

**โครงงานระบบสารสนเทศ**

**(PROJECT IN INFORMATION SYSTEM)**

**การพัฒนาแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน**

จัดทำโดย

นางสาวจุฬาลักษณ์ เชยชื่นจิตร รหัสนักศึกษา 5510513032

อาจารย์ที่ปรึกษา

ดร.จันทวรรณ ปิยะวัฒน์

โครงงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา 477-403 โครงงานระบบสารสนเทศ 2

(PROJECT IN INFORMATION SYSTEM II)

ภาคเรียนที่ 2 ปีการศึกษา 2558

ภาควิชาบริหารธุรกิจ สาขาระบบสารสนเทศ

คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

**บทคัดย่อ**

การบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน เป็นการบันทึกข้อมูลส่วนบุคคล จัดทำขึ้นเพื่อบันทึกกิจกรรมการใช้จ่าย รวมทั้งช่วยเป็นข้อมูลในการวิเคราะห์การใช้จ่ายในชีวิตประจำวัน เพื่อจะให้ทราบถึงพฤติกรรมการใช้จ่ายของตัวเราเอง และนำมาปรับเปลี่ยนพฤติกรรมในการใช้จ่ายให้คุ้มค่ามากขึ้น

ดังนั้นในโครงงานนี้ จึงได้มีแนวคิดที่จะพัฒนาแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวันบนโทรศัพท์มือถือในระบบปฏิบัติการ Android มาช่วยในการบันทึก จากเดิมที่จะต้องบันทึกในรูปแบบของสมุดหรือกระดาษ เปลี่ยนไปเป็นการบันทึกบนโทรศัพท์มือถือแทน เพื่อช่วยให้การบันทึกบัญชีรายรับ-รายจ่าย สามารถดำเนินได้อย่างต่อเนื่องและมีประสิทธิภาพมากยิ่งขึ้น

**กิตติกรรมประกาศ**

โครงงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา 477-403 โครงงานระบบสารสนเทศ 2 (PROJECT IN INFORMATION SYSTEM II) โดยมีจุดประสงค์เพื่อพัฒนาแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน เพื่อให้ง่ายต่อการบันทึกรายรับ-รายจ่าย ทำให้สะดวกต่อการบันทึก แล้วยังสามารถให้ทราบถึงพฤติกรรมการใช้จ่ายเงินของผู้ใช้อีกด้วย

คณะผู้จัดทำขอขอบพระคุณ ดร.จันทวรรณ ปิยะวัฒน์ อาจารย์ที่ปรึกษาโครงงาน ที่กรุณาให้แนวความคิด และคำแนะนำ ตลอดจนการช่วยแก้ปัญหาต่างๆที่เกิดขึ้นระหว่างการดำเนินงานโครงงาน รวมทั้งช่วยตรวจสอบและปรับปรุงเนื้อหาอันเป็นประโยชน์ต่อการจัดทำโครงงานนี้

คณะผู้จัดทำหวังว่า เนื้อหาในโครงงานฉบับนี้จะเป็นประโยชน์ต่อผู้สนใจไม่มากก็น้อย หากโครงงานฉบับนี้ผิดพลาดประการใดต้องขออภัยไว้ ณ โอกาสนี้ด้วย

คณะผู้จัดทำ

**สารบัญ**

**เรื่อง**  **หน้า**

บทคัดย่อ ก

กิตติกรรมประกาศ ข

สารบัญ ค

บทที่ 1 บทนำ

1.1 ชื่อโครงงาน 1

1.2 ชื่อผู้จัดทำโครงงาน 1

1.3 อาจารย์ที่ปรึกษาโครงงาน 1

1.4 หลักการและเหตุผล/แนวคิด/ที่มาและความสำคัญของโครงงาน 1

1.5 เป้าหมาย/จุดมุ่งหมาย/วัตถุประสงค์ของการศึกษาค้นคว้า 10

1.6 กลุ่มเป้าหมาย 11

1.7 กิจกรรมหลักของผู้ใช้ในการใช้งานระบบหรือซอฟต์แวร์ 11

1.8 ขั้นตอนการดำเนินงาน 12

1.9 แผนการปฏิบัติโครงงาน 13

1.10 ขอบเขตในการพัฒนา Application 14

1.11 ผลที่คาดว่าจะได้รับ 14

1.12 ตารางแผนการดำเนินงาน 15

1.13 วิเคราะห์แอพพลิเคชั่นตัวอย่าง 17

บทที่ 2 ทฤษฎีแนวคิดที่เกี่ยวข้องและเครื่องมือที่ใช้ในการพัฒนา

* 1. ทฤษฎีแนวคิดที่เกี่ยวข้อง 21
  2. เครื่องมือที่ใช้ในการพัฒนา 23

**สารบัญ (ต่อ)**

**เรื่อง หน้า**

2.2.1 Android Studio 23

2.2.2 SQLite 24 2.2.3 Illustrator CS6 25

2.2.4 Adobe Photoshop CS6 26

2.2.5 Java 27

2.2.6 Microsoft Access 28

2.2.7 Balsamiq Mockups 29

บทที่ 3 การวิเคราะห์ระบบ

* 1. User Reguirement 30
  2. Hierarchical Task Analysis (HTA) 45
  3. ER Diagram 52
  4. Data Dictionary 52
  5. Essential use case 54
  6. Screen Design 57

บทที่ 4 ผลการดำเนินโครงงาน

การพัฒนาแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน 66

ความสามารถของแอพพลิเคชั่น 66

แอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน 67

**บทที่ 5 สรุปผลการดำเนินงาน**

สรุปผลโครงงาน 71

ข้อดี 71

ข้อจำกัด 71

**สารบัญ (ต่อ)**

**เรื่อง หน้า**

ข้อเสนอแนะ 71

**อ้างอิง**  72

**ภาคผนวก ก ซอร์สโค้ดของแอพพลิเคชั่น 74**

**บทที่ 1**

**บทนำ**

**1.1 ชื่อโครงงาน**

Expense Record: การพัฒนาแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน

Expense Record: (Application Development for Daily Income and Expenses Account)

**1.2 ชื่อผู้ทำโครงงาน**

นางสาวจุฬาลักษณ์       เชยชื่นจิตร      5510513032

**1.3 ชื่อที่ปรึกษาโครงงาน**

ดร.จันทวรรณ ปิยะวัฒน์

**1.4 หลักการและเหตุผล/แนวคิด/ที่มาและความสำคัญของโครงงาน**

จากสภาวะสังคมปัจจุบันที่เต็มไปด้วยกระแสวัตถุนิยมและความฟุ่มเฟือย ทำให้คนไทยหลงเดินทางผิดไปตามกระแสนิยมจนกลายเป็นปัญหา โดยเฉพาะปัญหาหนี้สิน อย่างไรก็ตามคนไทยยังมีทางออก ซึ่งการจะดำรงชีวิตให้อยู่รอดภายใต้สังคมในปัจจุบัน แนวทางหนึ่งที่ประชาชนไทยควรยึดถือคือการพึ่งตนเอง รู้จักความพอประมาณ และไม่ประมาท ตามแนวปรัชญา “เศรษฐกิจพอเพียง” ของพระบาทสมเด็จพระเจ้าอยู่หัวที่ทรงมองเห็นถึงความสำคัญของการสร้างภูมิคุ้มกันให้กับตัวเอง รู้จักความพอมีพอกิน พอมีพอใช้ คำนึงถึงหลักเหตุผลและการประมาณตนเอง พร้อมกับทรงเตือนสติประชาชนคนไทยไม่ให้ประมาท โดยเฉพาะการใช้จ่ายเงินอันเป็นปัจจัยสำคัญในการดำเนินชีวิต

ตารางที่จะแสดงต่อไปนี้เป็นตารางสถิติแสดงรายได้เฉลี่ยต่อเดือนต่อครัวเรือนจำแนกตามภาคและจังหวัด พ.ศ.2539-2556 และตารางแสดงค่าใช้จ่ายเฉลี่ยต่อเดือนต่อครัวเรือนจำแนกตามภาคและจังหวัด พ.ศ.2549-2557 (สำนักงานสถิติแห่งชาติ พบว่า)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ภาค และจังหวัด | 2539  (1996) | 2541  (1998) | 2543  (2000) | 2545  (2002) | 2547  (2004) | 2549  (2006) | 2550  (2007) | 2552  (2009) | 2554  (2011) | 2556  (2013) | Region and province |
| **ทั่วราชอาณาจักร** | 10,779 | 12,492 | 12,150 | 13,736 | 14,963 | 17,787 | 18,660 | 20,904 | 23,236 | 25,194 | Whole Kingdom |
| **ตอนพิเศษ** | 21,947 | 24929 | 25242 | 28239 | 28135 | 33088 | 35007 | 37732 | 41631 | 43058 | Greater Bangkok |
| กรุงเทพมหานคร | 21550 | 26054 | 26909 | 29589 | 29843 | 36658 | 39020 | 42380 | 48951 | 49191 | Bangkok |
| ภาค และจังหวัด | 2539  (1996) | 2541  (1998) | 2543  (2000) | 2545  (2002) | 2547  (2004) | 2549  (2006) | 2550  (2007) | 2552  (2009) | 2554  (2011) | 2556  (2013) | Region and province |
| สมุทรปราการ | 17146 | 18100 | 15745 | 19680 | 19946 | 20382 | 21302 | 23359 | 23798 | 29575 | Samut Prakan |
| นนทบุรี | 36888 | 24211 | 24566 | 29119 | 26658 | 31152 | 32743 | 34626 | 35120 | 30664 | Nonthaburi |
| ปทุมธานี | 15539 | 21793 | 19282 | 22838 | 21530 | 25143 | 26107 | 26686 | 21616 | 33461 | Pathum Thani |
| **ภาคกลาง** | 10907 | 12643 | 13012 | 14128 | 16355 | 19279 | 18932 | 20960 | 20822 | 26114 | Central Region |
| พระนครศรีอยุธยา | 11376 | 12918 | 14904 | 13319 | 14980 | 19676 | 21676 | 25820 | 22302 | 26482 | Phra Nakhan Si Ayutthaya |
| อ่างทอง | 11414 | 10878 | 12544 | 11653 | 12855 | 18300 | 17704 | 25506 | 21140 | 28641 | Ang Thong |
| ลพบุรี | 9331 | 10587 | 10649 | 11010 | 15003 | 19935 | 16852 | 22405 | 17178 | 23426 | Lop buri |
| สิงห์บุรี | 9525 | 10786 | 11894 | 14434 | 14788 | 18510 | 20558 | 25419 | 26068 | 28115 | Sing Buri |
| ชัยนาท | 9575 | 11159 | 12693 | 11119 | 13383 | 13058 | 13995 | 18719 | 20181 | 32754 | Chai Nat |
| สระบุรี | 10582 | 13281 | 2319 | 14677 | 18742 | 20468 | 22363 | 21947 | 27114 | 32834 | Saraburi |
| ชลบุรี | 12223 | 14075 | 13807 | 16797 | 22286 | 21869 | 22260 | 24052 | 23007 | 28367 | Chon Buri |
| ระยอง | 13254 | 12491 | 14739 | 13255 | 18962 | 20018 | 25090 | 22983 | 21929 | 30401 | Rayong |
| จันทบุรี | 12411 | 15548 | 11527 | 15959 | 15897 | 20606 | 18866 | 19442 | 24278 | 27284 | Chanthaburi |
| ตราด | 11696 | 11703 | 7426 | 13822 | 14080 | 20286 | 16664 | 16949 | 28118 | 21653 | Trat |
| ฉะเชิงเทรา | 11820 | 14010 | 13781 | 14829 | 16938 | 16770 | 20665 | 21252 | 23031 | 34548 | Chachoengsao |
| ปราจีนบุรี | 8117 | 9852 | 10477 | 12720 | 15032 | 16031 | 18263 | 22548 | 25338 | 21039 | Prachin Bri |
| นครนายก | 9526 | 11809 | 12595 | 10722 | 13096 | 14104 | 15983 | 17503 | 17042 | 23391 | Nakhon Nayok |
| สระแก้ว | 7189 | 6958 | 8355 | 9951 | 10777 | 11577 | 13593 | 15525 | 17781 | 24805 | Sa Kaeo |
| ราชบุรี | 12621 | 12868 | 15174 | 14692 | 19867 | 23202 | 17576 | 20994 | 18713 | 25742 | Ratchaburi |
| กาญจนบุรี | 10230 | 13082 | 11680 | 15380 | 12122 | 15606 | 15326 | 17571 | 15210 | 18746 | Kanchanaburi |
| สุพรรณบุรี | 7427 | 12409 | 13334 | 12984 | 15781 | 14783 | 15112 | 15797 | 15928 | 17260 | Suphan Buri |
| นครปฐม | 14737 | 15178 | 18297 | 18674 | 20701 | 33835 | 25447 | 24989 | 22955 | 30856 | Nakhon Pathom |
| สมุทรสาคร | 15997 | 17565 | 13774 | 16437 | 15347 | 19555 | 18735 | 20978 | 20850 | 23658 | Samut Sakhon |
| ภาค และจังหวัด | 2539  (1996) | 2541  (1998) | 2543  (2000) | 2545  (2002) | 2547  (2004) | 2549  (2006) | 2550  (2007) | 2552  (2009) | 2554  (2011) | 2556  (2013) | Region and province |
| สมุทรสงคราม | 10925 | 11540 | 13072 | 13876 | 12528 | 16923 | 12634 | 18363 | 15068 | 16257 | Samut Songkhram |
| เพชรบุรี | 10591 | 14135 | 12390 | 15993 | 13040 | 15042 | 17855 | 17440 | 20026 | 21784 | Phetchaburi |
| ประจวบคีรีขันธ์ | 7923 | 10017 | 10840 | 11666 | 13806 | 16238 | 17932 | 16934 | 17477 | 28459 | Prachuap Khiri Khan |
| **ภาคเหนือ** | 8331 | 9779 | 8652 | 9530 | 10885 | 13146 | 13568 | 15727 | 17350 | 19267 | Northern Region |
| เชียงใหม่ | 9806 | 10349 | 9243 | 9582 | 12586 | 14904 | 14386 | 16141 | 18323 | 14393 | Chiang Mai |
| ลำพูน | 8735 | 8722 | 10231 | 11328 | 11843 | 14361 | 14104 | 20048 | 18778 | 22505 | Lamphun |
| ลำปาง | 8208 | 14177 | 9735 | 9759 | 10576 | 15585 | 13531 | 14020 | 18454 | 19262 | Lampang |
| อุตรดิตถ์ | 8181 | 10044 | 7270 | 8581 | 10940 | 14220 | 15001 | 15865 | 17602 | 21344 | Uttaradit |
| แพร่ | 8261 | 8522 | 9366 | 10048 | 11333 | 12591 | 14045 | 13034 | 18157 | 19418 | Phrae |
| น่าน | 6913 | 7387 | 7334 | 8130 | 10751 | 10590 | 11407 | 14624 | 16249 | 17598 | Nan |
| พะเยา | 6129 | 7491 | 5928 | 9520 | 9643 | 10352 | 11348 | 13189 | 14457 | 15491 | Phayao |
| เชียงราย | 7955 | 8924 | 7930 | 8369 | 8920 | 10495 | 13736 | 13278 | 15034 | 13510 | Chiang Rai |
| แม่ฮ่องสอน | 6187 | 5934 | 6317 | 6696 | 8676 | 6544 | 7245 | 7936 | 9024 | 8821 | Mae Hong Son |
| นครสวรรค์ | 8551 | 9500 | 7602 | 9660 | 10200 | 14953 | 15141 | 17127 | 21561 | 24158 | Nakhon Sawan |
| อุทัยธานี | 8203 | 7887 | 9598 | 6519 | 9817 | 13258 | 12036 | 17425 | 18487 | 20150 | Uthai Thani |
| กำแพงเพชร | 8142 | 9769 | 12354 | 12583 | 12776 | 15337 | 15559 | 18115 | 18672 | 22083 | Kamphaeng Phet |
| ตาก | 7604 | 9423 | 9674 | 7763 | 9549 | 10963 | 10791 | 12453 | 12936 | 17020 | Tak |
| สุโขทัย | 8321 | 10475 | 7156 | 8299 | 11427 | 13843 | 12720 | 15411 | 18098 | 24121 | Sukhothai |
| พิษณุโลก | 7586 | 12068 | 8965 | 10637 | 12751 | 14117 | 13364 | 15779 | 17474 | 27490 | Phitsanulok |
| พิจิตร | 10324 | 11185 | 8318 | 10887 | 10951 | 13488 | 15603 | 21821 | 18130 | 21664 | Phichit |
| เพชรบูรณ์ | 8204 | 8051 | 8435 | 9548 | 9572 | 11323 | 12914 | 17096 | 15678 | 19918 | Phetchabun |
| **ภาคตะวันออกเฉียงเหนือ** | 7388 | 8546 | 7765 | 9279 | 10139 | 11815 | 12995 | 15358 | 18217 | 19181 | Northeastern Region |
| ภาค และจังหวัด | 2539  (1996) | 2541  (1998) | 2543  (2000) | 2545  (2002) | 2547  (2004) | 2549  (2006) | 2550  (2007) | 2552  (2009) | 2554  (2011) | 2556  (2013) | Region and province |
| นครราชสีมา | 8803 | 9418 | 8415 | 9494 | 11398 | 13320 | 14177 | 19158 | 19399 | 22479 | Nakhon Ratchasima |
| บุรีรัมย์ | 6446 | 8365 | 8137 | 7861 | 8632 | 9698 | 10263 | 13734 | 17316 | 15624 | Buri Rum |
| สุรินทร์ | 6517 | 6398 | 6226 | 6675 | 8095 | 11850 | 12257 | 11310 | 18287 | 21351 | Surin |
| ศรีสะเกศ | 7149 | 7669 | 7593 | 7564 | 8745 | 9406 | 10782 | 10666 | 13944 | 16207 | Si Sa Ket |
| อุบลราชธานี | 6950 | 10267 | 9874 | 12258 | 11697 | 12035 | 14534 | 16747 | 21660 | 22344 | Ubon Ratchathani |
| ยโสธร | 6582 | 7478 | 4826 | 6045 | 9410 | 9969 | 10040 | 11813 | 16767 | 14418 | Yasothon |
| ชัยภูมิ | 7166 | 6220 | 7683 | 8364 | 9161 | 11551 | 11253 | 12380 | 15794 | 18641 | Chaiyaphum |
| อำนาจเจริญ | 7027 | 8139 | 7711 | 9307 | 11423 | 13134 | 11889 | 14763 | 15975 | 17273 | Amnat Charoen |
| บึงกาฬ | - | - | - | - | - | - | - | - | - | 20207 | Bungkan |
| หนองบัวลำภู | 7147 | 8386 | 5532 | 7326 | 8318 | 11030 | 11889 | 14012 | 17139 | 15390 | Nong Bua Lam Phu |
| ขอนแก่น | 9438 | 11521 | 9757 | 11554 | 13088 | 11349 | 15065 | 19779 | 16030 | 18095 | Khon Kaen |
| อุดรธานี | 7766 | 7744 | 6841 | 9885 | 10789 | 14097 | 17273 | 16684 | 22017 | 25966 | Udon Thani |
| เลย | 6875 | 9905 | 7387 | 9289 | 10013 | 12083 | 13765 | 13411 | 17958 | 16338 | Loei |
| หนองคาย | 8733 | 12548 | 7123 | 11001 | 11466 | 12877 | 12885 | 14467 | 16979 | 18058 | Nong Khai |
| มหาสารคาม | 5171 | 8444 | 8746 | 9642 | 10178 | 13646 | 15812 | 19845 | 25461 | 21644 | Maha Sarakham |
| ร้อยเอ็ด | 6746 | 5987 | 7207 | 9442 | 9817 | 10849 | 11779 | 14932 | 20167 | 20505 | Roi Et |
| กาฬสินธุ์ | 6079 | 6908 | 6469 | 8736 | 8908 | 12470 | 12507 | 11748 | 17293 | 13921 | Kalasin |
| สกลนคร | 7747 | 7660 | 7751 | 10227 | 8971 | 11295 | 11957 | 16679 | 15326 | 16854 | Sakon Nakhon |
| นครพนม | 6196 | 7909 | 5345 | 7613 | 8168 | 9927 | 10009 | 12859 | 14053 | 14310 | Nakhon Phanom |
| มุดดาหาร | 7454 | 9890 | 9485 | 9236 | 9284 | 13497 | 13406 | 17370 | 19048 | 16494 | Mukdahan |
| **ภาคใต้** | 9846 | 11461 | 11186 | 12487 | 14469 | 18668 | 19716 | 22926 | 27326 | 27504 | Southern Region |
| นครศรีธรรมราช | 9681 | 11196 | 10789 | 12317 | 14079 | 17377 | 18087 | 23296 | 29970 | 27479 | Nakhon Si Thammarat |
| กระบี่ | 8758 | 11986 | 9928 | 13318 | 16877 | 18446 | 18852 | 23647 | 33350 | 27276 | Krabi |
| ภาค และจังหวัด | 2539  (1996) | 2541  (1998) | 2543  (2000) | 2545  (2002) | 2547  (2004) | 2549  (2006) | 2550  (2007) | 2552  (2009) | 2554  (2011) | 2556  (2013) | Region and province |
| พังงา | 8696 | 7650 | 10217 | 12428 | 16965 | 29617 | 22211 | 22356 | 24470 | 26168 | Phangnga |
| ภูเก็ต | 15437 | 19313 | 20702 | 26363 | 25082 | 25630 | 25084 | 28515 | 26048 | 31857 | Phuket |
| สุราษฎร์ธานี | 12771 | 11967 | 10800 | 14056 | 16150 | 21747 | 26207 | 30604 | 34417 | 36865 | Surat Thani |
| ระนอง | 9359 | 9033 | 9410 | 10765 | 14328 | 18679 | 21620 | 19274 | 26049 | 32292 | Ranong |
| ชุมพร | 9874 | 10232 | 10454 | 10922 | 11482 | 17232 | 19003 | 25179 | 28022 | 27540 | Chumphan |
| สงขลา | 11089 | 15057 | 13505 | 14192 | 15481 | 22093 | 22342 | 27356 | 26711 | 26824 | Songkhla |
| สตูล | 8977 | 8924 | 9711 | 12702 | 11826 | 14307 | 17328 | 20695 | 21047 | 05400 | Satun |
| ตรัง | 11733 | 12973 | 12782 | 14332 | 17038 | 20801 | 23650 | 21992 | 36245 | 33270 | Trang |
| พัทลุง | 8474 | 12315 | 11550 | 10820 | 14985 | 18045 | 18670 | 19381 | 25202 | 26169 | Phattalung |
| ปัตตานี | 6876 | 10737 | 9759 | 9702 | 11837 | 12547 | 11840 | 13511 | 16122 | 20199 | Pattani |
| ยะลา | 7115 | 7560 | 9932 | 10018 | 11886 | 14354 | 13698 | 19619 | 21859 | 22483 | Yala |
| นราธิวาส | 6715 | 6281 | 7802 | 7603 | 9240 | 11586 | 13148 | 11244 | 16834 | 16773 | Narathiwat |

**ตารางที่1.1 แสดงรายได้เฉลี่ยต่อเดือนจำแนกตามภาคและจังหวัด พ.ศ.2539-2556**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ภาค และจังหวัด** | **2539**  **(1996)** | **2541**  **(1998)** | **2543**  **(2000)** | **2545**  **(2002)** | **2547**  **(2004)** | **2549**  **(2006)** | **2550**  **(2007)** | **2552**  **(2009)** | **2554**  **(2011)** | **2556**  **(2013)** | **2557**  **(2014)** | **Region and province** |
| **ทั่วราชอาณาจักร** | 9,190 | 10,389 | 9,848 | 10,889 | 12,297 | 14,311 | 14,500 | 16,205 | 17,403 | 19,061 | 20,892 | Whole Kingdom |
| **ตอนพิเศษ** | **17,419** | **19,820** | **19,582** | **21,087** | **21,716** | **24,194** | **23,996** | **27,985** | **27,566** | **32,425** | **31,606** | **Greater Bangkok** |
| กรุงเทพมหานคร | 17,846 | 20,780 | 20,448 | 21,919 | 22,982 | 25,749 | 25,615 | 30,334 | 29,970 | 35,024 | 34,426 | Bangkok |
| สมุทรปราการ | 12,421 | 14,005 | 14,297 | 14,836 | 15,669 | 15,683 | 15,910 | 18,450 | 20,158 | 26,193 | 22,747 | Samut Prakan |
| นนทบุรี | 23,365 | 19,647 | 19,088 | 22,702 | 20,357 | 25,950 | 26,414 | 29,487 | 29,099 | 26,947 | 30,812 | Nonthaburi |
| ปทุมธานี | 12,091 | 16,515 | 17,239 | 17,836 | 17,112 | 21,731 | 19,468 | 22,460 | 18,993 | 29,514 | 30,197 | Pathum Thani |
| **ภาค และจังหวัด** | **2539**  **(1996)** | **2541**  **(1998)** | **2543**  **(2000)** | **2545**  **(2002)** | **2547**  **(2004)** | **2549**  **(2006)** | **2550**  **(2007)** | **2552**  **(2009)** | **2554**  **(2011)** | **2556**  **(2013)** | **2557**  **(2014)** | **Region and province** |
| **ภาคกลาง** | **9,510** | **10,861** | **10,389** | **11,227** | **13,270** | **15,373** | **15,168** | **17,106** | **16,954** | **19,728** | **21,144** | **Central Region** |
| พระนครศรีอยุธยา | 10,188 | 11,942 | 11,924 | 11,335 | 11,071 | 15,859 | 16,757 | 20,717 | 17,534 | 20,494 | 20,410 | Phra Nakhon Si Ayutthaya |
| อ่างทอง | 8,063 | 9,300 | 9,799 | 8,834 | 10,249 | 13,318 | 13,431 | 20,773 | 216,591 | 21,183 | 19,634 | Ang Thong |
| ลพบุรี | 6,848 | 9,299 | 9,023 | 8,826 | 13,287 | 15,870 | 14,462 | 16,764 | 14,875 | 15,945 | 14,911 | Lop Buri |
| สิงห์บุรี | 9,863 | 10,589 | 10,589 | 11,710 | 12,899 | 16,209 | 14,988 | 17,787 | 19,207 | 22,118 | 19,632 | Sing Buri |
| ชัยนาท | 9,034 | 10,191 | 8,794 | 9,910 | 12,214 | 13,358 | 12,841 | 14,272 | 17,392 | 17,767 | 17,091 | Chai Nat |
| สระบุรี | 8,665 | 10,604 | 9,987 | 10,814 | 13,492 | 17,063 | 15,467 | 17,378 | 19,377 | 22,765 | 22,811 | Saraburi |
| ชลบุรี | 11,107 | 13,070 | 12,353 | 13,635 | 15,614 | 19,770 | 21,048 | 21,312 | 20,573 | 24,934 | 25,704 | Chon Buri |
| ระยอง | 11,530 | 11,334 | 10,587 | 9,967 | 14,649 | 17,215 | 18,165 | 18,559 | 18,546 | 21,873 | 23,303 | Rayong |
| จันทบุรี | 9,832 | 12,381 | 9,873 | 13,266 | 13,823 | 17,279 | 16,449 | 17,290 | 18,861 | 17,597 | 20,650 | Chanthaburi |
| ตราด | 10,798 | 9,265 | 7,673 | 9,998 | 10,660 | 13,296 | 13,463 | 13,365 | 17,051 | 16,706 | 18,127 | Trat |
| ฉะเชิงเทรา | 12,356 | 13,308 | 12,735 | 13,157 | 15,398 | 14,646 | 16,231 | 19,009 | 17,958 | 26,071 | 23,342 | Chachoengsao |
| ปราจีนบุรี | 6,110 | 8,692 | 9,696 | 11,100 | 13,118 | 14,539 | 16,471 | 18,472 | 22,883 | 18,315 | 20,790 | Prachin Buri |
| นครนายก | 9,228 | 10,452 | 10,651 | 9,255 | 12,141 | 13,091 | 13,584 | 13,574 | 13,807 | 17,697 | 17,483 | Nakhon Nayok |
| สระแก้ว | 6,728 | 6,773 | 7,215 | 6,871 | 7,489 | 8,342 | 9,975 | 12,396 | 15,130 | 18,571 | 20,227 | Sa Kaeo |
| ราชบุรี | 11,219 | 11,641 | 11,682 | 13,205 | 15,984 | 16,961 | 14,786 | 16,951 | 13,847 | 17,253 | 17,515 | Ratchaburi |
| กาญจนบุรี | 9,710 | 10,143 | 8,683 | 9,702 | 10,701 | 13,821 | 12,457 | 15,669 | 13,777 | 17,301 | 17,187 | Kanchanaburi |
| สุพรรณบุรี | 6,643 | 9,258 | 6,392 | 9,783 | 12,901 | 12,645 | 11,892 | 11,924 | 11,908 | 13,450 | 14,869 | Suphan Buri |
| นครปฐม | 12,286 | 14,296 | 15,348 | 15,441 | 18,674 | 19,070 | 18,139 | 17,308 | 15,981 | 15,981 | 25,876 | Nakhon Pathom |
| สมุทรสาคร | 11,699 | 12,543 | 11,537 | 13,146 | 12,759 | 14,197 | 14,076 | 18,074 | 18,164 | 18,565 | 22,560 | Samut Sakhon |
| สมุทรสงคราม | 9,145 | 10,256 | 11,640 | 11,489 | 10,451 | 11,833 | 9,918 | 17,830 | 15,601 | 16,558 | 19,511 | Samut Songkhram |
| **ภาค และจังหวัด** | **2539**  **(1996)** | **2541**  **(1998)** | **2543**  **(2000)** | **2545**  **(2002)** | **2547**  **(2004)** | **2549**  **(2006)** | **2550**  **(2007)** | **2552**  **(2009)** | **2554**  **(2011)** | **2556**  **(2013)** | **2557**  **(2014)** | **Region and province** |
| เพชรบุรี | 7,913 | 9,369 | 9,494 | 10,005 | 10,736 | 13,144 | 13,256 | 14,509 | 18,275 | 18,469 | 20,170 | Phetchaburi |
| ประจวบคีรีขันธ์ | 8,912 | 8,355 | 8,607 | 8,983 | 11,528 | 16,477 | 15,977 | 15,782 | 17,236 | 22,393 | 25,915 | Prachuap Khiri Khan |
| **ภาคเหนือ** | **7,043** | **8,209** | **7,318** | **7,747** | **9,361** | **11,185** | **10,990** | **12,051** | **13,668** | **14,066** | **15,286** | **Northern Region** |
| เชียงใหม่ | 9,040 | 8,600 | 7,477 | 8,465 | 11,121 | 12,606 | 12,480 | 14,234 | 15,607 | 11,704 | 13,732 | Chiang Mai |
| ลำพูน | 7,148 | 8,037 | 8,480 | 9,128 | 10,247 | 12,791 | 12,789 | 16,476 | 16,605 | 20,381 | 20,063 | Lamphun |
| ลำปาง | 6,708 | 9,884 | 7,712 | 6,925 | 8,660 | 12,767 | 11,360 | 10,744 | 14,693 | 15,102 | 16,021 | Lampang |
| อุตรดิตถ์ | 6,546 | 7,841 | 6,729 | 6,707 | 8,754 | 12,974 | 11,217 | 10,275 | 13,433 | 13,550 | 15,220 | Uttaradit |
| แพร่ | 5,418 | 6,532 | 7,960 | 8,005 | 9,480 | 10,454 | 11,491 | 10,695 | 14,135 | 15,479 | 15,455 | Phare |
| น่าน | 6,663 | 6,450 | 6,699 | 7,009 | 10,479 | 11,626 | 11,578 | 12,290 | 14,280 | 14,078 | 13,920 | Nan |
| พะเยา | 5,910 | 6,603 | 6,136 | 7,387 | 7,905 | 9,815 | 10,495 | 10,399 | 12,348 | 11,245 | 11,835 | Phayao |
| เชียงราย | 7,302 | 7,486 | 6,487 | 7,678 | 8,428 | 10,484 | 10,702 | 10,337 | 12,726 | 11,068 | 11,231 | Chiang Rai |
| แม่ฮ่องสอน | 5,045 | 5,249 | 5,225 | 5,884 | 7,684 | 6,147 | 5,333 | 5,283 | 7,224 | 7,405 | 9,686 | Mae Hong Son |
| นครสวรรค์ | 8,991 | 10,402 | 7,723 | 7,854 | 8,307 | 10,045 | 10,699 | 11,683 | 13,848 | 14,268 | 16,193 | Nakhon Sawan |
| อุทัยธานี | 6,551 | 7,000 | 7,247 | 5,732 | 8,125 | 10,260 | 9,726 | 12,000 | 12,710 | 13,083 | 15,690 | Uthai Thani |
| กำแพงเพชร | 7,258 | 9,116 | 10,366 | 10,437 | 11,554 | 13,196 | 12,461 | 13,423 | 13,540 | 16,360 | 17,880 | Kamphaeng Phet |
| ตาก | 6,920 | 7,816 | 7,805 | 6,422 | 8,799 | 9,568 | 8,874 | 9,615 | 10,261 | 13,149 | 13,919 | Tak |
| สุโขทัย | 5,489 | 6,598 | 5,892 | 6,330 | 8,347 | 9,933 | 8,695 | 11,501 | 13,315 | 15,312 | 14,905 | Sukhothai |
| พิษณุโลก | 5,523 | 9,889 | 7,224 | 8,827 | 11,208 | 11,416 | 11,276 | 12,163 | 15,163 | 19,928 | 19,003 | Phitsanulok |
| พิจิตร | 7,918 | 8,825 | 8,179 | 9,024 | 9,454 | 10,840 | 10,160 | 14,798 | 14,623 | 14,791 | 18,687 | Phichit |
| เพชรบูรณ์ | 5,225 | 6,977 | 5,980 | 6,185 | 7,829 | 10,392 | 10,946 | 12,862 | 11,423 | 13,789 | 15,599 | Phetchabun |
| **ภาคตะวันออกเฉียงเหนือ** | **6,661** | **7,237** | **6,546** | **7,550** | **8,484** | **10,316** | **10,920** | **12,260** | **14,375** | **15,092** | **16,284** | **Northeastern Region** |
| **ภาค และจังหวัด** | **2539**  **(1996)** | **2541**  **(1998)** | **2543**  **(2000)** | **2545**  **(2002)** | **2547**  **(2004)** | **2549**  **(2006)** | **2550**  **(2007)** | **2552**  **(2009)** | **2554**  **(2011)** | **2556**  **(2013)** | **2557**  **(2014)** | **Region and province** |
| นครราชสีมา | 7,402 | 6,910 | 6,524 | 7,405 | 9,141 | 10,154 | 11,305 | 13,939 | 14,251 | 15,618 | 17,771 | Nakhon Ratchasima |
| บุรีรัมย์ | 6,452 | 6,687 | 6,784 | 6,677 | 7,585 | 8,632 | 8,537 | 10,427 | 13,629 | 14,343 | 13,952 | Buri Ram |
| สุรินทร์ | 6,961 | 6,540 | 5,888 | 6,027 | 7,020 | 12,028 | 12,116 | 12,038 | 14,401 | 18,584 | 17,069 | Surin |
| ศรีสะเกษ | 7,094 | 7,631 | 6,174 | 6,385 | 7,085 | 9,316 | 9,070 | 8,725 | 11,672 | 13,368 | 14,609 | Si Sa Ket |
| อุบลราชธานี | 5,750 | 8,037 | 7,747 | 8,053 | 8,619 | 9,255 | 10,399 | 11,343 | 16,030 | 13,587 | 15,478 | Ubon Ratchathani |
| ยโสธร | 5,999 | 7,152 | 5,916 | 5,744 | 8,927 | 8,914 | 9,209 | 10,101 | 12,842 | 12,549 | 15,867 | Yasothon |
| ชัยภูมิ | 4,902 | 4,895 | 5,940 | 7,269 | 7,579 | 8,842 | 9,344 | 10,324 | 12,145 | 15,022 | 17,163 | Chaiyaphum |
| อำนาจเจริญ | 6,477 | 7,296 | 6,363 | 7,335 | 9,865 | 10,361 | 10,224 | 12,772 | 14,862 | 14,162 | 17,529 | Amnat Charoen |
| บึงกาฬ |  |  |  |  |  |  |  |  |  | 20,283 | 20,073 | Bungkan |
| หนองบัวลำภู | 6,514 | 7,452 | 5,272 | 6,062 | 7,594 | 11,250 | 11,717 | 13,302 | 16,381 | 13,664 | 17,622 | Nong Bua Lam Phu |
| ขอนแก่น | 7,442 | 9,251 | 7,322 | 8,215 | 10,642 | 9,335 | 11,247 | 13,892 | 12,866 | 13,915 | 15,700 | Khon Kaen |
| อุดรธานี | 7,503 | 5,492 | 5,169 | 7,859 | 8,365 | 12,257 | 14,759 | 17,307 | 19,086 | 21,439 | 21,620 | Udon Thani |
| เลย | 6,607 | 9,026 | 6,560 | 7,487 | 8,500 | 12,187 | 12,341 | 11,181 | 15,155 | 14,916 | 15,789 | Loei |
| หนองคาย | 8,174 | 11,616 | 7,547 | 9,038 | 10,075 | 12,281 | 12,131 | 13,598 | 15,517 | 15,670 | 18,361 | Nong Khai |
| มหาสารคาม | 5,098 | 7,151 | 7,803 | 7,977 | 8,061 | 10,579 | 11,605 | 12,692 | 16,636 | 15,506 | 14,254 | Maha Sarakham |
| ร้อยเอ็ด | 6,690 | 5,432 | 6,462 | 8,146 | 9,282 | 10,053 | 10,638 | 12,176 | 15,476 | 15,348 | 17,762 | Roi Et |
| กาฬสินธุ์ | 6,648 | 7,033 | 6,713 | 8,805 | 7,011 | 10,232 | 10,099 | 9,186 | 13,028 | 12,445 | 12,178 | Kalasin |
| สกลนคร | 6,143 | 6,721 | 6,019 | 8,998 | 8,004 | 11,006 | 10,916 | 12,623 | 12,337 | 13,021 | 13,475 | Sakon Nakhon |
| นครพนม | 5,535 | 6,982 | 5,486 | 7,090 | 8,704 | 10,256 | 9,652 | 10,868 | 12,049 | 11,447 | 12,562 | Nakhon Phanom |
| มุดดาหาร | 6,905 | 9,119 | 8,309 | 8,615 | 8,792 | 12,211 | 11,818 | 13,188 | 15,637 | 13,146 | 15,203 | Mukdahan |
| **ภาค และจังหวัด** | **2539**  **(1996)** | **2541**  **(1998)** | **2543**  **(2000)** | **2545**  **(2002)** | **2547**  **(2004)** | **2549**  **(2006)** | **2550**  **(2007)** | **2552**  **(2009)** | **2554**  **(2011)** | **2556**  **(2013)** | **2557**  **(2014)** | **Region and province** |
| **ภาคใต้** | **8,793** | **9,676** | **9,268** | **10,701** | **12,891** | **15,260** | **15,875** | **17,298** | **19,686** | **20,372** | **21,016** | **Southern Region** |
| นครศรีธรรมราช | 9,328 | 9,122 | 8,746 | 11,368 | 12,815 | 14,918 | 14,990 | 18,763 | 20,756 | 19,364 | 20,661 | Nakhon Si Thammarat |
| กระบี่ | 8,757 | 10,275 | 10,225 | 12,792 | 15,737 | 14,849 | 15,553 | 19,377 | 19,812 | 20,836 | 32,837 | Krabi |
| พังงา | 7,115 | 8,410 | 8,573 | 9,832 | 11,280 | 15,979 | 17,870 | 15,100 | 19,272 | 17,900 | 21,719 | Phangnga |
| ภูเก็ต | 14,006 | 16,864 | 18,054 | 18,689 | 19,665 | 19,492 | 19,329 | 23,783 | 21,255 | 25,338 | 28,111 | Phuket |
| สุราษฎร์ธานี | 10,157 | 9,980 | 9,909 | 11,360 | 14,841 | 16,245 | 18,133 | 19,720 | 22,826 | 28,119 | 26,410 | Surat Thani |
| ระนอง | 9,258 | 9,020 | 9,350 | 9,733 | 12,910 | 15,101 | 16,358 | 16,218 | 23,602 | 22,587 | 18,247 | Ranong |
| ชุมพร | 8,849 | 9,400 | 8,551 | 8,754 | 10,667 | 14,044 | 15,074 | 16,452 | 20,567 | 22,552 | 23,985 | Chumphon |
| สงขลา | 9,042 | 11,385 | 9,761 | 11,601 | 14,738 | 18,111 | 18,668 | 18,929 | 20,612 | 21,670 | 21,744 | Songkhla |
| สตูล | 8,015 | 7,820 | 8,056 | 9,233 | 10,296 | 11,737 | 14,716 | 18,043 | 18,954 | 22,715 | 19,715 | Satun |
| ตรัง | 9,966 | 11,274 | 9,888 | 11,208 | 13,768 | 18,370 | 19,149 | 17,085 | 22,687 | 20,922 | 17,107 | Trang |
| พัทลุง | 8,156 | 10,190 | 9,367 | 9,865 | 12,391 | 18,124 | 15,750 | 17,613 | 18,112 | 16,636 | 18,514 | Phattalung |
| ปัตตานี | 6,363 | 7,910 | 9,415 | 9,496 | 11,097 | 11,490 | 12,531 | 13,784 | 14,012 | 16,668 | 14,809 | Pattani |
| ยะลา | 6,687 | 8,401 | 8,623 | 9,583 | 9,937 | 13,077 | 11,990 | 10,580 | 14,777 | 12,522 | 12,370 | Yala |
| นราธิวาส | 7,016 | 6,603 | 6,325 | 7,193 | 9,040 | 8,201 | 9,711 | 10,486 | 13,901 | 13,728 | 11,986 | Narathiwat |

**ตารางที่ 1.2 แสดงค่าใช้จ่ายเฉลี่ยต่อเดือน จำแนกตามภาคและจังหวัด พ.ศ.2547-2557**

จากตารางข้างบนสังเกตเห็นได้ว่า รายได้เฉลี่ยต่อเดือนต่อครัวเรือนจำแนกตามภาค พ.ศ.2539-2556 มีอัตราที่เพิ่มสูงขึ้นเรื่อยๆ เช่นเดียวกันกับค่าใช้จ่ายเฉลี่ยต่อเดือนต่อครัวเรือนจำแนกตามภาค พ.ศ.2549-2557ที่มีอัตราเพิ่มสูงขึ้นเช่นเดียวกัน แสดงให้เห็นว่าคนไทยมีการใช้จ่ายเงินที่เพิ่มขึ้น ไม่ค่อยบริหารการเงิน ทำให้ความคล่องตัวในด้านการเงินลดลงมีการกู้หนี้ยืมสินเกิดขึ้นส่งผลให้จำนวนหนี้มีอัตราที่สูงขึ้นเช่นกัน ดังนั้นวิธีที่จะใช้ให้สามารถบริหารการเงินใช้จ่ายเงินได้อย่างเป็นระเบียบมากยิ่งขึ้นคือการทำบัญชีรายรับ-รายจ่ายทำให้สามารถรับรู้เงินเข้าและเงินออกจึงสามารถควบคุมการใช้จ่ายเงินได้

(ครูจรรยา   ประจันตะเสน (2558) พบว่า) การจัดทำบัญชี หมายถึง การจดบันทึกรายการเกี่ยวกับการเงินทุกรายการ ทั้งที่ได้รับเข้ามาและจ่ายออกไป เพื่อศึกษาเปรียบเทียบการดำเนินการว่าได้กำไรหรือขาดทุนมากน้อยแค่ไหน

        (นางทิชากร สำราญชลารักษ์ (2558) พบว่า) การทำบัญชีรายรับ-รายจ่าย หมายถึง การจดบันทึกรายการข้อมูลด้านการเงินของการปฏิบัติงาน ทั้งที่เกี่ยวกับรายการที่รับเข้ามาและรายการที่ต้องจ่ายออกไป เพื่อให้มีข้อมูลที่เกี่ยวข้องทางด้านการเงิน ตลอดจนผลของการดำเนินงานนั้นไว้ว่าคงเหลือเงินหรือไม่ จำนวนเท่าไร และเปรียบเทียบผลการดำเนินกิจการว่าได้กำไร หรือขาดทุนเพียงไร

        การทำบัญชีครัวเรือนเป็นการจดบันทึกรายรับรายจ่ายประจำวันของครัวเรือน และสามารถนำข้อมูลมาวางแผนการใช้จ่ายเงินในอนาคตได้อย่างเหมาะสม ทำให้เกิดการออม การใช้จ่ายเงินอย่างประหยัดคุ้มค่า ไม่ฟุ่มเฟือย

(กรมตรวจบัญชีสหกรณ์ (2558) พบว่า) รายได้ หมายถึง ผลตอบแทนที่กิจการได้รับจากการขายสินค้าหรือบริการตามปกติของกิจการรวมทั้ง ผลตอบแทนอื่นๆ ที่ไม่ได้เกิดจากการดำเนินงานตามปกติ รายได้แบ่งออกเป็น 2 ประเภท ดังนี้

1.       รายได้จากการขาย (Sale revenue) หมายถึง รายได้ที่เกิดจากการขายสินค้าหรือบริการอันเป็นรายได้จากการดำเนินงานตาม ปกติ เช่น กิจการซื้อขายสินค้ารายได้ของกิจการ คือ รายได้จากการขายสินค้า ส่วนกิจการให้บริการ เช่นซ่อมเครื่องไฟฟ้า รายได้ของกิจการ คือ รายได้ค่าซ่อม

2.       รายได้อื่น (Other incomes) หมายถึง รายได้ที่มิได้เกิดจากการดำเนินงานตามปกติของกิจการซึ่งเป็นรายได้ที่ไม่ใช้ รายได้จากการขายสินค้าหรือบริการ

(ครูเพียงเดือน วิทยาประดิษฐ์ (2558) พบว่า) ค่าใช้จ่าย หมายถึง ต้นทุนส่วนที่หักออกจากรายได้ในรอบระยะเวลาที่ดำเนินการงานหนึ่ง

การบันทึกบัญชีรายรับ-รายจ่ายในปัจจุบันมีความสะดวกยิ่งขึ้นโดยไม่จำเป็นต้องจดในกระดาษและสามารถบันทึกได้ทุกที่ทุกเวลา  นั้นคือการบันทึกบัญชีรายรับ-รายจ่ายบนโทรศัพท์สมาร์ทโฟนต่างๆ  ซึ่งปัจจุบันสมาร์ทโฟนมีความสำคัญเปรียบเสมือนเป็นส่วนหนึ่งในการดำเนินชีวิตประจำวันและได้รับการพูดถึงกันอย่างมากว่าคนไทยติดสมาร์ทโฟน(Insight Express บริษัทผู้วิจัยตลาดในสหรัฐฯพบว่า)  คนไทยติดสมาร์ทโฟน 98% และขาดสมาร์ทโฟนไม่ได้สมาร์ทโฟนเข้ามามีบทบาทสำคัญอย่างยิ่งในชีวิตประจำวัน สามารถอ่านสมาร์ทโฟนแทนหนังสือพิมพ์รายวัน สามารถเช็คอีเมล์ หรือติดตามความเคลื่อนไหวผ่านโซเชียลเน็ตเวิร์ค กลายเป็นกิจวัตรประจำวันของคนไทยไปเสียแล้ว

**1.5 เป้าหมาย/จุดมุ่งหมาย/วัตถุประสงค์ของการศึกษาค้นคว้า**

* เพื่อใช้บันทึกรายรับ รายจ่ายในชีวิตประจำวัน
* เพื่อทำให้การบันทึกรายรับ รายจ่ายทำได้ ไม่จำกัดสถานที่และเวลา
* คนไทยจะรู้จักบริหารการเงิน
* คนไทยมีนิสัยการใช้เงินที่ดีขึ้น
* จะได้รู้จักการแบ่งเงินเป็นสัดส่วน
* เพื่อใช้ดูรายการรับ-จ่ายต่าง ๆ ที่เกิดขึ้นได้

**1.6**     **กลุ่มเป้าหมาย**

* **(User characteristics, i.e., age, experience, education, etc.)**
* ผู้ที่มีความต้องการใช้เงินจำนวนมาก
* ผู้ที่มีเงินเข้าออกเป็นประจำ
* ผู้ที่มีรายได้น้อยหรือมีรายได้จำกัดต้องจัดสรรเงินให้เพียงพอต่อการใช้จ่ายในแต่ละเดือน
* **(User needs, interests, and goals)**
* เพิ่มความสะดวกสบายในการจดบันทึกข้อมูลมากขึ้น เพียงกรอกตัวเลขลงไปเท่านั้น นอกจากนี้แล้วผู้ใช้งานยังสามารถกำหนดเป้าหมาย วางแผนการเงินได้จากแอพพลิเคชั่นนี้ได้อีกด้วย ซึ่งเป็นประโยชน์มากสำหรับผู้ใช้งาน
* สามารถลดค่าใช้จ่ายที่ไม่จำเป็นได้ เพราะเราสามารถทราบถึงรายละเอียดค่าใช้จ่ายทั้งหมดจากแอพพลิเคชั่นนี้
* จัดสรรเงินออมและเงินลงทุน ได้อย่างมีประสิทธิภาพ
* รู้พฤติกรรมการใช้จ่ายรายเดือนของตัวเองจากการใช้แอพพลิเคชั่นนี้
* **(User environment and context)**
* การใช้แอพพลิเคชั่นนี้สามารถบันทึกรายรับรายจ่ายของเราได้ตลอดเวลา ซึ่งคนส่วนใหญ่ใช้งานโทรศัพท์เป็นประจำอยู่แล้ว เมื่อมีรายรับรายจ่ายก็สามารถบันทึกได้ในทันที มีความสะดวกสบาย
* **(User computer settings, i.e., connection speed, resolution, etc.)**
* สามารถบันทึกรายได้ ค่าใช้จ่ายได้ในทันที ทุกที่ทุกเวลาที่ต้องการ เพื่อเก็บรายละเอียดการใช้จ่ายเงินได้อย่างครบถ้วน

**1.7 กิจกรรมหลักของผู้ใช้ในการใช้งานระบบหรือซอฟต์แวร์นี้**

* การเพิ่มรายการรายรับ-รายจ่าย
* การเพิ่มข้อมูลต่างๆลงในแอพพลิเคชั่น
* การรับรู้ยอดของรายได้และค่าใช้จ่ายในแต่ละวัน
* สามารถโชว์กราฟค่าใช้จ่ายและรายรับได้

**1.8 ขั้นตอนการดำเนินงาน**

**วิธีการดำเนินงานโครงงาน ประกอบด้วยขั้นตอนต่างๆดังนี้**

1. **Planning**
   * วางแผนและวางขอบเขตโครงงาน
   * ศึกษาเครื่องมือที่เกี่ยวข้องกับการทำแอพพลิเคชั่น
   * ศึกษาภาษาที่จะใช้ในการเขียนโค้ดลงในแอพพลิเคชั่น
2. **Analysis**

* วิเคราะห์รูปแบบการจัดเก็บข้อมูล
* วิเคราะห์ผู้ที่มีความเกี่ยงข้องกับการใช้แอพพลิเคชั่น

1. **Design**

* ออกแบบ Logo ของแอพพลิเคชั่น
* ออกแบบ Interface ของแอพพลิเคชั่น

1. **Coding**

* ทำการสร้าง Interface ตามที่ออกแบบไว้
* ทำการสร้างแอพพลิเคชั่นขึ้นมา

1. **Testing**

* ทดสอบการทำงานของ Application ผ่านผู้ใช้งานแอพพลิเคชั่นจริง
* ปรับปรุงแอพพลิเคชั่นหากเกิดข้อผิดพลาด

1. **Implementation**

* นำแอพพลิเคชั่นไปใช้งานจริง

1. **Documentation**

* จัดทำเอกสาร

**เครื่องมือ วัสดุอุปกรณ์**

1. **HARDWARE**

* Asus In search of incredible X555L Inter Core i5-5200U,up to 2.7GHz
* Flash drive SanDisk 32gb

1. **Software**

* Android Studio
* Adobe Photoshop CS6
* SQLite
* Java Development kit
* Adobe Illustrator CS6
* Microsoft PowerPoint
* Microsoft Access
* Balsamiq mockups

1. **Smartphone**

* Samsung Galaxy Grand Prime

**สถานที่ปฏิบัติงาน**

1. ห้องปฏิบัติการคอมพิวเตอร์ คณะวิทยาการจัดการมหาวิทยาลัยสงขลานครินทร์

2. หอพักนักศึกษา มหาวิทยาลัยสงขลานครินทร์

**1.9 แผนการปฏิบัติโครงงาน**

1. จัดทำโครงร่างโครงงานและคำถามสัมภาษณ์ผู้ใช้งาน
2. สัมภาษณ์ผู้ใช้งานจำนวน 5 คน ลำทำการถอดเทป สกัดเนื้อหาจากเทปและวิเคราะห์แอพพลิเคชั่น
3. จัดทำ UML
4. จัดทำ UML และนำเสนอ Proposal พร้อมกับส่งรายงาน Proposal
5. ออกแบบ Hierarchical Task Analysis (HTA)
6. ออกแบบ Database
7. จัดทำ Data Dictionary
8. จัดทำ Essential use case
9. ออกแบบ Screen Design
10. ส่งเอกสารรายงานโครงงาน
11. ปรับแก้เอกสารรายงานโครงงาน
12. นำเสนอโครงงาน

**1.10 ขอบเขตในการพัฒนา Application**

1. สามารถบันทึก แก้ไข ลบ รายรับได้

2. สามารถบันทึก แก้ไข ลบ รายจ่ายได้

3. สามารถดูยอดเงินคงเหลือรวมได้

4. สามารถแสดงกราฟกิจกรรมรายรับ-รายจ่ายได้

**1.11 ผลที่คาดว่าจะได้รับ**

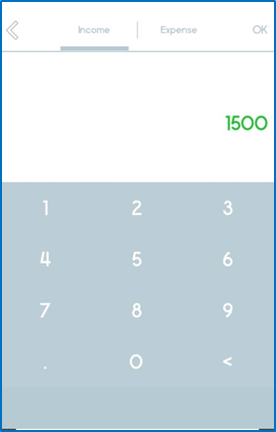
1. เพื่ออำนวยความสะดวกในการบันทึกรายรับ-รายจ่ายที่เกิดขึ้นในแต่ละงาน
2. เพื่อช่วยในการวิเคราะห์การใช้จ่ายเงินของผู้ใช้ได้
3. เพื่อเพิ่มความสะดวกในการตรวจสอบการใช้จ่ายย้อนหลังได้
4. ช่วยให้คนไทยรู้จักบริหารการเงิน
5. ช่วยให้คนไทยมีนิสัยในการใช้เงินที่ดีขึ้น
6. ช่วยให้สามารถแบ่งเงินเป็นสัดส่วนตามการใช้งานของแต่ละเดือน

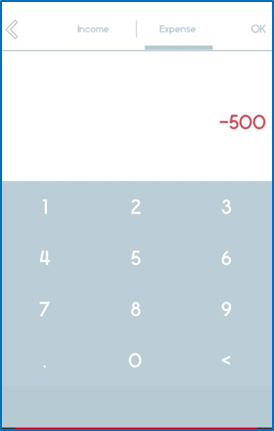
**1.12 ตารางแผนการดำเนินงานโครงงาน 1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ขั้นตอนการดำเนินงาน** | **เดือน (2558-2559)** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **ก.ย.** | | | | **ต.ค.** | | | | **พ.ย.** | | | | | **ธ.ค.** | | | | **ม.ค.** | | | | **ก.พ.** | | | | **มี.ค.** | | | | | **เม.ย.** | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| **Planning** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| วางแผนและวางขอบเขตโครงงาน |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| ศึกษาการใช้เครื่องมือและภาษาที่เกี่ยวข้องกับการทำแอพพลิเคชั่น |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| **Analysis** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| วิเคราะห์กระบวนการทำงานของแอพพลิเคชั่น |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| วิเคราะห์รูปแบบการจัดเก็บข้อมูล |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| วิเคราะห์ผู้มีส่วนเกี่ยวข้องกับระบบ |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| **Design** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ออกแบบ Interface ของแอพพลิเคชั่น |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| ออกแบบ Logo แอพพลิเคชั่น |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| **Coding** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ทำการสร้างแอพพลิเคชั่นตามที่ออกแบบไว้ |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| **ขั้นตอนการดำเนินงาน** | **เดือน (2558-2559)** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ก.ย. | | | | ต.ค. | | | | | พ.ย. | | | | ธ.ค. | | | | ม.ค. | | | | ก.พ. | | | | | มี.ค. | | | | เม.ย. | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| **Testing** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ทดสอบการทำงานของแอพพลิเคชั่นผ่านผู้ใช้จริง |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| ปรับปรุงแอพพลิเคชั่นหากเกิดข้อผิดพลาด |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| **Implementation** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| นำแอพพลิเคชั่นไปใช้งานจริง |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |
| **Document** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| จัดทำเอกสาร |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |

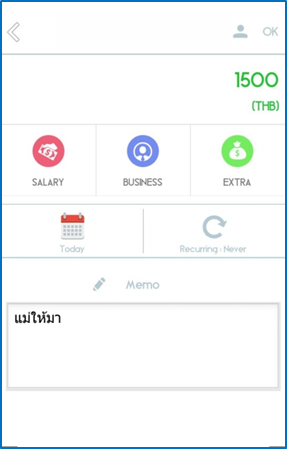
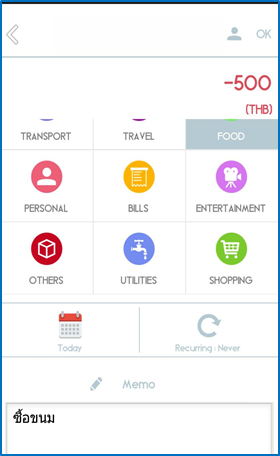
**1.13 วิเคราะห์แอพพลิเคชั่นตัวอย่าง**

1. สามารบันทึกรายรับ-รายจ่ายได้

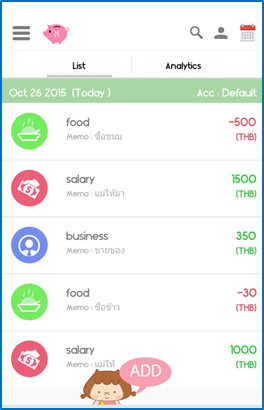
****

****

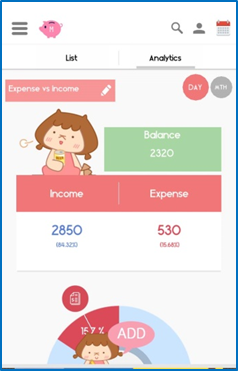
1. สามารถบันทึกแยกเป็นหมวดหมู่ได้และสามารถใส่รายละเอียดของรายการได้

****

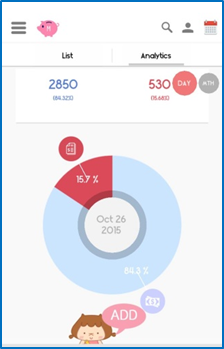
1. สามารถดูรายการที่บันทึกในแต่ละวันได้

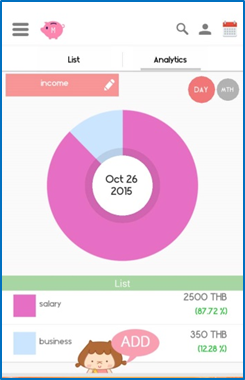
****

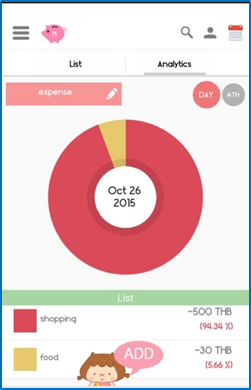
1. คำนวณและแสดงยอดเงินคงเหลือได้



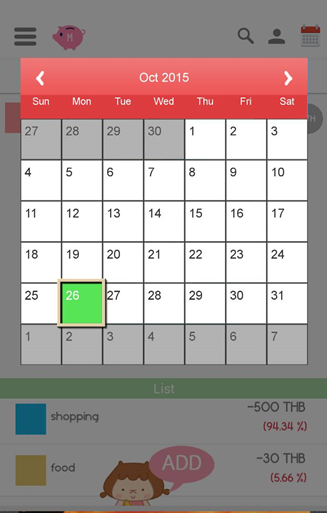
1. แสดงผลออกมาในรูปแผนภูมิวงกลมได้ โดยสามารถดูได้ทั้งแบบเปรียบระหว่างรายรับ-รายจ่ายและสามารถดูแยกกันระหว่างรายรับ-รายจ่ายได้

****

****

****

1. สามารถเลือกวันที่ที่ต้องการดูผลได้



**ข้อดี**

1. สามารถบันทึกแยกเป็นหมวดหมู่ได้และมีไอคอนสำหรับบอกหมวดหมู่
2. สามารถแสดงผลของการบันทึกรายการและยอดเงินออกมาในรูปแบบแผนภูมิวงกลมได้
3. มีฟังก์ชันการทำงานเท่าที่จำเป็นและใช้งานง่าย
4. มีภาษาให้เลือก2ภาษาทั้งภาษาไทยและอังกฤษทำให้ง่ายต่อความเข้าใจและใช้งานของผู้ใช้

**ข้อเสีย**

1. ต้องมีค่าใช้จ่ายเพิ่มเติมถ้าหากต้องการฟังก์ชั่นการทำงานที่ดีกว่านี้

**บทที่ 2**

**ทฤษฎีแนวคิดที่เกี่ยวข้องและเครื่องมือที่ใช้ในการพัฒนา**

**2.1 ทฤษฎีแนวคิดที่เกี่ยวข้อง**

จากแนวความคิดดังกล่าวที่ได้กล่าวมาข้างต้น ทางกลุ่มจึงได้จัดทำแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน เพื่อให้ผู้ใช้ได้สามารถรู้จักการบันทึกรายรับ-รายจ่าย เพื่อที่จะทำให้ผู้ใช้สามารถรับรู้ยอดเงิน ณ ปัจจุบัน และสามารถทราบว่าในแต่ละวันผู้ใช้ได้ใช้จ่ายอะไรไปบ้าง เพื่อนำมาปรับปรุงแก้ไขการใช้จ่ายในครั้งต่อไป อีกทั้งเป็นการทำให้ผู้ใช้สามารถมีเงินเก็บหรือเป็นคนที่รู้จักการใช้จ่ายเงินมากขึ้น สามารถทำให้ผู้ใช้รู้รายละเอียดของรายจ่ายและนำไปตัดค่าใช้จ่ายที่ไม่เป็นประโยชน์ออกไป

ทฤษฎีแนวคิดที่เกี่ยวข้องในการพัฒนาแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวันมีส่วนสอดคล้องกับการทำบัญชีทั่วไป

**2.1 บัญชีครัวเรือน**

บัญชีครัวเรือน ถือเป็นเครื่องมือประเภทหนึ่งของปรัชญาเศรษฐกิจพอเพียง ซึ่งเป็นปรัชญาของพระบาทสมเด็จพระเจ้าอยู่หัวภูมิพลอดุลยเดช หมายถึง เศรษฐกิจที่มีความพอดีต่อตนเอง พอกินพอใช้ไม่เดือนร้อน ทำตามความเหมาะสม โดยยึดทางสายกลาง มีเหตุมีผล โดยมีความรู้และคุณธรรมเป็นพื้นฐาน และสามารถสร้างภูมิคุ้มกันกับตัวเองได้

บัญชีครัวเรือน เป็นการบันทึกรายรับ-รายจ่ายส่วนบุคคลหรืออาจจะเป็นของธุรกิจขนาดย่อม เพื่อให้ทราบถึงกำไรหรือขาดทุนจากการประกอบธุรกิจนั้นๆ

บัญชีครัวเรือนกับชีวิตประจำวัน การดำเนินชีวิตประจำวันย่อมมีรายรับและรายจ่ายต่างๆที่เกิดขึ้น รายรับได้มาจากการทำมาหาเลี้ยงชีพทั้งจากอาชีพหลักและอาชีพรองส่วนรายจ่ายก็ได้แก่ค่าใช้จ่ายต่างๆได้แค่ค่าอุปโภคบริโภคที่จำเป็นต่อการดำรงชีวิต

การบันทึกรายรับ-รายจ่ายนั้น เป็นประโยชน์อย่างมาก ซึ่งจะเปรียบเสมือนเป็นเครื่องมือที่ช่วยให้จัดเก็บเงินได้อย่างเป็นระบบ จะทำให้สามารถเห็นกิจกรรมและพฤติกรรมการใช้จ่ายเงิน รวมถึงการเปลี่ยนแปลงต่างๆที่เกิดขึ้น จนวิเคราะห์นิสัยการใช้เงินของแต่ละบุคคล ช่วยให้สร้างนิสัยการใช้เงินที่ดีขึ้น เพราะกำหนดเป้าหมายได้ชัดเจน และยังสร้างวินัยทางการขึ้น จัดการเงินได้อย่างมีประสิทธิภาพมากขึ้น เราสามารถบันทึกรายละเอียดการใช้จ่ายแบบง่ายๆในสมุดได้ดังต่อไปนี้

1. วันที่ ใช้บันทึกวันที่ เดือน ปี ที่ได้มีการรับเงินหรือจ่ายเงินจริง
2. รายการ ใช้บันทึกคำอธิบายหรือรายระเอียดที่รับเงินหรือจ่ายเงิน และจำเป็นที่จะต้องบอกจำนวนเงินที่เกิดขึ้นจากการรับหรือจ่ายเงินนั้นด้วย ซึ่งสามารถแบ่งออกเป็น 2 ส่วนหลักๆคือ
   * + - รายรับ ใช้บันทึกจำนวนเงิน รายได้ที่หามาได้ทั้งหมด เช่น เงินเดือน เงินที่ได้จากการขายสินค้า เงินที่ได้จากการทำงานต่างๆ
       - รายจ่าย ใช้บันทึกจำนวนเงิน ค่าใช้จ่ายที่ทำให้เงินลดลงไป เช่น ค่าอาหาร ค่าเดินทาง ค่าที่พัก ค่าน้ำ ค่าไฟ เป็นต้น
3. เงินคงเหลือ ใช้บันทึกจำนวนเงิน ที่เป็นส่วนต่างที่เกิดขึ้นจากรายรับหรือรายจ่าย ควรมีการบันทึกทุกครั้งที่มีการรับหรือจ่ายเกิดขึ้น เงินคงเหลือนี้จะบอกถึงจำนวนเงินปัจจุบันที่มีอยู่ วิธีการคำนวณคือ

* คงเหลือจากรายรับ = เงินคงเหลือล่าสุด + รายรับ
* คงเหลือจากรายจ่าย = เงินคงเหลือล่าสุด – รายจ่าย

1. รายละเอียดเพิ่มเติม ใช้บันทึกคำอธิบายและรายละเอียดเพิ่มเติมจากรายการ
2. รวมรายรับ-รายจ่ายและยอดคงเหลือ ใช้บันทึกจำนวนเงินของแต่ละรายการที่จะต้องนำจำนวนเงินไปคำนวณ

ทั้งนี้การบันทึกรายรับ-รายจ่ายในสมุดนั้น ต้องใช้ความระมัดระวังรอบคอบในการกรอกรายการและคำนวณเงินคงเหลือ อาจทำให้ได้รับผลลัพธ์ที่ผิดพลาดได้

ในปัจจุบันนี้ ยังสามารถประยุกต์ใช้กับโปรแกรมคอมพิวเตอร์สำเร็จรูปในการบันทึกรายรับ-รายจ่ายได้เอง เช่น ไมโครซอฟท์เอ็กเซล (Microsoft Excle) เพื่อเพิ่มความสะดวกในการคำนวณ เพียงกรอกรายละเอียดของกิจกรรมที่รับหรือจ่าย และจำนวนเงินที่รับหรือจ่ายในแต่ละครั้ง ส่วนคงเหลือ เราสามารถใช้สูตรในการคำนวณได้ โดยไม่ต้องกรอกเอง จึงช่วยลดความยุ่งยากและผิดพลาดในการคำนวณต่างๆได้ ทั้งนี้ผู้ใช้งานควรมีความรู้วิธีการใช้งานโปรแกรมด้วย

* 1. **เครื่องมือที่ใช้ในการพัฒนา**

**2.2.1 Android Studio**

****

**ภาพที่ 2.1 สัญลักษณ์โปรแกรม** Android Studio

Android Studio เป็น IDE Tool ล่าสุดจาก Google ไว้สำหรับพัฒนาโปรแกรม Android โดยเฉพาะ โดยพัฒนาจากแนวคิดพื้นฐานมาจาก InteliJ IDEA คล้ายๆ กับการทำงานของ Eclipse และ Android ADT Plugin โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนา App บน Android ให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถ Preview ตัว App มุมมองที่แตกต่างกันบน Smart Phone แต่ล่ะรุ่น สามารถแสดงผลบางอย่างได้ทันทีโดนไม่ต้องทำการรัน App บน Emulator รวมทั้งยังแก้ไขปรับปรุงในเรื่องของความเร็วของ Emulator ที่ยังเจอปัญหากันอยู่ในปัจจุบัน โดยมีความสามารถในการทำงานเด่นๆดังต่อไปนี้

* มีความยืดหยุ่นในการใช้งานด้วยใช้ Gradle-baseed
* การสร้างตัวแปรและการสร้างไฟล์ APK ในหลาย ๆ แม่แบบ
* แม่แบบที่ช่วยอำนวยความสะดวกในการใช้งานคุณสมบัติที่ถูกใช้งานบ่อย ๆ
* ตัวช่วยแก้ไขรูปแบบ (Layout) ที่รองรับการลากและวาง
* เครื่องมือที่จะตรวจจับประสิทธิภาพการใช้งาน,การทำงานร่วมกันรุ่นและปัญหาอื่น ๆ
* การสนับสนุนบิวท์อินสำหรับแพลตฟอร์ม Google Cloud ทำให้มันง่ายต่อการรวม Google Cloud Messaging และ App Engine
* และอื่น ๆ อีกมากมาย

**ข้อดีของ Android Studio**

1. สนับสนุนการติดตั้งหลากหลายระบบปฏิบัติการ (Platform) Windows, Mac OS X และ Linux
2. เครื่องมือช่วยพัฒนาการออกแบบส่วนติดต่อกับผู้ใช้และสามารถจะพรีวิวให้ดูแบบหลายๆขนาดหน้าจอได้พร้อม เป็นแบบ Live Preview
3. ระบบการคาดเดาการพิมพ์โค้ดที่ชาญฉลาด
4. ลดขั้นตอนที่เคยพัฒนาใน eclipse
5. ความเร็วของ Android Visual Device
   * 1. **SQLite**

****

**ภาพที่ 2.2 สัญลักษณ์โปรแกรม** SQLite

SQLite มันก็คือ Library ตัวหนึ่งที่มีขนาดเล็กมาก คอยจัดการDatabase ให้กับเรา โดยใช้Syntax SQL เหมือนกับ MySQL เลย Data Type ที่ SQLite ซับพอร์ตก็คือ text เหมือนกับ String ใน Java ,Integer เก็บค่าตัวเลขเหมือนกับ long และ REAL เก็บค่าเหมือนกับ double สำหรับใน SQLite ใน Android นั้นมันติดมากับเครื่อง Android ทุกเครื่อง เวลาเขียนแอพพลิเคชั่นเพื่อเชื่อมต่อกับSQLite ก็ไม่จำเป็นต้องโหลดหรือทำอะไรเพิ่ม เพราะว่าทาง Android นั้นก็ได้เตรียม Class ต่างอำนวยความสะดวกให้แล้ว

ฐานข้อมูลSQLite เป็น Database ขนาดเล็กที่ได้รับความนิยมอย่างมากกับ Application ที่ทำงานบน Smart Phone ประเภทต่าง ๆ รูปแบบการทำงานของ SQLite เป็นแบบ Standalone ทำงานอยู่ใน Application นั้น ๆ SQLite มีโครงสร้างง่ายต่อการจัดเก็บและนำไปใช้ และไฟล์ที่จัดเก็บนั้นก็มีขนาดเล็กมาก เกือบเท่ากับการเก็บข้อมูลจริง เพราะฉะนั้น SQLite Database จึงเหมาะสมกับ Application ที่ทำงานบน Smartphone อย่างยิ่ง โดยเฉพาะ อันเนื่องจากข้อจำกัดทางด้าน Hardware และ Memory รวมทั้งความสามารถในการ Process ข้อมูลต่าง ๆ ใน Smartphone ย่อมน้อยกว่า PC Desktop เป็นธรรมดา

ใน Android จะมี Package ของ android. Database ซึ่งเป็น Library ที่สามารถเรียกใช้งานเกี่ยวกับ Database ของ SQLite ได้ในทันที โดยการเริ่มทำงานบน SQLite จะเริ่มต้นการการประกาศ Class ของ SQLite Open Helper ซึ่งเป็น Class ที่ทำหน้าที่ติดต่อกับฐานข้อมูลได้

* + 1. **Illustrator CS6**

****

**ภาพที่ 2.3 สัญลักษณ์โปรแกรม Illustrator CS6**

Illustrator เป็นโปรแกรมที่ใช้ในการวาดภาพที่มีลักษณะเป็นลาย เส้น หรือเวกเตอร์และยังสามารถรวมภาพกราฟิก ทั้งแบบเวกเตอร์และแบบบิทแมพเข้าด้วยกัน ให้เป็นงานกราฟิกที่มีทั้งภาพเป็นเส้นที่คมชัด

Illustrator คือ โปรแกรมที่ใช้ในการวาดภาพ โดยจะสร้างภาพที่มีลักษณะเป็นลายเส้น หรือที่เรียกว่า Vector Graphic จัดเป็นโปรแกรมระดับมืออาชีพที่ใช้กันเป็นมาตรฐานในการออกแบบระดับสากล สามารถทำงานออกแบบต่างๆ ได้หลากหลาย ไม่ว่าจะเป็นสิ่งพิมพ์ บรรจุภัณฑ์ เว็บ และภาพเคลื่อนไหวตลอดจนการสร้างภาพเพื่อใช้เป็นภาพประกอบในการทำงานอื่นๆ เช่น การ์ตูน ภาพประกอบหนังสือ เป็นต้น

การประมวลผลภาพกราฟฟิกของคอมพิวเตอร์ ในความเป็นจริงแล้วภาพกราฟฟิกที่เก็บในคอมพิวเตอร์นั้น จะมีวิธีการประมวลผลภาพ 2แบบแตกต่างกันไปตามแต่ละโปรแกรมดังนี้ คือ

1. การประมวลผลแบบอาศัยการคำนวณทางคณิตศาสตร์

ตัวอย่างเช่น รูปล้อรถจักรยาน ถ้าเป็นการเก็บแบบเวกเตอร์ เครื่องจะเก็บข้อมูลที่เป็นสูตรทางคณิตศาสตร์แต่สามารถอธิบายได้คือ รูปล้อจักรยาน คือเส้นวงกลมที่เกิดจากการวัดความห่างจากจุดศูนย์กลางจุดหนึ่งไปยังบริเวณ รอบๆ ด้วยระยะห่างทีเท่ากัน โดยมีสีและตำแหน่งของสีที่แน่นอน ฉะนั้นไม่ว่าเราจะมีการเคลื่อนย้ายที่หรือย่อขยายขนาดของภาพ ภาพจะไม่เสียรูปทรงในเชิงเลขาคณิต เช่น โปรแกรม Illustrator ใช้วิธีนี้ในการเก็บ เป็นต้น

ตัวอย่างงานกราฟฟิกของภาพแบบเวกเตอร์งานกราฟฟิกในแบบเวกเตอร์นี้จะเป็น ลักษณะของภาพลายเส้น ซึ่งงานเหล่านี้จะเน้นถึงความ คมชัดของเส้นเป็นหลัก เช่น ภาพโลโก้ ตราบริษัท และภาพลายเส้นแบบคลิปอาร์ต โปรแกรมเหล่านี้ได้แก่ Illustrator, CorelDraw และ Freehand เป็นต้น

2. การเก็บและแสดงผลแบบบิตแม็พ (Bitmap)

เป็นการประมวลผลแบบอาศัยการอ่านค่าสีในแต่ละพิกเซล ซึ่งเหมาะกับภาพที่มีโทนสีที่ใกล้เคียงกัน เช่น ภาพถ่าย โปรแกรม Photoshop ใช้วิธีนี้ การประมวลผลแบบบิตแม็พนี้เราเรียกอีกอย่างว่า Raser image เป็นการเก็บข้อมูลดิบ คือค่า 0 และ 1 ใช้การแสดงผลเป็นพิกเซล โดยแต่ละพิกเซลจะมีการเก็บค่าสีที่เจาะจงในแต่ละตำแหน่ง

**สิ่งที่ Illustrator สามารถทำได้ก็คือ**

งานสิ่งพิมพ์ - ไม่ว่าจะเป็นงานโฆษณา โบร์ชัวร์ นามบัตร หนังสือ หรือนิตยสาร เรียกได้ว่าเกือบทุกสิ่งพิมพ์ที่ต้องการความคมชัด

งานออกแบบทางกราฟฟิก - การสร้างภาพสามมิติ การออกแบบปกหนังสือ การออกแบบสกรีน CD-ROM และการออกแบบการ์ดอวยพร ฯลฯ

งานทางด้านการ์ตูน - ในการสร้างภาพการ์ตูนต่างๆนั้น โปรแกรม Illustrator ได้เข้ามามีบทบาทและช่วยในการวาดรูปได้เป็นอย่างดี

งานเว็บไซต์บนอินเตอร์เน็ต - ใช้สร้างภาพตกแต่งเว็บไซต์ไม่ว่าจะเป็น Background หรือปุ่มตอบโต้ แถบหัวเรื่องตลอดจนภาพประกอบต่างๆ ที่ปรากฏบนหน้าเว็บ

* + 1. **Adobe Photoshop CS6**



**ภาพที่ 2.4 สัญลักษณ์โปรแกรม** Adobe Photoshop CS6

โปรแกรม Photo shop เป็นโปรแกรมในตระกูล Adobe ที่ใช้สำหรับตกแต่งภาพถ่ายและภาพกราฟฟิก ได้อย่างมีประสิทธิภาพ ไม่ว่าจะเป็นงานด้านสิ่งพิมพ์ นิตยสาร และงานด้านมัลติมีเดีย อีกทั้งยังสามารถ retouching ตกแต่งภาพและการสร้างภาพ สามารถใช้โปรแกรม Photoshop ในการตกแต่งภาพ การใส่ Effect ต่าง ๆให้กับภาพ และตัวหนังสือ การทำภาพขาวดำ การทำภาพถ่ายเป็นภาพเขียน การนำภาพมารวมกัน การ Retouch ตกแต่งภาพต่าง Adobe Photoshopสามารถใช้ในการทำการแก้ไขภาพ ตกแต่งภาพ ซ้อนภาพในรูปแบบต่างๆและการใส่ข้อความประกอบลงในภาพ

**Adobe Photoshop CS6** เป็นโปรแกรมในชุด Creative Suite 6 หรือเรียกสั้น ๆ ว่า CS6 โปรแกรมสร้างและแก้ไขรูปภาพ อย่างมืออาชีพ โดยเฉพาะนักออกแบบในทุกวงการย่อมรู้จักโปรแกรมตัวนี้ดี โปรแกรม Photoshop เป็นโปรแกรมที่มีเครื่องมือมากมายเพื่อสนับสนุนการสร้างงานประเภทสิ่งพิมพ์ งานวิดีทัศน์ งานนำเสนอ งานมัลติมีเดีย ตลอดจนงานออกแบบและพัฒนาเว็บไซต์ ขณะนี้โปรแกรม Photoshop ได้พัฒนามาถึงรุ่น Adobe Photoshop CS6 พร้อมกับเครื่องมือที่สร้างความสะดวกสบายให้กับผู้ใช้อย่างมากมาย

**ความสามารถพื้นฐานของ Adobe Photoshop ที่ควรทราบ**

* ตกแต่งหรือแก้ไขรูปภาพ
* ตัดต่อภาพบางส่วน หรือที่เรียกว่า crop ภาพ
* เปลี่ยนแปลงสีของภาพ จากสีหนึ่งเป็นอีกสีหนึ่งได้
* สามารถลากเส้น แบบฟรีสไตล์ หรือใส่รูปภาพ สี่เหลี่ยม วงกลม หรือสร้างภาพได้อย่างอิสระ
* มีการแบ่งชั้นของภาพเป็น Layer สามารถเคลื่อนย้ายภาพได้เป็นอิสระต่อกัน
* การทำ cloning ภาพ หรือการทำภาพซ้ำในรูปภาพเดียวกัน
* เพิ่มเติมข้อความ ใส่ effect ของข้อความได้
* Brush หรือแปรงทาสี ที่สามารถเลือกรูปแบบสำเร็จรูปในการสร้างภาพได้และอื่นๆ อีกมากมาย
  + 1. **Java**



**ภาพที่ 2.5 สัญลักษณ์โปรแกรม** **Java**

Java คือภาษาคอมพิวเตอร์ที่ได้รับการออกแบบ สำหรับการใช้บนอินเตอร์เน็ต โดยมีส่วนของการ "look and feel" แบบภาษา C++ แต่ง่ายกว่าการใช้ C++ และสามารถสร้างมุมมองโดยโปรแกรมได้ Java สามารถใช้ในการสร้างการประยุกต์แบบสมบูรณ์ ซึ่งสามารถเรียกใช้ได้เฉพาะเครื่องคอมพิวเตอร์ หรือการกระจายระหว่างเครื่องแม่ข่ายกับลูกข่ายในระบบเครือข่าย และสามารถสร้างโมดูลการประยุกต์ขนาดเล็กหรือ applet สำหรับเป็นส่วนของเว็บเพจ applet ทำให้มีความเป็นได้ในคอมไพล์จะแปลงด้านการตอบสนองของผู้ใช้กับเว็บเพ็จคุณลักษณะคือ**โ**ปรแกรมมีขนาดเล็กในระบบเครือข่ายการที่โปรแกรมเป็น Java byte code ซึ่งสามารถเรียกใช้งานได้ทุกที่ภายในเครือข่าย Java virtual machine เป็นตัวแปร byte code ให้เป็นโปรแกรมเพื่อใช้งานบนเครื่องคอมพิวเตอร์มีความหมายว่า platformที่แตกต่างกันของคอมพิวเตอร์ สามารถใช้โปรแกรมนี้ได้

คำสั่งเป็นแบบ "Robust" มีความหมายว่า อ็อบเจคของ Java ไม่มีการอ้างอิงข้อมูล หรืออ็อบเจ็คจากภายนอกซึ่งแตกต่างจาก C++ และภาษาอื่น ๆ เป็นการทำให้มั่นใจไม่มีการเก็บตำแหน่งของข้อมูล ในโปรแกรมประยุกต์อื่น หรือในระบบปฏิบัติการที่ทำให้โปรแกรมไม่ทำงาน Java virtual machine ทำการตรวจสอบแต่ละอ็อบเจ็คที่ใช้ในโปรแกรม

Java เป็นเหมือนกับอ็อบเจคหนึ่ง สามารถได้รับประโยชน์จาก class หรือคำสั่งเนื่องจากอ็อบเจ็คมีคุณสมบัติเป็น "นาม" ซึ่งทำให้ติดต่อกับผู้ใช้ได้ ในขณะนี้ภาษาดั้งเดิมมีคุณสมบัติเป็น "กริยา" ดังนั้น method จะได้การรับรู้เป็นความสามารถของอ็อบเจ็คหรือพฤติกรรม

* + 1. **Microsoft Access**



**ภาพที่ 2.6 สัญลักษณ์โปรแกรม Microsoft Access**

ไมโครซอฟท์แอคเซส (Microsoft Access) คือ โปรแกรมเพื่อพัฒนาระบบฐานข้อมูล มีตารางเก็บข้อมูลและสร้างแบบสอบถามได้ง่าย มีวัตถุคอนโทลให้เรียกใช้ในรายงานและฟอร์ม สร้างมาโครและโมดูลด้วยภาษาเบสิก เพื่อประมวลผลตามหลักภาษาโครงสร้าง สามารถใช้โปรแกรมนี้เป็นเพียงระบบฐานข้อมูลให้โปรแกรมจากภายนอกเรียกใช้ก็ได้

การใช้ Access จะทำให้คุณสามารถ

* เพิ่มข้อมูลใหม่ลงในฐานข้อมูล เช่น รายการใหม่ในสินค้าคงคลัง
* แก้ไขข้อมูลที่มีอยู่ในฐานข้อมูล เช่น การเปลี่ยนตำแหน่งที่ตั้งปัจจุบันของรายการ
* ลบข้อมูล ถ้ารายการถูกขายออกหรือละทิ้งแล้ว
* จัดระเบียบและดูข้อมูลด้วยวิธีต่างๆ
* ใช้ข้อมูลร่วมกันกับผู้อื่นผ่านทางรายงาน ข้อความอีเมล์ อินทราเน็ต หรืออินเทอร์เน็ต
  + 1. **Balsamiq Mockups**



Balsamiq Mockup คือ การสร้างโครงงานออกแบบของเราแบบคร่าวๆเพื่อให้เห็นภาพในสิ่งที่เราคิดสร้างสรรค์ออกมาให้ได้เข้าใจง่ายขึ้น ทำให้การคุยงานกับลูกค้ากับหลายๆฝ่ายนั้นเข้าใจตรงกัน โดยใช้ Mockup ในการนำเสนอ ก่อนจะไปทำจริง

**ข้อดี**

* ใช้งานง่ายการลากวางทา ได้รวดเร็ว
* มีของให้ใช้เยอะมากมาย ทา ให้เราไม่ต้องเขียนอะไรใหม่เองเลย
* สามารถเพิ่มรูปแบบใหม่ๆมาใช้งานได้เช่น iPhone
* สามารถ Group / Ungroup หรือ Lock ได้คล้ายใน Flash เพื่อสะดวกในการออกแบบ
* สามารถนำออกเป็นแบบภาพ PNG ได้

**บทที่ 3**

**การวิเคราะห์ระบบ**

โปรแกรมบันทึกรายรับ-รายจ่ายบนโทรศัพท์มือถือในระบบปฏิบัติการ Android นี้เป็นแอพพลิเคชั่นส่วนบุคคลจะเน้นการใช้งานที่ง่ายและสะดวกต่อการใช้งาน ไม่ซับซ้อน สามารถใช้ได้ทันทีไม่ต้องรู้อะไรมาก สามารถเปิดใช้งานได้ตลอดเวลาที่ต้องการ สามารถตรวจสอบค่าใช้จ่ายในแต่ละโปรเจคและนำมาวิเคราะห์ ดูพฤติกรรมการใช้จ่ายได้ทันที และยังสามารถวางแผนการใช้งานได้ทันที

**3.1 User Reguirement**

**3.1.1 บทสัมภาษณ์ User Reguirement**

**ประวัติผู้ให้สัมภาษณ์**

* **ชื่อ-นามสกุล** ภก.ณรงค์ศักดิ์ สืบสาย
* **อายุ** 24 ปี
* **สถานะภาพ** โสด
* **อาชีพ** เภสัชกรปฏิบัติการ (ร้านยา)
* **รายได้** 50000 บาท/เดือน

**ด้านความสนใจและพฤติกรรม**

* **คุณคิดว่าการจดบันทึกรายรับรายจ่ายสำคัญกับคุณหรือไม่ อย่างไร**

**ตอบ** มีความสำคัญมาก เพราะจะเป็นตัวชี้วัดที่บอกเราได้ว่ารายได้ในแต่ละเดือนเราหมดไปกับอะไรบ้าง อย่างละกี่ % อันไหนเป็นรายจ่ายประจำที่เราต้องกำหนดไว้ตายตัว อันไหนคือรายจ่ายที่เราพอจะประหยัดหรือตัดออกได้บ้าง ทำให้เรารู้และเห็นได้ชัดเจนเลยว่าช่องทางและที่มาที่ไปของเงินหมุนเวียนไหลเข้าไหลออกในแต่ละเดือนของเราเป็นอย่างไร เพื่อจะได้ปรับเปลี่ยนพฤติกรรมและการบริหารวิธีการใช้เงินของเราอย่างไรให้คุ้มค่าและเกิดประโยชน์สูงสุดในเดือนถัดไป

* **คุณได้มีการจดบันทึกรายรับรายจ่ายหรือไม่**

**ตอบ** มีครับ เพราะโดยส่วนตัวแล้วนอกจากหารายได้จากการทำงานแล้วยังมีหน้าที่ส่งเสียดูแลน้องๆและแม่ที่บ้านอีกด้วย จึงต้องมีการแบ่งเงินออกเป็นส่วนๆ เพื่อให้การใช้เงินไม่เกิดการติดขัด เมื่อถึงคราวจำเป็นต้องใช้เงิน จึงมีการทำรายรับ-รายจ่ายอยู่ทุกเดือนเป็นปกติอยู่แล้ว

* **คุณจดบันทึกรายรับ-รายจ่ายมานานเท่าไหร่**

**ตอบ** นับตั้งแต่เริ่มทำงานเดือนแรกเลย ตั้งแต่จบมาใหม่ๆ ก็คือเดือน พฤษภาคม ปี 2557 นับจนถึงปัจจุบันก็ราวๆ 1 ปีครึ่งแล้วครับ และทุกวันนี้ก็ยังคงทำอยู่

* **คุณมีการจดบันทึกรายรับ-รายจ่ายบ่อยแค่ไหน**

**ตอบ** จะนับเป็นรายเดือนไปครับ เดือนละครั้ง แล้วสรุปยอดดุลรายรับ-รายจ่าย

* **คุณมักจะจดบันทึกรายรับรายจ่ายเวลาไหน**

**ตอบ** จะมีการทำบัญชีรายรับ/รายจ่ายตอนสิ้นเดือนของทุกๆเดือนครับ ค่าใช้จ่ายระหว่างเดือนก็จะจดไว้แล้วรวมทีเดียว ส่วนรายรับก็ตามใบสลิปรับเงินเดือนเลย

* **คุณเคยลืมจดบันทึกรายรับรายจ่ายที่ต้องทำในชีวิตประจำวันบ้างหรือไม่**

**ตอบ** ปกติจะรวบรวมใบเสร็จค่าใช้จ่ายทุกอย่างเลยแล้วมานับสรุปยอดทีเดียว แต่โดยมากจะไม่ได้ทำรายรับรายจ่ายประจำวันอยู่แล้ว คือใช้วิธีการแบ่งเงินออกเป็นส่วนๆตั้งแต่ต้นเดือน ว่าส่วนนี้คือเงินสำหรับจ่ายค่าห้อง ส่วนนี้ส่งให้กับที่บ้าน ส่วนนี้ใช้ตั้งตัวสร้างอนาคต ส่วนนี้สำหรับค่ากิน/ใช้แต่ละวัน เป็นต้น ดังนั้นรายรับรายจ่ายจะสรุปทีเดียวเลยคือสิ้นเดือน ไม่ได้ทำทุกวัน แต่ก็ไม่เคยลืมครับ ทำอยู่แบบนี้ทุกสิ้นเดือนอยู่แล้ว

* + **หากคุณลืมจดบันทึกรายรับรายจ่ายคุณจะทำอย่างไร**

**ตอบ** หากมีการลืมก็คงตั้งต้นตามเงินที่คงเหลืออยู่จริง ณ ขณะนั้นแล้วค่อยไปรวมกับรายรับของเดือนถัดไป แล้วเริ่มบันทึกข้อมูลรายรับ-รายจ่ายใหม่ แต่ก็ยังไม่เคยลืมทำสรุปตั้งแต่ทำงานมาจนถึงทุกวันนี้

* + **คุณลืมจดบันทึกรายรับรายจ่ายบ่อยแค่ไหน**

**ตอบ** ยังไม่เคยลืมทำบันทึกรายรับ-รายจ่ายเลยครับ อาจจะเพราะผมทำสรุปเป็นรายเดือนด้วยแหละ โอกาสลืมจึงมีน้อยมาก

**ข้อมูลการใช้ Smart Phone**

* **คุณใช้Smart Phoneยี่ห้ออะไร**

**ตอบ** iPhone 5s 16 GB

* **Smart Phone ของคุณระบบปฏิบัติการอะไร**

**ตอบ** ระบบปฏิบัติการ iOs

* **คุณใช้Smart Phone บ่อยหรือไม่**

**ตอบ** ใช้ตลอดทุกวันและเป็นกิจวัตรประจำวันไปแล้วครับ โดยเฉพาะพวกงานประชุมวิชาการ ข่าวสารทางวิชาชีพต่างๆ หรือติดต่อในเวลาทำงานประสานงานกัน รวมถึงคุยแชทกับเพื่อนๆในโซเชียลด้วย

* + **คุณใช้Smart Phone กี่ครั้งต่อวัน**

**ตอบ** เกือบๆจะทั้งวันเลย เพราะต้องตามงานตลอด

* + **คุณใช้ Smart Phone ช่วงเวลาไหน**

**ตอบ** มักจะเป็นช่วงเวลาทำงาน เนื่องจากนอกเวลางานใช้เวลาส่วนใหญ่ไปกับการเตรียมตัวสอนพิเศษ(หารายได้เสริม) และอ่านหนังสือเรียนนิติศาสตร์ มสธ.

* + **คุณใช้Smart Phone แต่ละครั้งนานเท่าไร**

**ตอบ** ประมาณ 10-15 นาที แล้วแต่ความสำคัญหรือธุระของงาน แต่ก็คอยตรวจเช็คอยู่บ่อยๆเวลาใครทักแชทมาหรือติดต่อมา

* **คุณใช้Smart Phone ทำอะไรบ่อยที่สุด**

**ตอบ** ติดต่องานทางไลน์กลุ่ม และติดตามข่าวการประชุมวิชาการทางวิชาชีพ

**ปัญหาของผู้ใช้**

* **คุณใช้อะไรในการจดบันทึกรายได้และค่าใช้จ่าย**

**ตอบ** โปรแกรม Microsoft Excell เนื่องจากสามารถสร้างฟังก์ชันคำนวณได้เลย ไม่ต้องมานั่งทดเลขในเศษกระดาษ และยังสามารถคาดการณ์ล่วงหน้าได้ด้วยว่าเราจะมีรายรับ-รายจ่ายในอนาคตประมาณเท่าไหร่ คงเหลือประมาณเท่าไหร่ แบ่งสัดส่วนค่าใช้จ่ายเป็น % อย่างไรได้บ้าง

* + **กรณีจดบันทึกรายการกิจกรรมที่ต้องทำลงในสมุดหรือเศษกระดาษ เกิดปัญหาการใช้หรือไม่ อย่างไร**

**ตอบ** อาจเกิดปัญหาในแง่ของเปลืองเนื้อที่ในการจัดเก็บข้อมูล และการแก้ไขก็ทำได้ยากในกรณีที่การบันทึกข้อมูลมีการผิดพลาด การบันทึกอาจเกิดข้อผิดพลาดได้เช่น ลงรายรับผิด รายจ่ายผิด ยากแก่การตรวจสอบ นอกจากนี้ยังต้องใช้เครื่องคิดเลขช่วยคำนวณ การวางแผนใช้เงินแลดูยุ่งยากซับซ้อนหลายขั้นตอนเกินไป และที่สำคัญข้อมูลไม่เป็นความลับเท่าที่ควร การรวบรวมการจัดเก็บเอกสารก็ค่อนข้างเปลืองเนื้อที่ ความล่าช้าในการบันทึกก็มูล และความยุ่งยากลำบากในการจะดึงข้อมูลต่างๆมาใช้ ไม่เหมือนทำด้วยโปรแกรม กรอกข้อมูลเสร็จสามารถผูกสูตรให้คำนวณได้เสร็จสรรพเลย ลดขั้นตอนการทำงานได้เป็นอย่างมาก

* **เพราะเหตุใดคุณจึงมีการบันทึกรายรับ-รายจ่าย**

**ตอบ** เพื่อต้องการทราบความเป็นไปและแนวโน้มการใช้เงินในแต่ละเดือน เพื่อใช้เป็นข้อมูลประกอบการตัดสินใจวางแผนการใช้เงินในเดือนถัดไป และต้องการมองภาพรวมช่องทางการได้มาของรายได้ และช่องทางการใช้จ่ายไปในแต่ละเดือน อาจปรับเปลี่ยนแก้ไขการใช้เงินให้เป็นระบบระเบียบได้ดียิ่งขึ้น

* **คุณพกอุปกรณ์ในการบันทึกรายรับ-รายจ่ายติดตัวตลอดเวลาหรือไม่**

**ตอบ** ไม่ได้พกอยู่ตลอดเวลา เพราะมีการจัดทำดุลรายรับและรายจ่ายเพียงเดือนละครั้งเท่านั้น

* + **กรณีไม่พกตลอดเวลา คุณใช้อะไรในการจดบันทึกรายได้และค่าใช้จ่ายแทนอุปกรณ์**

**ตอบ** จะมีการเก็บพวกใบเสร็จค่าใช้จ่ายต่างๆ ข้อมูลสลิปเงินเดือน และหลักฐานอื่นๆ รวบรวมเอาไว้แล้วมาดุลงบในแต่เดือนอีกที แต่โดยทั่วไปใช้วิธีแบ่งรายจ่ายออกเป็นส่วนๆอยู่แล้ว เวลาตรวจเช็คก็นับจากเงินคงเหลือในแต่ละส่วนที่ได้แบ่งเอาไว้ตั้งแต่ต้นๆเดือน แล้วนำคงเหลือจากทุกส่วนมารวมกันหักลบออกจากรายรับ ก็จะตามเช็คย้อนกลับรายจ่ายของทั้งเดือนได้

* **คุณคิดว่าการจดบนกระดาษกับจดบนแอพพลิเคชั่นอย่างไหนดีกว่ากัน**

**ตอบ** การจดบนแอพลิเคชั่นน่าจะเป็นผลดีกว่า เพราะง่ายและสะดวกในการจัดเก็บ ความผิดพลาดจากการจดบันทึกก็มีน้อย การดึงข้อมูลนำมาใช้ก็สะดวก โอกาสผิดพลาดในการดุลงบรายรับ-รายจ่ายต่อเดือนก็น้อยกว่า รวมจำนวนรายได้รายจ่ายโดยใช้โปรแกรมเข้าช่วย ย่อมให้ผลที่ถูกต้องแม่นยำกว่าการมานั่งกดเครื่องคิดเลขเข้ารายการทีละรายการ โอกาสอ่านค่าเลขผิดพลาดมีสูงมากและเกิดข้อผิดพลาดได้ง่ายด้วย

**ด้านแอพพลิเคชั่น**

* **ท่านเคยใช้แอพพลิเคชั่นในจดบันทึกรายรับ-รายจ่ายหรือไม่**

**ตอบ** เคยใช้แต่โปรแกรม Microsoft Excell ตามปกติ ไม่ได้มีการติดตั้งแอพลิเคชั่นใดๆลงบนเครื่อง Smart Phone เพื่อการจดบันทึกรายรับรายจ่าย

* **คุณคิดว่าคุณมีความพึงพอใจกับแอพที่ใช้ในปัจจุบันหรือไม่ อย่างไร**

**ตอบ** ปกติที่ใช้อยู่ก็เป็นโปรแกรมพื้นฐานธรรมดา ก็ปรากฏว่าได้ผลดีอยู่แล้ว ตรงตามความต้องการที่จะใช้งาน

* **หากมีการพัฒนาแอพพลิเคชั่นสำหรับบันทึกรายรับ-รายจ่ายขึ้นมา ท่านอยากให้ตัวแอพพลิเคชั่นมีลักษณะแบบใด**

**ตอบ** ควรเป็นแอพลิเคชั่นในลักษณะที่เวลากรอกรายรับแล้วสามารถคำนวณออกมาได้เลยว่ารายจ่ายแต่ละส่วนตามที่เราต้องการแบ่งออกเป็นกี่ % ของรายรับนั้นจะถูกจ่ายออกไปในแต่ละส่วนอย่างละเท่าไหร่ และเมื่อลงรายจ่ายเสร็จแล้วก็สามารถคำนวณออกมาได้เลยว่ามียอดคงเหลือเท่าไหร่ %ของรายจ่ายคงเหลือ และมีการบอกแนวโน้มในอนาคตให้เราได้ด้วยว่าหากเรามีรายได้คงที่อยู่เท่านี้ เราจะมีเงินเก็บปีละเท่าไหร่ ต้องจ่ายภาษีเงินได้ขั้นไหน เป็นต้น

* **ถ้ามีแอพพลิเคชั่นตัวนี้ขึ้นมา นอกจากจะใช้ในการบันทึกรายรับ-รายจ่ายแล้ว คุณอยากให้ใส่ข้อมูลอะไรลงไปอีกบ้าง**

**ตอบ** ข้อมูลการต้องชำระภาษีเงินได้ส่วนบุคคลว่าต้องจ่ายขั้นไหน เป็นจำนวนเงินเท่าไหร่

* **อยากให้มีลูกเล่นอะไรบ้างในแอพลิเคชันตัวนี้**

**ตอบ** สามารถแสดงข้อมูลออกมาในรูปกราฟหรือแผนภูมิที่แสดงแนวโน้มหรือการเปลี่ยนแปลงในรายรับ/รายจ่ายของแต่ละเดือน มีการเปรียบเทียบอย่างเห็นได้ชัดเจน ทำให้มองง่ายขึ้นและใช้ประกอบการตัดสินใจได้ดีขึ้น ดีกว่าการนั่งอ่านข้อมูลเปรียบเทียบแต่เชิงตัวเลข ทำให้มองภาพได้ไม่ชัดเจนเท่ากับการใช้กราฟหรือแผนภูมิเปรียบเทียบทางสถิติ

* **แล้วรูปที่อยากเห็น ดีไซน์ อยากให้มีอะไรพิเศษจะไปเล่น**

**ตอบ** ควรจะมีลูกเล่นที่มีการแจ้งเตือนหรือบอกถึงสถานะทางการเงินในเดือนนี้ว่าอยู่ในระดับใด อาจแจ้งเตือนเป็นตัวอักษรหรือขึ้นสีต่างๆก็ได้ เพื่อคอยย้ำเตือนว่าเราควรทำอย่างไรต่อไปกับการบริหารจัดการเรื่องการใช้เงิน จะได้วางแผนถูกและป้องกันปัญหาที่อาจเกิดจากการใช้เงินในอนาคตได้

* **แล้วถ้าให้สรุปโดยภาพรวม ว่าข้อมูลที่ต้องการได้จาก App นี้ คืออะไรบ้าง**

**ตอบ** - ยอดดุลรายรับ/รายจ่ายในแต่ละเดือน โดยแสดงออกมาในรูปของแผนภาพ หรือแผนภูมิต่างๆ

- ค่าใช้จ่ายในแต่ละรายการว่าควรจัดสรรปันส่วนอย่างละเท่าไหร่หลังจากกรอกข้อมูลรายรับลงไป

- ตัวแจ้งเตือนบ่งบอกสถานะการบริการเงินในแต่ละเดือนหลังจากกรอกข้อมูลค่าใช้จ่ายที่แท้จริงลงไป

- ค่าชำระภาษีเงินได้ส่วนบุคคลที่ต้องยื่นต่อกรมสรรพากรในปลายปี

**3.1.2 บทสัมภาษณ์ User Reguirement**

**ประวัติผู้ให้สัมภาษณ์**

* **ชื่อ-นามสกุล** นายธิชากร โชติรัตน์
* **อายุ** 21 ปี
* **สถานะภาพ** โสด
* **อาชีพ** นักศึกษามหาวิทยาลัยราชภัฏภูเก็ต คณะครุศาสตร์ ชั้นปีที่ 4
* **รายได้** 8000 บาท/เดือน

**ด้านความสนใจและพฤติกรรม**

* **คุณคิดว่าการจดบันทึกรายรับรายจ่ายสำคัญกับคุณหรือไม่ อย่างไร**

**ตอบ** มีความสำคัญเพราะจะได้ทราบว่าเรามีรายรับต่อเดือนกี่บาท และรายจ่ายต่อเดือนกี่บาท และเราใช้จ่ายอะไรไปบ้าง อะไรที่ไม่จำเป็นเราก็จะรู้ และสามารถที่จะตัดส่วนที่ไม่จำเป็นออกไปได้

* **คุณได้มีการจดบันทึกรายรับรายจ่ายหรือไม่**

**ตอบ** ผมจะมีการบันทึกรายรับรายจ่ายตลอดคับ เพื่อที่จะได้รู้ว่าใช้จ่ายอะไรไปบ้างแล้ว

* **คุณจดบันทึกรายรับ-รายจ่ายมานานเท่าไหร่**

**ตอบ** ประมาณ 2 ปีได้มั่ง

* **คุณมีการจดบันทึกรายรับ-รายจ่ายบ่อยแค่ไหน**

**ตอบ** เป็นครั้งคราว เพราะผมอาจจะลืมบ้าง

* **คุณมักจะจดบันทึกรายรับรายจ่ายเวลาไหน**

**ตอบ** เวลาตอนที่ผมว่างครับ

* **คุณเคยลืมจดบันทึกรายรับรายจ่ายที่ต้องทำในชีวิตประจำวันบ้างหรือไม่**

**ตอบ** มีบ้าง

* + **หากคุณลืมจดบันทึกรายรับรายจ่ายคุณจะทำอย่างไร**

**ตอบ** จะบันทึกย้อนหลัง

* + **คุณลืมจดบันทึกรายรับรายจ่ายบ่อยแค่ไหน**

**ตอบ** 2-3 วัน

**ข้อมูลการใช้ Smart Phone**

* **คุณใช้Smart Phoneยี่ห้ออะไร**

**ตอบ** iPhone 5s 16 GB

* **Smart Phone ของคุณระบบปฏิบัติการอะไร**

**ตอบ** ระบบปฏิบัติการ iOs

* **คุณใช้Smart Phone บ่อยหรือไม่**

**ตอบ** ใช้ตลอดทุกวันและเป็นกิจวัตรประจำวันไปแล้วครับ

* + **คุณใช้Smart Phone กี่ครั้งต่อวัน**

**ตอบ** เกือบๆจะทั้งวันเลย

* + **คุณใช้ Smart Phone ช่วงเวลาไหน**

**ตอบ** ตลอดเวลา ยกเว้นเวลานอนหลับ

* + **คุณใช้Smart Phone แต่ละครั้งนานเท่าไร**

**ตอบ** 2 ช.ม.

* **คุณใช้Smart Phone ทำอะไรบ่อยที่สุด**

**ตอบ** แชทกับเพื่อน

**ปัญหาของผู้ใช้**

* **คุณใช้อะไรในการจดบันทึกรายได้และค่าใช้จ่าย**

**ตอบ** สมุด

* + **กรณีจดบันทึกรายการกิจกรรมที่ต้องทำลงในสมุดหรือเศษกระดาษ เกิดปัญหาการใช้หรือไม่ อย่างไร**

**ตอบ** หายครับ

* **เพราะเหตุใดคุณจึงมีการบันทึกรายรับ-รายจ่าย**

**ตอบ** จะได้ทราบรายรับ-รายจ่ายของผมได้ตลอดเวลา เพราะผมยังเรียนอยู่ต้องขอเงินจากพ่อแม่ ในแต่ละเดือนแม่ก็จะกำหนดว่าเดือนหนึ่งแม่จะให้ 8,000 บาท ผมต้องบริหารจัดการเอาเองว่าจะทำยังไงให้พอใช้ในแต่ละเดือนครับ

* **คุณพกอุปกรณ์ในการบันทึกรายรับ-รายจ่ายติดตัวตลอดเวลาหรือไม่**

**ตอบ** ไม่ได้พกอยู่ตลอดเวลา

* + **กรณีไม่พกตลอดเวลา คุณใช้อะไรในการจดบันทึกรายได้และค่าใช้จ่ายแทนอุปกรณ์**

**ตอบ** ใช้ความจำและรวมถึงกับการเก็บใบเสร็จค่าใช้จ่ายทั้งหมดที่ทำในแต่ละวัน

* **คุณคิดว่าการจดบนกระดาษกับจดบนแอพพลิเคชั่นอย่างไหนดีกว่ากัน**

**ตอบ** ผมคิดว่าเป็นกระดาษนะครับ เพราะผมไม่เคยบันทึกกับแอพพลิเคชั่นผมจึงไม่ทราบว่ามันจะใช้งานยังไง ใช้งานง่ายหรือเปล่า อย่างนั้นผมก็ตอบเป็นกระดาษแล้วกันครับ ถ้ามีโอกาสผมได้ลองใช้แอพพลิเคชั่นรายรับ-รายจ่าย ผมจะพิจารณาอีกครั้งครับ

**ด้านแอพพลิเคชั่น**

* **ท่านเคยใช้แอพพลิเคชั่นในจดบันทึกรายรับ-รายจ่ายหรือไม่**

**ตอบ** ไม่เคย

* **คุณคิดว่าคุณมีความพึงพอใจกับแอพที่ใช้ในปัจจุบันหรือไม่ อย่างไร**

**ตอบ** -

* **หากมีการพัฒนาแอพพลิเคชั่นสำหรับบันทึกรายรับ-รายจ่ายขึ้นมา ท่านอยากให้ตัวแอพพลิเคชั่นมีลักษณะแบบใด**

**ตอบ** สามารถเตือนได้เวลาวันไหนที่เราไม่ได้บันทึก และสามารถคำนวณเงิน ณ เวลาปัจจุบันได้

* **ถ้ามีแอพพลิเคชั่นตัวนี้ขึ้นมา นอกจากจะใช้ในการบันทึกรายรับ-รายจ่ายแล้ว คุณอยากให้ใส่ข้อมูลอะไรลงไปอีกบ้าง**

**ตอบ** อยากให้การคำนวณกราฟแสดงผลลัพธ์ออกมา

* **อยากให้มีลูกเล่นอะไรบ้างในแอพลิเคชันตัวนี้**

**ตอบ** สามารถมีการตกแต่งและมีการจำลองรายรับรายจ่ายล่วงหน้า

* **แล้วรูปที่อยากเห็น ดีไซน์ อยากให้มีอะไรพิเศษจะไปเล่น**

**ตอบ** สามารถสร้างภาพพื้นหลังที่สวยๆๆได้ มีการจัดแบ่งเป็นหมวดหมู่

* **แล้วถ้าให้สรุปโดยภาพรวม ว่าข้อมูลที่ต้องการได้จาก App นี้ คืออะไรบ้าง**

**ตอบ** อยากได้ความสะดวกสบายในการบันทึก อยากได้แอพที่ใช้งานง่าย มีการทำงานที่ดี และมีการคำนวณรายจ่าย-รายรับ ออกมาอย่างถูกต้อง และมีการแจ้งเตือนเมื่อผมไม่ได้บันทึก

**3.1.3 บทสัมภาษณ์ User Reguirement**

**ประวัติผู้ให้สัมภาษณ์**

* **ชื่อ-นามสกุล** นายสถาพร คงอินทร์
* **อายุ** 26 ปี
* **สถานะภาพ** โสด
* **อาชีพ** รับจ้าง
* **รายได้** 12,000 บาท/เดือน

**ด้านความสนใจและพฤติกรรม**

* **คุณคิดว่าการจดบันทึกรายรับรายจ่ายสำคัญกับคุณหรือไม่ อย่างไร**

**ตอบ** สำคัญ เพราะ จะได้รู้รายรับรายจ่ายของแต่ละเดือน เพื่อนำเอาข้อมูลมาวางแผนการใช้จ่ายแต่ละเดือนและวางแผนการเก็บออมเพื่อไว้ใช้ในอนาคต และจะได้รู้ว่าแต่ละเดือนมีเงินเก็บไม่ หรือสิ่งไหนบ้างที่จ่ายไปแล้วไม่จำเป็น จะได้วางแผนในการจ่ายเดือนต่อไป สิ่งไหนไม่จำเป็นจะได้ตัดออก

* **คุณได้มีการจดบันทึกรายรับรายจ่ายหรือไม่**

**ตอบ** บันทึก เพื่อที่จะบันทึกรายจ่ายของแต่ละวัน แล้วมารวมยอดเป็นหนึ่งเดือน แล้วมาเปรียบเทียบกับรายได้

* **คุณจดบันทึกรายรับ-รายจ่ายมานานเท่าไหร่**

**ตอบ** ผมบันทึกมาสักประมาณ 2 ปีได้ แต่ไม่ถึงกับว่าบันทึกทุกวัน อาจจะมีลืมบ้างในบางครั้ง

* **คุณมีการจดบันทึกรายรับ-รายจ่ายบ่อยแค่ไหน**

**ตอบ** เกือบทุกวัน อยากจะจดบันทึกทุกวัน แต่บางวันก็ลืม บางวันก็ไม่สะดวก

* **คุณมักจะจดบันทึกรายรับรายจ่ายเวลาไหน**

**ตอบ** ตอนกลางคืนหรือก่อนนอน

* **คุณเคยลืมจดบันทึกรายรับรายจ่ายที่ต้องทำในชีวิตประจำวันบ้างหรือไม่**

**ตอบ** ลืมบ่อยมากเลย แล้ววิธีแก้ของผมก็คือการที่กลับไปดูวันก่อนๆ แล้วทำให้คิดออกว่าเราได้ใช้จ่ายอะไรไปบ้าง และมีรายรับอะไรเข้ามาบ้าง

* + **หากคุณลืมจดบันทึกรายรับรายจ่ายคุณจะทำอย่างไร**

**ตอบ** จะบันทึกย้อนหลัง

* + **คุณลืมจดบันทึกรายรับรายจ่ายบ่อยแค่ไหน**

**ตอบ** 2-3 วัน

**ข้อมูลการใช้ Smart Phone**

* **คุณใช้Smart Phoneยี่ห้ออะไร**

**ตอบ** SUMSUNG

* **Smart Phone ของคุณระบบปฏิบัติการอะไร**

**ตอบ** ระบบปฏิบัติการ Android

* **คุณใช้Smart Phone บ่อยหรือไม่**

**ตอบ** ใช้ตลอดทุกวัน

* + **คุณใช้Smart Phone กี่ครั้งต่อวัน**

**ตอบ** หลายครั้ง

* + **คุณใช้ Smart Phone ช่วงเวลาไหน**

**ตอบ** เช้า พักเที่ยง และตอนเย็นๆ

* + **คุณใช้Smart Phone แต่ละครั้งนานเท่าไร**

**ตอบ** 4-5 นาที

* **คุณใช้Smart Phone ทำอะไรบ่อยที่สุด**

**ตอบ** เล่นเฟสบุ๊ค

**ปัญหาของผู้ใช้**

* **คุณใช้อะไรในการจดบันทึกรายได้และค่าใช้จ่าย**

**ตอบ** ผมใช้สมุดในการจดบันทึก เพราะสมุดเวลาเราเขียนลงไปมันสามารถให้เราเขียนได้ดี เราสามารถนึกได้ว่าในแต่ละวันเราทำอะไรไปบ้าง จ่ายอะไรไปบ้าง แต่ก็มีข้อเสียอยู่ดีกว่าเราจะได้มาบันทึก เราก็อาจจะลืมไปบ้างอยู่ดี

* + **กรณีจดบันทึกรายการกิจกรรมที่ต้องทำลงในสมุดหรือเศษกระดาษ เกิดปัญหาการใช้หรือไม่ อย่างไร**

**ตอบ** หายครับ

* **เพราะเหตุใดคุณจึงมีการบันทึกรายรับ-รายจ่าย**

**ตอบ** เพราะผมจากทราบการใช้งานของเงินในแต่ละเดือน ผมจะต้องจัดสรรเงินในแต่ละเดือนว่าในแต่ละเดือนผมจะค่าผ่อนรถ มีค่าใช้จ่ายในส่วนต่างๆ จะต้องจัดสรรให้พอในแต่ละเดือน ถ้าผมไม่ทำบัญชีรายรับ-รายจ่าย ผมจะไม่รู้เลยว่าในแต่ละเดือนผมจะเหลือเงินเท่าไหร และจะได้รู้ว่าผมใช้จ่ายในสิ่งที่จำเป็นหรือไม่ และผมสามารถนำบัญชีรายรับ-รายจ่ายนำมาวางแผนการใช้จ่ายเงินในเดือนถัดไปได้ถูกต้อง และผมสามารถวางแผนเงินเก็บไว้ใช้ในอนาคตได้

* **คุณพกอุปกรณ์ในการบันทึกรายรับ-รายจ่ายติดตัวตลอดเวลาหรือไม่**

**ตอบ** ไม่ตลอดเวลา แต่ใช้การเก็บใบเสร็จรายการที่ซื้อไว้แทนเพื่อจะนำมาลงบันทึกในสมุดในเวลาที่ผมสะดวก

* + **กรณีไม่พกตลอดเวลา คุณใช้อะไรในการจดบันทึกรายได้และค่าใช้จ่ายแทนอุปกรณ์**

**ตอบ** ใช้ความจำ

* **คุณคิดว่าการจดบนกระดาษกับจดบนแอพพลิเคชั่นอย่างไหนดีกว่ากัน**

**ตอบ** จดบนแอพพลิเคชั่นหน้าจะดีกว่า เพราะมันสะดวกต่อการใช้งานและสามารถที่จะจดบันทึกได้ตลอดเวลาไม่ว่าผมจะอยู่ที่ไหน หรือว่าว่างตอนไหน เพราะผมก็พกโทรศัพท์ติดตัวตลอดเวลาอยู่แล้ว เมื่อว่างก็สามารถเอาโทรศัพท์มาจดบันทึกได้

**ด้านแอพพลิเคชั่น**

* **ท่านเคยใช้แอพพลิเคชั่นในจดบันทึกรายรับ-รายจ่ายหรือไม่**

**ตอบ** ไม่เคย

* **คุณคิดว่าคุณมีความพึงพอใจกับแอพที่ใช้ในปัจจุบันหรือไม่ อย่างไร**

**ตอบ** -

* **หากมีการพัฒนาแอพพลิเคชั่นสำหรับบันทึกรายรับ-รายจ่ายขึ้นมา ท่านอยากให้ตัวแอพพลิเคชั่นมีลักษณะแบบใด**

**ตอบ** สิ่งที่ผมอยากได้ก็คือ แอพสามารถทีจะมีการแสดงการบันทึกรายรับ-รายจ่ายให้ผมเห็นได้ และสามารถแสดงยอดเงินคงเหลือที่มีอยู่ในขณะนี้ให้ได้ทราบว่าตอนนี้ผมเหลือเงินอยู่เท่าไหรในบัญชีรายรับ-รายจ่าย และสามารถแสดงการสรุปยอดในแต่ละเดือนให้ได้ด้วย และแอพพลิเคชั่นมีการใช้งานง่าย และสะดวกต่อผู้ใช้งาน และอยากให้มีพื้นหลังที่ดูสบายตา อาจจะมีให้มีการตกแต่งบัญชีรายรับ-รายจ่าย ของตัวเองตามแบบที่ผมชอบอะไรประมาณนี้ครับ

* **ถ้ามีแอพพลิเคชั่นตัวนี้ขึ้นมา นอกจากจะใช้ในการบันทึกรายรับ-รายจ่ายแล้ว คุณอยากให้ใส่ข้อมูลอะไรลงไปอีกบ้าง**

**ตอบ** อยากให้มีการแสดงกราฟ มีการแจ้งเตือนสิ่งที่เราจำเป็นที่จะจ่ายทุกเดือน และแจ้งเตือนว่าเราเก็บเงินส่วนนี้ไว้นะ เช่น ผ่อนรถ ผมก็ต้องผ่อนรถทุกเดือน อยากให้แอพนี้ช่วยเตือนผมว่าจะต้องเก็บเงินส่วนนี้ไว้นะและจะครบกำหนดผ่อนในวันนี้ประมาณนี้ครับ

* **อยากให้มีลูกเล่นอะไรบ้างในแอพลิเคชันตัวนี้**

**ตอบ** สามารถแสดงเป็นแผนภาพกราฟได้ และสามารถแสดงผลลัพธ์ที่ได้ว่าเดือนนี้เหลือเงินกี่บาทได้

* **แล้วรูปที่อยากเห็น ดีไซน์ อยากให้มีอะไรพิเศษจะไปเล่น**

**ตอบ** อยากให้แอพตัวนี้สามารถมีการใส่ภาพโปรไฟล์ได้ด้วย สามารถตกแต่งบัญชีรายรับ-รายจ่ายได้อย่างที่ผมต้องการ

* **แล้วถ้าให้สรุปโดยภาพรวม ว่าข้อมูลที่ต้องการได้จาก App นี้ คืออะไรบ้าง**

**ตอบ** อยากได้ข้อมูลทางการเงิน ณ เวลาปัจจุบัน และมีความสะดวกกับผมที่ไม่ต้องไปจดในสมุดแล้วต้องมานั่งคิดเลขอยู่อีกว่าตอนนี้เหลือเงินเท่าไหร แต่ถ้ามีแอพพลิเคชั่นนี้ขึ้นมาผมก็สะดวกมากขึ้นครับ

**วิเคราะห์ User Requirement**

**ด้านความสนใจและพฤติกรรม**

* **คุณคิดว่าการจดบันทึกรายรับรายจ่ายสำคัญกับคุณหรือไม่ อย่างไร**

มีความสำคัญเนื่องจาก

* จะเป็นตัวชี้วัดที่บอกเราได้ว่ารายได้ในแต่ละเดือนเราหมดไปกับอะไรบ้าง
* จะได้ปรับเปลี่ยนพฤติกรรมและการบริหารวิธีการใช้เงินของเราอย่างไรให้คุ้มค่าและเกิดประโยชน์สูงสุดในเดือนถัดไป
* จะได้รู้รายรับรายจ่ายของแต่ละเดือน
* เพื่อนำเอาข้อมูลมาวางแผนการใช้จ่ายแต่ละเดือนและวางแผนการเก็บออมเพื่อไว้ใช้ในอนาคต
* สามารถที่จะตัดส่วนที่ไม่จำเป็นในการใช้จ่ายออกไปได้
* **คุณได้มีการจดบันทึกรายรับรายจ่ายหรือไม่**
* มีการจดบันทึก

**ข้อมูลการใช้ Smart Phone**

* **คุณใช้Smart Phoneยี่ห้ออะไร**
* iPhone
* SUMSUNG
* **Smart Phone ของคุณระบบปฏิบัติการอะไร**
* ระบบปฏิบัติการ iOS
* ระบบปฏิบัติการ Android
* **คุณใช้Smart Phone ทำอะไรบ่อยที่สุด**
* ติดต่องานทางไลน์กลุ่ม
* เล่นเฟชบุ๊ค
* ติดตามข่าว

**ปัญหาของผู้ใช้**

* **คุณใช้อะไรในการจดบันทึกรายได้และค่าใช้จ่าย**
* Microsoft Excel
* สมุด
* **กรณีจดบันทึกรายการกิจกรรมที่ต้องทำลงในสมุดหรือเศษกระดาษ เกิดปัญหาการใช้หรือไม่ อย่างไร**
* สมุดหาย
* การบันทึกข้อมูลมีการผิดพลาด
* มีความยุ่งยากในการบันทึก
* มีความล่าช้าในการบันทึก
* มีความล่าช้าในการเอาข้อมูลมาใช้งาน
* **เพราะเหตุใดคุณจึงมีการบันทึกรายรับ-รายจ่าย**
* เพื่อต้องการทราบความเป็นไปและแนวโน้มการใช้เงินในแต่ละเดือน
* เพื่อใช้เป็นข้อมูลประกอบการตัดสินใจวางแผนการใช้เงินในเดือนถัดไป
* ต้องการมองภาพรวมช่องทางการได้มาของรายได้
* **คุณพกอุปกรณ์ในการบันทึกรายรับ-รายจ่ายติดตัวตลอดเวลาหรือไม่**
* ไม่ได้พกพาตลอดเวลา
  + **กรณีไม่พกตลอดเวลา คุณใช้อะไรในการจดบันทึกรายได้และค่าใช้จ่ายแทนอุปกรณ์**
  + มีการเก็บพวกใบเสร็จค่าใช้จ่ายต่างๆ
  + .ใช้ความจำ
  + **คุณคิดว่าการจดบนกระดาษกับจดบนแอพพลิเคชั่นอย่างไหนดีกว่ากัน**
  + บันทึกกับแอพพลิเคชั่นดีกว่าบันทึกในสมุด

**ด้านแอพพลิเคชั่น**

* **ท่านเคยใช้แอพพลิเคชั่นในจดบันทึกรายรับ-รายจ่ายหรือไม่**
* ไม่เคยใช้
* **หากมีการพัฒนาแอพพลิเคชั่นสำหรับบันทึกรายรับ-รายจ่ายขึ้นมา ท่านอยากให้ตัวแอพพลิเคชั่นมีลักษณะแบบใด**
* เป็นแอพลิเคชั่นในลักษณะที่เวลากรอกรายรับแล้วสามารถคำนวณออกมาได้เลยว่ารายจ่ายแต่ละส่วนตามที่เราต้องการแบ่งออกเป็นกี่ % ของรายรับ
* สามารถคำนวณออกมาได้เลยว่ามียอดคงเหลือเท่าไหร่ %ของรายจ่ายคงเหลือ
* **อยากให้มีลูกเล่นอะไรบ้างในแอพลิเคชั่นตัวนี้**
* สามารถแสดงข้อมูลออกมาในรูปกราฟหรือแผนภูมิได้
* มีการเปรียบเทียบกันได้ชัดระหว่างรายรับ-รายจ่าย
* **แล้วถ้าให้สรุปโดยภาพรวม ว่าข้อมูลที่ต้องการได้จาก App นี้ คืออะไรบ้าง**
* ยอดดุลรายรับ/รายจ่ายในแต่ละเดือน โดยแสดงออกมาในรูปของแผนภาพ หรือแผนภูมิต่างๆ
* อยากได้ความสะดวกสบายในการบันทึก
* อยากได้แอพที่ใช้งานง่าย มีการทำงานที่ดี และมีการคำนวณรายจ่าย-รายรับ ออกมาอย่างถูกต้อง
* อยากได้ข้อมูลทางการเงิน ณ เวลาปัจจุบัน

**3.2 Hierarchical Task Analysis (HTA)**

Hierarchical Task Analysis (HTA) คือ การลำดับขั้นงานการวิเคราะห์ เป็นชนิดที่ใช้กันอย่างแพร่หลายของการวิเคราะห์งานที่เป็นงานระดับสูงมีการแตกออกเป็นลำดับชั้นของงานย่อย HTA บางครั้งจะเรียกว่าการสลายตัวตามลำดับชั้น

HTA ประกอบด้วยลำดับชั้นของงานงานย่อยและการดำเนินการร่วมกับแผนการที่อธิบายถึงการทำ

งานของระบบงานที่ผู้ใช้จะสร้างขึ้นมา

แก้ไขหรือลบรายการบันทึก

บันทึกรายจ่าย

**บันทึกรายรับ-รายจ่าย**

HI.1

HI.6

HI.5

HI.4

HI.3

HI.2

แสดงผลการบันทึก

รายงาน

บันทึกรายรับ

เปิดแอพพลิเคชั่น

HI.2.5

HI.2.4

กรอกหมวดหมู่รายรับ

HI.2.3

กรอกวันที่บันทึกรายรับ

HI.2.2

HI.2.1

กรอกจำนวนเงิน

เลือกเมนูรายรับ

เพิ่มข้อมูล

HI.2

บันทึกรายรับ

เพิ่มข้อมูล

เลือกเมนูรายจ่าย

กรอกหมวดหมู่รายจ่าย

กรอกวันที่บันทึกรายจ่าย



กรอกจำนวนเงิน

HI.3.5

HI.3.4

HI.3.3

HI.3.2

HI.3.1

HI.3

บันทึกรายจ่าย

รายงานยอดคงเหลือแต่ละเดือน

เลือกกราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน

เลือกรายงานรายรับ

HI.4.6

HI.4.4

HI.4.2

HI.4

รายงาน

HI.5.3.3.1

HI.5.3.2.1

HI.5.3.1.1

แสดงผลยอดคงเหลือแต่ละเดือน

แสดงผลกราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน

แสดงผลรายงานรายรับ

HI.5.3.3

HI.5.3.2

HI.5.3.1

รายงานยอดคงเหลือแต่ละเดือน

เลือกกราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน

เลือกรายงานรายรับ

HI.5.2.2

HI.5.1.2

รายรับ

HI.5.2.1

HI.5.1.1

เลือกรายจ่าย

เลือกรายรับ

แสดงผลรายจ่าย

HI.5.3

รายงาน

แสดงผลรายรับ

HI.5.1

HI.5

HI.5.2

รายจ่าย

แสดงผลการบันทึก

กดบันทึก

เลือกรายการที่ต้องการลบหรือแก้ไข

เลือกกิจกรรม

HI.6

HI.6.4

HI.6.3

HI.6.2

HI.6.1

แก้ไขหรือลบรายการที่เลือก

แก้ไขหรือลบรายการบันทึก

**3.3 ER Diagram**

ExRec\_ID

ExRec\_Name

ExRec\_Amt

ExRec\_Date

ExCat\_ID

**PK**

BudRec\_ID

BudRec\_Name

BudRec\_Amt

BudRec \_Date

BudCat \_ID

Budget Record

**PK**

**PK**

Expense Record

**PK**

ExCat\_ID

ExCat\_Name

Expense Category

**FK**

Budget Category

BudCat\_ID

BudCat \_Name

**FK**

**รูปที่ 3.1 ER Diagram**

**3.4 Data Dictionary**

**Data Dictionary** คือ พจนานุกรมข้อมูล ที่แสดงรายละเอียดตารางข้อมูลต่างๆ ในฐานข้อมูล (Database) ซึ่งประกอบด้วยรีเลชั่น (Relation Name), แอตทริบิวต์ (Attribute), ชื่อแทน (Aliases Name), รายละเอียดข้อมูล (Data Description), แอตทริบิวโดเมน (Attribute Domain), ฯลฯ ทำให้สามารถค้นหารายละเอียดที่ต้องการได้สะดวกมากยิ่งขึ้น พจนานุกรมข้อมูลเป็นการผสมผสานระหว่างรูปแบบของพจนานุกรมโดยทั่วไปและรูปแบบของข้อมูลในระบบงานคอมพิวเตอร์ เพื่ออธิบายชนิดของข้อมูลแต่ละตัวว่าเป็น ตัวเลข อักขระ ข้อความ หรือวันที่ เป็นต้น เพื่อช่วยในการอธิบายรายละเอียดต่างๆ ในการอ้างอิงหรือค้นหาที่เกี่ยวกับข้อมูล หรือจะเรียกง่ายๆ ว่า Data Dictionary คือ เอกสารที่ใช้อธิบายฐานข้อมูลหรือการจัดเก็บฐานข้อมูล ซึ่ง Data Dictionary มีประโยชน์ ดังนี้

- จัดเก็บรายละเอียดข้อมูล

- แสดงความหมายที่เกี่ยวกับระบบ

- ทำเอกสารที่บอกคุณลักษณะของระบบ

- หาข้อบกพร่องและสิ่งที่หายไปจากระบบ

ส่วนประกอบของ Data Dictionary

1. ข้อมูลย่อย (Data Element): ส่วนประกอบพื้นที่ ที่ไม่สามารถแบ่งให้เล็กลงได้อีก

2. โครงสร้างข้อมูล (Data Structure): สร้างขึ้นโดยการนำส่วนย่อยของข้อมูล ตั้งแต่ 1 ตัวขึ้นไป ที่สัมพันธ์กันมารวมเข้าด้วยกัน

**การออกแบบ Data Dictionary**

สามารถสร้างฐานข้อมูลออกมาได้ 4 ตาราง มีรายละเอียดดังนี้

* ตาราง Expense Category ใช้เก็บข้อมูลของหมวดหมู่รายจ่าย เพื่อนำไปใช้กับรายการต่างๆ

|  |  |  |  |
| --- | --- | --- | --- |
| โครงสร้างฐานข้อมูลตาราง Expanse Category | | | |
| Name | Type | Value | Description |
| \*ExCat\_ID | INTEGER | 1<=a<=10 | รหัสของหมวดหมู่รายจ่าย |
| ExCat\_Name | TEXT | 1<=a<=100 | ชื่อของหมวดหมู่รายจ่าย |

**ตารางที่ 3.1 Expense Category**

* ตาราง Expense Record ใช้เก็บข้อมูลของรายจ่าย

|  |  |  |  |
| --- | --- | --- | --- |
| โครงสร้างฐานข้อมูลตาราง Expense Record | | | |
| Name | Type | Value | Description |
| \*ExRec\_ID | INTEGER | 1<=a<=20 | รหัสของการบันทึกรายจ่าย |
| ExRec\_Name | TEXT | 1<=a<=100 | ชื่อของการบันทึกรายจ่าย |
| ExRec\_Amt | INTEGER | 1<=a<=10 | ยอดเงินของการบันทึกรายจ่าย |
| ExRec\_Date | date | 1<=a<=10 | วันที่บันทึกรายจ่าย |
| ExCat\_ID(FK) | INTEGER | 1<=a<=10 | รหัสของหมวดหมู่รายจ่าย |

**ตารางที่ 3.2 Expense Record**

* ตาราง Budget Record ใช้เก็บข้อมูลของงบรายรับ

|  |  |  |  |
| --- | --- | --- | --- |
| โครงสร้างฐานข้อมูลตาราง Budget Record | | | |
| Name | Type | Value | Description |
| \* BudRec\_ID | INTEGER | 1<=a<=20 | รหัสของการบันทึกรายรับ |
| BudRec\_Name | TEXT | 1<=a<=100 | ชื่อของการบันทึกรายรับ |
| BudRec\_Amt | INTEGER | 1<=a<=10 | ยอดเงินของการบันทึกรายรับ |
| BudRec\_Date | date | 1<=a<=10 | วันที่บันทึกรายรับ |
| BudCat\_ID (FK) | INTEGER | 1<=a<=10 | รหัสของหมวดหมู่รายรับ |

**ตารางที่ 3.3 Budget Record**

* ตาราง Budget Category ใช้เก็บข้อมูลของหมวดหมู่รายรับ เพื่อนำไปใช้กับรายการต่างๆ

|  |  |  |  |
| --- | --- | --- | --- |
| โครงสร้างฐานข้อมูลตาราง Budget Category | | | |
| Name | Type | Value | Description |
| \*BudCat\_ID | INTEGER | 1<=a<=10 | รหัสของหมวดหมู่รายรับ |
| BudCat\_Name | TEXT | 1<=a<=100 | ชื่อของหมวดหมู่รายรับ |

**ตารางที่ 3.4 Budget Category**

**3.5 Essential use case**

**Essential use case:** การสร้างรายรับ **HI2 UC1**

|  |  |
| --- | --- |
| User Action | System Response |
| Open the Application |  |
|  | Show a menu bar |
|  | Show budget and Add เพิ่มข้อมูล button |
| Press add เพิ่มเมนู button |  |
|  | Show create a budget form |
| Enter a budget date |  |
| Enter a budget category |  |
| Enter a budget amount |  |
| Press save a budget butter |  |
|  | Show a budget list |

**Essential use case:** สร้างบันทึกรายจ่าย **HI3 UC2**

|  |  |
| --- | --- |
| User Action | System Response |
| Open the Application |  |
|  | Show a menu bar |
|  | Show expense and Add เพิ่มข้อมูล button |
| Press add เพิ่มเมนู button |  |
|  | Show create a expense form |
| Enter a expense date |  |
| Enter a expense category |  |
| Enter a expense amount |  |
| Press save a expense butter |  |
|  | Show a expense list |

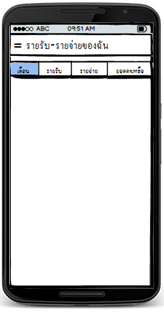
**Essential use case:** รายงาน **HI4 UC3**

|  |  |
| --- | --- |
| User Action | System Response |
| Open the Application |  |
|  | Show a menu bar |
|  | Show Report and  Add เลือกรายงานรายรับ button  Add กราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน button  Add รายงานยอดคงเหลือแต่ละเดือน button |
| Press Add เลือกรายงานรายรับ button  Add กราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน button  Add รายงานยอดคงเหลือแต่ละเดือน button |  |
|  | Show report เลือกรายงานรายรับ  Show report กราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน  Show report รายงานยอดคงเหลือแต่ละเดือน |

**Essential use case:** การแก้ไขหรือลบรายการบันทึก  
**HI6 UC5**

|  |  |
| --- | --- |
| User Action | System Response |
| Open the Application |  |
|  | Show a splash screen |
|  | Show the current activity list |
| Selected activity |  |
|  | Show a activity |
|  | Show the record list Show create a revenue button and  Show create a expense button  Show analysis of record |
| Select the record list |  |
|  | Show the Edit and Delete menu |
| Select the Edit or Delete menu |  |
|  | Show the Edit or Delete |
| Enter the Edit or Delete record |  |
|  | Show confirms the Edit or Delete button |
| Select Yes or No button |  |
|  | Show a record list |

**3.6 Screen Design**



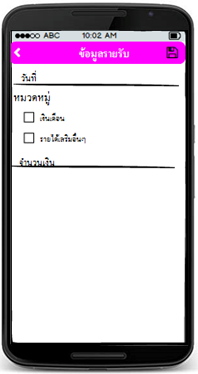
**ภาพที่ 3.2 Show รายรับ-รายจ่ายของฉัน**



**ภาพที่ 3.3 Show menu bar**



**ภาพที่ 3.4 แสดงหน้า list รายรับ**



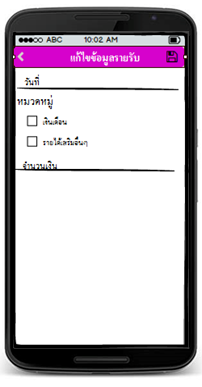
**ภาพที่ 3.5 แสดงหน้าบันทึกงบรายรับ**



**ภาพที่ 3.6 แสดงหน้า list รายรับ**



**ภาพที่ 3.7 แสดงข้อมูลรายรับที่บันทึกแล้ว**



**ภาพที่ 3.8 แสดงหน้าแก้ไขข้อมูลรายรับ**



**ภาพที่ 3.9 แสดงหน้าลบข้อมูลรายรับ**



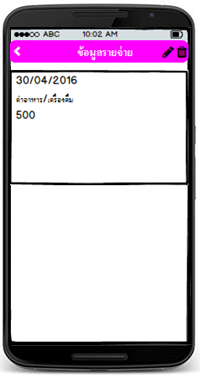
**ภาพที่ 3.10 แสดงหน้า list รายจ่าย**



**ภาพที่ 3.11 แสดงหน้าบันทึกงบรายจ่าย**



**ภาพที่ 3.12 แสดงหน้า list รายจ่าย**



**ภาพที่ 3.13 แสดงข้อมูลรายจ่ายที่บันทึกแล้ว**

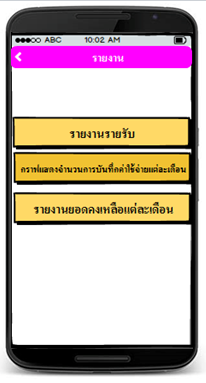


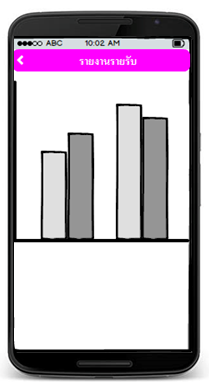
**ภาพที่ 3.14 แสดงหน้าแก้ไขข้อมูลรายจ่าย**



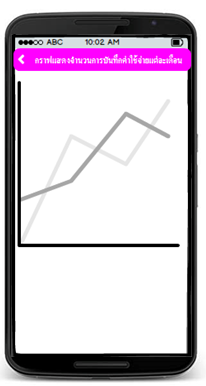
**ภาพที่ 3.15 แสดงหน้าลบข้อมูลรายจ่าย**

**ภาพที่ 3.16 แสดงหน้ารายงาน**

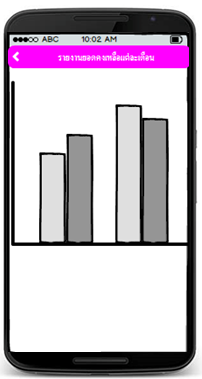




**ภาพที่ 3.17 แสดงกราฟรายงานรายรับ**



**ภาพที่ 3.18 แสดงกราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน**



**ภาพที่ 3.19 แสดงกราฟรายงานยอดคงเหลือแต่ละเดือน**

**บทที่ 4**

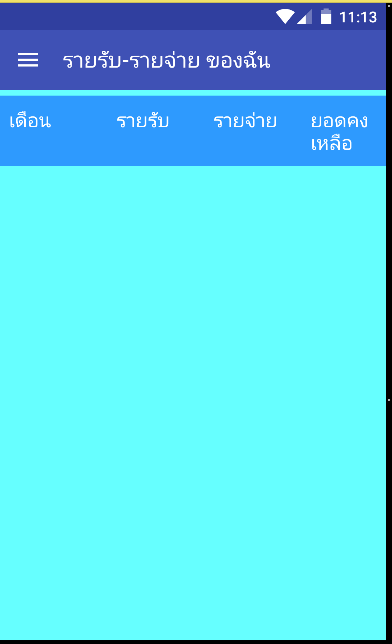
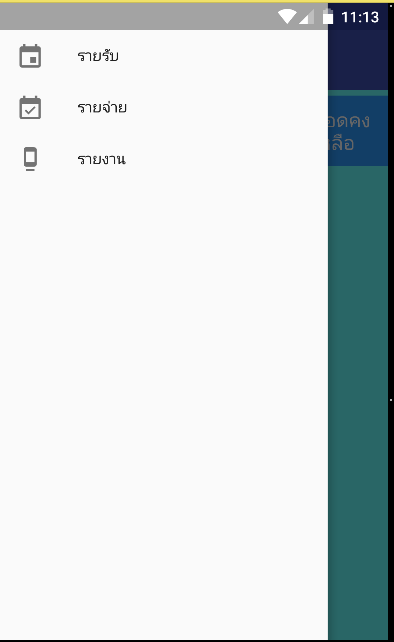
**ผลการดำเนินโครงงาน**

**การพัฒนาแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน**

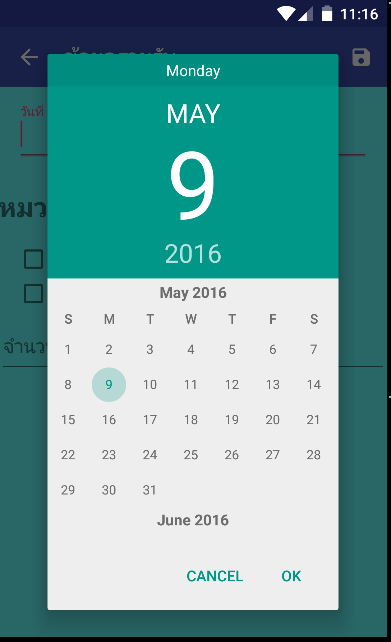
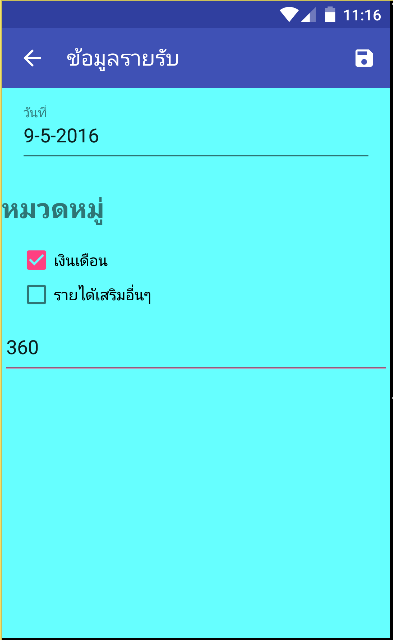
แอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน เป็นแอพพลิเคชั่นที่ได้รับการพัฒนามาจากระบบปฏิบัติการแอนดรอยด์ สามารถทำงานได้ทั้งแบบออนไลน์และออฟไลน์ และแอพพลิเคชั่นรายรับ-รายจ่ายในชีวิตประจำวันได้พัฒนาขึ้นเพื่อสร้างความสะดวกสบายให้คนใช้ในการบันทึกรายรับ-รายจ่ายได้ง่ายยิ่งขึ้น เนื่องจากคนส่วนใหญ่จะไม่ค่อยได้บันทึกค่าใช้จ่ายในแต่ละวัน จึงไม่ทราบถึงรายละเอียดที่ใช้จ่ายไป โดยแอพพลิเคชั่นจะมีการแบ่งหมวดหู่ของรายรับ-รายจ่ายให้ง่ายที่ผู้ใช้งานรู้ว่าตัวเองใช้จ่ายไปกับอะไรเยอะที่สุด และเมื่อเราบันทึกข้อมูลรายรับ-รายจ่ายแล้ว แอพพลิเคชั่นสามารถคำนวนออกมาเป็นกราฟเปรียบเทียบค่าใช้จ่ายให้กับผู้ใช้งานได้

**ความสามารถของแอพพลิเคชั่น**

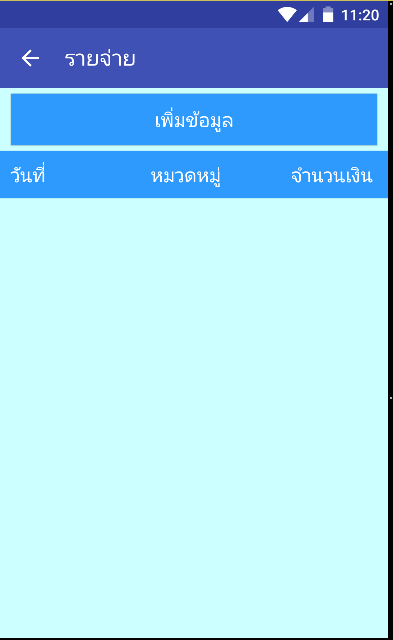
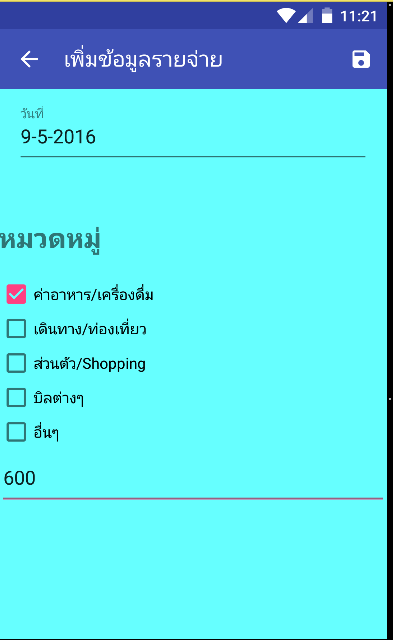
1. แอพพลิเคชั่นสามารถบันทึกรายรับได้
2. แอพพลิเคชั่นสามารถบันทึกรายจ่ายได้
3. แอพพลิเคชั่นสามารถโชว์ข้อมูลรายรับ-รายจ่ายได้
4. แอพพลิเคชั่นสามารถแก้ไขข้อมูลรายรับ-รายจ่ายได้
5. แอพพลิเคชั่นสามารถลบข้อมูลรายรับ-รายจ่ายได้
6. แอพพลิเคชั่นสามารถแสดงกราฟของรายรับได้
7. แอพพลิเคชั่นแสดงกราฟของรายจ่ายได้

**แอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน**

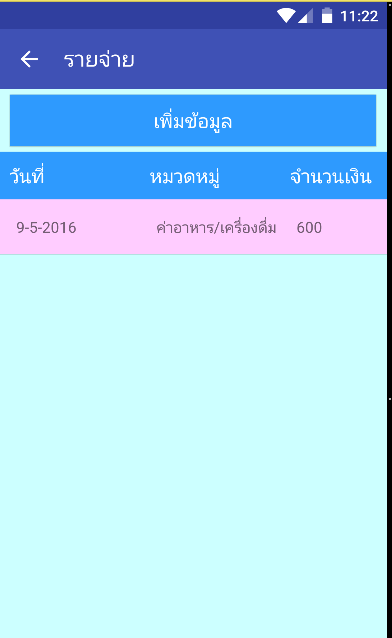
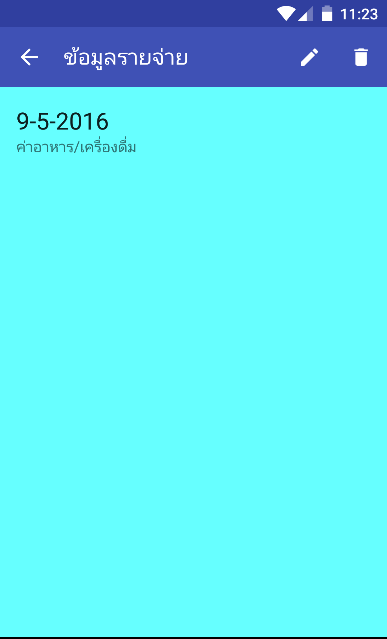
**ภาพที่ 4.1 หน้าแรกของแอพพลิเคชั่น ภาพที่ 4.2 หน้าเมนูของแอพพลิเคชั่น**



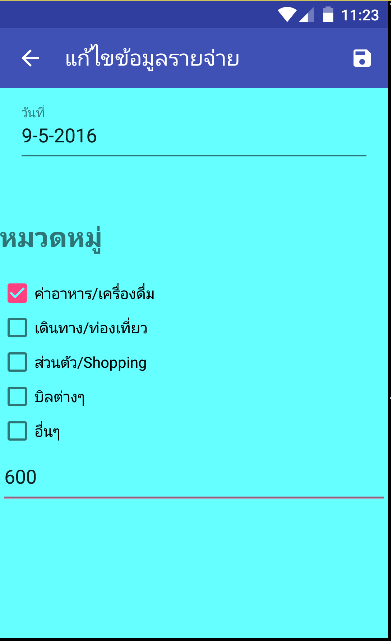
**ภาพที่ 4.3 หน้าแสดงข้อมูลรายรับ ภาพที่ 4.4 หน้าบันทึกข้อมูลรายรับ**



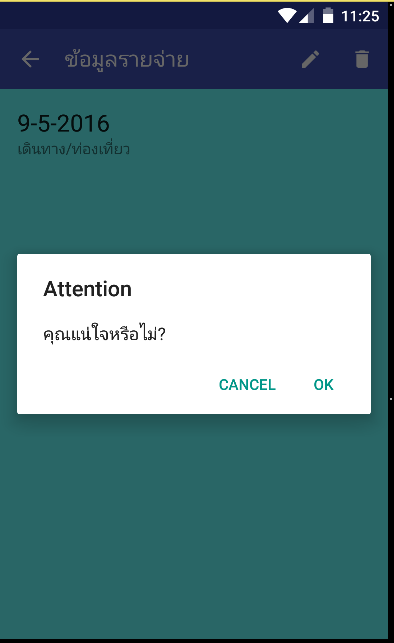
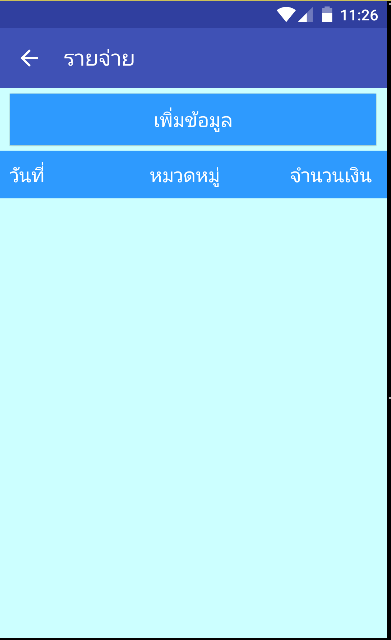
**ภาพที่ 4.5 หน้าแสดงข้อมูลรายจ่าย ภาพที่ 4.6 หน้าแสดงบันทึกข้อมูลรายจ่าย**



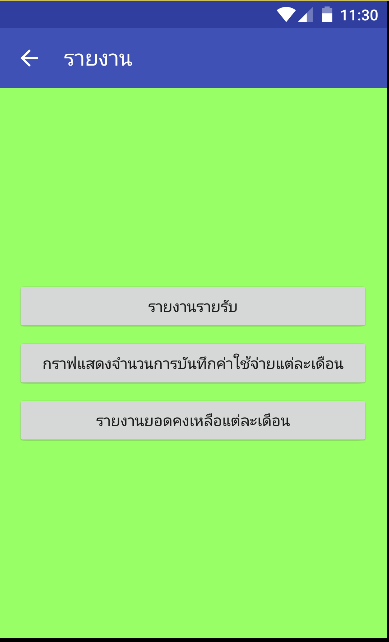
**ภาพที่ 4.7 หน้าแสดงข้อมูลรายจ่ายที่บันทึกไว้ ภาพที่ 4.8 หน้าแสดงข้อมูลรายจ่าย**



**ภาพที่ 4.9 หน้าแก้ไขข้อมูลรายจ่าย**



**ภาพที่ 4.10 หน้าลบข้อมูลรายจ่าย**



**ภาพที่ 4.11 หน้าแสดงรายงาน ภาพที่ 4.12 หน้าแสดงกราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน**

**บทที่ 5**

**สรุปผลการดำเนินงาน**

จากการได้พัฒนาโครงงานแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน โดยวัตถุประสงค์ในการทำแอพพลิเคชั่นรายรับ-รายจ่ายก็คือการที่ผู้ใช้งานสามารถบันทึกรายรับ-รายจ่ายได้ทุกที่ไม่ว่าจะเป็นเวลาไหน สะดวกสบาย การพัฒนาแอพพลิเคชั่นนี้มีการพัฒนาที่เป็นขั้นตอน โดยการเริ่มจากการศึกษาขอบเขตของโครงงาน ศึกษาความต้องการใช้แอพพลิเคชั่นรายรับ-รายจ่าย การวิเคราะห์ระบบและออกแบบระบบซึ่งเป็นส่วนสำคัญในการพัฒนาแอพพลิเคชั่นให้มีความถูกต้องและแม่นยำในการใช้งานแอพพลิเคชั่นได้มากยิ่งขึ้น จึงมีการสรุปผลการดำเนินงานได้ดังนี้

**สรุปผลโครงงาน**

การพัฒนาแอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน ได้มีการพัฒนาและดำเนินการตามที่ผู้จัดทำได้ตั้งเป้าหมายไว้ โดยในการพัฒนาแอพพลิเคชั่นได้มีเครื่องมือและเทคโนโลยีที่นำมาใช้ ได้แค่ Android Studio , Active Android เทคโนโลยีที่กล่าวมานี้เป็นเทคโนโลยีที่ช่วยให้การพัฒนาแอพพลิเคชั่นสำเร็จได้ด้วยดีเป็นไปตามเป้าหมายที่ผู้จัดทำได้วางแผนขึ้น เพื่อที่จะให้แอพพลิเคชั่นสามารถเป็นสิ่งอำนวยความสะดวกในการใช้งานของผู้ใช้งานได้เป็นอย่างดี

**ข้อดี**

1. อำนวยความสะดวกให้ผู้ใช้งานในการบันทึกรายรับ-รายจ่าย
2. ผู้ใช้งานสามารถบันทึกรายรับ-รายจ่ายในเวลาไหนก็ได้
3. ผู้ใช้งานสามารถรู้จำนวนเงินในแต่ละวันที่จ่ายไปได้

**ข้อจำกัด**

1. การแสดงผลในโทรศัพท์บางเครื่องอาจจะผิดเพี้ยนไปจากรูปแบบ
2. การแสดงผลของกราฟอาจจะไม่สมบูรณ์

**ข้อเสนอแนะ**

ควรจะบันทึกข้อมูลรายรับ-รายจ่ายทุกวัน เพื่อความแน่นอนของข้อมูล

**อ้างอิง**

สำนักงานสถิติแห่งชาติ. ๒๕๕๘. รายได้รายจ่ายและหนี้สินครัวเรือน. (ออนไลน์). แหล่งที่มา : <http://service.nso.go.th/nso/web/statseries/statseries11.html>. ๒๓ กันยายน ๒๕๕๘

จรรยา ประจันตะเสน. ๒๕๕๘. การจัดทำบัญชี. (ออนไลน์). แหลงที่มา : http://www.thaigoodview.com/library/teachershow/lopburi/janya\_p/egg/sec02p04.html. ๒๓ กันยายน ๒๕๕๘

ทิชากร สำราญชลารักษ์. ๒๕๕๘. การทำบัญชีรายรับ-รายจ่าย. (ออนไลน์). แหล่งที่มา :

https://sites.google.com/site/kruticha/bth-reiyn/ngan-thurkic/hnwy-thi-4-kin-xyu-xyang-chlad/kar-banthuk-baychi-rayrab-ray-cay. ๒๓ กันยายน ๒๕๕๘

กรมตรวจบัญชีสหกรณ์. ๒๕๕๘. รายได้.(ออนไลน์). แหล่งที่มา : http://www.cad.go.th/ewt\_news.php?nid=2469&filename=index. ๒๓ กันยายน ๒๕๕๘

เพียงเดือน วิทยาประดิษฐ์. ๒๕๕๘. ค่าใช้จ่าย. (ออนไลน์). แหล่งที่มา : http://www.thaigoodview.com/library/teachershow/nakhonsithamrat/peangduan\_w/bunnambunshe/sec03p06.html. ๒๓ กันยายน ๒๕๕๘

Chai Phonbopit.๒๕๕๘.ตัวอย่างการใช้งาน SQLite บน Android (ออนไลน์) .แหล่งที่มา:

http://www.devahoy.com/2014/05/android-sqlite-tutorial-part-1. ๑๖ ตุลาคม ๒๕๕๘

ThaiCreate.Com Team.๒๕๕๘.การเขียนแอนดรอยด์เพื่อติดต่อกับฐานข้อมูลของ SQLite (ออนไลน์). แหล่งที่มา : [http://www.thaicreate.com/mobile/android-sqlite-database.html. ๑๖](http://www.thaicreate.com/mobile/android-sqlite-database.html.%20๑๖) ตุลาคม ๒๕๕๘

100Y Design.๒๕๕๘.Illustrator (ออนไลน์). แหล่งที่มา :

[http://www.100ydesign.com/column.php?id=000103. ๑๖ ตุลาคม](http://www.100ydesign.com/column.php?id=000103.%20๑๖%20ตุลาคม) ๒๕๕๘

jaray hatta . ๒๕๕๘.Adobe Photoshop(ออนไลน์). แหล่งที่มา : [http://teacherjaray.blogspot.com/2014/05/blog-post.html. ๑๕](http://teacherjaray.blogspot.com/2014/05/blog-post.html.%20๑๕) ตุลาคม ๒๕๕๘

Georgios Christou (as part of COST Action 294: MAUSE); Carla Saraiva.๒๕๕๕.Hierarchical Task Analysis (ออนไลน์).แหล่งที่มา: www.usabilitybok.org/hierarchical-task-analysis. ๓ ธันวาคม ๒๕๕๘

MindPHP.๒๕๕๖.ความหมาย Data Dictionary (ออนไลน์).แหล่งที่มา: www.mindphp.com/คู่มือ/73-คืออะไร/2727-data-dictionary.html. ๓ ธันวาคม ๒๕๕๘

**ภาคผนวก ก**

**ซอร์สโค้ดของแอพพลิเคชั่น**

**ภาคผนวก ก**

**ซอร์สโค้ดของแอพพลิเคชั่น**

แอพพลิเคชั่นเพื่อบันทึกรายรับ-รายจ่ายในชีวิตประจำวัน เป็นแอพพลิเคชั่นที่ถูกพัฒนาบนระบบปฏิบัติการแอนดรอยด์ โดยใช้โปรแกรม Android Studio และเขียนด้วยภาษาจาวาเป็นหลัก และเทคโนโลยีที่นำมาใช้จะใช้ Active Android และ SQLite ซึ้งในขั้นตอนนี้ผู้จัดทำจะอธิบายการทำงานของโค้ดที่ใช้ไม่ว่าจะเป็นในส่วนของรายรับ รายจ่าย และรายงาน ทั้งไฟล์ .java และ .xml ดังต่อไปนี้

AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="org.usablelabs.duedo"**>  
  
 <**application  
 android:name=".DueDoApplication"  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="Expense"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".ListActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity android:name=".FormActivity"** />  
 <**activity android:name=".ShowActivity"** />  
 <**activity android:name=".ShowExpenseActivity"** />  
  
 <**meta-data  
 android:name="AA\_DB\_NAME"  
 android:value="DueDu.db"** />  
 <**meta-data  
 android:name="AA\_DB\_VERSION"  
 android:value="1"** />  
 <**meta-data  
 android:name="AA\_MODELS"  
 android:value="org.usablelabs.duedo.Task,org.usablelabs.duedo.Expense,org.usablelabs.duedo.Income"** />  
  
 <**activity android:name=".ReportActivity"** />  
 <**activity android:name=".FormExpenseActivity"** />  
 <**activity android:name=".FormIncomeActivity"** />  
 <**activity android:name=".Form2ExpenseActivity"** />  
 <**activity android:name=".IncomeActivity"** />  
 <**activity android:name=".BarChartActivity"** />  
 <**activity android:name=".LineChartActivity"** />  
 <**activity android:name=".ShowIncomeActivity"** />  
 <**activity android:name=".ExpenseActivity"**></**activity**>  
 </**application**>  
  
</**manifest**>

BaseActivity.java

**package** org.usablelabs.duedo;  
  
**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 {  
  
 **protected static final int** NEW\_TASK = 1;  
 @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, FormActivity.class);* **new** DrawerBuilder()  
 .withActivity(**this**)  
 .withToolbar(toolbar\_main)  
 .addDrawerItems(  
 **new** PrimaryDrawerItem()  
 .withName(**"รายรับ"**)  
 .withIcon(MaterialDesignIconic.Icon.gmi\_calendar)  
 .withSelectable(**false**)  
 .withIdentifier(1)  
 ).addDrawerItems(  
 **new** PrimaryDrawerItem()  
 .withName(**"รายจ่าย"**)  
 .withIcon(MaterialDesignIconic.Icon.gmi\_calendar\_check)  
 .withSelectable(**false**)  
 .withIdentifier(2)  
 ).addDrawerItems(  
 **new** PrimaryDrawerItem()  
 .withName(**"รายงาน"**)  
 .withIcon(MaterialDesignIconic.Icon.gmi\_dock)  
 .withSelectable(**false**)  
 .withIdentifier(3)

).withSelectedItem(-1)  
 .withOnDrawerItemClickListener(**new** Drawer.OnDrawerItemClickListener() {  
 @Override  
 **public boolean** onItemClick(View view, **int** position, IDrawerItem drawerItem) {  
 **switch** (drawerItem.getIdentifier()) {  
 **case** 1:  
 startActivityForResult(**new** Intent(getApplicationContext(), IncomeActivity.**class**), NEW\_TASK);  
 **break**;  
 **case** 2:  
 startActivityForResult(**new** Intent(getApplicationContext(), ExpenseActivity.**class**), NEW\_TASK);  
 **break**;  
 **case** 3:  
 startActivityForResult(**new** Intent(getApplicationContext(), ReportActivity.**class**), NEW\_TASK);  
 **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(IIcon icon) {  
 **return new** IconicsDrawable(**this**).icon(icon).color(Color.WHITE).sizeDp(24).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);  
 }  
}

Income.java

**package** org.usablelabs.duedo;  
  
**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 = **"Tasks"**)  
**public class** Income **extends** Model {  
  
 @Column(name = **"title"**)  
 **public** String **title**;  
  
 @Column(name = **"content"**)  
 **public** String **content**;  
  
 @Column(name = **"v8"**)  
 **public boolean v8**;  
  
 @Column(name = **"v9"**)  
 **public boolean v9**;  
  
 @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<Income> getAll() {  
 **return new** Select().from(Income.**class**).orderBy(**"updatedAt DESC"**).execute();  
 }  
  
 **public void** saveWithTimestamp() {  
 Date now = **new** Date();  
 **updatedAt** = now;  
 **if** (**createdAt** == **null**)  
 **createdAt** = now;  
 save();  
 }  
}

IncomeActivity.java

**package** org.usablelabs.duedo;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**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.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
**public class** IncomeActivity **extends** BaseActivity {  
  
 **private static final int *SHOW\_TASK*** = 2;  
  
 **private** ArrayList<Income> **income**;  
  
 **private** ListView **listView**;  
 **private** TextView **emptyLabel**;  
 **private** IncomeAdapter **adapter**;  
 **private** Button **btn\_add**;  
 **private int index\_protect** = 0;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_income***);  
 setDrawer(**true**);  
 setTitle(**"รายรับ"**);  
  
 **listView** = (ListView) findViewById(R.id.***listView***);  
 **emptyLabel** = (TextView) findViewById(R.id.***emptyLabel***);  
 *//setView();* **btn\_add** = (Button)findViewById(R.id.***btn\_add1***);  
 **btn\_add**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(IncomeActivity.**this**,FormIncomeActivity.**class**);  
 startActivity(intent);  
 }  
 });  
  
 setView();  
 }

**private void** setView() {  
 **income** = **new** ArrayList<Income>(Income.*getAll*());  
 **if** (**income**.isEmpty()) {  
 **listView**.setVisibility(View.***GONE***);  
 **emptyLabel**.setVisibility(View.***VISIBLE***);  
 } **else** {  
 **emptyLabel**.setVisibility(View.***GONE***);  
 **listView**.setVisibility(View.***VISIBLE***);  
 **adapter** = **new** IncomeAdapter(**this** , **income**);  
 **listView**.setAdapter(**adapter**);  
 **listView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Intent intent = **new** Intent(IncomeActivity.**this**, ShowIncomeActivity.**class**);  
 intent.putExtra(**"id"**, **income**.get(position).getId());  
 startActivityForResult(intent, ***SHOW\_TASK***);  
 }  
 });  
 }  
 }  
  
 @Override  
 **protected void** onRestart() {  
 **super**.onRestart();  
  
 **index\_protect** = 0;  
 Log.*e*(**"LOG"**, **"onrestrat"**);  
 **income** = **new** ArrayList<Income>(Income.*getAll*());  
 Log.*e*(**"SIZE"**,**income**.size()+**""**);  
 **adapter** = **new** IncomeAdapter(**this** , **income**);  
 **listView**.setAdapter(**adapter**);  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 onBackPressed();  
 **break**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **class** IncomeAdapter **extends** ArrayAdapter<Income> {  
  
 String **olddate**;  
 **private** ArrayList<Income> **alltasks**;  
  
 **public** IncomeAdapter(Context context, ArrayList<Income> tasks) {  
 **super**(context, 0, tasks);  
 **alltasks** = tasks;  
 }

@Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 *//Task task = getItem(position);* **if** (convertView == **null**) {  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.***item\_list\_main***, parent, **false**);  
 }  
 TextView tv = (TextView) convertView.findViewById(R.id.***txt\_title***);  
 tv.setText(**alltasks**.get(position).**title**);  
  
 TextView tv3 = (TextView) convertView.findViewById(R.id.***txt\_content***);  
 tv3.setText(**alltasks**.get(position).**content**);  
  
 String protect = **""**;  
  
  
  
 **if**(**alltasks**.get(position).**v8**){  
 protect += **"เงินเดือน"**;  
 **index\_protect**++;  
 }  
  
  
 **if**(**alltasks**.get(position).**v9**){  
 **if**(**index\_protect**==0)  
 protect += **"รายได้เสริมอื่นๆ"**;  
 **else** protect += **", รายได้เสริมอื่นๆ"**;  
 **index\_protect**++;  
 }  
  
 **index\_protect** = 0;  
  
 TextView tv2 = (TextView) convertView.findViewById(R.id.***txt\_name***);  
 tv2.setText(protect);  
  
 **return** convertView;  
 }  
 }  
}

Activity\_income.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="#66FFFF"  
 tools:context="org.usablelabs.duedo.IncomeActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
  
 <**Button  
 android:id="@+id/btn\_add1"  
 android:layout\_below="@+id/toolbar\_main"  
 android:background="#2E9AFE"  
 android:layout\_marginTop="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:text="เพิ่มข้อมูล"**/>  
  
 <**LinearLayout  
 android:layout\_below="@+id/btn\_add1"  
 android:id="@+id/layout\_head"  
 android:layout\_width="match\_parent"  
 android:background="#2E9AFE"  
 android:layout\_marginTop="5dp"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"**>  
  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="130dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="วันที่"**/>  
  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="130dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="หมวดหมู่"**/>  
  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="จำนวนเงิน"**/>  
 </**LinearLayout**>

<**ListView  
 android:id="@+id/listView"  
 android:scrollbars="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/layout\_head"  
 android:visibility="gone"**/>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceLarge"  
 android:text="ไม่มีข้อมูลรายรับ"  
 android:id="@+id/emptyLabel"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"  
 android:textColor="@android:color/darker\_gray"  
 android:visibility="gone"**/>  
  
</**RelativeLayout**>

ไฟล์ IncomeActivity.java เป็นไฟล์ที่เขียนคำสั่งให้มีการแสดงหน้าของ activity\_income.xml ซึ่งจะประกอบไปด้วยปุ่มเพิ่มข้อมูลเพื่อที่จะไปแสดงหน้า FormIncome.java และยังเป็นหน้า list ที่ใช้ในการแสดงข้อมูลเมื่อบันทึกเสร็จ เช่น

**btn\_add** = (Button)findViewById(R.id.***btn\_add1***);  
 **btn\_add**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(IncomeActivity.**this**,FormIncomeActivity.**class**);  
 startActivity(intent);  
 }  
 });  
  
 setView();  
 }

FormIncomeActivity.java

**package** org.usablelabs.duedo;  
  
  
**import** android.app.Activity;  
**import** android.app.AlertDialog;  
**import** android.app.FragmentTransaction;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
  
**import** com.mikepenz.material\_design\_iconic\_typeface\_library.MaterialDesignIconic;  
  
  
**public class** FormIncomeActivity **extends** BaseActivity {  
  
 **private** Income **db** = **null**;  
  
 **private** EditText **titleEdit**; *// date* **private** EditText **ContentEdit**; *//จำนวนเงิน* CheckBox **check8**,**check9**;  
  
 **private static final int *MenuItem\_SaveID*** = 1;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_form\_income***);  
 setDrawer(**true**);  
 setTitle(**"ข้อมูลรายรับ"**);  
  
 **titleEdit** = (EditText) findViewById(R.id.***dateEdit***);  
 **ContentEdit** = (EditText) findViewById(R.id.***date2Edit***);  
 **check8** = (CheckBox)findViewById(R.id.***checkBox8***);  
 **check9** = (CheckBox)findViewById(R.id.***checkBox9***);  
  
 }  
  
 **public void** onStart() {  
 **super**.onStart();  
 EditText dateEdit = (EditText) findViewById(R.id.***dateEdit***);  
 dateEdit.setOnFocusChangeListener(**new** View.OnFocusChangeListener() {  
 @Override  
 **public void** onFocusChange(View v, **boolean** hasFocus) {  
 **if** (hasFocus) {  
 DateDialog dialog = **new** DateDialog(v);  
 FragmentTransaction ft = getFragmentManager().beginTransaction();  
 dialog.show(ft, **"DatePicker"**);  
 }  
 }  
 });  
  
 }

@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***:  
 **this**.finish();  
 **break**;  
 **case *MenuItem\_SaveID***:  
 save();  
 **break**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **private void** save() {  
 **if** (**titleEdit**.getText().length() > 0) {  
 **if** (**db** == **null**)  
 **db** = **new** Income();  
 **db**.**title** = **titleEdit**.getText().toString();  
 **db**.**content** = **ContentEdit**.getText().toString();  
 **db**.**v8** = **check8**.isChecked();  
 **db**.**v9** = **check9**.isChecked();  
 **db**.saveWithTimestamp();  
 setResult(Activity.***RESULT\_OK***, **new** Intent().putExtra(**"id"**, **db**.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();  
 }  
 }  
}

activity\_form\_income.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:descendantFocusability="beforeDescendants"  
 android:focusableInTouchMode="true"  
 android:background="#66FFFF"  
 tools:context="org.usablelabs.duedo.FormIncomeActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />

<**LinearLayout  
 android:id="@+id/layout1"  
 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/dateEdit"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="วันที่"** />  
  
 </**android.support.design.widget.TextInputLayout**>  
  
 </**LinearLayout**>  
  
 <**TextView  
 android:id="@+id/txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="left"  
 android:layout\_alignRight="@+id/textView4"  
 android:layout\_alignEnd="@+id/textView4"  
 android:text="หมวดหมู่"  
 android:layout\_marginTop="150dp"  
 android:textStyle="bold"  
 android:textSize="25sp"  
 android:layout\_weight="0.05"** />  
  
 <**LinearLayout  
 android:id="@+id/layout2"  
 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/txt"  
 android:layout\_gravity="center\_horizontal"**>  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="เงินเดือน"  
 android:id="@+id/checkBox8"**/>  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="รายได้เสริมอื่นๆ"  
 android:id="@+id/checkBox9"**/>  
  
  
 </**LinearLayout**>  
  
 <**android.support.design.widget.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 </**android.support.design.widget.TextInputLayout**>  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:id="@+id/date2Edit"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="จำนวนเงิน"  
 android:layout\_below="@+id/layout2"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
  
</**RelativeLayout**>

</**LinearLayout**>  
  
 <**android.support.design.widget.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 </**android.support.design.widget.TextInputLayout**>  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:id="@+id/date2Edit"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="จำนวนเงิน"  
 android:layout\_below="@+id/layout2"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
  
</**RelativeLayout**>

ไฟล์ FormIncomeActivity.java เป็นไฟล์ที่เขียนคำสั่งให้มีการแสดงหน้าของ activity\_form\_income.xml ซึ่งจะแสดงหน้าของฟอร์มรายรับจะมีการให้บันทึกวันที่ หมวดหมู่ และจำนวนเงิน แล้วก็ save

ShowIncomeActivity.java

**package** org.usablelabs.duedo;  
  
**import** android.app.Activity;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AlertDialog;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.TextView;  
  
**import** com.mikepenz.material\_design\_iconic\_typeface\_library.MaterialDesignIconic;  
  
**public class** ShowIncomeActivity **extends** BaseActivity {  
  
 **private** Income **task** = **null**;  
 **private** TextView **titleView**;  
 **private** TextView **contentView**;  
  
 **private static final int *MenuItem\_EditID*** = 1;  
 **private static final int *MenuItem\_DeleteID*** = 2;  
 **private static final int *EDIT\_TASK*** = 14;  
 **long id**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_show\_income***);  
 setDrawer(**true**);  
 setTitle(**"ข้อมูลรายรับ"**);  
  
 **titleView** = (TextView) findViewById(R.id.***titleView***);  
 **contentView** = (TextView) findViewById(R.id.***contentView***);  
  
 **id** = getIntent().getLongExtra(**"id"**, 0);  
 setView(**id**);  
 }  
  
 **private void** setView(**long** id) {  
 **if** (id > 0)  
 **task** = Income.*load*(Income.**class**, id);  
 **if** (**task** != **null**) {  
 **titleView**.setText(**task**.**title**);  
  
 String protect = **""**;  
  
 **int** num = 0;  
 **if**(**task**.**v8**){  
 **if**(num>0) {  
 protect += **"เงินเดือน"**;  
 }**else**{  
 protect += **"เงินเดือน"**;  
 }  
 num++;  
 }

**if**(**task**.**v9**){  
 **if**(num>0) {  
 protect += **", รายได้เสริมอื่นๆ"**;  
 }**else**{  
 protect += **"รายได้เสริมอื่นๆ"**;  
 }  
 num++;  
 }  
  
 **contentView**.setText(protect);  
 } **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***:  
 Log.*e*(**"Edit"**,**"Edit"**);  
 Intent intent = **new** Intent(**this**, FormIncomeActivity.**class**);  
 intent.putExtra(**"id"**, **task**.getId());  
 startActivity(intent);  
 **this**.finish();  
 **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);  
 }  
}

Activity\_show\_income.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="#FFCCFF"  
 tools:context=".ShowIncomeActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
  
 <**LinearLayout  
 android:id="@+id/layout\_head"  
 android:layout\_width="match\_parent"  
 android:background="#e46dde"  
 android:layout\_marginTop="50dp"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"**>  
  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="130dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="วันที่"**/>  
  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="130dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="หมวดหมู่"**/>  
  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="จำนวนเงิน"**/>  
 </**LinearLayout**>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/titleView"  
 android:layout\_marginLeft="5dp"  
 android:paddingTop="95dp"**/>  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="260dp"  
 android:id="@+id/contentView"  
 android:paddingTop="95dp"** />  
 <**ListView  
 android:id="@+id/listView"  
 android:scrollbars="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:visibility="gone"**/>  
  
  
</**RelativeLayout**>

ไฟล์ ShowIncomeActivity.java เป็นไฟล์ที่เขียนคำสั่งให้มีการแสดงหน้าของ

activity\_show\_income.xml ซึ่งจะแสดงหน้าของข้อมูลรายรับที่ได้บันทึกไปแล้วจะมีการแสดงข้อมูล

ของวันที่ หมวดหมู่ และจำนวนเงิน และจะเป็นหน้าที่สามารถกดปุ่ม Edit และ Delete ได้และจะทำ

การกดปุ๋ม Edit เพื่อไปยังหน้าแก้ไขข้อมูลของรายรับ และกดปุ๋ม Delete เพื่อทำการลบข้อมูลรายรับ

ที่บันทึกไว้แล้ว

Expense.java

**package** org.usablelabs.duedo;  
  
**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 = **"Expense"**)  
**public class** Expense **extends** Model {  
  
 @Column(name = **"date"**)  
 **public** String **date**;  
  
 @Column(name = **"v1"**)  
 **public boolean v1**;  
  
 @Column(name = **"v2"**)  
 **public boolean v2**;  
  
 @Column(name = **"v3"**)  
 **public boolean v3**;  
  
 @Column(name = **"v4"**)  
 **public boolean v4**;  
  
 @Column(name = **"v5"**)  
 **public boolean v5**;  
  
 @Column(name = **"v6"**)  
 **public** String **v6**;  
  
 @Column(name = **"createdAt"**, index = **true**)  
 **public** Date **createdAt** = **null**;  
  
 @Column(name = **"updatedAt"**, index = **true**)  
 **public** Date **updatedAt** = **null**;  
  
 **public static** List<Expense> getAll() {  
 **return new** Select().from(Expense.**class**).orderBy(**"updatedAt DESC"**).execute();  
 }  
  
 **public void** saveWithTimestamp() {  
 Date now = **new** Date();  
 **updatedAt** = now;  
 **if** (**createdAt** == **null**)  
 **createdAt** = now;  
 save();  
 }  
}

ExpenseActivity.java

**package** org.usablelabs.duedo;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**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.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
**public class** ExpenseActivity **extends** BaseActivity {  
  
 **private static final int *SHOW\_TASK*** = 2;  
  
 **private** ArrayList<Expense> **expense**;  
  
 **private** ListView **listView**;  
 **private** TextView **emptyLabel**;  
 **private** ExpenseAdapter **adapter**;  
 **private** Button **btn\_add**;  
 **private int index\_protect** = 0;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_expense***);  
 setDrawer(**true**);  
 setTitle(**"รายจ่าย"**);  
  
 **listView** = (ListView) findViewById(R.id.***listView***);  
 **emptyLabel** = (TextView) findViewById(R.id.***emptyLabel***);  
 *//setView();* **btn\_add** = (Button)findViewById(R.id.***btn\_add***);  
 **btn\_add**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(ExpenseActivity.**this**,FormExpenseActivity.**class**);  
 startActivity(intent);  
 }  
 });  
  
 setView();  
 }  
  
 **private void** setView() {  
 **expense** = **new** ArrayList<Expense>(Expense.*getAll*());  
 **if** (**expense**.isEmpty()) {  
 **listView**.setVisibility(View.***GONE***);  
 **emptyLabel**.setVisibility(View.***VISIBLE***);  
 } **else** {  
 **emptyLabel**.setVisibility(View.***GONE***);  
 **listView**.setVisibility(View.***VISIBLE***);  
 **adapter** = **new** ExpenseAdapter(**this**, **expense**);  
 **listView**.setAdapter(**adapter**);  
 **listView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Intent intent = **new** Intent(ExpenseActivity.**this**, ShowExpenseActivity.**class**);  
 intent.putExtra(**"id"**, **expense**.get(position).getId());  
 startActivityForResult(intent, ***SHOW\_TASK***);  
 }  
 });  
 }  
 }  
  
 @Override  
 **protected void** onRestart() {  
 **super**.onRestart();  
  
 **index\_protect** = 0;  
 Log.*e*(**"LOG"**, **"onrestrat"**);  
 **expense** = **new** ArrayList<Expense>(Expense.*getAll*());  
 Log.*e*(**"SIZE"**,**expense**.size()+**""**);  
 **adapter** = **new** ExpenseAdapter(**this**, **expense**);  
 **listView**.setAdapter(**adapter**);  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 onBackPressed();  
 **break**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **class** ExpenseAdapter **extends** ArrayAdapter<Expense> {  
  
 String **olddate**;  
 **private** ArrayList<Expense> **alltasks**;  
  
 **public** ExpenseAdapter(Context context, ArrayList<Expense> tasks) {  
 **super**(context, 0, tasks);  
 **alltasks** = tasks;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 *//Task task = getItem(position);* **if** (convertView == **null**) {  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.***item\_list\_expense***, parent, **false**);  
 }  
 TextView tv = (TextView) convertView.findViewById(R.id.***txt\_date***);  
 tv.setText(**alltasks**.get(position).**date**);  
  
 TextView tv3 = (TextView) convertView.findViewById(R.id.***txt\_total***);  
 tv3.setText(**alltasks**.get(position).**v6**);  
  
 String protect = **""**;  
  
 Log.*e*(**"V1"**,**alltasks**.get(position).**v1**+**""**);  
 Log.*e*(**"V2"**,**alltasks**.get(position).**v2**+**""**);  
 Log.*e*(**"V3"**,**alltasks**.get(position).**v3**+**""**);  
 Log.*e*(**"V4"**,**alltasks**.get(position).**v4**+**""**);  
  
 **if**(**alltasks**.get(position).**v1**){  
 protect += **"ค่าอาหาร/เครื่องดื่ม"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v2**){  
 **if**(**index\_protect**==0) {  
 protect += **"เดินทาง/ท่องเที่ยว"**;  
 }**else**{  
 protect += **", เดินทาง/ท่องเที่ยว"**;  
 }  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v3**){  
 **if**(**index\_protect**==0)  
 protect += **"ส่วนตัว/Shopping"**;  
 **else** protect += **", ส่วนตัว/Shopping"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v4**){  
 **if**(**index\_protect**==0)  
 protect += **"บิลต่างๆ"**;  
 **else** protect += **", บิลต่างๆ"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v5**){  
 **if**(**index\_protect**==0)  
 protect += **"อื่นๆ"**;  
 **else** protect += **", อื่นๆ"**;  
 **index\_protect**++;  
 }  
  
 **index\_protect** = 0;  
  
 TextView tv2 = (TextView) convertView.findViewById(R.id.***txt\_protect***);  
 tv2.setText(protect);  
  
 **return** convertView;  
 }  
 }  
}

**private void** setView() {  
 **expense** = **new** ArrayList<Expense>(Expense.*getAll*());  
 **if** (**expense**.isEmpty()) {  
 **listView**.setVisibility(View.***GONE***);  
 **emptyLabel**.setVisibility(View.***VISIBLE***);  
 } **else** {  
 **emptyLabel**.setVisibility(View.***GONE***);  
 **listView**.setVisibility(View.***VISIBLE***);  
 **adapter** = **new** ExpenseAdapter(**this**, **expense**);  
 **listView**.setAdapter(**adapter**);  
 **listView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Intent intent = **new** Intent(ExpenseActivity.**this**, ShowExpenseActivity.**class**);  
 intent.putExtra(**"id"**, **expense**.get(position).getId());  
 startActivityForResult(intent, ***SHOW\_TASK***);  
 }  
 });  
 }  
 }  
  
 @Override  
 **protected void** onRestart() {  
 **super**.onRestart();  
  
 **index\_protect** = 0;  
 Log.*e*(**"LOG"**, **"onrestrat"**);  
 **expense** = **new** ArrayList<Expense>(Expense.*getAll*());  
 Log.*e*(**"SIZE"**,**expense**.size()+**""**);  
 **adapter** = **new** ExpenseAdapter(**this**, **expense**);  
 **listView**.setAdapter(**adapter**);  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 onBackPressed();  
 **break**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **class** ExpenseAdapter **extends** ArrayAdapter<Expense> {  
  
 String **olddate**;  
 **private** ArrayList<Expense> **alltasks**;  
  
 **public** ExpenseAdapter(Context context, ArrayList<Expense> tasks) {  
 **super**(context, 0, tasks);  
 **alltasks** = tasks;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 *//Task task = getItem(position);* **if** (convertView == **null**) {  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.***item\_list\_expense***, parent, **false**);  
 }  
 TextView tv = (TextView) convertView.findViewById(R.id.***txt\_date***);  
 tv.setText(**alltasks**.get(position).**date**);  
  
 TextView tv3 = (TextView) convertView.findViewById(R.id.***txt\_total***);  
 tv3.setText(**alltasks**.get(position).**v6**);  
  
 String protect = **""**;  
  
 Log.*e*(**"V1"**,**alltasks**.get(position).**v1**+**""**);  
 Log.*e*(**"V2"**,**alltasks**.get(position).**v2**+**""**);  
 Log.*e*(**"V3"**,**alltasks**.get(position).**v3**+**""**);  
 Log.*e*(**"V4"**,**alltasks**.get(position).**v4**+**""**);  
  
 **if**(**alltasks**.get(position).**v1**){  
 protect += **"ค่าอาหาร/เครื่องดื่ม"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v2**){  
 **if**(**index\_protect**==0) {  
 protect += **"เดินทาง/ท่องเที่ยว"**;  
 }**else**{  
 protect += **", เดินทาง/ท่องเที่ยว"**;  
 }  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v3**){  
 **if**(**index\_protect**==0)  
 protect += **"ส่วนตัว/Shopping"**;  
 **else** protect += **", ส่วนตัว/Shopping"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v4**){  
 **if**(**index\_protect**==0)  
 protect += **"บิลต่างๆ"**;  
 **else** protect += **", บิลต่างๆ"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v5**){  
 **if**(**index\_protect**==0)  
 protect += **"อื่นๆ"**;  
 **else** protect += **", อื่นๆ"**;  
 **index\_protect**++;  
 }  
  
 **index\_protect** = 0;  
  
 TextView tv2 = (TextView) convertView.findViewById(R.id.***txt\_protect***);  
 tv2.setText(protect);  
  
 **return** convertView;  
 }  
 }  
}

@Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 *//Task task = getItem(position);* **if** (convertView == **null**) {  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.***item\_list\_expense***, parent, **false**);  
 }  
 TextView tv = (TextView) convertView.findViewById(R.id.***txt\_date***);  
 tv.setText(**alltasks**.get(position).**date**);  
  
 TextView tv3 = (TextView) convertView.findViewById(R.id.***txt\_total***);  
 tv3.setText(**alltasks**.get(position).**v6**);  
  
 String protect = **""**;  
  
 Log.*e*(**"V1"**,**alltasks**.get(position).**v1**+**""**);  
 Log.*e*(**"V2"**,**alltasks**.get(position).**v2**+**""**);  
 Log.*e*(**"V3"**,**alltasks**.get(position).**v3**+**""**);  
 Log.*e*(**"V4"**,**alltasks**.get(position).**v4**+**""**);  
  
 **if**(**alltasks**.get(position).**v1**){  
 protect += **"ค่าอาหาร/เครื่องดื่ม"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v2**){  
 **if**(**index\_protect**==0) {  
 protect += **"เดินทาง/ท่องเที่ยว"**;  
 }**else**{  
 protect += **", เดินทาง/ท่องเที่ยว"**;  
 }  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v3**){  
 **if**(**index\_protect**==0)  
 protect += **"ส่วนตัว/Shopping"**;  
 **else** protect += **", ส่วนตัว/Shopping"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v4**){  
 **if**(**index\_protect**==0)  
 protect += **"บิลต่างๆ"**;  
 **else** protect += **", บิลต่างๆ"**;  
 **index\_protect**++;  
 }  
  
 **if**(**alltasks**.get(position).**v5**){  
 **if**(**index\_protect**==0)  
 protect += **"อื่นๆ"**;  
 **else** protect += **", อื่นๆ"**;  
 **index\_protect**++;  
 }

**index\_protect** = 0;  
  
 TextView tv2 = (TextView) convertView.findViewById(R.id.***txt\_protect***);  
 tv2.setText(protect);  
  
 **return** convertView;  
 }  
 }  
}

Activity\_expense.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="#CCFFFF"  
 tools:context="org.usablelabs.duedo.ExpenseActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
  
 <**Button  
 android:id="@+id/btn\_add"  
 android:layout\_below="@+id/toolbar\_main"  
 android:background="#2E9AFE"  
 android:layout\_marginTop="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:text="เพิ่มข้อมูล"**/>  
  
 <**LinearLayout  
 android:layout\_below="@+id/btn\_add"  
 android:id="@+id/layout\_head"  
 android:layout\_width="match\_parent"  
 android:background="#2E9AFE"  
 android:layout\_marginTop="5dp"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"**>  
  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="130dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="วันที่"**/>  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="130dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="หมวดหมู่"**/>  
  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="จำนวนเงิน"**/>  
 </**LinearLayout**>  
  
 <**ListView  
 android:id="@+id/listView"  
 android:scrollbars="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/layout\_head"  
 android:visibility="gone"**/>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceLarge"  
 android:text="ไม่มีข้อมูลรายจ่าย"  
 android:id="@+id/emptyLabel"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"  
 android:textColor="@android:color/darker\_gray"  
 android:visibility="gone"**/>  
  
</**RelativeLayout**>

<**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="จำนวนเงิน"**/>  
 </**LinearLayout**>  
  
 <**ListView  
 android:id="@+id/listView"  
 android:scrollbars="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/layout\_head"  
 android:visibility="gone"**/>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceLarge"  
 android:text="ไม่มีข้อมูลรายจ่าย"  
 android:id="@+id/emptyLabel"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"  
 android:textColor="@android:color/darker\_gray"  
 android:visibility="gone"**/>  
  
</**RelativeLayout**>

ไฟล์ ExpenseActivity.java เป็นไฟล์ที่เขียนคำสั่งให้มีการแสดงหน้าของ activity\_expense.xml ซึ่งจะประกอบไปด้วยปุ่มเพิ่มข้อมูลเพื่อที่จะไปแสดงหน้า FormExpense.java และยังเป็นหน้า list ที่ใช้ในการแสดงข้อมูลเมื่อบันทึกเสร็จ เช่น

**btn\_add** = (Button)findViewById(R.id.***btn\_add***);  
 **btn\_add**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(ExpenseActivity.**this**,FormExpenseActivity.**class**);  
 startActivity(intent);  
 }  
 });  
  
 setView();  
 }

FormExpenseActivity.java

**package** org.usablelabs.duedo;  
  
  
**import** android.app.Activity;  
**import** android.app.AlertDialog;  
**import** android.app.FragmentTransaction;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
  
**import** com.mikepenz.material\_design\_iconic\_typeface\_library.MaterialDesignIconic;  
  
  
**public class** FormExpenseActivity **extends** BaseActivity {  
  
 **private** Expense **db** = **null**;  
  
 **private** EditText **titleEdit**; *// date* **private** EditText **ContentEdit**; *//จำนวนเงิน* CheckBox **check1**,**check2**,**check3**,**check4**,**check5**;  
  
 **private static final int *MenuItem\_SaveID*** = 1;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_form\_expense***);  
 setDrawer(**true**);  
 setTitle(**"เพิ่มข้อมูลรายจ่าย"**);  
  
 **titleEdit** = (EditText) findViewById(R.id.***sexdateEdit***);  
  
 **check1** = (CheckBox)findViewById(R.id.***checkBox1***);  
 **check2** = (CheckBox)findViewById(R.id.***checkBox2***);  
 **check3** = (CheckBox)findViewById(R.id.***checkBox3***);  
 **check4** = (CheckBox)findViewById(R.id.***checkBox4***);  
 **check5** = (CheckBox)findViewById(R.id.***checkBox5***);  
  
 **ContentEdit** = (EditText) findViewById(R.id.***total***);  
 }  
  
 **public void** onStart() {  
 **super**.onStart();  
 EditText dateEdit = (EditText) findViewById(R.id.***sexdateEdit***);  
 dateEdit.setOnFocusChangeListener(**new** View.OnFocusChangeListener() {  
 @Override  
 **public void** onFocusChange(View v, **boolean** hasFocus) {  
 **if** (hasFocus) {  
 DateDialog dialog = **new** DateDialog(v);  
 FragmentTransaction ft = getFragmentManager().beginTransaction();  
 dialog.show(ft, **"DatePicker"**);  
 }  
 }  
 });  
  
 }  
  
 @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***:  
 **this**.finish();  
 **break**;  
 **case *MenuItem\_SaveID***:  
 save();  
 **break**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **private void** save() {  
 **if** (**titleEdit**.getText().length() > 0) {  
 **if** (**db** == **null**)  
 **db** = **new** Expense();  
 **db**.**date** = **titleEdit**.getText().toString();  
 **db**.**v1** = **check1**.isChecked();  
 Log.*e*(**"c1"**,**check1**.isChecked()+**""**);  
 **db**.**v2** = **check2**.isChecked();  
 Log.*e*(**"c1"**,**check2**.isChecked()+**""**);  
 **db**.**v3** = **check3**.isChecked();  
 Log.*e*(**"c1"**,**check3**.isChecked()+**""**);  
 **db**.**v4** = **check4**.isChecked();  
 Log.*e*(**"c1"**,**check4**.isChecked()+**""**);  
 **db**.**v5** = **check5**.isChecked();  
 Log.*e*(**"c1"**,**check5**.isChecked()+**""**);  
 **db**.**v6** = **ContentEdit**.getText().toString();  
 **db**.saveWithTimestamp();  
 setResult(Activity.***RESULT\_OK***, **new** Intent().putExtra(**"id"**, **db**.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 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***:  
 **this**.finish();  
 **break**;  
 **case *MenuItem\_SaveID***:  
 save();  
 **break**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **private void** save() {  
 **if** (**titleEdit**.getText().length() > 0) {  
 **if** (**db** == **null**)  
 **db** = **new** Expense();  
 **db**.**date** = **titleEdit**.getText().toString();  
 **db**.**v1** = **check1**.isChecked();  
 Log.*e*(**"c1"**,**check1**.isChecked()+**""**);  
 **db**.**v2** = **check2**.isChecked();  
 Log.*e*(**"c1"**,**check2**.isChecked()+**""**);  
 **db**.**v3** = **check3**.isChecked();  
 Log.*e*(**"c1"**,**check3**.isChecked()+**""**);  
 **db**.**v4** = **check4**.isChecked();  
 Log.*e*(**"c1"**,**check4**.isChecked()+**""**);  
 **db**.**v5** = **check5**.isChecked();  
 Log.*e*(**"c1"**,**check5**.isChecked()+**""**);  
 **db**.**v6** = **ContentEdit**.getText().toString();  
 **db**.saveWithTimestamp();  
 setResult(Activity.***RESULT\_OK***, **new** Intent().putExtra(**"id"**, **db**.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();  
 }  
 }  
}

Activity\_form\_expense.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:descendantFocusability="beforeDescendants"  
 android:focusableInTouchMode="true"  
 android:background="#66FFFF"  
 tools:context="org.usablelabs.duedo.Form2ExpenseActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
  
 <**LinearLayout  
 android:id="@+id/layout1"  
 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/sexdateEdit"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="วันที่"** />  
  
 </**android.support.design.widget.TextInputLayout**>  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 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"  
 android:id="@+id/view"  
 android:layout\_centerVertical="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"**>

<**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="left"  
 android:layout\_alignRight="@+id/textView4"  
 android:layout\_alignEnd="@+id/textView4"  
 android:text="หมวดหมู่"  
 android:layout\_marginBottom="20dp"  
 android:textStyle="bold"  
 android:textSize="25sp"  
 android:layout\_weight="0.05"** />  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"** >  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="ค่าอาหาร/เครื่องดื่ม"  
 android:id="@+id/checkBox1"**/>  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="เดินทาง/ท่องเที่ยว"  
 android:id="@+id/checkBox2"**/>  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="ส่วนตัว/Shopping"  
 android:id="@+id/checkBox3"** />  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="บิลต่างๆ"  
 android:id="@+id/checkBox4"**/>  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="อื่นๆ"  
 android:id="@+id/checkBox5"** />  
  
 </**LinearLayout**>  
 </**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**>  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:id="@+id/total"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="จำนวนเงิน"  
 android:layout\_marginTop="400dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
  
  
  
</**RelativeLayout**>

<**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="left"  
 android:layout\_alignRight="@+id/textView4"  
 android:layout\_alignEnd="@+id/textView4"  
 android:text="หมวดหมู่"  
 android:layout\_marginBottom="20dp"  
 android:textStyle="bold"  
 android:textSize="25sp"  
 android:layout\_weight="0.05"** />  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"** >  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="ค่าอาหาร/เครื่องดื่ม"  
 android:id="@+id/checkBox1"**/>  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="เดินทาง/ท่องเที่ยว"  
 android:id="@+id/checkBox2"**/>  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="ส่วนตัว/Shopping"  
 android:id="@+id/checkBox3"** />  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="บิลต่างๆ"  
 android:id="@+id/checkBox4"**/>  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="อื่นๆ"  
 android:id="@+id/checkBox5"** />  
  
 </**LinearLayout**>  
 </**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**>

<**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:id="@+id/total"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="จำนวนเงิน"  
 android:layout\_marginTop="400dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
  
  
  
</**RelativeLayout**>

<**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:id="@+id/total"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="จำนวนเงิน"  
 android:layout\_marginTop="400dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
</**RelativeLayout**>

ไฟล์ FormExpenseActivity.java เป็นไฟล์ที่เขียนคำสั่งให้มีการแสดงหน้าของ activity\_form\_expense.xml ซึ่งจะแสดงหน้าของฟอร์มรายจ่ายจะมีการให้บันทึกวันที่ หมวดหมู่ และจำนวนเงิน แล้วก็ save

ShowExpenseActivity.java

**package** org.usablelabs.duedo;  
  
**import** android.app.Activity;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AlertDialog;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.TextView;  
  
**import** com.mikepenz.material\_design\_iconic\_typeface\_library.MaterialDesignIconic;  
  
**public class** ShowExpenseActivity **extends** BaseActivity {  
  
 **private** Expense **task** = **null**;  
 **private** TextView **titleView**;  
 **private** TextView **contentView**;  
  
 **private static final int *MenuItem\_EditID*** = 1;  
 **private static final int *MenuItem\_DeleteID*** = 2;  
 **private static final int *EDIT\_TASK*** = 14;  
 **long id**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_show***);  
 setDrawer(**true**);  
 setTitle(**"ข้อมูลรายจ่าย"**);  
  
 **titleView** = (TextView) findViewById(R.id.***titleView***);  
 **contentView** = (TextView) findViewById(R.id.***contentView***);  
  
 **id** = getIntent().getLongExtra(**"id"**, 0);  
 setView(**id**);  
 }  
  
 **private void** setView(**long** id) {  
 **if** (id > 0)  
 **task** = Expense.*load*(Expense.**class**, id);  
 **if** (**task** != **null**) {  
 **titleView**.setText(**task**.**date**);  
  
 String protect = **""**;  
  
 **int** num = 0;  
 **if**(**task**.**v1**){  
 **if**(num>0) {  
 protect += **"ค่าอาหาร/เครื่องดื่ม"**;  
 }**else**{  
 protect += **"ค่าอาหาร/เครื่องดื่ม"**;  
 }  
 num++;  
 }  
  
 **if**(**task**.**v2**){  
 **if**(num>0) {  
 protect += **", เดินทาง/ท่องเที่ยว"**;  
 }**else**{  
 protect += **"เดินทาง/ท่องเที่ยว"**;  
 }  
 num++;  
 }  
  
 **if**(**task**.**v3**){  
 **if**(num>0) {  
 protect += **", ส่วนตัว/Shopping"**;  
 }**else**{  
 protect += **"ส่วนตัว/Shopping"**;  
 }  
 num++;  
 }  
  
 **if**(**task**.**v4**){  
 **if**(num>0) {  
 protect += **", บิลต่างๆ"**;  
 }**else**{  
 protect += **"บิลต่างๆ"**;  
 }  
 num++;  
 }  
  
 **if**(**task**.**v5**){  
 **if**(num>0) {  
 protect += **", อื่นๆ"**;  
 }**else**{  
 protect += **"อื่นๆ"**;  
 }  
 num++;  
 }  
  
 **contentView**.setText(protect);  
 } **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***:  
 Log.*e*(**"Edit"**,**"Edit"**);  
 Intent intent = **new** Intent(**this**, Form2ExpenseActivity.**class**);  
 intent.putExtra(**"id"**, **task**.getId());  
 startActivity(intent);  
 **this**.finish();  
 **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;  
 }  
 }  
 }\*/*}

@Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_show***);  
 setDrawer(**true**);  
 setTitle(**"ข้อมูลรายจ่าย"**);  
  
 **titleView** = (TextView) findViewById(R.id.***titleView***);  
 **contentView** = (TextView) findViewById(R.id.***contentView***);  
  
 **id** = getIntent().getLongExtra(**"id"**, 0);  
 setView(**id**);  
 }  
  
 **private void** setView(**long** id) {  
 **if** (id > 0)  
 **task** = Expense.*load*(Expense.**class**, id);  
 **if** (**task** != **null**) {  
 **titleView**.setText(**task**.**date**);  
  
 String protect = **""**;  
  
 **int** num = 0;  
 **if**(**task**.**v1**){  
 **if**(num>0) {  
 protect += **"ค่าอาหาร/เครื่องดื่ม"**;  
 }**else**{  
 protect += **"ค่าอาหาร/เครื่องดื่ม"**;  
 }  
 num++;  
 }  
  
 **if**(**task**.**v2**){  
 **if**(num>0) {  
 protect += **", เดินทาง/ท่องเที่ยว"**;  
 }**else**{  
 protect += **"เดินทาง/ท่องเที่ยว"**;  
 }  
 num++;  
 }  
  
 **if**(**task**.**v3**){  
 **if**(num>0) {  
 protect += **", ส่วนตัว/Shopping"**;  
 }**else**{  
 protect += **"ส่วนตัว/Shopping"**;  
 }  
 num++;  
 }  
  
 **if**(**task**.**v4**){  
 **if**(num>0) {  
 protect += **", บิลต่างๆ"**;  
 }**else**{  
 protect += **"บิลต่างๆ"**;  
 }  
 num++;  
 }  
  
 **if**(**task**.**v5**){  
 **if**(num>0) {  
 protect += **", อื่นๆ"**;  
 }**else**{  
 protect += **"อื่นๆ"**;  
 }  
 num++;  
 }  
  
 **contentView**.setText(protect);  
 } **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***:  
 Log.*e*(**"Edit"**,**"Edit"**);  
 Intent intent = **new** Intent(**this**, Form2ExpenseActivity.**class**);  
 intent.putExtra(**"id"**, **task**.getId());  
 startActivity(intent);  
 **this**.finish();  
 **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;  
 }  
 }  
 }\*/*}

**if**(**task**.**v5**){  
 **if**(num>0) {  
 protect += **", อื่นๆ"**;  
 }**else**{  
 protect += **"อื่นๆ"**;  
 }  
 num++;  
 }  
  
 **contentView**.setText(protect);  
 } **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***:  
 Log.*e*(**"Edit"**,**"Edit"**);  
 Intent intent = **new** Intent(**this**, Form2ExpenseActivity.**class**);  
 intent.putExtra(**"id"**, **task**.getId());  
 startActivity(intent);  
 **this**.finish();  
 **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;  
 }  
 }  
 }\*/*}

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:orientation="vertical"  
 android:background="#66FFFF"  
 tools:context="org.usablelabs.duedo.ShowExpenseActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceLarge"  
 android:text=""  
 android:id="@+id/titleView"  
 android:layout\_below="@+id/toolbar\_main"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"** />  
  
 <**ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/scrollView"  
 android:layout\_below="@+id/titleView"  
 android:layout\_centerHorizontal="true"**>  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=""  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:scrollbars="vertical"  
 android:scrollHorizontally="false"  
 android:id="@+id/contentView"** />  
 </**ScrollView**>  
  
</**RelativeLayout**>

ไฟล์ ShowExpenseActivity.java เป็นไฟล์ที่เขียนคำสั่งให้มีการแสดงหน้าของ activity\_show\_expense.xml ซึ่งจะแสดงหน้าของข้อมูลรายจ่ายที่ได้บันทึกไปแล้วจะมีการแสดงข้อมูลของวันที่ หมวดหมู่ และจำนวนเงิน และจะเป็นหน้าที่สามารถกดปุ่ม Edit และ Delete ได้และจะทำ การกดปุ๋ม Edit เพื่อไปยังหน้าแก้ไขข้อมูลของรายจ่าย และกดปุ๋ม Delete เพื่อทำการลบข้อมูลรายจ่าย ที่บันทึกไว้แล้ว

Task.java

**package** org.usablelabs.duedo;  
  
**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 = **"Tasks"**)  
**public class** Task **extends** Model {  
  
 @Column(name = **"title"**)  
 **public** String **title**;  
  
 @Column(name = **"content"**)  
 **public** String **content**;  
  
 @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();  
 }  
}

ListActivity.java

**package** org.usablelabs.duedo;  
  
**import** android.app.Activity;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
**public class** ListActivity **extends** BaseActivity {  
  
 **private static final int *SHOW\_TASK*** = 2;  
  
 **private** ArrayList<Task> **tasks**;  
  
 **private** ListView **listView**;  
 **private** TextView **emptyLabel**;  
 **private** TasksAdapter **adapter**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_list***);  
 setDrawer(**false**);  
 setTitle(**"รายรับ-รายจ่าย ของฉัน"**);  
  
  
  
 **listView** = (ListView) findViewById(R.id.***listView***);  
 **emptyLabel** = (TextView) findViewById(R.id.***emptyLabel***);  
  
 *//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<Task>(Task.*getAll*());  
 **if** (**tasks**.isEmpty()) {  
 **listView**.setVisibility(View.***GONE***);  
 **emptyLabel**.setVisibility(View.***VISIBLE***);  
 } **else** {  
 **emptyLabel**.setVisibility(View.***GONE***);  
 **listView**.setVisibility(View.***VISIBLE***);  
 **adapter** = **new** TasksAdapter(**this**, **tasks**);  
 **listView**.setAdapter(**adapter**);  
 **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***);  
 }  
 });  
 }  
 }  
  
 @Override  
 **protected void** onRestart() {  
 **super**.onRestart();  
  
 Log.*e*(**"LOG"**, **"onrestrat"**);  
 **tasks** = **new** ArrayList<Task>(Task.*getAll*());  
 Log.*e*(**"SIZE"**,**tasks**.size()+**""**);  
 **adapter** = **new** TasksAdapter(**this**, **tasks**);  
 **listView**.setAdapter(**adapter**);  
 }  
}  
  
**class** TasksAdapter **extends** ArrayAdapter<Task> {  
  
 String **olddate**;  
 **private** ArrayList<Task> **alltasks**;  
  
 **public** TasksAdapter(Context context, ArrayList<Task> tasks) {  
 **super**(context, 0, tasks);  
 **alltasks** = tasks;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 *//Task task = getItem(position);* **if** (convertView == **null**) {  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.***item\_list\_main***, parent, **false**);  
 }  
 TextView tv = (TextView) convertView.findViewById(R.id.***txt\_title***);  
 tv.setText(**alltasks**.get(position).**title**);  
  
 TextView tv2 = (TextView) convertView.findViewById(R.id.***txt\_content***);  
 tv2.setText(**alltasks**.get(position).**content**);  
  
  
 **return** convertView;  
 }  
}

**private void** setView() {  
 **tasks** = **new** ArrayList<Task>(Task.*getAll*());  
 **if** (**tasks**.isEmpty()) {  
 **listView**.setVisibility(View.***GONE***);  
 **emptyLabel**.setVisibility(View.***VISIBLE***);  
 } **else** {  
 **emptyLabel**.setVisibility(View.***GONE***);  
 **listView**.setVisibility(View.***VISIBLE***);  
 **adapter** = **new** TasksAdapter(**this**, **tasks**);  
 **listView**.setAdapter(**adapter**);  
 **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***);  
 }  
 });  
 }  
 }  
  
 @Override  
 **protected void** onRestart() {  
 **super**.onRestart();  
  
 Log.*e*(**"LOG"**, **"onrestrat"**);  
 **tasks** = **new** ArrayList<Task>(Task.*getAll*());  
 Log.*e*(**"SIZE"**,**tasks**.size()+**""**);  
 **adapter** = **new** TasksAdapter(**this**, **tasks**);  
 **listView**.setAdapter(**adapter**);  
 }  
}  
  
**class** TasksAdapter **extends** ArrayAdapter<Task> {  
  
 String **olddate**;  
 **private** ArrayList<Task> **alltasks**;  
  
 **public** TasksAdapter(Context context, ArrayList<Task> tasks) {  
 **super**(context, 0, tasks);  
 **alltasks** = tasks;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 *//Task task = getItem(position);* **if** (convertView == **null**) {  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.***item\_list\_main***, parent, **false**);  
 }  
 TextView tv = (TextView) convertView.findViewById(R.id.***txt\_title***);  
 tv.setText(**alltasks**.get(position).**title**);  
  
 TextView tv2 = (TextView) convertView.findViewById(R.id.***txt\_content***);  
 tv2.setText(**alltasks**.get(position).**content**);  
  
  
 **return** convertView;  
 }  
}

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="#66FFFF"  
 tools:context="org.usablelabs.duedo.ListActivity"**>  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
 <**LinearLayout  
 android:layout\_below="@+id/toolbar\_main"  
 android:id="@+id/layout\_head"  
 android:layout\_width="match\_parent"  
 android:background="#2E9AFE"  
 android:layout\_marginTop="5dp"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"**>  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="100dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="เดือน"**/>  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="90dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="รายรับ"**/>  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="90dp"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="รายจ่าย"**/>  
 <**TextView android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:text="ยอดคงเหลือ"**/>  
 </**LinearLayout**>  
 <**ListView  
 android:id="@+id/listView"  
 android:scrollbars="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/layout\_head"  
 android:visibility="gone"**/>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceLarge"  
 android:text="ไม่มีข้อมูล"  
 android:id="@+id/emptyLabel"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"  
 android:textColor="@android:color/darker\_gray"  
 android:visibility="gone"**/>  
</**RelativeLayout**>

ReportActivity.java

**package** org.usablelabs.duedo;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
  
**public class** ReportActivity **extends** BaseActivity {  
  
 Button **btn\_1**, **btn\_2**, **btn\_3**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_report***);  
 setTitle(**"รายงาน"**);  
 setDrawer(**true**);  
  
  
 **btn\_1** = (Button)findViewById(R.id.***btn\_1***);  
 **btn\_1**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(ReportActivity.**this**, BarChartActivity.**class**);  
 startActivity(intent);  
 }  
 });  
  
 **btn\_2** = (Button)findViewById(R.id.***btn\_2***);  
 **btn\_2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(ReportActivity.**this**, LineChartActivity.**class**);  
 startActivity(intent);  
 }  
 });  
  
 }  
  
 }

activity\_report.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout 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="#99FF66"  
 tools:context="org.usablelabs.duedo.ReportActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
   
 <**LinearLayout  
 android:layout\_height="match\_parent"  
 android:layout\_width="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:padding="16dp"**>  
  
 <**Button  
 android:id="@+id/btn\_1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="5dp"  
 android:text="รายงานรายรับ"** />  
  
  
 <**Button  
 android:id="@+id/btn\_2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="5dp"  
 android:text="กราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน"** />  
  
 <**Button  
 android:id="@+id/btn\_3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="รายงานยอดคงเหลือแต่ละเดือน"**/>  
  
 </**LinearLayout**>  
  
</**LinearLayout**>

BarChartActivity.java

**package** org.usablelabs.duedo;  
  
**import** android.graphics.Color;  
**import** android.os.Bundle;  
**import** android.util.Log;  
  
**import** com.github.mikephil.charting.charts.BarChart;  
**import** com.github.mikephil.charting.data.BarData;  
**import** com.github.mikephil.charting.data.BarDataSet;  
**import** com.github.mikephil.charting.data.BarEntry;  
**import** com.github.mikephil.charting.interfaces.datasets.IBarDataSet;  
  
**import** java.text.DateFormat;  
**import** java.text.ParseException;  
**import** java.text.SimpleDateFormat;  
**import** java.util.ArrayList;  
**import** java.util.Date;  
  
**public class** BarChartActivity **extends** BaseActivity {  
  
 **private** ArrayList<Income> **income**;  
 **private** ArrayList<Integer> **month**;  
 ArrayList<String> **xAxis**;  
 ArrayList<Integer> **yes**;  
 ArrayList<Integer> **no**;  
 ArrayList<IBarDataSet> **dataSets** = **new** ArrayList<IBarDataSet>();  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_bar\_chart***);  
 setTitle(**"รายงานรายรับ"**);  
 setDrawer(**true**);  
 BarChart chart = (BarChart) findViewById(R.id.***chart***);  
 **income** = **new** ArrayList<Income>(Income.*getAll*());  
 **month** = **new** ArrayList<>();  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
  
 **yes** = **new** ArrayList<>();  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
  
 **no** = **new** ArrayList<>();  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
  
 getXAxisValues();  
  
 BarData data = **new** BarData(**xAxis**, getDataSet());  
 chart.setData(data);  
 chart.setDescription(**""**);  
 chart.animateXY(2000, 2000);  
 chart.invalidate();  
 }  
  
 **private** ArrayList<IBarDataSet> getDataSet() {  
 ArrayList<IBarDataSet> dataSets = **null**;  
  
 ArrayList<BarEntry> valueSet1 = **new** ArrayList<>();  
 ArrayList<BarEntry> valueSet2 = **new** ArrayList<>();  
 **int** num = 0;  
 **for** (**int** i=0;i<**yes**.size();i++){  
 Log.*e*(i+**":"**, **yes**.get(i)+**" , "**+**no**.get(i));  
 **if**(**yes**.get(i)>0 || **no**.get(i)>0) {  
 *//Log.e("Yes", yes.get(i) + "");* BarEntry v1e1 = **new** BarEntry(**yes**.get(i), num); *// Jan* valueSet1.add(v1e1);  
  
 *//Log.e("No", no.get(i) + "");* BarEntry v2e1 = **new** BarEntry(**no**.get(i), num); *// Jan* valueSet2.add(v2e1);  
 num++;  
 }  
 }  
  
 BarDataSet barDataSet1 = **new** BarDataSet(valueSet1, **"เงินเดือน"**);  
 barDataSet1.setColor(Color.*rgb*(255, 0, 127));  
 BarDataSet barDataSet2 = **new** BarDataSet(valueSet2, **"รายได้เสริมอื่นๆ"**);  
 barDataSet2.setColor(Color.*rgb*(51, 255, 255));  
  
 dataSets = **new** ArrayList<>();  
 dataSets.add(barDataSet1);  
 dataSets.add(barDataSet2);  
 **return** dataSets;  
 }  
  
 **private void** getXAxisValues() {  
 **xAxis** = **new** ArrayList<>();  
 **for** (**int** i=0;i<**income**.size();i++){  
  
 DateFormat df = **new** SimpleDateFormat(**"dd-MM-yyyy"**);  
 Date startDate = **null**;  
 **try** {  
 startDate = df.parse(**income**.get(i).**title**);  
 String newDateString = df.format(startDate);  
 System.***out***.println(newDateString);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
  
 startDate.getMonth();  
 **month**.set(startDate.getMonth(), **month**.get(startDate.getMonth()) + 1);  
 **if**(**income**.get(i).**v8**){  
 **no**.set(startDate.getMonth(),**no**.get(startDate.getMonth())+1);  
 }**else**{  
 **yes**.set(startDate.getMonth(), **yes**.get(startDate.getMonth()) + 1);  
 }  
 }  
  
 **for**(**int** i=0;i<12;i++){  
 String value = **""**;  
 **if**(**month**.get(i)>0){  
 **switch** (i){  
 **case** 0:  
 value = **"JAN"**;  
 **break**;  
 **case** 1:  
 value = **"FAB"**;  
 **break**;  
 **case** 2:  
 value = **"MAR"**;  
 **break**;  
 **case** 3:  
 value = **"APR"**;  
 **break**;  
 **case** 4:  
 value = **"MAY"**;  
 **break**;  
 **case** 5:  
 value = **"JUN"**;  
 **break**;  
 **case** 6:  
 value = **"JUL"**;  
 **break**;  
 **case** 7:  
 value = **"AUG"**;  
 **break**;  
 **case** 8:  
 value = **"SEP"**;  
 **break**;  
 **case** 9:  
 value = **"OCT"**;  
 **break**;  
 **case** 10:  
 value = **"NOV"**;  
 **break**;  
 **case** 11:  
 value = **"DEC"**;  
 **break**;  
 }  
 Log.*e*(**"VALUE"**,value);  
 **xAxis**.add(value);  
 }  
 }  
  
 }  
  
}

**no** = **new** ArrayList<>();  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
 **no**.add(0);  
  
 getXAxisValues();  
  
 BarData data = **new** BarData(**xAxis**, getDataSet());  
 chart.setData(data);  
 chart.setDescription(**""**);  
 chart.animateXY(2000, 2000);  
 chart.invalidate();  
 }  
  
 **private** ArrayList<IBarDataSet> getDataSet() {  
 ArrayList<IBarDataSet> dataSets = **null**;  
  
 ArrayList<BarEntry> valueSet1 = **new** ArrayList<>();  
 ArrayList<BarEntry> valueSet2 = **new** ArrayList<>();  
 **int** num = 0;  
 **for** (**int** i=0;i<**yes**.size();i++){  
 Log.*e*(i+**":"**, **yes**.get(i)+**" , "**+**no**.get(i));  
 **if**(**yes**.get(i)>0 || **no**.get(i)>0) {  
 *//Log.e("Yes", yes.get(i) + "");* BarEntry v1e1 = **new** BarEntry(**yes**.get(i), num); *// Jan* valueSet1.add(v1e1);  
  
 *//Log.e("No", no.get(i) + "");* BarEntry v2e1 = **new** BarEntry(**no**.get(i), num); *// Jan* valueSet2.add(v2e1);  
 num++;  
 }  
 }  
  
 BarDataSet barDataSet1 = **new** BarDataSet(valueSet1, **"เงินเดือน"**);  
 barDataSet1.setColor(Color.*rgb*(255, 0, 127));  
 BarDataSet barDataSet2 = **new** BarDataSet(valueSet2, **"รายได้เสริมอื่นๆ"**);  
 barDataSet2.setColor(Color.*rgb*(51, 255, 255));  
  
 dataSets = **new** ArrayList<>();  
 dataSets.add(barDataSet1);  
 dataSets.add(barDataSet2);  
 **return** dataSets;  
 }  
  
 **private void** getXAxisValues() {  
 **xAxis** = **new** ArrayList<>();  
 **for** (**int** i=0;i<**income**.size();i++){  
  
 DateFormat df = **new** SimpleDateFormat(**"dd-MM-yyyy"**);  
 Date startDate = **null**;  
 **try** {  
 startDate = df.parse(**income**.get(i).**title**);  
 String newDateString = df.format(startDate);  
 System.***out***.println(newDateString);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
  
 startDate.getMonth();  
 **month**.set(startDate.getMonth(), **month**.get(startDate.getMonth()) + 1);  
 **if**(**income**.get(i).**v8**){  
 **no**.set(startDate.getMonth(),**no**.get(startDate.getMonth())+1);  
 }**else**{  
 **yes**.set(startDate.getMonth(), **yes**.get(startDate.getMonth()) + 1);  
 }  
 }  
  
 **for**(**int** i=0;i<12;i++){  
 String value = **""**;  
 **if**(**month**.get(i)>0){  
 **switch** (i){  
 **case** 0:  
 value = **"JAN"**;  
 **break**;  
 **case** 1:  
 value = **"FAB"**;  
 **break**;  
 **case** 2:  
 value = **"MAR"**;  
 **break**;  
 **case** 3:  
 value = **"APR"**;  
 **break**;  
 **case** 4:  
 value = **"MAY"**;  
 **break**;  
 **case** 5:  
 value = **"JUN"**;  
 **break**;  
 **case** 6:  
 value = **"JUL"**;  
 **break**;  
 **case** 7:  
 value = **"AUG"**;  
 **break**;  
 **case** 8:  
 value = **"SEP"**;  
 **break**;  
 **case** 9:  
 value = **"OCT"**;  
 **break**;  
 **case** 10:  
 value = **"NOV"**;  
 **break**;  
 **case** 11:  
 value = **"DEC"**;  
 **break**;  
 }  
 Log.*e*(**"VALUE"**,value);  
 **xAxis**.add(value);  
 }  
 }  
  
 }  
  
}

String newDateString = df.format(startDate);  
 System.***out***.println(newDateString);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
  
 startDate.getMonth();  
 **month**.set(startDate.getMonth(), **month**.get(startDate.getMonth()) + 1);  
 **if**(**income**.get(i).**v8**){  
 **no**.set(startDate.getMonth(),**no**.get(startDate.getMonth())+1);  
 }**else**{  
 **yes**.set(startDate.getMonth(), **yes**.get(startDate.getMonth()) + 1);  
 }  
 }  
  
 **for**(**int** i=0;i<12;i++){  
 String value = **""**;  
 **if**(**month**.get(i)>0){  
 **switch** (i){  
 **case** 0:  
 value = **"JAN"**;  
 **break**;  
 **case** 1:  
 value = **"FAB"**;  
 **break**;  
 **case** 2:  
 value = **"MAR"**;  
 **break**;  
 **case** 3:  
 value = **"APR"**;  
 **break**;  
 **case** 4:  
 value = **"MAY"**;  
 **break**;  
 **case** 5:  
 value = **"JUN"**;  
 **break**;  
 **case** 6:  
 value = **"JUL"**;  
 **break**;  
 **case** 7:  
 value = **"AUG"**;  
 **break**;  
 **case** 8:  
 value = **"SEP"**;  
 **break**;  
 **case** 9:  
 value = **"OCT"**;  
 **break**;  
 **case** 10:  
 value = **"NOV"**;  
 **break**;  
 **case** 11:  
 value = **"DEC"**;  
 **break**;  
 }  
 Log.*e*(**"VALUE"**,value);  
 **xAxis**.add(value);  
 }  
 }  
  
 }  
  
}

activity\_bar\_chart.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="#FFCCFF"  
 tools:context=".BarChartActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="10dp"  
 android:layout\_marginTop= "5dp"  
 android:layout\_below="@id/toolbar\_main"**>  
  
 <**com.github.mikephil.charting.charts.BarChart  
 android:id="@+id/chart"  
 android:layout\_width="match\_parent"  
 android:layout\_height="300dp"** />

</**RelativeLayout**>

</**RelativeLayout**>

LineChartActivity.java

**package** org.usablelabs.duedo;  
  
**import** android.os.Bundle;  
**import** android.util.Log;  
  
**import** com.github.mikephil.charting.charts.LineChart;  
**import** com.github.mikephil.charting.data.Entry;  
**import** com.github.mikephil.charting.data.LineData;  
**import** com.github.mikephil.charting.data.LineDataSet;  
**import** com.github.mikephil.charting.utils.ColorTemplate;  
  
**import** java.text.DateFormat;  
**import** java.text.ParseException;  
**import** java.text.SimpleDateFormat;  
**import** java.util.ArrayList;  
**import** java.util.Date;  
  
**public class** LineChartActivity **extends** BaseActivity {  
  
 **private** ArrayList<Expense> **expense**;  
 **private** ArrayList<Integer> **month**;  
 ArrayList<String> **xAxis**;  
 ArrayList<Integer> **yes**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_line\_chart***);  
 setTitle(**"กราฟแสดงจำนวนการบันทึกค่าใช้จ่ายแต่ละเดือน"**);  
 setDrawer(**true**);  
 *//BarChart chart = (BarChart) findViewById(R.id.chart);* **expense** = **new** ArrayList<Expense>(Expense.*getAll*());  
 **month** = **new** ArrayList<>();  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
 **month**.add(0);  
  
 **yes** = **new** ArrayList<>();  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
 **yes**.add(0);  
  
 getXAxisValues();  
  
 LineChart lineChart = (LineChart) findViewById(R.id.***chart***);  
 *// creating list of entry* LineDataSet dataset = **new** LineDataSet(getDataSet(), **"จำนวนการบันทึก"**);  
 dataset.setColor(ColorTemplate.*rgb*(**"#99004C"**));  
 dataset.setCircleColor(ColorTemplate.*rgb*(**"#FF0000"**));  
 LineData data = **new** LineData(**xAxis**, dataset);  
  
 lineChart.setData(data);  
 lineChart.setDescription(**""**);  
  
  
 */\*chart.setDescription("");  
 chart.animateXY(2000, 2000);  
 chart.invalidate();\*/* }  
  
 **private** ArrayList<Entry> getDataSet() {  
 ArrayList<Entry> dataSets = **new** ArrayList<>();  
  
 **int** num = 0;  
 **for** (**int** i=0;i<**yes**.size();i++){  
 **if**(**yes**.get(i)>0) {  
 Log.*e*(**"Yes : "**,**yes**.get(i)+**""**);  
 dataSets.add(**new** Entry(**yes**.get(i), num));  
 num++;  
 }  
 }  
  
 **return** dataSets;  
 }  
  
 **private void** getXAxisValues() {  
 **xAxis** = **new** ArrayList<>();  
 **for** (**int** i=0;i<**expense**.size();i++){  
  
 DateFormat df = **new** SimpleDateFormat(**"dd-MM-yyyy"**);  
 Date startDate = **null**;  
 **try** {  
 startDate = df.parse(**expense**.get(i).**date**);  
 String newDateString = df.format(startDate);  
 System.***out***.println(newDateString);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
  
 startDate.getMonth();  
 **month**.set(startDate.getMonth(), **month**.get(startDate.getMonth()) + 1);  
  
 **yes**.set(startDate.getMonth(), **yes**.get(startDate.getMonth()) + 1);  
  
 }  
  
 **for**(**int** i=0;i<12;i++){  
 String value = **""**;  
 **if**(**month**.get(i)>0){  
 Log.*e*(**"Month"**,i+**""**);  
 **switch** (i){  
 **case** 0:  
 value = **"JAN"**;  
 **break**;  
 **case** 1:  
 value = **"FAB"**;  
 **break**;  
 **case** 2:  
 value = **"MAR"**;  
 **break**;  
 **case** 3:  
 value = **"APR"**;  
 **break**;  
 **case** 4:  
 value = **"MAY"**;  
 **break**;  
 **case** 5:  
 value = **"JUN"**;  
 **break**;  
 **case** 6:  
 value = **"JUL"**;  
 **break**;  
 **case** 7:  
 value = **"AUG"**;  
 **break**;  
 **case** 8:  
 value = **"SEP"**;  
 **break**;  
 **case** 9:  
 value = **"OCT"**;  
 **break**;  
 **case** 10:  
 value = **"NOV"**;  
 **break**;  
 **case** 11:  
 value = **"DEC"**;  
 **break**;  
 }  
 Log.*e*(**"VALUE"**,value);  
 **xAxis**.add(value);  
 }  
 }  
  
 }  
  
}

getXAxisValues();  
  
 LineChart lineChart = (LineChart) findViewById(R.id.***chart***);  
 *// creating list of entry* LineDataSet dataset = **new** LineDataSet(getDataSet(), **"จำนวนการบันทึก"**);  
 dataset.setColor(ColorTemplate.*rgb*(**"#99004C"**));  
 dataset.setCircleColor(ColorTemplate.*rgb*(**"#FF0000"**));  
 LineData data = **new** LineData(**xAxis**, dataset);  
  
 lineChart.setData(data);  
 lineChart.setDescription(**""**);  
  
  
 */\*chart.setDescription("");  
 chart.animateXY(2000, 2000);  
 chart.invalidate();\*/* }  
  
 **private** ArrayList<Entry> getDataSet() {  
 ArrayList<Entry> dataSets = **new** ArrayList<>();  
  
 **int** num = 0;  
 **for** (**int** i=0;i<**yes**.size();i++){  
 **if**(**yes**.get(i)>0) {  
 Log.*e*(**"Yes : "**,**yes**.get(i)+**""**);  
 dataSets.add(**new** Entry(**yes**.get(i), num));  
 num++;  
 }  
 }  
  
 **return** dataSets;  
 }  
  
 **private void** getXAxisValues() {  
 **xAxis** = **new** ArrayList<>();  
 **for** (**int** i=0;i<**expense**.size();i++){  
  
 DateFormat df = **new** SimpleDateFormat(**"dd-MM-yyyy"**);  
 Date startDate = **null**;  
 **try** {  
 startDate = df.parse(**expense**.get(i).**date**);  
 String newDateString = df.format(startDate);  
 System.***out***.println(newDateString);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
  
 startDate.getMonth();  
 **month**.set(startDate.getMonth(), **month**.get(startDate.getMonth()) + 1);  
  
 **yes**.set(startDate.getMonth(), **yes**.get(startDate.getMonth()) + 1);  
  
 }  
  
 **for**(**int** i=0;i<12;i++){  
 String value = **""**;  
 **if**(**month**.get(i)>0){  
 Log.*e*(**"Month"**,i+**""**);  
 **switch** (i){

**case** 0:  
 value = **"JAN"**;  
 **break**;  
 **case** 1:  
 value = **"FAB"**;  
 **break**;  
 **case** 2:  
 value = **"MAR"**;  
 **break**;  
 **case** 3:  
 value = **"APR"**;  
 **break**;  
 **case** 4:  
 value = **"MAY"**;  
 **break**;  
 **case** 5:  
 value = **"JUN"**;  
 **break**;  
 **case** 6:  
 value = **"JUL"**;  
 **break**;  
 **case** 7:  
 value = **"AUG"**;  
 **break**;  
 **case** 8:  
 value = **"SEP"**;  
 **break**;  
 **case** 9:  
 value = **"OCT"**;  
 **break**;  
 **case** 10:  
 value = **"NOV"**;  
 **break**;  
 **case** 11:  
 value = **"DEC"**;  
 **break**;  
 }  
 Log.*e*(**"VALUE"**,value);  
 **xAxis**.add(value);  
 }  
 }  
  
 }  
  
}

**case** 0:  
 value = **"JAN"**;  
 **break**;  
 **case** 1:  
 value = **"FAB"**;  
 **break**;  
 **case** 2:  
 value = **"MAR"**;  
 **break**;  
 **case** 3:  
 value = **"APR"**;  
 **break**;  
 **case** 4:  
 value = **"MAY"**;  
 **break**;  
 **case** 5:  
 value = **"JUN"**;  
 **break**;  
 **case** 6:  
 value = **"JUL"**;  
 **break**;  
 **case** 7:  
 value = **"AUG"**;  
 **break**;  
 **case** 8:  
 value = **"SEP"**;  
 **break**;  
 **case** 9:  
 value = **"OCT"**;  
 **break**;  
 **case** 10:  
 value = **"NOV"**;  
 **break**;  
 **case** 11:  
 value = **"DEC"**;  
 **break**;  
 }  
 Log.*e*(**"VALUE"**,value);  
 **xAxis**.add(value);  
 }  
 }  
  
 }  
  
}

activity\_line\_chart.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="#66FFFF"  
 tools:context=".LineChartActivity"**>  
  
 <**include android:id="@+id/toolbar\_main" layout="@layout/toolbar\_main"** />  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="10dp"  
 android:layout\_marginTop= "5dp"  
 android:layout\_below="@id/toolbar\_main"**>  
  
 <**com.github.mikephil.charting.charts.LineChart  
 android:id="@+id/chart"  
 android:layout\_width="match\_parent"  
 android:layout\_height="300dp"** />  
 </**RelativeLayout**>  
</**RelativeLayout**>