setTitle(R.string.contactDetail);
    }

```



```

firstView = (TextView) findViewById(R.id.firstView);
lastView = (TextView) findViewById(R.id.lastView);
long id = getIntent().getLongExtra("id", 0);
setView(id);
}

```

```

private void setView(long id) {
    if (id > 0)
        contactTB = ContactTB.load(ContactTB.class, id);
    if (contactTB != null) {
        firstView.setText(contactTB.firstName);
        lastView.setText(contactTB.lastName);

    } else {
        finish();
    }
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    addMenuItem(menu, MenuItem_EditID, R.string.edit,
buildDrawable(MaterialDesignIconic.Icon.gmi_edit));
    addMenuItem(menu, MenuItem_DeleteID, R.string.delete,
buildDrawable(MaterialDesignIconic.Icon.gmi_delete));
    return true;
}

```

@Override

```

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

        case MenuItem_EditID:

            Intent intent = new Intent(this, FormContactDefault.class);
            intent.putExtra("id", contactTB.getId());
            startActivityForResult(intent, EDIT_TASK);

            break;

        case MenuItem_DeleteID:

            AlertDialog.Builder alert = new AlertDialog.Builder(this);
            alert.setTitle(android.R.string.dialog_alert_title);
            alert.setMessage(R.string.are_you_sure);
            alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {

                @Override

                public void onClick(DialogInterface dialog, int which) {

                    contactTB.delete();

                    setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));

                    finish();

                }

            });

            alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {

                @Override

                public void onClick(DialogInterface dialog, int which) {

                }

            });

            alert.show();

            break;

    }
}

```

```

        return super.onOptionsItemSelected(item);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (resultCode == Activity.RESULT_OK) {
            Bundle extras = data.getExtras();
            switch (requestCode) {
                case EDIT_TASK:
                    if (extras != null && extras.getLong("id", 0) > 0) {
                        setView(contactTB.getId());
                        setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));
                    }
                    break;
            }
        }
    }
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ ShowContact.java สร้างขึ้นมาเพื่อให้แอปพลิเคชันแสดงรายละเอียดของผู้ติดต่อ ที่ผู้ใช้ได้บันทึกไว้ เพื่อไว้เลือกใช้งาน โดยทำงานร่วมกันกับหน้าแสดงผล contact_show.xml

contact_show.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical"

    android:background="@drawable/bkcontact"

    tools:context="project.ann.pee.studentappointment.ShowActivity">

    <include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />

    <TextView

        android:id="@+id/nameView"

        android:text="First Name :"

        android:textSize="15dp"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:paddingLeft="16dp"

        android:paddingTop="20dp"

        android:paddingBottom="16dp"

        android:layout_centerVertical="true"

        android:layout_alignEnd="@+id/lastNameview" />

    <TextView

        android:id="@+id/lastNameview"

        android:text="Last Name :"

        android:textSize="15dp"
```

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/nameView"
android:layout_alignParentStart="true"
android:layout_marginStart="30dp" />

```

<TextView

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="firstName"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
android:id="@+id/firstView"

android:textColor="@color/colorPrimary"
android:layout_above="@+id/lastNameview"
android:layout_toEndOf="@+id/nameView" />

```

<TextView

```

android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="lastName"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
android:textColor="@color/colorPrimary"
android:scrollbars="vertical"
android:scrollHorizontally="false"
android:id="@+id/lastView"

```

```

        android:layout_below="@+id/nameView"
        android:layout_alignStart="@+id/firstView" />
</RelativeLayout>

```

ShowAct.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import org.w3c.dom.Text;

public class ShowAct extends BaseActivity {
    private Act    act = null;
    private TextView actView;
    private static final int MenuItem_EditID = 1;
    private static final int MenuItem_DeleteID = 2;
    private static final int EDIT_TASK = 10;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

```

```

setContentView(R.layout.act_show);

setDrawer(true);

setTitle(R.string.actshow);


actView = (TextView) findViewById(R.id.actView);


long id = getIntent().getLongExtra("id", 0);
setView(id);
}

private void setView(long id) {

    if (id > 0)

        act = Act.load(Act.class, id);

    if (act != null) {

        actView.setText(act.activityName);

    } else {

        finish();

    }

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

    addItem(menu, MenuItem_EditID, R.string.edit,
buildDrawable(MaterialDesignIconic.Icon.gmi_edit));

    addItem(menu, MenuItem_DeleteID, R.string.delete,
buildDrawable(MaterialDesignIconic.Icon.gmi_delete));

    return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

```

case MenuItem_EditID:

```
Intent intent = new Intent(this, FormActDefault.class);
```

```
intent.putExtra("id", act.getId());
```

```
startActivityForResult(intent, EDIT_TASK);
```

```
break;
```

case MenuItem_DeleteID:

```
AlertDialog.Builder alert = new AlertDialog.Builder(this);
```

```
alert.setTitle(android.R.string.dialog_alert_title);
```

```
alert.setMessage(R.string.are_you_sure);
```

```
alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(DialogInterface dialog, int which) {
```

```
        act.delete();
```

```
        setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));
```

```
        finish();
```

```
    }
```

```
});
```

```
alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(DialogInterface dialog, int which) {
```

```
    }
```

```
});
```

```
alert.show();
```

```
break;
```

```
}
```

```
return super.onOptionsItemSelected(item);
```

```
}
```



```

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (resultCode == Activity.RESULT_OK) {

        Bundle extras = data.getExtras();

        switch (requestCode) {

            case EDIT_TASK:

                if (extras != null && extras.getLong("id", 0) > 0) {

                    setContentView(act.getId());

                    setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));

                }

                break;

            }

        }

    }

}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ ShowAct.java สร้างขึ้นมาเพื่อให้แอปพลิเคชันแสดงรายละเอียดของกิจกรรม ที่ผู้ใช้ได้บันทึกไว้ เพื่อไว้เลือกใช้งาน โดยทำงานร่วมกันกับหน้าแสดงผล act_show.xml

act_show.xml

```

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="@drawable/bk"

    android:orientation="vertical"

    tools:context="project.ann.pee.studentappointment.ShowActivity">

```

```
<include
    android:id="@+id/toolbar_main"
    layout="@layout/toolbar_main"></include>
```

```
<TextView
    android:id="@+id/timeEnd_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:paddingBottom="16dp"
    android:paddingLeft="60dp"
    android:paddingTop="20dp"
    android:text="Activity :"
    android:textSize="15dp" />
```

```
<TextView
    android:id="@+id/actView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/timeEnd_name"
    android:layout_toEndOf="@+id/timeEnd_name"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="@dimen/activity_horizontal_margin"
    android:scrollHorizontally="false"
    android:scrollbars="vertical"
```

```

        android:text="Activity"
        android:textColor="@color/primary" />
</RelativeLayout>

```

ShowLocation.java

```

package project.ann.pee.studentappointment;

import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;

public class ShowLocation extends BaseActivity {

    private LocationsTB locationsTB = null;
    private TextView locationView;
    private static final int MenuItem_EditID = 1;
    private static final int MenuItem_DeleteID = 2;
    private static final int EDIT_TASK = 10;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);

setContentView(R.layout.location_show);

setDrawer(true);

setTitle(R.string.location);


locationView = (TextView) findViewById(R.id.locationView);

long id = getIntent().getLongExtra("id", 0);

setView(id);
}

private void setView(long id) {

    if (id > 0)

        locationsTB = LocationsTB.load(LocationsTB.class, id);

    if (locationsTB != null) {

        locationView.setText(locationsTB.locationName);

    } else {

        finish();

    }

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

    addMenuItem(menu, MenuItem_EditID, R.string.edit,

buildDrawable(MaterialDesignIconic.Icon.gmi_edit));

    addMenuItem(menu, MenuItem_DeleteID, R.string.delete,

buildDrawable(MaterialDesignIconic.Icon.gmi_delete));

    return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

```

case MenuItem_EditID:

```

Intent intent = new Intent(this, FormLocationDefault.class);
intent.putExtra("id", locationsTB.getId());
startActivityForResult(intent, EDIT_TASK);

break;

```

case MenuItem_DeleteID:

```

AlertDialog.Builder alert = new AlertDialog.Builder(this);
alert.setTitle(android.R.string.dialog_alert_title);
alert.setMessage(R.string.are_you_sure);
alert.setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        locationsTB.delete();

        setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));
        finish();
    }
});

alert.setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
    }
});

alert.show();

break;
}

return super.onOptionsItemSelected(item);
}

```

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (resultCode == Activity.RESULT_OK) {
        Bundle extras = data.getExtras();
        switch (requestCode) {
            case EDIT_TASK:
                if (extras != null && extras.getLong("id", 0) > 0) {
                    setView(locationsTB.getId());
                    setResult(Activity.RESULT_OK, new Intent().putExtra("refreshNeeded", true));
                }
                break;
            }
        }
    }
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ ShowLocation.java สร้างขึ้นมาเพื่อให้แอปพลิเคชันแสดงรายละเอียดของสถานที่ ที่ผู้ใช้ได้บันทึกไว้ เพื่อไว้เลือกใช้งาน โดยทำงานร่วมกันกับหน้าแสดงผล

location_show.xml

Location_show.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical"

    android:background="@drawable/bk4"

    tools:context="project.ann.pee.studentappointment.ShowActivity">

    <include android:id="@+id/toolbar_main" layout="@layout/toolbar_main" />


    <TextView

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:text="Location"

        android:paddingLeft="16dp"

        android:paddingRight="16dp"

        android:paddingBottom="16dp"

        android:textColor="@color/primary"

        android:scrollbars="vertical"

        android:scrollHorizontally="false"

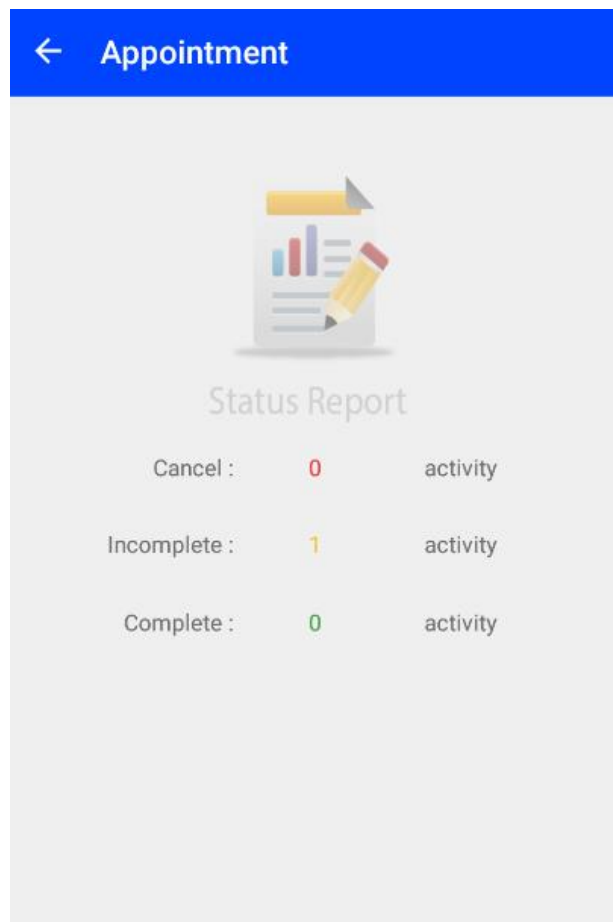
        android:id="@+id/locationView"

        android:layout_alignTop="@+id/textView"

        android:layout_alignStart="@+id/textView"

        android:layout_marginStart="91dp" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Location :"
    android:textSize="15dp"
    android:id="@+id/textView"
    android:layout_gravity="center_vertical"
    android:layout_centerVertical="true"
    android:layout_alignParentStart="true"
    android:layout_marginStart="56dp" />
</RelativeLayout>
```



report.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bkreport"
    android:orientation="vertical"
    tools:context="project.ann.pee.studentappointment.ShowActivity">
```

```

<include
    android:id="@+id/toolbar_main"
    layout="@layout/toolbar_main" />

<TextView
    android:id="@+id/Incomplete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/reportIncomplete"
    android:layout_toStartOf="@+id/reportIncomplete"
    android:text="Incomplete :" />

<TextView
    android:id="@+id/Complete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/reportComplete"
    android:layout_toStartOf="@+id/reportComplete"
    android:text="Complete :" />

<TextView
    android:id="@+id/reportComplete"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/reportIncomplete"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="29dp"
    android:gravity="center"
    android:text="reportComplete"

```

```
android:textColor="@color/md_green_600" />
```

```
<TextView
```

```
    android:id="@+id/reportIncomplete"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/reportCancle"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="28dp"
    android:gravity="center"
    android:text="reportIncomple"
    android:textColor="@color/md_yellow_700" />
```

```
<TextView
```

```
    android:id="@+id/Cancle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/reportIncomplete"
    android:layout_alignEnd="@+id/Incomplete"
    android:text="Cancel :" />
```

```
<TextView
```

```
    android:id="@+id/reportCancle"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/reportIncomplete"
    android:layout_centerVertical="true"
    android:gravity="center"
    android:text="reportCancle"
```

```
android:textColor="@color/md_red_600" />
```

```
<TextView
```

```
    android:id="@+id/ac1"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignStart="@+id/ac2"
```

```
    android:layout_alignTop="@+id/reportCancle"
```

```
    android:text="activity " />
```

```
<TextView
```

```
    android:id="@+id/ac2"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignStart="@+id/ac2"
```

```
    android:layout_alignTop="@+id/reportIncomplete"
```

```
    android:text="activity " />
```

```
<TextView
```

```
    android:id="@+id/ac2"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignTop="@+id/reportComplete"
```

```
    android:layout_marginStart="17dp"
```

```
    android:layout_toEndOf="@+id/reportComplete"
```

```
    android:text="activity " />
```

```
</RelativeLayout>
```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ Report.xml สร้างขึ้นมาเพื่อให้แอปพลิเคชันแสดงรายงานจำนวนกิจกรรมนัดหมาย แยกตามสถานะของกิจกรรม คือ Complete Incomplete และ Cancel โดยทำงานร่วมกับจาวาคลาส Report1.java โดยลักษณะการเขียนหน้าแสดงผลนี้ใช้ TextView เขียนภายใต้แท็ก <RelativeLayout> และปิดด้วย </RelativeLayout> ซึ่งลักษณะการเขียนคล้ายคลึงกับโค้ดการแสดงผลอื่น ๆ ก่อนนี้ที่ได้อธิบายไปข้างต้นแล้ว

Report1.java

```

package project.ann.pee.studentappointment;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import com.activeandroid.query.Select;
import org.w3c.dom.Text;
import java.util.ArrayList;

public class Report1 extends BaseActivity{

    private Task task ;
    private TextView reportIncomplete;
    private TextView reportComplete;
    private TextView reportCancel;
    private static int countIncomplete;
    private static int countComplete;
    private static int countCancel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.report);

        setDrawer(true);

        setTitle(R.string.tasks);

        //countIncomplete

        reportIncomplete = (TextView) findViewById(R.id.reportIncomplete);

        countIncomplete = Task.countIncomplete();
    }

```

```

        reportIncomplete.setText(Integer.toString(countIncomplete));

//countComplete

        reportComplete = (TextView) findViewById(R.id.reportComplete);

        countComplete = Task.countComplete();

        reportComplete.setText(Integer.toString(countComplete));

//countCancel

        reportCancel = (TextView) findViewById(R.id.reportCancel);

        countCancel = Task.countCancel();

        reportCancel.setText(Integer.toString(countCancel));
    }
}

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ Report1.java สร้างขึ้นมาเพื่อให้แอปพลิเคชันแสดงรายงานของจำนวนกิจกรรมนัดหมายแยกตามสถานะของกิจกรรมนัดหมายตั้งแต่ Incomplete Complete และ Cancel ซึ่งจะทำงานร่วมกับหน้าแสดงผล report.xml สามารถอธิบายซอร์สโค้ดได้ดังนี้

หลังจากการ import library แล้วก็เริ่มด้วยการประกาศคลาส และประกาศตัวแปรต่าง ๆ

```

public class Report1 extends BaseActivity{

    private Task task ;

    private TextView reportIncomplete;

    private TextView reportComplete;

    private TextView reportCancel;

    private static int countIncomplete;

    private static int countComplete;

    private static int countCancel;

```

ส่วนถัดมาคือการเรียกใช้หน้าแสดงผล report โดยใช้คำสั่ง

@Override

```

protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);

setContentView(R.layout.report);

setDrawer(true);

setTitle(R.string.tasks);

```

และส่วนสำคัญของหน้านี้คือการนับจำนวนกิจกรรมโดยแยกตามสถานะ Incomplete Complete และ Cancel โดยลักษณะการเขียนคำสั่งคือ ก่อนหน้านี้ได้ประกาศให้สถานะ Incomplete Complete และ Cancel มีลักษณะเป็น int หรือในรูปของตัวเลขนั่นเอง เพื่อให้นับได้ หลังจากนั้นก็มาเขียนคำสั่งให้นับในตาราง task ว่าแต่ละสถานะมีจำนวนกี่กิจกรรม ซอร์สโค้ดคือ

```
//countIncomplete
```

```

reportIncomplete = (TextView) findViewById(R.id.reportIncomplete);

countIncomplete = Task.countIncomplete();

reportIncomplete.setText(Integer.toString(countIncomplete));

```

```
//countComplete
```

```

reportComplete = (TextView) findViewById(R.id.reportComplete);

countComplete = Task.countComplete();

reportComplete.setText(Integer.toString(countComplete));

```

```
//countCancel
```

```

reportCancel = (TextView) findViewById(R.id.reportCancel);

countCancel = Task.countCancel();

reportCancel.setText(Integer.toString(countCancel));

```

หลังจากนับแล้วก็ให้ส่งค่าไปแสดงที่หน้าแสดงผล ตัวอย่างโค้ดคือ **reportCancel** = (TextView)
 findViewById(R.id.**reportCancel**);



about_us.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bkabout"
    android:orientation="vertical"
    tools:context="project.ann.pee.studentappointment.ShowActivity">

    <include
        android:id="@+id/toolbar_main"
        layout="@layout/toolbar_main"></include>
```

```

<ImageView
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:src="@drawable/logo3d"
    android:layout_below="@+id/toolbar_main"
    android:layout_centerHorizontal="true"
    android:id="@+id/imageView" />

<TextView
    android:id="@+id/Version"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/version"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />

<TextView
    android:layout_width="250dp"
    android:layout_height="match_parent"
    android:text="@string/about"
    android:layout_centerHorizontal="true"
    android:id="@+id/textView3" android:layout_below="@id/Version"/>

</RelativeLayout>

```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ about_us.xml สร้างขึ้นมาเพื่อให้แอปพลิเคชันแสดงรายละเอียดของแอปพลิเคชัน โดยทำงานร่วมกันกับจาวาคลาส AboutUS.java ซึ่งจะเห็นว่าหน้าแสดงผลนี้มีการดึงรูปภาพมาแสดง ลักษณะของโค้ดจะเขียนโดยใช้ <ImageView xxxx /> ซึ่งเป็นการเขียนเพื่อดึง source ของรูปภาพที่ถูกเก็บไว้ที่ drawable มาแสดง และสามารถกำหนดขนาดของรูปภาพได้ ภายใต้โค้ดชุด <ImageView /> ดังต่อไปนี้

```
<ImageView
```

```
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:src="@drawable/logo3d"
    android:layout_below="@+id/toolbar_main"
    android:layout_centerHorizontal="true"
    android:id="@+id/imageView" />
```

ส่วนข้อความที่แสดงใช้การเขียนแบบ TextView ที่ได้อธิบายไว้ก่อนหน้านี้แล้ว

AboutUs.java

```
package project.ann.pee.studentappointment;
```

```
import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;

import org.w3c.dom.Text;
```

```
public class AboutUs extends BaseActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.about_us);
    }
}
```

```
setDrawer(false);  
setTitle("About Us");  
}  
}
```

ในส่วนของซอร์สโค้ด AboutUs.java ไม่มีคำสั่งอะไร เป็นเพียงการสั่งงานให้เรียกหน้าแสดงผลมาโชว์ เพราะหน้านี้เป็นเพียงการแสดงรายละเอียดเกี่ยวกับแอปพลิเคชัน ไม่มีการรับข้อมูลใด ๆ จากผู้ใช้

StudentAppointmentApplication.java

```
package project.ann.pee.studentappointment;

import android.app.Application;
import com.activeandroid.ActiveAndroid;

public class StudentAppointmentApplication extends Application{

    @Override

    public void onCreate(){

        super.onCreate();

        ActiveAndroid.initialize(this);

    }

}
```

จากซอร์สโค้ด สามารถอธิบายได้ว่า ไฟล์ StudentAppointmentApplication.java สร้างขึ้นมาเพื่อติดตั้ง activeandroid ให้สามารถทำงานภายในแอปพลิเคชัน

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="project.ann.pee.studentappointment">

    <application
        android:name=".StudentAppointmentApplication"
        android:allowBackup="true"
        android:icon="@mipmap/logoapp3d"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity
            android:name=".ListActivity"
            android:label="@string/app_name"
            android:uiOptions="splitActionBarWhenNarrow">

            <intent-filter>

                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />

            </intent-filter>

        </activity>

        <activity android:name=".ShowActivity" />
        <activity android:name=".FormActDefault" />
        <activity android:name=".FormActivity" />
        <activity android:name=".FormLocationDefault" />

```

```

<activity android:name=".FormContactDefault" />
<activity android:name=".ActList" />
<activity android:name=".ShowAct" />
<activity android:name=".LocationList" />
<activity android:name=".ShowLocation" />
<activity android:name=".ShowContact" />
<activity android:name=".ContactList" />
<activity android:name=".Report1" />
<activity android:name=".AboutUs" />

<meta-data
    android:name="AA_DB_NAME"
    android:value="SA.db" />
<meta-data
    android:name="AA_DB_VERSION"
    android:value="1" />
<meta-data
    android:name="AA_MODELS"
    android:value="project.ann.pee.studentappointment.Task,
project.ann.pee.studentappointment.Act,
project.ann.pee.studentappointment.LocationsTB,project.ann.pee.studentappointment.ContactTB,proj
ect.ann.pee.studentappointment.StatusTB" />
</application>
</manifest>

```

จากซอร์สโค้ดสามารถอธิบายได้ว่า ไฟล์นี้คือส่วนของรายละเอียดของแอปพลิเคชันและการเชื่อมโยง
หน้าของแอปพลิเคชัน

จากซอร์สโค้ดด้านบนสามารถอธิบายได้ดังนี้ ส่วนอื่น ๆ เป็นแค่ส่วนที่บ่งบอกถึงรายละเอียดต่าง ๆ ของแอปพลิเคชันเพียงเท่านั้น ทั้งนี้จะขออธิบายในส่วนที่จำเป็นคือ การประกาศเรียกการกระทำที่เกี่ยวข้องกับการใช้ฟังก์ชัน Intent(การเชื่อมโยงหรือเรียกหน้าใหม่) ซึ่งจะใช้คำสั่งดังนี้

```
<activity android:name=".FormActDefault" />
```

อธิบายคือใช้แท็ก <activity android:name=".ชื่อไฟล์จาวาคลาส” จากตัวอย่างเป็นการสั่งให้เมื่อมีการเรียกหน้าหลัก ให้มีการเชื่อมโยงหน้านั้นได้ เมื่อมีการกำหนด activity ภายใน AndroidManifest.xml ได้อย่างถูกต้องแล้วและถ้าหากไม่มีการกำหนดในส่วนนี้และจะไม่สามารถเชื่อมโยงหน้าที่เราต้องการเชื่อมโยงกับหน้าปัจจุบันได้หรือกล่าวได้อีกอย่างหนึ่งคือจาวาคลาสที่ถูกเขียนจะไม่สามารถทำงานได้ถ้าไม่มีการประกาศในหน้า AndroidManifest

และในส่วนต่อไปคือการประกาศค่าต่าง ๆ ของฐานข้อมูล ไม่ว่าจะเป็นชื่อฐานข้อมูล เวอร์ชันของฐานข้อมูล ดังซอร์สโค้ดนี้

```
<meta-data
```

```
    android:name="AA_DB_NAME"
```

```
    android:value="SA.db" />
```

```
<meta-data
```

```
    android:name="AA_DB_VERSION"
```

```
    android:value="1" />
```

```
<meta-data
```

```
    android:name="AA_MODELS"
```

```
    android:value="project.ann.pee.studentappointment.Task,  
project.ann.pee.studentappointment.Act,  
project.ann.pee.studentappointment.LocationsTB,project.ann.pee.studentappointment.ContactTB,proj  
ect.ann.pee.studentappointment.StatusTB" /> และในส่วนนี้คือการประกาศตาราง(Table)ทั้งหมดที่ได้  
เขียนไว้ หากมีการเพิ่มตารางก็ต้องเขียนเพิ่มไปในส่วนนี้ด้วยเช่นกัน
```


String.xml

ไฟล์ String.xml อยู่ในส่วนโฟลเดอร์ values เป็นการเก็บค่าที่เกี่ยวข้องกับตัวอักษร หัวข้อ ต่าง ๆ ที่จะแสดงในหน้าแอปพลิเคชัน เพื่อเรียกใช้ โดยไม่ต้องเขียนซ้ำ ๆ ทุกหน้า

```
<resources>

    <string name="app_name">StudentAppointment</string>

    <string name="tasks">Appointment</string>

    <string name="actshow">Activity</string>

    <string name="act">Activities</string>

    <string name="locations">Locations</string>

    <string name="task_title">Title</string>

    <string name="new_task">New Appointment</string>

    <string name="task">Task</string>

    <string name="save">Save</string>

    <string name="refresh">Refresh</string>

    <string name="no_task">No Appointment</string>

    <string name="no_act">No Activity</string>

    <string name="no_location">No Location</string>

    <string name="no_contact">No Contact</string>

    <string name="task_content">Content</string>

    <string name="unsaved_exit_alert">You have unsaved data. Are you sure you want to exit?</string>

    <string name="title_is_required">Date is required.</string>

    <string name="edit">Edit</string>

    <string name="delete">Delete</string>

    <string name="are_you_sure">Are you sure?</string>

    <string name="edit_task">Edit Task</string>

    <string name="contact_person">Contact Person</string>

    <string name="contactDetail">Contact Detail</string>

    <string name="time_start">Time Start</string>
```

```

<string name="time_end">Time End</string>
<string name="date_start">Date Start</string>
<string name="date_end">Date End</string>
<string name="actDefault">ชื่อกิจกรรม</string>
<string name="new_activity">New activity</string>
<string name="new_location">New location</string>
<string name="new_status">New status</string>
<string name="new_contact">New contact</string>
<string name="edit_activity">Edit activity</string>
<string name="edit_location">Edit location</string>
<string name="edit_status">Edit status</string>
<string name="edit_contact">Edit contact</string>
<string name="spinner_title">Spinner</string>
<string name="location">Location</string>
<string name="locationDefault">ชื่อสถานที่</string>
<string name="status">Your Status</string>
<string name="firsNameDefault">ชื่อ</string>
<string name="lastNameDefault">สกุล</string>
<string name="activity_is_required">activity is required.</string>
<string name="location_is_required">location is required.</string>
<string name="contact_is_required">name is required.</string>
<string name="setting_Activity">Activity</string>
<string name="setting_Location">Location</string>
<string name="setting_Contact">Contact</string>
<string name="about_us">About Us</string>
<string name="setting_home">HOME</string>

```

```

<string name="about">Student appointment เป็นแอปพลิเคชันสำหรับนักเรียนและนักศึกษาเพื่อบันทึก

```

กิจกรรมนัดหมายที่เกี่ยวข้องกับการเรียน หรือกิจกรรมทั่วไป</string>

<string name="version">Version 0.9 </string>

<string name="FormAct">Activity :</string>

<string name="FormContact">Contact :</string>

<string name="FormLocation">Location :</string>

<string name="FormStatus">Status :</string>

<string name="action_settings">Settings</string>

</resources>

Style.xml

ไฟล์ Style.xml เป็นการเก็บค่า สไตล์ของแอปพลิเคชัน เพื่อไว้เรียกใช้งานโดยจาวาคลาส

```
<resources>

    <!-- Base application theme. -->

    <style name="AppTheme" parent="MaterialDrawerTheme.Light.DarkToolbar">

        <!-- Customize your theme here. -->

        <item name="colorPrimary">@color/colorPrimary</item>

        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>

        <item name="colorAccent">@color/colorAccent</item>

    </style>

</resources>
```

Colors.xml

ไฟล์ Colors.xml เป็นไฟล์สำหรับเก็บค่าสี ต่าง ๆ ที่จะใช้ในการแสดงผลแอปพลิเคชัน

```
<?xml version="1.0" encoding="utf-8"?>

<resources>

    <color name="colorPrimary">#0044ff</color>

    <color name="colorPrimaryDark">#000000</color>

    <color name="colorAccent">#3366ff</color>

    <color name="colorbk">#ffffff</color>

    <color name="colorcalenbk">#282828</color>

</resources>
```

0.SQL

ไฟล์ 0.SQL คือการเขียนคำสั่งเพื่อเพิ่มข้อมูลลงในฐานข้อมูลในตอนแรกที่ตั้งแอฟพลิเคชัน ดังข้อ
สัปดาห์ต่อไป

```
insert into Act(activityName) values("Nothing");
insert into Act(activityName) values("ประชุมเชียร์");
insert into Act(activityName) values("ส่งการบ้าน");
insert into Act(activityName) values("ประชุม");
insert into Act(activityName) values("ทำงานกลุ่ม");
insert into Act(activityName) values("ติวหนังสือ");
insert into Act(activityName) values("เรียนชดเชย");
insert into Act(activityName) values("สอบ");
insert into Act(activityName) values("พบอาจารย์");
insert into Location(locationName) values("Nothing");
insert into Contact(firstName) values("Nothing");

insert into StatusTBs(statusName) values("Incomplete")
insert into StatusTBs(statusName) values("Complete")
insert into StatusTBs(statusName) values("Cancel")
```

BaseActivity.java

```

package project.ann.pee.studentappointment;

import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import com.mikepenz.iconics.IconicsDrawable;
import com.mikepenz.iconics.typeface.IIcon;
import com.mikepenz.material_design_iconic_typeface_library.MaterialDesignIconic;
import com.mikepenz.materialdrawer.Drawer;
import com.mikepenz.materialdrawer.DrawerBuilder;
import com.mikepenz.materialdrawer.model.PrimaryDrawerItem;
import com.mikepenz.materialdrawer.model.interfaces.IDrawerItem;

public abstract class BaseActivity extends AppCompatActivity {

    private Toolbar mToolBar;

    protected static final int NEW_TASK = 1;
    protected static final int NEW = 2;
    protected static final int NEW2 = 3;
    protected static final int HOME = 4;

```

```

protected static final int NEW4 = 5;

protected static final int NEW6 = 6;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

}

protected void setDrawer(Boolean upEnabled) {

    Toolbar toolbar_main = (Toolbar) findViewById(R.id.toolbar_main);

    setSupportActionBar(toolbar_main);

    ActionBar supportActionBar = getSupportActionBar();

    if (supportActionBar != null)

        if (upEnabled) {

            supportActionBar.setDisplayHomeAsUpEnabled(true);

        } else {

            final Intent intent = new Intent(this, ActList.class);

            final Intent intent1 = new Intent(this, LocationList.class);

            final Intent intent2 = new Intent(this, ContactList.class);

            final Intent intent3 = new Intent(this, ListActivity.class);

            final Intent intent4 = new Intent(this, AboutUs.class);

            final Intent intent5 = new Intent(this, Report1.class);

            new DrawerBuilder()

                .withActivity(this)

                .withToolbar(toolbar_main)

                .addDrawerItems(

                    new PrimaryDrawerItem()

                        .withName(R.string.setting_home)

                        .withIcon(MaterialDesignIconic.Icon.gmi_home)

                        .withSelectable(false)

                        .withIdentifier(4)

```



```

).withSelectedItem(-1)
.addDrawerItems(
    new PrimaryDrawerItem()
        .withName(R.string.setting_Activity)
        .withIcon(MaterialDesignIconic.Icon.gmi_local_activity)
        .withSelectable(false)
        .withIdentifier(1)
).withSelectedItem(-1)
.addDrawerItems(
    new PrimaryDrawerItem()
        .withName(R.string.setting_Contact)
        .withIcon(MaterialDesignIconic.Icon.gmi_nature_people)
        .withSelectable(false)
        .withIdentifier(3)
).withSelectedItem(-1)
.addDrawerItems(
    new PrimaryDrawerItem()
        .withName(R.string.setting_Location)
        .withIcon(MaterialDesignIconic.Icon.gmi_my_location)
        .withSelectable(false)
        .withIdentifier(2)
).withSelectedItem(-1)
.addDrawerItems(
    new PrimaryDrawerItem()
        .withName("Report appointment status")
        .withIcon(MaterialDesignIconic.Icon.gmi_reader)
        .withSelectable(false)
        .withIdentifier(6)

```

```

)
.addDrawerItems(
    new PrimaryDrawerItem()
        .withName(R.string.about_us)
        .withIcon(MaterialDesignIconic.Icon.gmi_apps)
        .withSelectable(false)
        .withIdentifier(5)
).withSelectedItem(-1)
.withOnDrawerItemClickListener(new Drawer.OnDrawerItemClickListener() {
    @Override
    public boolean onItemClick(View view, int position, IDrawerItem drawerItem) {
        switch (drawerItem.getIdentifier()) {
            case 1:
                startActivityForResult(intent, NEW_TASK);
                break;
            case 2:
                startActivityForResult(intent1, NEW);
                break;
            case 3:
                startActivityForResult(intent2, NEW2);
                break;
            case 4:
                startActivityForResult(intent3, HOME);
                break;
            case 5:
                startActivityForResult(intent4, NEW4);
                break;
            case 6:
                startActivityForResult(intent5, NEW6);

```

```

        break;
    }
    return false;
}
})
.build();
}
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == android.R.id.home) onBackPressed();
    return super.onOptionsItemSelected(item);
}

protected Drawable buildDrawable(Icon icon) {
    return new IconicsDrawable(this).icon(icon).color(Color.WHITE).sizeDp(70).paddingDp(4);
}

protected void addMenuItem(Menu menu, int id, int labelId, Drawable icon) {
    MenuItem menuItem = menu.add(Menu.NONE, id, Menu.NONE, labelId);
    menuItem.setIcon(icon);
    menuItem.setShowAsActionFlags(MenuItem.SHOW_AS_ACTION_IF_ROOM);
}
}

```

สำหรับไฟล์ BaseActivity.java คือคำสั่งที่มีการใช้งานบ่อยในแอปพลิเคชัน จาว่าหน้าต่าง ๆ สามารถเรียกใช้งานได้เลย โดยไม่ต้องเขียนโค้ดซ้ำ ๆ ทุกหน้า ในซอร์สโค้ดนี้จะเป็นการเขียนเพื่อกำหนดลักษณะของเมนูบาร์ กำหนดขนาดไซส์ของไอคอน ซึ่งลักษณะของการใช้งานคือจากคลาสหน้าต่าง ๆ ที่ต้องการใช้งานคำสั่งภายใน BaseActivity ก็ใช้คำสั่งโดยการเปิดคลาสแล้ว extends BaseActivity ตัวอย่างเช่น public class FormActivity extends BaseActivity เป็นต้น เพื่อสร้างเมนูบาร์และควบคุมขนาดไอคอนในการทำงานหน้า

เพิ่มกิจกรรมนัดหมาย ยกตัวอย่างซอร์สโค้ดภายใน BaseActivity ที่เขียนเพื่อกำหนดสีและขนาดของไอคอน ดังนี้

```
protected Drawable buildDrawable(Icon icon) {  
    return new IconicsDrawable(this).icon(icon).color(Color.WHITE).sizeDp(70).paddingDp(4);  
}
```

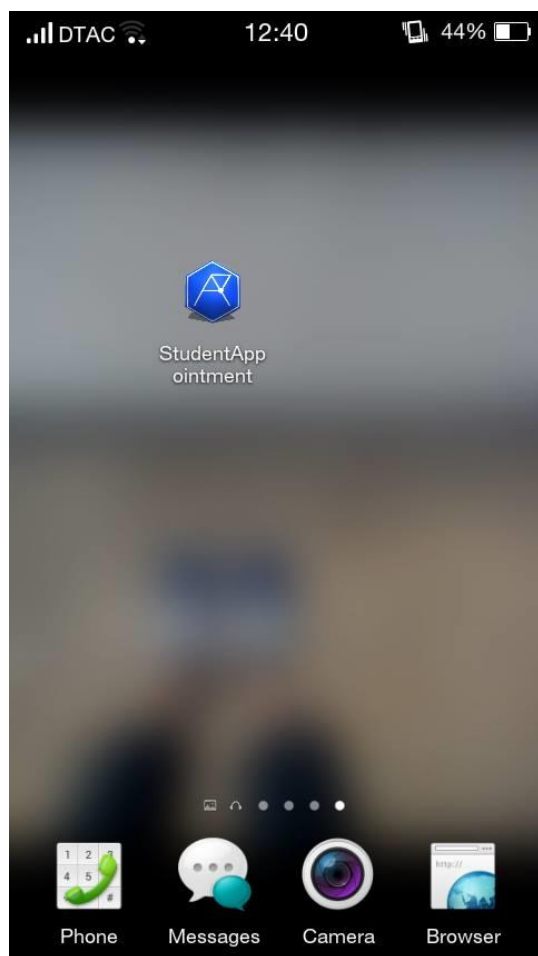
เป็นต้น

สามารถดาวน์โหลด source code ทั้งหมดได้ที่ Github.com ตามลิงค์ต่อไปนี้
<https://github.com/Pee3011/StudentAppointment>

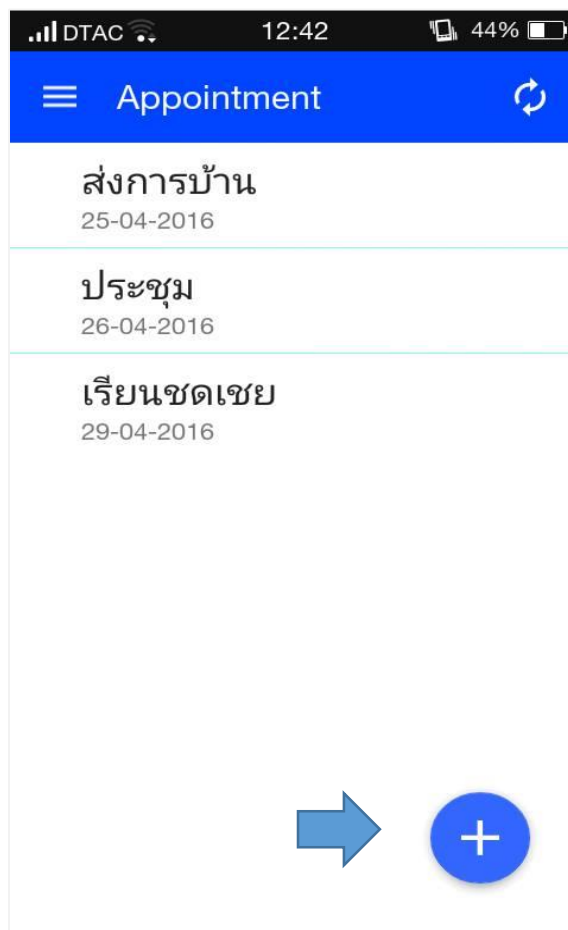
ภาคผนวก ข

คู่มือการใช้งานแอปพลิเคชัน Student Appointment

คู่มือการใช้งานแอปพลิเคชัน Student Appointment



กดไอคอนแอปพลิเคชันเพื่อเข้าสู่
แอปพลิเคชัน



กดไอคอนบวก เพื่อเพิ่มกิจกรรมนัด
หมาย



กลับไปที่หน้าแสดงลิสต์รายการนัดหมาย

← New Appoint... ↻ 📁

กดไอคอน save เมื่อต้องการบันทึกกิจกรรมนัดหมาย

Activity : Nothing

Contact : Nothing

Location : Nothing

กด dropdown เพื่อเลือกกิจกรรม ผู้ติดต่อ และสถานที่

Content ใส่รายละเอียดเพิ่มเติม

Date Start

กดปุ่มบวก กรณีที่ไม่มีกิจกรรม หรือผู้ติดต่อ หรือสถานที่ที่ต้องการและต้องการเพิ่มรายการใหม่ลงไป

กดเพื่อเลือกวันที่นัดหมาย

Time Start

กดเพื่อเลือกเวลานัดหมาย

Date End

กดเพื่อเลือกวันที่สิ้นสุดนัดหมาย

Time End

กดเพื่อเลือกเวลาสิ้นสุดนัดหมาย

Status : Incomplete ▼

กดไอคอนนี้เมื่อต้องการตั้งค่าและดูรายงานสรุปจำนวนกิจกรรมนัดหมาย



ส่งการบ้าน
25-04-2016

ประชุม
26-04-2016

เรียนสดเชย
29-04-2016

กดที่นัดหมายที่ต้องการเพื่อดู
รายละเอียดของนัดหมาย

กดไอคอนดินสอ เมื่อต้องการแก้ไขรายละเอียดของ
กิจกรรมนัดหมายรวมถึงแก้ไขสถานะของกิจกรรมนัด
หมาย หรือ กด ไอคอนถังขยะเมื่อต้องการลบนัดหมายนี้



ส่งการบ้าน
วิชาการตลาด

Location : ห้อง ก316

Contact : อ.เพ็ญศรี

Date Start : 25-04-2016

Time Start : 13:44

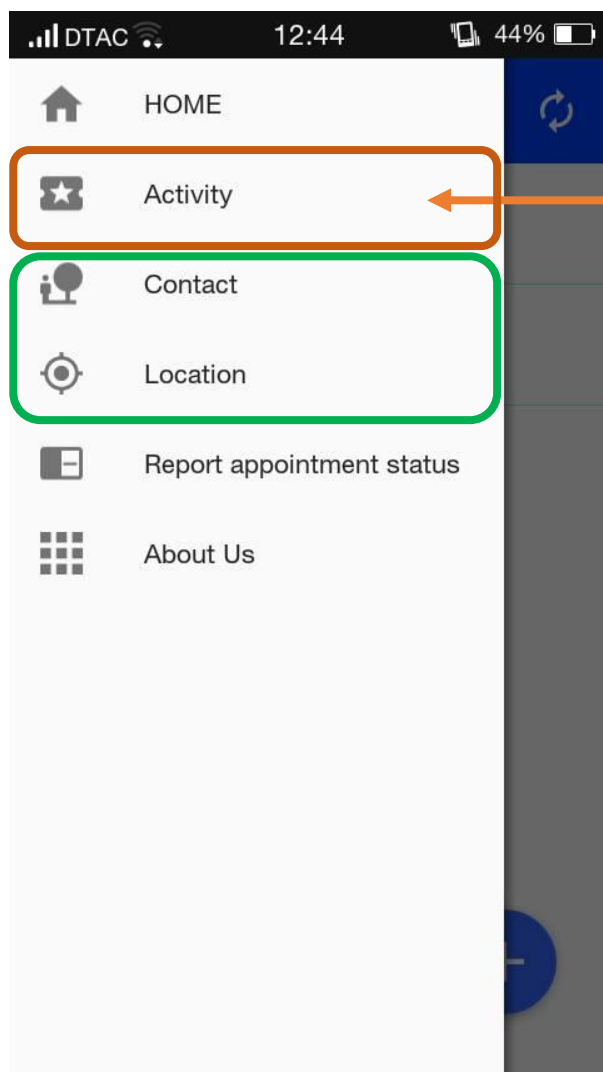
Date End : 25-04-2016

Time End : 16:30

Status : Incomplete

หน้าแสดงรายละเอียดนัดหมาย

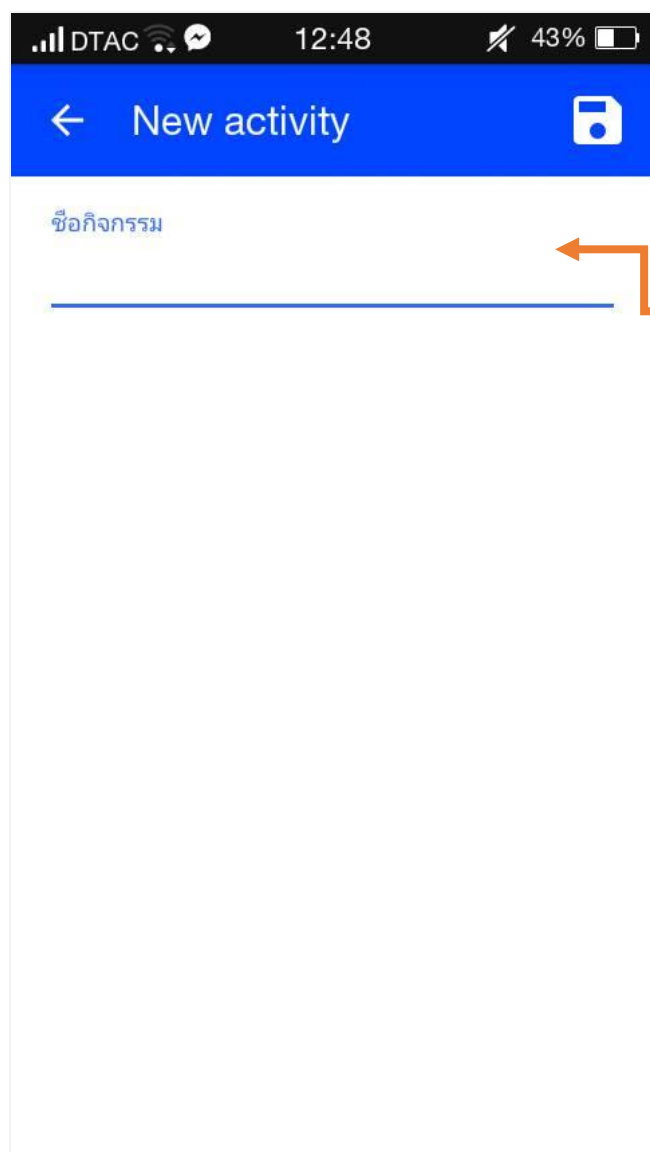
การจัดการกับรายชื่อกิจกรรมนัดหมาย



กดเลือก Activity

กดปุ่มบวกเมื่อต้องการเพิ่มชื่อ
กิจกรรม





กรอกชื่อกิจกรรม แล้วกด
บันทึกที่ไอคอน save

กดที่กิจกรรมที่ต้องการเพื่อดู
รายละเอียด

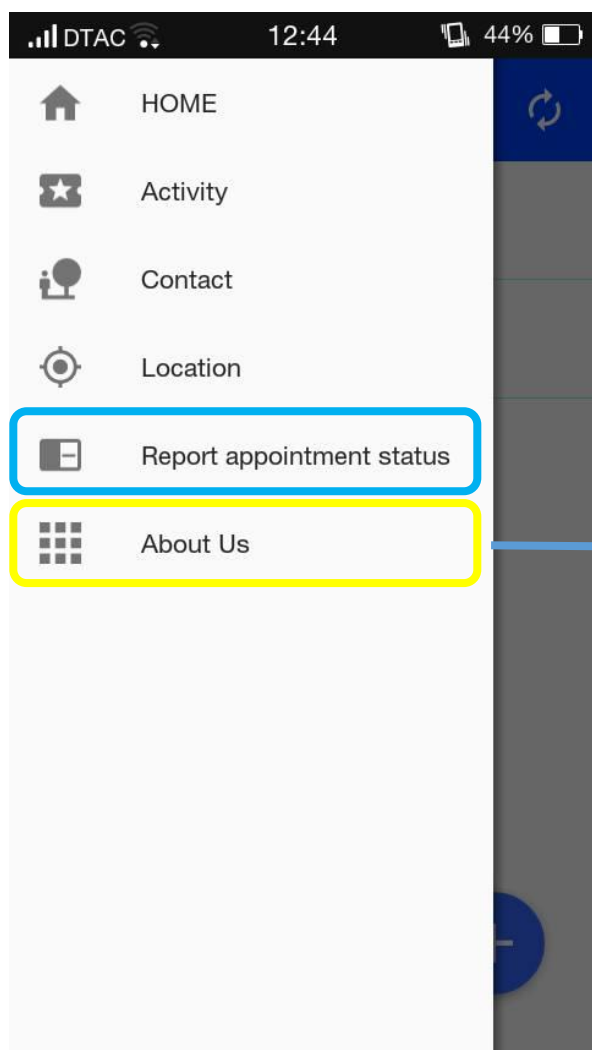




กดไอคอนดินสอ เมื่อต้องการแก้ไขกิจกรรม หรือ กด
ไอคอนถังขยะเมื่อต้องการลบกิจกรรมนี้

หน้าแสดงชื่อกิจกรรม

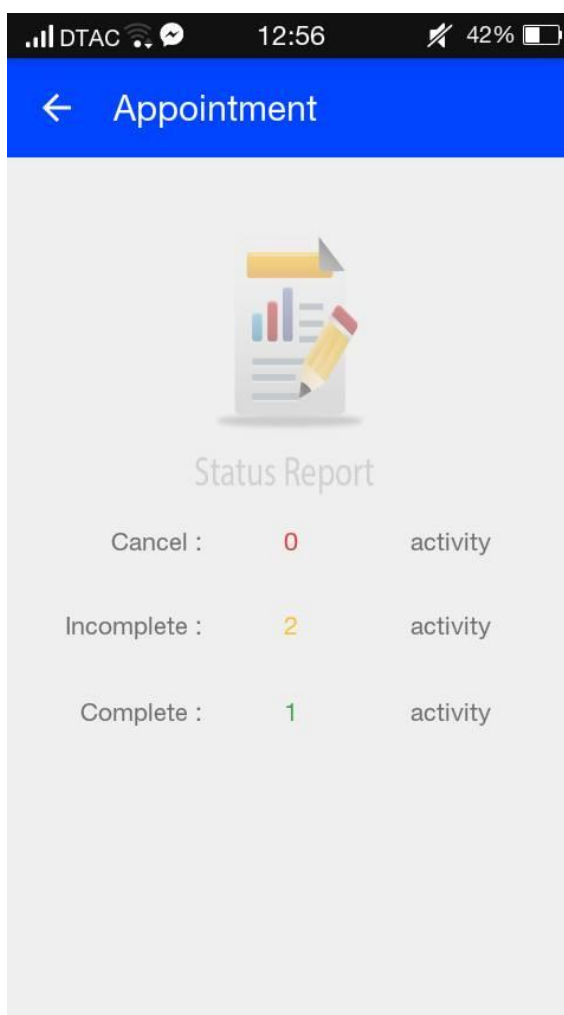
****ในส่วนของการจัดการ ชื่อผู้ติดต่อและชื่อสถานที่มีลักษณะการใช้งานเหมือนกัน
กับการจัดการกิจกรรม**



กดเลือก Report appointment status
เพื่อดูรายงานจำนวนกิจกรรมนัดหมาย แยก
ตามสถานะ

กดเลือก About Us เพื่อดูรายละเอียดของแอปพลิเคชัน

หน้าแสดงรายงานจำนวนกิจกรรมนัดหมาย
แยกตามสถานะ





หน้าแสดงรายละเอียดของแอปพลิเคชัน
Student appointment