



แอปพลิเคชันเกมเวทีทอง
บริษัท สนุกออนไลน์ จำกัด
GAME WAETEETHONG APPLICATION
SANOOK ONLINE LIMITED

นายชุตีภัทร์ โชคพิพัฒน์พร

โครงงานสหกิจนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีมัลติมีเดีย
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย – ญี่ปุ่น

พ.ศ. 2556

แอปพลิเคชันเกมเวทีทอง
บริษัท สนุก ออนไลน์ จำกัด
GAME WAETEETHONG APPLICATION
SANOOK ONLINE LIMITED

นายชุตติภัทร์ โชคพิพัฒน์พร

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาเทคโนโลยีมัลติมีเดีย
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
พ.ศ. 2556

คณะกรรมการสอบ

.....ประธานกรรมการสอบ

(อาจารย์ ดร. ภาสกร อภิรักษ์วรพินิต)

.....กรรมการสอบ

(อาจารย์ ภัทมะ เจริญพงษ์)

.....อาจารย์ที่ปรึกษา

(อาจารย์ ชาญ จารุงศรีรังสี)

.....ประธานสหกิจศึกษาสาขาวิชา

(อาจารย์ ภูวดล ศิริกองธรรม)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น

ชื่อโครงการ	แอปพลิเคชันเกมเวทีทอง
ผู้เขียน	นาย ชุตติภัทร์ โชคพิพัฒน์พร
คณะวิชา	เทคโนโลยีสารสนเทศ สาขาวิชาเทคโนโลยีมัลติมีเดีย
อาจารย์ที่ปรึกษา	อาจารย์ชาญ จารวงค์รังสี
พนักงานที่ปรึกษา	นาย จิรวัดน์ กรณียวิทยาการ
ชื่อบริษัท	บริษัท สนุก ออนไลน์ จำกัด
ประเภทธุรกิจ / สินค้า	ธุรกิจการให้บริการบนอินเทอร์เน็ต

บทสรุป

ได้พัฒนาแอปพลิเคชันเกมเวทีทองในระบบปฏิบัติการAndroid โดยได้ทำการศึกษาแอปพลิเคชันประเภทเกมทายคำจากภาพที่มีอยู่แล้วใน Play Store และนำมาวิเคราะห์ข้อดีข้อเสีย รวมถึงสอบถามความต้องการของผู้ใช้ใน Features ต่างๆที่ต้องการให้มีในแอปพลิเคชันประเภทเกมทายคำจากภาพ เพื่อนำมาพัฒนาแอปพลิเคชันเกมเวทีทองให้ตรงตามความต้องการ เมื่อพัฒนาแอปพลิเคชันเสร็จเรียบร้อยแล้ว ได้นำแอปพลิเคชันไปทดสอบ ซึ่งแอปพลิเคชันสามารถทำงานและแสดงผลได้ถูกต้องตรงตามที่ต้องการ

ในการสหกิจศึกษาคครั้งนี้ได้เรียนรู้ถึงการทำงานจริงของ Developer ในบริษัทขนาดกลาง, ได้เรียนรู้เทคนิคใหม่ๆในการเขียนโปรแกรม, ได้รู้ถึงงานโปรแกรมเมอร์ที่บริษัทส่วนใหญ่ต้องการ, ได้พัฒนาทักษะของตนเองทั้งในด้านการเขียนโปรแกรม (Android, SQL, PHP, CSS) และการออกแบบ (เว็บไซต์และแอปพลิเคชัน) อีกทั้งได้รับรู้ถึงสถานการณ์และสภาพชีวิตในการดำเนินชีวิตที่ทำงานจริงและการทำงานร่วมกับผู้อื่น ซึ่งถือว่าเป็นประสบการณ์ที่สำคัญในการพัฒนาตนเองต่อไปในอนาคต

Project's name	Game Waeteethong Application
Writer	Mr. Chutipat Chokpipatporn
Faculty	Faculty of Information Technology, Multimedia Technology Program
Faculty Advisor	Mr. Chan Jaruvongrangsee
Job Supervisor	Mr. Jirawat Karanwittayakarn
Company's name	Sanook! Online Limited
Business Type	Provided services on the internet

Summary

This project is to develop a game application called Game Waeteethong that available in mobile platform, Android OS, by studying the word games that have published in Play Store and analysis those applications about their advantages and disadvantages. Then collecting requirement from users about features that they want in word game application in order to developing Game Waeteethong application to meet the requirements, before developing game application. When finished, testing the application to find errors and fix them, which can display result correctly and all features can response efficiently.

After going to cooperative education at Sanook! Online Limited, I had an experience about “how developer’s working in large company”, learning a lot of skills in Android and PHP programming, learning about skills of programmer that most companies required, practicing my programming skills such as PHP, SQL, Android, CSS ,and practicing design skills in Website and Android application. Moreover, I learned about real situation and real life during working time and in working with another people, which is the most important experience for developing myself in the future.

กิตติกรรมประกาศ

ในการปฏิบัติงานสหกิจศึกษาที่บริษัท สนุก ออนไลน์ จำกัด เป็นระยะเวลาทั้งสิ้น 4 เดือนนั้น ได้รับความกรุณาและการดูแลจากบุคคลหลายๆท่าน, ได้รับเรียนรู้การทำงานจริง, และได้รับความรู้ต่างมากมาย ทำให้การปฏิบัติงานครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณ คุณจิรวัดน์ กรณ์ชัยวิทยาการ พนักงานที่ปรึกษา ซึ่งให้การดูแลและให้คำปรึกษาต่างๆในการทำงาน รวมถึงตอบข้อสงสัยต่างๆอย่างกระจ่าง พร้อมทั้งแนะนำเทคนิคในการทำงานซึ่งสามารถนำไปใช้ได้จริง รวมไปถึงพนักงานท่านอื่นๆ ที่ให้การดูแลและตอบข้อสงสัย รวมถึงเป็นที่ปรึกษาและคอยให้ข้อมูลในด้านต่างๆในการทำงาน ทำให้การสหกิจศึกษาตลอด 4 เดือนที่ผ่านมาเป็นความทรงจำที่น่าประทับใจ ขอขอบพระคุณทุกท่านไว้ ณ โอกาสนี้ด้วย

TNI

สารบัญ

หน้า

บทสรุป	ก
Summary	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญภาพประกอบ	ซ
บทที่	
1. บทนำ	1
1.1 ชื่อและที่ตั้งของสถานประกอบการ	1
1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร	2
1.3 รูปแบบการจัดการองค์กรและการบริหารองค์กร	4
1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย	5
1.5 พนักงานที่ปรึกษาและตำแหน่งของพนักงานที่ปรึกษา	5
1.6 ระยะเวลาที่ปฏิบัติงาน	5
1.7 ที่มาและความสำคัญของปัญหา	5
1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงการ	6
1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย	6
1.10 นิยามศัพท์เฉพาะ	7
2. ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน	8
2.1 ทฤษฎีที่ใช้ในการปฏิบัติงาน	8
2.1.1 โครงสร้างของแอปพลิเคชันบนระบบปฏิบัติการAndroid	8
2.1.2 ส่วนประกอบของแอปพลิเคชันบนระบบปฏิบัติการAndroid	9
2.1.3 วงจรชีวิตของ Android Activity	10
2.2 เทคโนโลยีที่ใช้ในการปฏิบัติงาน	12
2.2.1 Java	12

2.2.2 Javascript Object Notation : JSON	13
2.2.3 MySQL	14
2.2.4 Google Cloud Messaging : GCM	15
2.2.5 phpMyAdmin	16
2.2.6 Adobe DreamWeaver	17
2.2.7 File Transfer Protocol : FTP	19
2.2.8 GenyMotion	23
3. แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน	24
3.1 แผนการปฏิบัติงาน	24
3.2 รายละเอียดโครงการ	25
3.2.1 ประชากรและกลุ่มตัวอย่าง	25
3.2.2 การสร้างเครื่องมือที่ใช้ในโรงงาน	25
3.2.2.1 วิธีการติดตั้งโปรแกรม Adobe Master Collection CS6	25
3.2.2.2 วิธีการติดตั้งโปรแกรม AppServ	27
3.2.2.3 วิธีการติดตั้ง Java Development Kit : JDK	31
3.2.2.4 วิธีการติดตั้งโปรแกรม Eclipse	33
3.2.2.5 วิธีการติดตั้งโปรแกรม GenyMotion	34
3.2.3 การเก็บรวบรวมข้อมูล	37
3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงการ	38
3.3.1 งานที่ปฏิบัติ	38
3.3.1.1 Sanook Weather Application	38
3.3.1.2 Test SQLite Application	41
3.3.1.3 jsTwit Application	44
3.3.1.4 genPassword Application	47
3.3.2 โครงการแอปพลิเคชันเกมเวทีทอง	49
4. สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่างๆ	52
4.1 ขั้นตอนและผลการดำเนินงาน	52
4.1.1 ศึกษาและทำความเข้าใจเกี่ยวกับรูปแบบของช่วงปริศนาจ๊กมก	52
ในรายการเวทีทอง	

4.1.2 ศึกษาแอปพลิเคชันเกมทายคำจากภาพที่มีอยู่แล้วใน Play Store	52
4.1.3 วิเคราะห์ความต้องการของผู้ใช้งาน	54
4.1.4 ออกแบบ Mock Up ของแอปพลิเคชัน	55
4.1.5 สร้างเว็บหลังบ้านของแอปพลิเคชันเกมเวทีทอง	55
4.1.6 สร้างเดโมของแอปพลิเคชันเกมเวทีทอง	59
4.1.7 ทดสอบการทำงานของเดโมแอปพลิเคชัน	61
4.1.8 นำFeedbackและข้อเสนอแนะที่ได้จากการนำเสนอเดโมแอปพลิเคชัน ไปปรับปรุง	61
4.1.9 พัฒนาเดโมแอปพลิเคชันให้เป็นแอปพลิเคชันตัวเต็ม	61
4.1.10 ทดสอบการทำงานของแอปพลิเคชันเพื่อหาข้อผิดพลาดและนำไปปรับปรุง	61
4.2 ผลการวิเคราะห์	62
4.2.1 วิเคราะห์การทำงานและการตอบสนองของระบบเกม	62
4.2.2 วิเคราะห์การทำงานและการตอบสนองของ features ต่างๆในแอปพลิเคชัน	62
4.2.3 วิเคราะห์การทำงานและการตอบสนองของเว็บหลังบ้าน	62
4.3 วิจัยข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมาย การปฏิบัติงานหรือการจัดทำโครงการ	63
5. บทสรุปและข้อเสนอแนะ	68
5.2 สรุปผลการดำเนินงาน	68
5.2 แนวทางแก้ไขปัญหา	69
5.3 ข้อเสนอแนะจากการดำเนินงาน	69
เอกสารอ้างอิง	70
ภาคผนวก	73
ก. Code Web หลังบ้าน ของ Application เกมเวทีทอง	74
ข. Code Android ของ Application เกมเวทีทอง	103
ค. รายงานประจำสัปดาห์	179
ประวัติผู้จัดทำโครงการ	198

สารบัญตาราง

ตารางที่	หน้า
3.1 แผนการปฏิบัติสหกิจศึกษา	24
4.1 Data Dictionary ของฐานข้อมูล 578032	57
4.2 Data Dictionary ของฐานข้อมูล 578032db2	57



สารบัญภาพประกอบ

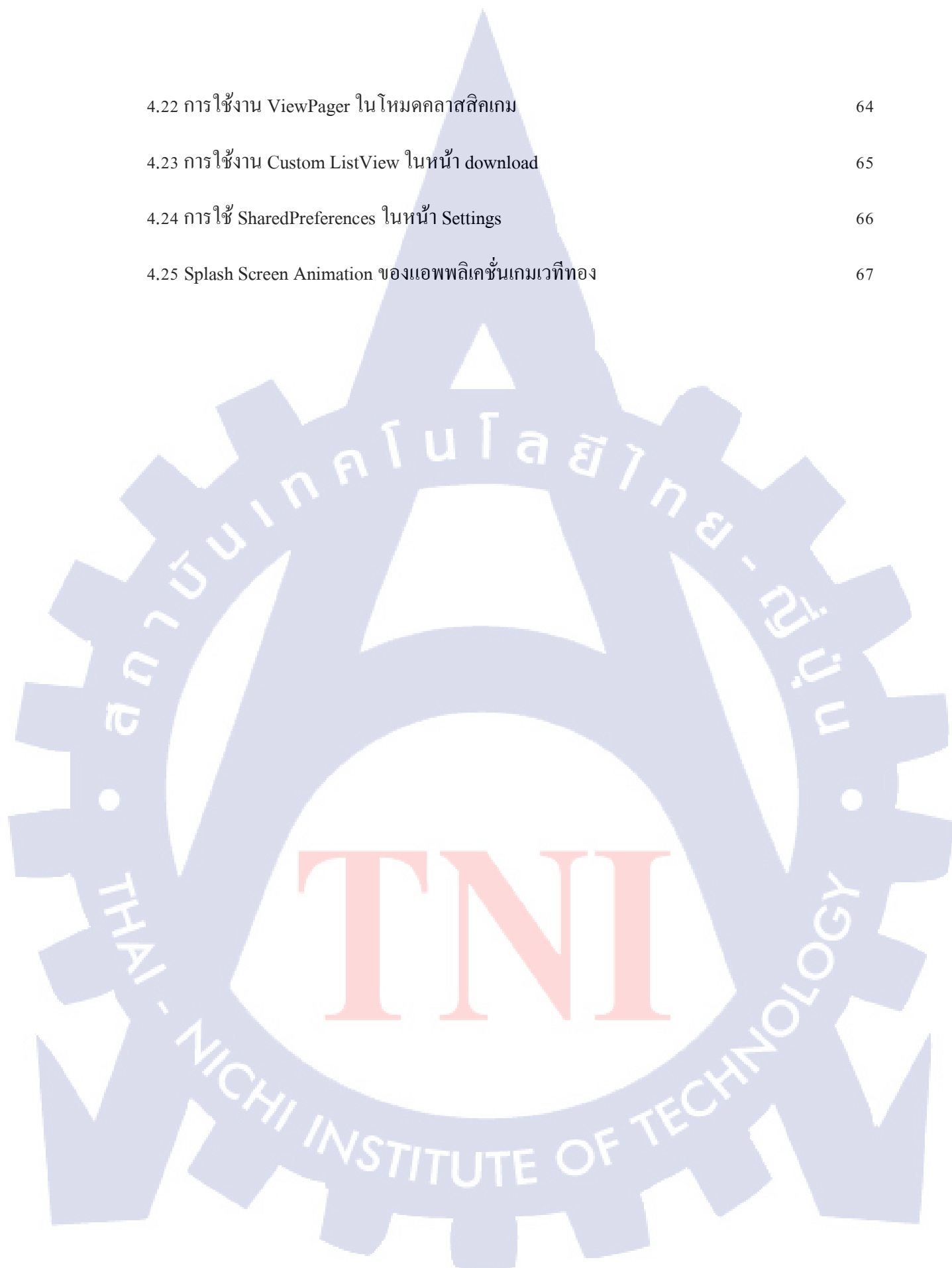
ภาพที่	หน้า
1.1 แผนที่บริษัท สนุก ออนไลน์ จำกัด	1
1.2 แผนผังองค์กร	4
2.1 โครงสร้างของแอปพลิเคชันบนระบบปฏิบัติการAndroid	9
2.2 Android Activity Life Cycle	11
2.3 การทำงานของ GCM	15
2.4 ตัวอย่างหน้าจอ phpMyAdmin	16
2.5 มุมมองแบบ Code ใน Dreamweaver	18
2.6 มุมมองแบบ Split ใน Dreamweaver	18
2.7 มุมมองแบบ Design ใน Dreamweaver	19
2.8 ลักษณะการทำงานของ FTP : File Transfer Protocol	22
2.9 Interface ของ GenyMotion	23
3.1 ขั้นตอนการลงโปรแกรม Adobe Master Collection CS6 (1)	25
3.2 ขั้นตอนการลงโปรแกรม Adobe Master Collection CS6 (2)	26
3.3 ขั้นตอนการลงโปรแกรม Adobe Master Collection CS6 (3)	26
3.4 ขั้นตอนการลงโปรแกรม Adobe Master Collection CS6 (4)	27
3.5 ขั้นตอนการติดตั้งโปรแกรม AppServ	27
3.6 แสดงรายละเอียดเงื่อนไขการ GNU License	28
3.7 เลือกปลายทางการติดตั้งโปรแกรม AppServ	28

3.8 เลือก Package Components ที่ต้องการติดตั้ง	29
3.9 แสดงการกำหนดค่าคอนฟิกค่า Apache Web Server	29
3.10 แสดงการกำหนดค่าคอนฟิกของ MySQL Database	30
3.11 แสดงหน้าจอขั้นตอนสิ้นสุดการติดตั้งโปรแกรม AppServ	30
3.12 ขั้นตอนการติดตั้ง JDK (1)	31
3.13 ขั้นตอนการติดตั้ง JDK (2)	31
3.14 ขั้นตอนการติดตั้ง JDK (3)	32
3.15 ขั้นตอนการติดตั้ง JDK (4)	32
3.16 ขั้นตอนการติดตั้งโปรแกรม Eclipse (1)	33
3.17 ขั้นตอนการติดตั้งโปรแกรม Eclipse (2)	33
3.18 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (1)	34
3.19 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (2)	34
3.20 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (3)	35
3.21 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (4)	35
3.22 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (5)	36
3.23 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (6)	36
3.24 หนังสือคู่มือเขียนแอปAndroid ฉบับรวมโค้ด	37
3.25 ไอคอนของแอปพลิเคชัน Sanook Weather	38
3.26 Widget ของแอปพลิเคชัน Sanook Weather	39
3.27 Interface ของแอปพลิเคชัน Sanook Weather ในหน้าแสดงสภาพอากาศ วันปัจจุบัน	39

3.28 Interface ของแอปพลิเคชัน Sanook Weather ในหน้าแสดงสภาพอากาศ ในสัปดาห์นี้	40
3.29 Interface ของแอปพลิเคชัน Sanook Weather ในหน้าตั้งค่าหน่วยของ อุณหภูมิที่จะให้แสดง	40
3.30 ไอคอนของแอปพลิเคชัน SQLite Test	41
3.31 SplashScreen ของแอปพลิเคชัน SQLite	41
3.32 Interface หน้าแรก (แสดงรายชื่อเพื่อน) ของแอปพลิเคชัน SQLite	42
3.33 Interface เมื่อกดไปที่รายชื่อเพื่อนในแอปพลิเคชัน SQLite	42
3.34 Interface หน้าแก้ไขข้อมูลของเพื่อนในแอปพลิเคชัน SQLite	43
3.35 Interface หน้าเพิ่มเพื่อนใหม่ในแอปพลิเคชัน SQLite	43
3.36 ไอคอนของแอปพลิเคชัน jsTwit	44
3.37 SplashScreen ของแอปพลิเคชัน jsTwit	44
3.38 Interface หน้าแรกของแอปพลิเคชัน jsTwit	45
3.39 หน้าจอ login เข้าสู่ twitter ของแอปพลิเคชัน jsTwit	45
3.40 Interface หลังจาก login	46
3.41 Interface หน้าแสดงข้อมูลข่าว จส.100	47
3.42 Interface ของโปรแกรม genPassword	48
3.43 Interface เมื่อกดปุ่ม generate	48
3.44 Interface เมื่อนำAPI Key ไปตรวจสอบ	48
3.45 รูปภาพปุ่มของโหมดคลาสสิกเกม	49
3.46 รูปภาพปุ่มของโหมดโกสต์มาสเตอร์	50

4.1 แอปพลิเคชัน SalahToonLite	52
4.2 Interface หน้าแรก	53
4.3 Interface หน้าเล่นเกม	54
4.4 MockUp ของแอปพลิเคชันเกมเวทีทอง	55
4.5 ฐานข้อมูลของตาราง game	56
4.6 ฐานข้อมูลตาราง mode	56
4.7 ฐานข้อมูลตาราง gcm_users	56
4.8 Interface หน้าแรกของเว็บหลังบ้าน	58
4.9 Interface หน้าเพิ่มด่านใหม่	58
4.10 Interface หน้าเลือก feed	58
4.11 หน้า feed	58
4.12 โลโก้แอปพลิเคชัน	59
4.13 หน้า SplashScreen	59
4.14 หน้า MainActivity	59
4.15 หน้า choosemode	59
4.16 หน้า download	59
4.17 หน้า Settings	59
4.18 หน้า HowToPlay	60
4.19 หน้า classicgame	60
4.20 หน้า ghostmaster	60
4.21 Notification ในโทรศัพท์มือถือ เมื่อมีการเพิ่มด่านใหม่	64

4.22 การใช้งาน ViewPager ในโหมดคลาสสิกเกม	64
4.23 การใช้งาน Custom ListView ในหน้า download	65
4.24 การใช้ SharedPreferences ในหน้า Settings	66
4.25 Splash Screen Animation ของแอปพลิเคชันเกมเวทีทอง	67



บทที่ 1

บทนำ

1.1 ชื่อและที่ตั้งของสถานประกอบการ

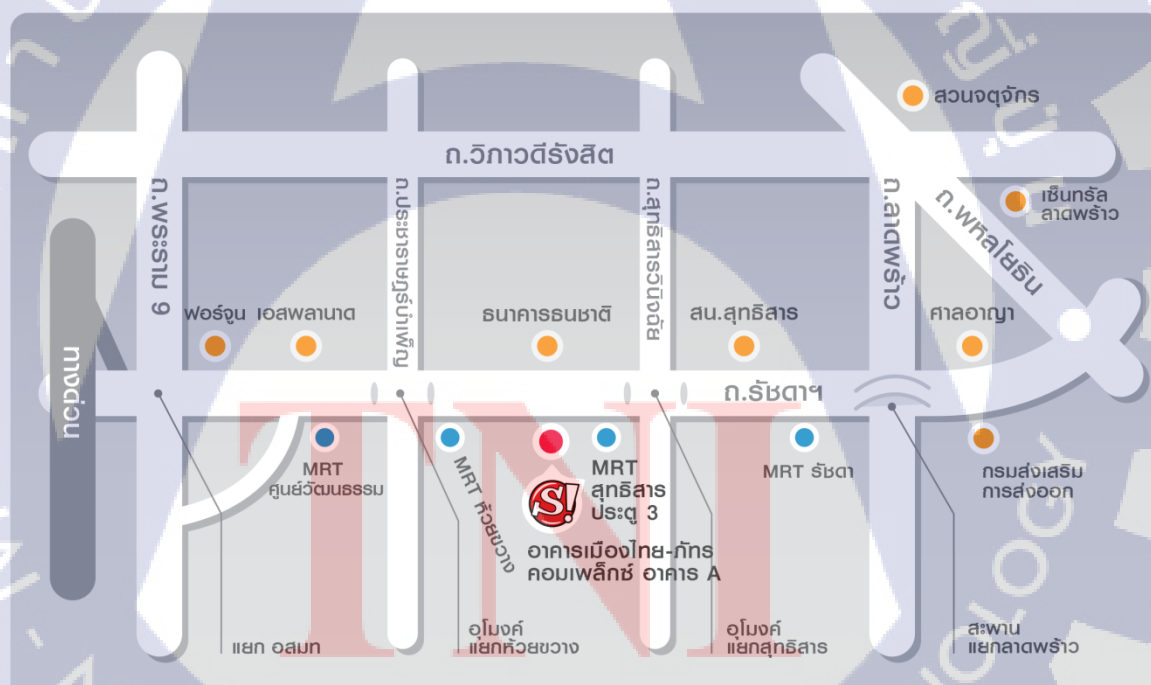
1.1.1) ชื่อของสถานประกอบการ : บริษัท สนุก ออนไลน์ จำกัด

1.1.2) ที่ตั้งของสถานประกอบการ :

อาคารเมืองไทย-ภัทร คอมเพล็กซ์ อาคาร A ชั้น 15-1 เลขที่ 252/19-20

ถนนรัชดาภิเษก แขวงห้วยขวาง เขตห้วยขวาง กรุงเทพฯ 10310 ประเทศไทย

โทรศัพท์ : +662-833-3000 โทรสาร : +662-833-3113



ภาพที่ 1.1 แผนที่บริษัท สนุก ออนไลน์ จำกัด

1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร

บริษัท สนุก ออนไลน์ จำกัด เป็นบริษัทที่ให้บริการบนอินเทอร์เน็ตแบบครบวงจรที่พร้อมให้บริการสำหรับบุคคลทั่วไปและในส่วนของภาคธุรกิจ เราครบครันด้วยข้อมูล ข่าวสาร ความบันเทิง รวมทั้งบริการผ่านสื่ออินเทอร์เน็ตอีกมากมายที่พร้อมจะตอบสนองผู้ใช้งานออนไลน์ได้ทุกที่ทุกเวลา

ปัจจุบันบริษัท สนุก ออนไลน์ จำกัด มีการให้บริการหลักๆ แบ่งออกเป็น 5 ประเภท ดังนี้

1) สนุก! พอร์ทัล / เว็บไซต์สนุกดอทคอม

เว็บไซต์อันดับหนึ่งของประเทศ อัดแน่นไปด้วยข้อมูลข่าวสาร, ความบันเทิงนำเสนอด้วยความรวดเร็วทันเหตุการณ์ อัปเดตตลอด 24 ชั่วโมง, เกาะติดกระแสฮิตในสังคม, ฟังเพลงใหม่, ข้อมูลภาพยนตร์, คลิปวิดีโอ, แชน, ดูดวง, เว็บบอร์ด, ความรัก, การเงิน, ไอที, กีฬา, และรถยนต์ ครอบคลุมทุกความสนใจทุกไลฟ์สไตล์ ไม่ตกกระแสบนโลกออนไลน์ที่ไม่เคยหยุดนิ่ง

2) สนุก! เกมส์

2.1) สนุก! เพลย์ทาวน์

โซเชี่ยลเน็ตเวิร์กใหม่! สำหรับคนรักเกมส์แห่งแรกในประเทศไทยอัดแน่นด้วยโซเชี่ยลเกมส์คุณภาพ หลากหลายแนว เล่นสนุกเข้าใจง่ายไม่ต้องแปล ทุกเกมส์เป็นภาษาไทย เป็นชุมชนยุคใหม่ของหนุ่มสาวไซเบอร์ที่มีไลฟ์สไตล์คล้ายกัน แชร์โพสต์ แชร์เรื่องราวแลกเปลี่ยนเทคนิคเกมส์ กับเพื่อนใหม่ๆ มิตรภาพดีๆ เริ่มต้นที่นี่

2.2) สนุก! เกมส์

เป็นผู้ให้บริการเกมส์ที่ใหญ่ที่สุดในประเทศไทย มีเกมส์เพลซให้เลือกเล่นมากกว่า 10,000 เกมส์ พร้อมทั้งให้บริการข้อมูลข่าวสารเกมส์, รีวิวเกมส์, รีวิวเกมส์, เทคนิคเกมส์ของเกมส์ออนไลน์ เกมส์พีซี เกมส์คอนโซล, คลิปวิดีโอเกมส์ใหม่, ภาพเกมส์, ภาพพรีดีจันเกมส์, และ คอสเพลย์ ซึ่งอัปเดตทุกวัน

3) สนุก! อีคอมเมิร์ซ

3.1) Sabuy.com

แหล่งรวมสินค้าคุณภาพ และสินค้าที่หลากหลายประกอบไปด้วยร้านค้ามากกว่า 1,500 ร้านค้า และสินค้ามากกว่า 200,000 ชิ้น ทั้งสินค้าแฟชั่น แบรินด์เนม Gadget และสินค้าทั่วไป พร้อมเปิดบริการใหม่เพื่อสร้างความมั่นใจให้ผู้ซื้อสินค้ามากยิ่งขึ้นด้วย Sabuy Guarantee กรอบคลุมไปด้วยการรับประกันความพึงพอใจ, การรับประกันได้สินค้าในเวลาที่กำหนดหรือหากสินค้าชำรุดหรือไม่ตรงตามคำบรรยาย สามารถเปลี่ยน/คืนสินค้าได้

3.2) Dealfish.co.th

เว็บไซต์ที่จะช่วยให้คุณประกาศซื้อ-ขาย หรือค้นหาสินค้าที่อยู่ในพื้นที่เดียวกับที่คุณต้องการ ได้อย่างง่ายดายและรวดเร็วที่สุด

3.3) สนุก! คุปอง

นำเสนอคุปองส่วนลดพิเศษสุดทั้งร้านอาหาร, ที่พัก, สถานบันเทิง, แหล่งช้อปปิ้ง และอีกมากมาย คุณจะได้อัปเดตข้อเสนอที่ดีที่สุดโดยไม่ต้องเสียเวลาค้นหาด้วยตัวเอง

4) สนุก! โมบาย

4.1) สนุก! โมบายเมจิก

เป็นการให้บริการเสริมผ่านโทรศัพท์มือถือ เช่น บริการดาวน์โหลด, บริการข้อมูลข่าวสารผ่านSMS, บริการ SMS Voting, บริการAudio Text 1900 เป็นต้น

4.2) สนุก! บนมือถือ

เป็นบริการย่อโลกข่าวสารสาระบันเทิงและช่วยให้เข้าถึงบริการจากสนุก! ที่คุณประทับใจ

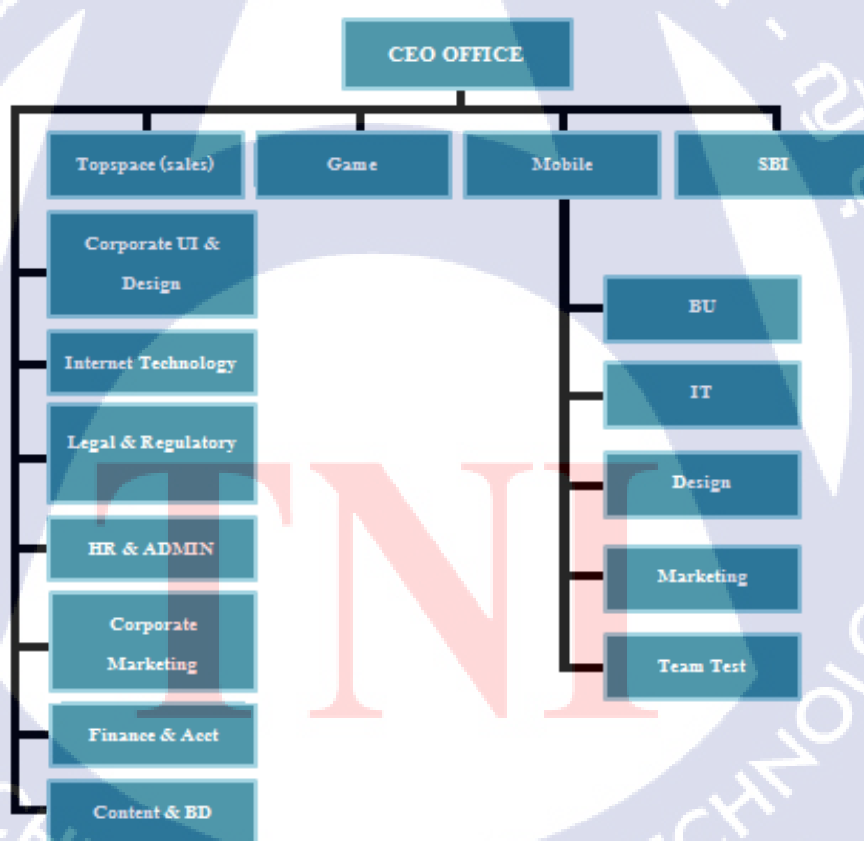
4.3) สนุก! เบราว์เซอร์

พัฒนามาเพื่อใช้ท่องโลกอินเทอร์เน็ตบนโทรศัพท์มือถือ

5) บริการโฆษณาออนไลน์

ให้บริการด้านการสื่อสารทางการตลาดออนไลน์แบบครบวงจรและให้บริการโฆษณาออนไลน์ที่สามารถตอบสนองต่อวัตถุประสงค์และความต้องการของลูกค้าทุกรายได้อย่างแม่นยำและครบวงจร

1.3 รูปแบบการจัดการองค์กรและการบริหารองค์กร



ภาพที่ 1.2 แผนผังองค์กร

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่งงาน : นักศึกษาฝึกงานด้าน Android Application

หน้าที่ : พัฒนาแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการ Android ทั้งการเขียนโปรแกรมตามที่ได้รับมอบหมายจากพี่เลี้ยงและการออกแบบ Interface ของโปรแกรมให้ครอบคลุมกับโทรศัพท์มือถือระบบปฏิบัติการ Android ซึ่งมีขนาดของหน้าจอที่แตกต่างกันไป

1.5 พนักงานที่ปรึกษา และตำแหน่งของพนักงานที่ปรึกษา

ชื่อ-นามสกุล : นาย จิรวัดน์ กรณ์ยวีทยาการ

ตำแหน่ง : ผู้ช่วยผู้จัดการด้านไอที

1.6 ระยะเวลาที่ปฏิบัติงาน

เริ่มต้นการปฏิบัติงาน : 3 มิถุนายน พ.ศ. 2556

สิ้นสุดการปฏิบัติงาน : 4 ตุลาคม พ.ศ. 2556

เป็นระยะเวลาทั้งสิ้น : 18 สัปดาห์ (4 เดือน)

1.7 ที่มาและความสำคัญของปัญหา

ในอดีตมีรายการเกมโชว์ยอดนิยมรายการหนึ่งชื่อว่า “เวทีทอง” ภายในรายการเวทีทองประกอบไปด้วยช่วงในรายการหลายช่วง โดยช่วงหนึ่งในรายการที่ถูกพูดถึงมากที่สุดคือช่วงปริศนาจ๊กกนั้นเอง ในช่วงปริศนาจ๊กกนี้ ผู้เข้าแข่งขันจะต้องทายคำศัพท์จากภาพให้ถูกต้องภายในระยะเวลาที่กำหนด แต่ละภาพที่นำมาใช้ได้สร้างความสนุกสนานให้กับผู้ชมและผู้เข้าแข่งขันเป็นอย่างมาก ทำให้รายการเวทีทองได้ออกอากาศในรายการโทรทัศน์เป็นเวลายาวนานจนกระทั่งในปีพ.ศ.2550 จึงถูกถอดออกไปจากผังรายการโทรทัศน์ รวมระยะเวลาที่ได้ออกอากาศทั้งสิ้นเป็นเวลากว่า 18 ปี

จากเหตุผลข้างต้น ผู้จัดทำจึงมีความต้องการที่จะนำรายการเวทีทองในช่วงปริศนาจ๊กกกลับมาอีกครั้งในรูปแบบของแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการ Android เพื่อคลายความคิดถึงและสร้างความเพลิดเพลินให้กับผู้ใช้

1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงการ

- 1) เพื่อศึกษาการใช้งาน Push Notification โดยใช้ GCM
- 2) เพื่อศึกษาการใช้งาน ViewPager ในAndroid
- 3) เพื่อศึกษาการ get json data จาก api มาเก็บไว้ใน Database ของโทรศัพท์มือถือ และนำข้อมูลที่เก็บอยู่ใน Database มาแสดงในหน้าจอโทรศัพท์มือถือ
- 4) เพื่อศึกษาการทำงานของ ArrayList ในAndroid
- 5) เพื่อศึกษาการดาวน์โหลดรูปภาพจาก Internet ผ่านURLของรูปภาพนั้น มาเก็บไว้ในหน่วยความจำของโทรศัพท์มือถือ
- 6) เพื่อศึกษาการใช้งาน Custom ListView ในAndroid
- 7) เพื่อศึกษาการใช้งาน SharedPreferences ในAndroid
- 8) เพื่อศึกษาการประยุกต์ใช้ Library ของโปรแกรมที่มีนักพัฒนาแอปพลิเคชันได้เขียนเอาไว้แล้ว มาใช้กับแอปพลิเคชันของเรา
- 9) เพื่อศึกษาการทำ Animation แบบง่ายๆให้กับ SplashScreen ของแอปพลิเคชัน
- 10) เพื่อฝึกการออกแบบ User Interface และ Layout ในหน้าต่างๆของแอปพลิเคชันให้รองรับกับโทรศัพท์มือถือและแท็บเล็ตที่ใช้ระบบปฏิบัติการAndroid อย่างครอบคลุมมากที่สุด

1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

- 1) สามารถนำความรู้ทุกอย่างที่ได้จากการพัฒนาแอปพลิเคชันนี้ ไปประยุกต์ใช้ได้ในอนาคต
- 2) ได้รับประสบการณ์ในการทำงานจริงในบริษัท
- 3) ได้เรียนรู้หลักการในการออกแบบ User Interface และ Layout ในหน้าต่างๆของแอปพลิเคชันบนโทรศัพท์มือถือและแท็บเล็ต เพื่ออำนวยความสะดวกต่อเหล่าโปรแกรมเมอร์ผู้รับผิดชอบเขียนแอปพลิเคชันนั้นๆให้ทำงานได้สะดวกยิ่งขึ้น

1.10 นิยามศัพท์เฉพาะ

Google Cloud Messaging (GCM) เป็นบริการที่อนุญาตให้นักพัฒนาแอปพลิเคชันสามารถส่งข้อมูลจาก Server ของตน ไปยังอุปกรณ์Android

Push คือการส่งข้อมูลไปยังโทรศัพท์มือถือ เช่น SMS และ Notification

DPI หรือ Dot Per Inch คือ ความหนาแน่นของหน้าจอ โดยวัดจาก จำนวนพิกเซลต่อ 1 นิ้ว เช่น โทรศัพท์มือถือมีขนาดจอที่เท่ากัน แต่จอหนึ่งมีความละเอียดมากกว่า อีกจอหนึ่งมีความละเอียดน้อยกว่า จอที่มีความละเอียดมากกว่าจะมีความหนาแน่นของหน้าจอเยอะกว่า ซึ่ง DPI แบ่งออกเป็น 6 แบบ ได้แก่ ldpi, mdpi, tvdpi, hdpi, xxhdpi และ xxxhdpi

หน่วย DP เป็นหน่วยของความละเอียดหน้าจอ ที่สมมติขึ้นมาในAndroidโดยอิงจาก Resolution, Size และ Density ของจอบนเครื่องนั้นๆ

Splash Screen คือ หน้า Intro ก่อนเข้าแอปพลิเคชัน Android อาจเป็นหน้าที่แสดงโลโก้ของบริษัท หรือสัญลักษณ์ของแอปพลิเคชัน

บทที่ 2

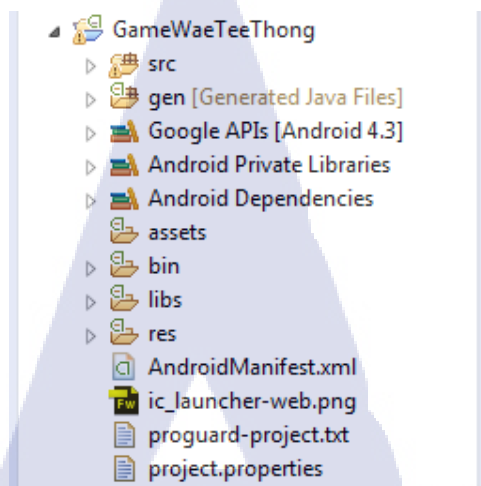
ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1 ทฤษฎีที่ใช้ในการปฏิบัติงาน

แอปพลิเคชันบนระบบปฏิบัติการ Android เขียนขึ้นมาจากภาษา Java โดยใช้ Android SDK ที่ทาง Google ได้ปล่อยออกมาให้นักพัฒนาแอปพลิเคชันสามารถนำไปใช้ได้ฟรี โดยใช้โปรแกรม Eclipse มาเป็นเครื่องมือในการพัฒนาโปรแกรม

2.1.1) โครงสร้างของแอปพลิเคชันบนระบบปฏิบัติการAndroid

- 1) src (Source Code) เป็นส่วนของ Source Code ที่เก็บไฟล์ .java
- 2) gen (Generated JAVA Files) เป็นส่วนที่ถูกสร้างขึ้นโดยอัตโนมัติ ภายในจะประกอบไปด้วยข้อความและ UI Element โดยถูกนำเข้ามานในโปรเจกผ่าน Android Plug-in ซึ่งไฟล์นี้เปรียบเสมือนการชี้ไปยังโฟลเดอร์ Drawable, Layout ,และ Values
- 3) res (Resource) เป็นส่วนของการแสดงผลของไฟล์อื่นๆที่นำมาเป็นส่วนประกอบร่วมกับ Code ที่เขียนขึ้น โดยมีการแบ่งโฟลเดอร์ไว้ตามชนิดของ Resource ที่ถูกสร้างขึ้นมาพร้อมกับโปรเจก
- 4) AndroidMainFest.xml เป็นโครงสร้างของ xml ไฟล์ ซึ่งจำกัดคุณสมบัติและการตั้งค่าต่างๆของแอปพลิเคชัน เช่น การกำหนดสิทธิ์ และเวอร์ชันของ Code เป็นต้น



ภาพที่ 2.1 โครงสร้างของแอปพลิเคชันบนระบบปฏิบัติการAndroid

2.1.2) ส่วนประกอบของแอปพลิเคชันบนระบบปฏิบัติการAndroid

1) Activity หรือ User Interface คือสิ่งที่ใช้ในการแสดงผลเพื่อให้ผู้ใช้งานได้เห็นและใช้งานได้ โดยหนึ่งแอปพลิเคชันสามารถมีได้มากกว่าหนึ่ง Activity

2) Service หรือ Service Provider (ผู้ให้บริการ Service) เป็นส่วนของการทำงานเบื้องหลัง โดย Service นั้นอาจมีการกระทำบางอย่าง เช่น ติดต่อรับส่งข้อมูลผ่านเครือข่าย หรือคำนวณค่าต่างๆและนำผลลัพธ์ไปแสดงในActivity

3) Broadcast Receiver เป็นตัวคอยรับและตอบสนองเหตุการณ์ต่างๆ เช่น แบตเตอรี่ต่ำ, ผู้ใช้เปลี่ยนภาษา เป็นต้น Broadcast Receiverไม่มีส่วนของการแสดงผลแต่จะเรียก Activity ขึ้นมาแสดงให้ผู้ใช้งาน

4) Content Provider (ผู้ให้บริการข้อมูล) คือกลุ่มของข้อมูลที่สร้างขึ้นจากแอปพลิเคชัน เพื่อให้แอปพลิเคชันอื่นๆสามารถนำไปใช้ได้ โดยการจัดเก็บข้อมูลของผู้ให้บริการข้อมูลนั้นจะอยู่ในลักษณะของไฟล์, SQLite, และอื่นๆ

2.1.3) วงจรชีวิตของAndroid Activity

Activity มีหน้าที่สร้างและควบคุมการทำงานของหน้าจอเพื่อโต้ตอบกับผู้ใช้ ซึ่งจะมีได้แค่ Activity เดียวเท่านั้นที่โต้ตอบกับผู้ใช้ในขณะนั้น โดยแต่ละ Activity จะมีวงจรชีวิตเป็นของตัวเอง แบ่งออกได้ดังนี้

- 1) Running : เป็นสถานะที่บ่งบอกว่า Activity กำลังทำงานอยู่ในขณะนั้น
- 2) Pause : เป็นสถานะที่บ่งบอกว่า Activity ปรากฏอยู่บนหน้าจอ แต่ยังไม่ได้ทำงานในขณะนั้น
- 3) Stop : เป็นสถานะที่บ่งบอกว่า Activity ไม่ปรากฏบนหน้าจอ
- 4) Kill : เป็นสถานะที่บ่งบอกว่า Activity ถูกทำลายไป

Android ถูกออกแบบมาเพื่อใช้งานกับอุปกรณ์มือถือ ซึ่งจะมีทรัพยากรของระบบที่จำกัด จึงจำเป็นต้องมีกลไกที่คอยจัดการบริหารทรัพยากรเหล่านั้น เรียกว่า Android Activity Lifecycle ซึ่งจะกำหนดสถานะหรืออีเวนต์ที่ Activity ต้องผ่านตั้งแต่ตอนที่ถูกสร้างขึ้นจนกว่าจะเสร็จสิ้นการทำงาน ดังรูปที่ 2.2 ซึ่งแต่ละ Activity จะตรวจสอบและตอบสนองกับอีเวนต์เหล่านี้โดย Instant Method ที่ Override จากคลาส Activity ได้แก่

1) onCreate จะถูกเรียกเมื่อ Activity ถูกสร้างขึ้นครั้งแรก ซึ่งปกติจะมีการสร้างส่วนติดต่อผู้ใช้หรือวิว (View) ต่างๆ เปิดไฟล์ข้อมูลตารางที่ใช้ใน Activity และกำหนดค่าเริ่มต้นของ Activity

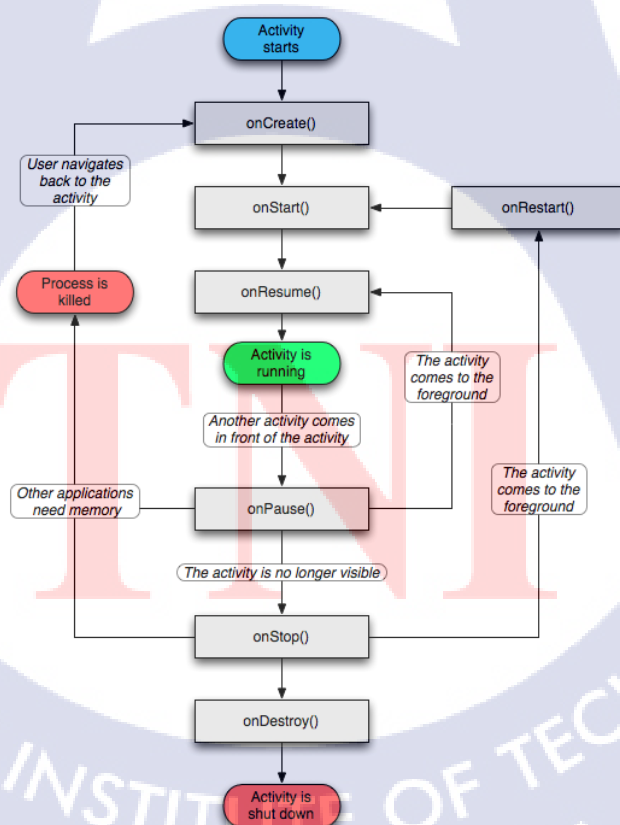
2) onStart จะถูกเรียกก่อนที่ Activity จะแสดงผลออกมาที่หน้าจอ เมื่อ onStart ทำงานเสร็จ ถ้า Activity สามารถเป็นโฟร์กราวนด์ Activity บนหน้าจอ การควบคุมจะถูกส่งต่อไปยัง onResume แต่ถ้า Activity ไม่สามารถเป็นโฟร์กราวนด์ Activity ได้ การควบคุมจะถูกส่งต่อไปยังเมธอด onStop

3) onResume จะถูกเรียกหลังจาก onStart ถ้า Activity เป็นโฟร์กราวนด์ Activity ณ จุดนี้ Activity จะทำงานและตอบสนองกับผู้ใช้ นอกจากนี้ onResume ยังถูกเรียกอีกครั้ง ถ้า Activity ที่ไม่ได้เป็นโฟร์กราวนด์กลับมาเป็นโฟร์กราวนด์ Activity อีกครั้ง

4) onPause จะถูกเรียกเมื่อระบบทำการนำ Activity ใหม่มาเป็นโฟร์กราวนด์ ซึ่ง Activity เดิมจะไม่สามารถเข้าถึงหน้าจอและจำเป็นต้องหยุดการดำเนินการที่ใช้แบตเตอรี่และ CPU Cycle ชั่วคราว โดย Activity จำเป็นต้องใช้ประโยชน์จากเมมโอดนี้ในการจัดเก็บสถานะหรือค่าต่างๆ ที่จำเป็นจะต้องใช้ในกรณีที่ Activity กลับมาเป็นโฟร์กราวนด์ Activity อีกครั้ง ซึ่ง Android ยังไม่รับประกันว่า Activity จะสามารถกลับมาเป็น โฟร์กราวนด์ได้เสมอไป เนื่องจากถ้าอุปกรณ์มีหน่วยความจำไม่เพียงพอและไม่มีหน่วยความจำเสมือน (Virtual memory) บนดิสก์ที่ขยายต่อได้ ดังนั้น Activity จะถูกทำลายและคืนทรัพยากรให้แก่ระบบ

5) onStop จะถูกเรียกเมื่อ Activity ไม่สามารถมองเห็นได้อีก หรือ Activity กำลังจะถูกทำลาย ถัดจากนี้เมธอด onRestart จะถูกเรียก ถ้า Activity จะกลับมาแสดงผลและตอบสนองกับผู้ใช้ หรือเมธอด onDestroy จะถูกเรียกเมื่อ Activity สิ้นสุดการทำงาน

6) onDestroy เมธอดสุดท้ายที่ Activity จะทำงานก่อนที่จะถูกทำลาย



ภาพที่ 2.2 Android Activity Life Cycle

2.2 เทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.2.1 Java

Java หรือ Java Programming Language คือภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทน ภาษา C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษา Objective-C แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนด้วย Java ได้

ภาษา Java เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้น คลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

ข้อดีของ ภาษา Java

- 1) ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้เราสามารถใช้อำนาจหรือชื่อ ต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น
- 2) โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารรถถูก Compile และ Run บนเครื่องพีซีธรรมดาได้
- 3) ภาษา Java มีการตรวจสอบข้อผิดพลาดทั้งตอน Compile Time และ Runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้ Debug โปรแกรมได้ง่าย
- 4) ภาษา Java มีความซับซ้อนน้อยกว่าภาษา C++ เมื่อเปรียบเทียบ Code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวน

Code น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น

5) ภาษา Java ถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วย Java มีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้น ด้วยภาษาอื่น เพราะ Java มี Security ทั้ง Low Level และ High Level ได้แก่ Electronic Signature, Public And Private Key Management, Access Control และ Certificates

6) มี IDE, Application Server, และ Library ต่าง ๆ มากมายสำหรับจาวาที่เราสามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย ทำให้เราสามารถลดค่าใช้จ่ายที่ต้องเสียไปกับการซื้อ Tool และ S/W ต่าง ๆ

ข้อเสียของ ภาษา Java

1) ทำงานได้ช้ากว่า Native Code (โปรแกรมที่ Compile ให้อยู่ในรูปของภาษาเครื่อง) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น C หรือ C++ ทั้งนี้ก็เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลาง ก่อน แล้วเมื่อโปรแกรมทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีก ทีหนึ่ง ทีละคำสั่ง (หรือกลุ่มของคำสั่ง) ณ Runtime ทำให้ทำงานช้ากว่า Native Code ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ Compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วยจาวา

2) Tool ที่มีในการใช้พัฒนาโปรแกรม Java มักไม่ค่อยเก่ง ทำให้หลายอย่างโปรแกรมเมอร์จะต้องเป็นคนทำเอง ทำให้ต้องเสียเวลาทำงานในส่วนที่ Tool ทำไม่ได้ ถ้าเราดู Tool ของ MS จะใช้งานได้ง่ายกว่า และพัฒนาได้เร็วกว่า (แต่เราต้องซื้อ Tool ของ MS และก็ต้องรันบน Platform ของ MS)

2.2.2 JavaScript Object Notation : JSON

JSON เป็นฟอร์แมตสำหรับแลกเปลี่ยนข้อมูลคอมพิวเตอร์ ฟอร์แมต JSON นั้นอยู่ในรูปข้อความธรรมดา (Plain Text) ที่ทั้งมนุษย์และโปรแกรมคอมพิวเตอร์สามารถอ่านเข้าใจได้

มาตรฐานของฟอร์แมต JSON คือ RFC 4627 มี Internet Media Type เป็น Application/JSON และมีนามสกุลของไฟล์เป็น .json แต่ส่วนใหญ่จะไม่ใช้ File นามสกุลนี้ เพราะส่วนใหญ่ที่ทำเป็น JSON จะเป็นเว็บ Dynamic เพื่อจะเรียก JSON ใช้งานร่วมกับ Ajax

ก่อนหน้านี้ Ajax โปรแกรมเมอร์จะนิยมใช้รับข้อมูลแบบ XML ซึ่งจะจับข้อมูลมาใส่ Tag ต่างๆ เป็นชุดๆ เพื่อง่ายต่อการจัดกลุ่มข้อมูล ซึ่ง JSON จะมีลักษณะเป็นชุดข้อมูลแบบ XML แต่การเขียนจะสั้นกว่าเยอะซึ่งทำให้ลดค่า Transfer Data ได้เล็กน้อย

ลักษณะCode ตัวอย่างของ JSON มีลักษณะดังนี้

```
{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "phoneNumbers": [
    "212 555-1234",
    "646 555-4567"
  ]
}
```

2.2.3 MySQL

MySQL คือ โปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัทMySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่งเอสคิวแอล (SQL = Structured Query Language) เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นๆ อย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้ เช่น ทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อ

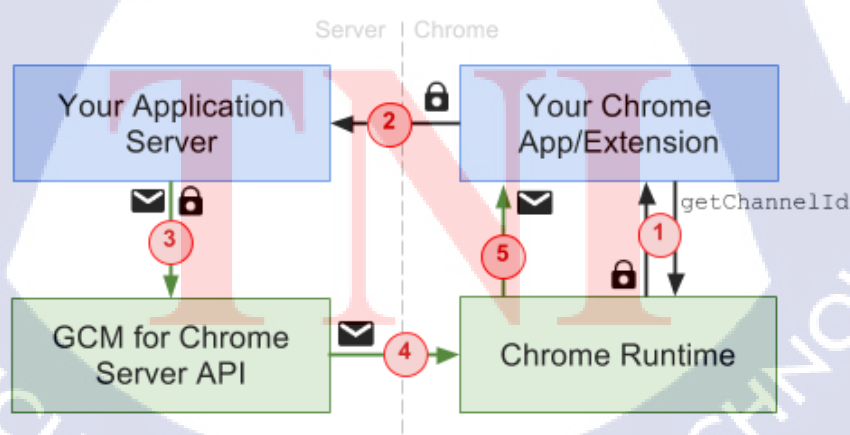
ให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษา PHP , ภาษา ASP.NET , หรือภาษา JSP เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษา VB ภาษา JAVA เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์สที่ถูกนำไปใช้งานมากที่สุด

ชนิดของข้อมูลที่สนับสนุน

- 1) ชนิดของข้อมูลที่เป็นตัวเลข เช่น BIT, TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT
- 2) ชนิดของข้อมูลที่เป็นวันที่และเวลา เช่น DATETIME, TIMESTAMP, YEAR
- 3) ชนิดของข้อมูลที่เป็นตัวอักษร เช่น CHAR, VARCHAR, BINARY, BLOB, TEXT, ENUM

2.2.4 Google Cloud Messaging : GCM

GCM เป็นบริการที่อนุญาตให้นักพัฒนาแอปพลิเคชันสามารถส่งข้อมูลจาก Server ของตน ไปยังอุปกรณ์ Android โดยระหว่างทางนักพัฒนาแอปพลิเคชันต้องส่งข้อมูลจากเซิร์ฟเวอร์ของตัวเองมายังเซิร์ฟเวอร์ GCM ของ Google ผ่าน API เพื่อให้ Google ส่ง Notification ไปยังปลายทางซึ่งก็คือโทรศัพท์มือถือนั่นเอง



ภาพที่ 2.3 การทำงานของ GCM

2.2.5 phpMyAdmin

phpMyAdmin คือโปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูล MySQL แทนการคีย์คำสั่ง เนื่องจากถ้าเราจะใช้ฐานข้อมูลที่เป็น MySQL บางครั้งจะมีความลำบากและยุ่งยากในการใช้งาน ดังนั้นจึงมีเครื่องมือในการจัดการฐานข้อมูล MySQL ขึ้นมาเพื่อให้สามารถจัดการ ตัวDBMS ที่เป็น MySQL ได้ง่ายและสะดวกยิ่งขึ้น โดย phpMyAdmin ก็ถือเป็นเครื่องมือชนิดหนึ่งในการจัดการนั่นเอง

phpMyAdmin เป็นส่วนต่อประสานที่สร้างโดยภาษาพีเอชพี ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้าง Table ใหม่ๆ และยังมี Function ที่ใช้สำหรับการทดสอบการ Query ข้อมูลด้วยภาษา SQL พร้อมกันนั้น ยังสามารถทำการ Insert Delete Update หรือแม้กระทั่งใช้ คำสั่งต่างๆ เหมือนกับกับการใช้ภาษา SQL ในการสร้างตารางข้อมูล

phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการข้อมูล MySQL ผ่าน Web Browser ได้โดยตรง phpMyAdmin ตัวนี้จะทำงานบน Web Server เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server ความสามารถของ phpMyAdmin คือ

- สร้างและลบ Database
- สร้างและจัดการ Table เช่น แทรก Record, ลบ Record, แก้ไข Record, ลบ Table, แก้ไข Field
- โหลดเท็กซ์ไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้
- หาผลสรุป (Query) ด้วยคำสั่ง SQL



ภาพที่ 2.4 ตัวอย่างหน้าจอ phpMyAdmin

2.2.6 Adobe DreamWeaver

Adobe Dreamweaver เป็นโปรแกรมแก้ไข HTML พัฒนาโดยบริษัทแมโครมีเดีย สำหรับการออกแบบเว็บไซต์ในรูปแบบ WYSIWYG กับการควบคุมของส่วนแก้ไขรหัส HTML ในการพัฒนาโปรแกรมที่มีการรวมทั้งสองแบบเข้าด้วยกันแบบนี้ ทำให้ Dreamweaver เป็นโปรแกรมที่แตกต่างจากโปรแกรมอื่นๆ ในประเภทเดียวกัน ในช่วงปลายปีทศวรรษ 2533 จนถึงปี พ.ศ. 2544 Dreamweaver มีสัดส่วนตลาดโปรแกรมแก้ไข HTML อยู่มากกว่า 70%

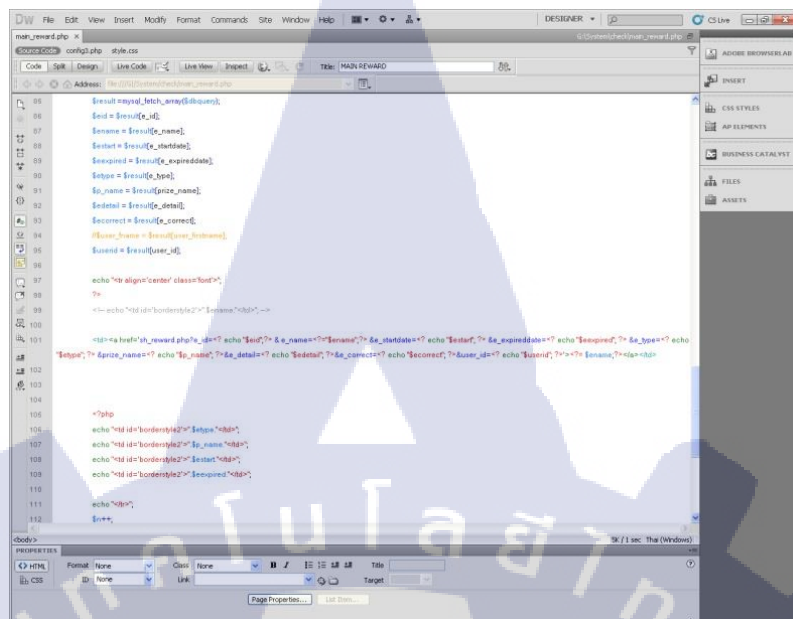
Dreamweaver มีทั้งในระบบปฏิบัติการแมคอินทอช และไมโครซอฟท์วินโดวส์ Dreamweaver ยังสามารถทำงานบนระบบปฏิบัติการแบบยูนิกซ์ ผ่านโปรแกรมจำลองอย่าง WINE ได้ รุ่นล่าสุดคือ Dreamweaver CS6

การทำงานกับภาษาต่างๆ

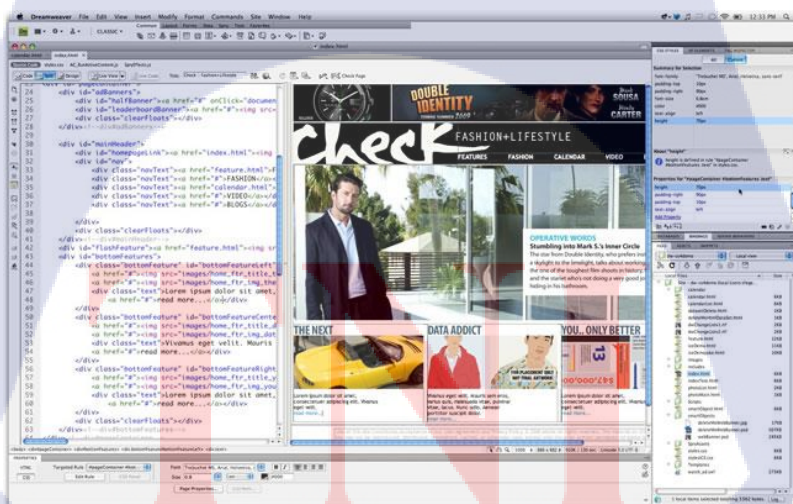
Dreamweaver สามารถทำงานกับภาษาคอมพิวเตอร์ในการเขียนเว็บไซต์แบบไดนามิก ซึ่งมีการใช้ HTML เป็นตัวแสดงผลของเอกสาร เช่น ASP, ASP.NET, PHP, JSP และ ColdFusion รวมถึงการจัดการฐานข้อมูลต่างๆ อีกด้วย และในเวอร์ชันล่าสุด (เวอร์ชัน 8) ยังสามารถทำงานร่วมกับ XML และ CSS ได้อย่างง่ายดาย

มุมมองในการทำงานกับ Dreamweaver

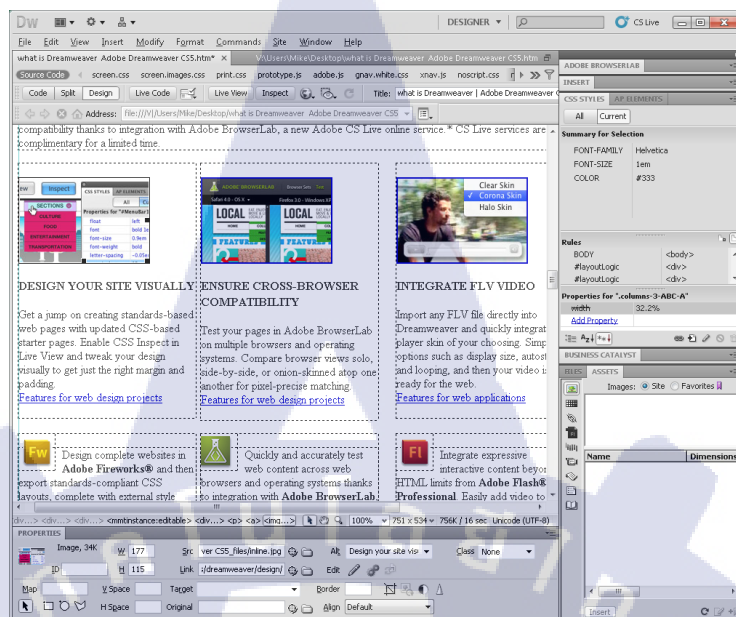
- มุมมองแบบ Code เป็นมุมมองที่ใช้ในการแก้ไขโค้ด HTML หากผู้ที่จะใช้มุมมอง Code ในการจัดการกับไฟล์ต้องมีความรู้เกี่ยวกับโค้ด HTML
- มุมมองแบบ split เป็นมุมมองที่แสดงให้เห็นทั้งโค้ด HTML และมุมมองปกติ ซึ่งจะบอกให้ทราบว่าการทำงานกับไฟล์ ขณะนั้นอยู่ในส่วนใดของโค้ด HTML กับมุมมองปกติ
- มุมมองแบบ Design เป็นมุมมองการทำงานปกติ ที่แสดงเหมือนจริงของเอกสารเว็บเพจ



ภาพที่ 2.5 มุมมองแบบ Code ใน Dreamweaver



ภาพที่ 2.6 มุมมองแบบ Split ใน Dreamweaver



ภาพที่ 2.7 มุมมองแบบ Design ใน Dreamweaver

2.2.7 File Transfer Protocol : FTP

FTP คือการแลกเปลี่ยนข้อมูลแบบ FTP หมายถึง การดาวน์โหลด หรือย้ายไฟล์ในระบบอินเทอร์เน็ตมายังเครื่องคอมพิวเตอร์ เช่นเดียวกันก็สามารถนำไฟล์จากเครื่องคอมพิวเตอร์ไปไว้ในพื้นที่เก็บข้อมูลโดยผ่านโปรแกรม FTP สามารถดาวน์โหลดไฟล์ได้หลายประเภทแล้วแต่ว่าผู้ใช้งานต้องการใช้งานไฟล์ประเภทใด จุดเด่นสามารถนำไปใช้ร่วมกับระบบการการทำเว็บไซต์, เป็นช่องทางเก็บหรือส่งข้อมูลหากลูกค้าหรือภายในองค์กรได้ ที่ Webmaster ใช้ นั่นคือ การ FTP อัปโหลดไฟล์ ขึ้นโฮส (Host)

เอฟทีพี (FTP) ย่อมาจากภาษาอังกฤษว่า File Transfer Protocol : FTP หมายถึง การดึงไฟล์จากคอมพิวเตอร์ในเครือข่ายอินเทอร์เน็ตมายังเครื่องคอมพิวเตอร์ของเราหรือของผู้ใช้คนอื่น โดยปกติการดึงไฟล์จากอินเทอร์เน็ต เราจะต้องติดต่อไปยังผู้ให้บริการ ซึ่งเรียกว่า เอฟทีพีไซต์ (FTP Site) เอฟทีพีไซต์ที่สำคัญในอดีตจะเป็นหน่วยงานของรัฐบาล หรือมหาวิทยาลัย FTP เป็นโปรแกรมที่ใช้สำหรับ Upload/Download หรือโครงสร้างของไฟล์และ Directory ใน Server ซึ่ง FTP มีอยู่ 2 ประเภท ได้แก่

- FTP ประเภทที่ 1 คือ โพรเวทเอฟทีพี (Private FTP) หรือเอฟทีพีเฉพาะกลุ่ม นิยมใช้ตามสถานศึกษาและภายในบริษัท ผู้ใช้บริการจะต้องมีรหัสผ่านเฉพาะจึงจะใช้งานได้

- FTP ประเภทที่ 2 คือ อะโนนิมัสเอฟทีพี (Anonymous FTP) เป็นFTPสาธารณะ ให้บริการดึงไฟล์ฟรีโดยไม่ต้องมีรหัสผ่าน เราสามารถติดต่อเว็บไซต์เหล่านี้ได้โดยผ่านเว็บเบราว์เซอร์ ซึ่งปัจจุบันเว็บไซต์สาธารณะมีอยู่เป็นจำนวนมาก การใช้FTPมีจุดมุ่งหมายหลักก็คือ การนำเอาไฟล์ข้อมูลต่างๆ ที่อยู่บนอินเทอร์เน็ตมาใช้งาน ข้อมูลเหล่านั้นอาจเป็นข้อมูลตัวหนังสือ รูปภาพ เสียง วิดีโอ หรือโปรแกรมสำเร็จรูป โดยเฉพาะโปรแกรมใหม่ๆ ที่บริษัทต่างๆ กิดค้นขึ้นมา และต้องการเผยแพร่ไปสู่สาธารณชน ผู้ใช้อินเทอร์เน็ตคนใดที่สนใจก็สามารถใช้FTPดึงเอาโปรแกรมเหล่านั้นมาใช้งานได้

ประเภทของแฟ้มข้อมูล ไฟล์ที่เราสามารถดึงมาใช้งานได้นั้น แบ่งออกเป็นหลายประเภท ดังนี้

- ไฟล์ข้อความ (Text File) เป็นไฟล์ที่แพร่หลายมากที่สุดในอินเทอร์เน็ต โดยจะบรรจุข้อความตัวอักษรภาษาต่างๆ เช่น ไทย อังกฤษ ญี่ปุ่น จีน และอื่นๆ เป็นต้น ปัจจุบันเวปไซด์ใดเว็บนิยมใช้ไฟล์ข้อความมาก เช่น หนังสือพิมพ์ นิตยสาร อีเมล เป็นต้น การสังเกตไฟล์ข้อความดูได้จากนามสกุล คือ txt , doc , html เป็นต้น การดึงไฟล์ข้อความจากอินเทอร์เน็ตทำได้ง่าย ใช้เวลาน้อย เพราะมีขนาดไม่ใหญ่มาก

- ไฟล์ภาพ (Graphics) เป็นภาพประกอบบนเว็บเพจ นิยมใช้กันมากรองจากไฟล์ข้อความ โดยปกติไฟล์ภาพจะมีขนาดใหญ่ การดึงไฟล์จึงต้องใช้เวลาานกว่าปกติ ตัวอย่างไฟล์สังเกตได้จากนามสกุลไฟล์ เช่น bmp , jpg , gif เป็นต้น

- ไฟล์เสียง ไฟล์เสียงได้รับความนิยมมากขึ้นตามลำดับโดยเฉพาะไฟล์เสียงจาก Real Audio ที่ให้เสียงต่อเนื่องเหมือนฟังรายการเพลงจากวิทยุทั่วไป และไฟล์ประเภท Mp3 ซึ่งเป็นเพลงต่างๆ ทั้งภาษาไทยและภาษาอังกฤษ ไฟล์เสียงที่นิยมใช้ประกอบในหน้าเว็บเพจ ตัวอย่างไฟล์เสียง เช่น au , ra , ram , wav เป็นต้น

FTP มีความสำคัญและประโยชน์อย่างไร

ในโลกธุรกิจปัจจุบันที่มีการแข่งขันกันมากในทุกๆ ด้าน ไม่ว่าจะเป็นด้านการผลิต ด้าน การตลาด การบริหาร การจัดการ และด้านที่ขาดไม่ได้ด้วยเช่นกัน คือ ด้านการสื่อสาร ซึ่งแต่ละธุรกิจมีความจำเป็นต้องใช้ ไม่ว่าจะเป็นการติดต่อสื่อสารกับลูกค้า การติดต่อสื่อสารภายในหรือระหว่างบริษัท ซึ่ง FTP มีส่วนช่วยอย่างมากในการสื่อสารต่างๆ FTP จะช่วยให้การถ่ายโอนข้อมูลเป็นไปได้อย่างมากขึ้น แต่ละบริษัทหรือหน่วยงานสามารถมีข้อมูลมากมายหลายรูปแบบที่ต้องการสื่อสารไปยังแหล่งอื่น หรือแม้แต่ต้องการข้อมูลที่เป็นประโยชน์จากแหล่งอื่นเข้ามาใช้ เช่น ข้อมูลข่าวสารประจำวัน บทความ ข้อมูลทางสถิติ ผลการทดลองทางวิทยาศาสตร์ เป็นต้น การจะเดินทางไปเอาข้อมูลต่างๆ เองก็ถือเป็นการใช้เวลาโดยเปล่าประโยชน์ ในเมื่อมีเทคโนโลยีเข้ามาช่วยเหลือแล้ว FTP จะเป็นตัวช่วยให้การได้รับข้อมูลเหล่านี้สามารถทำได้ง่ายยิ่งขึ้นเพียงอยู่หน้าจอคอมพิวเตอร์เท่านั้น ผู้ใช้งานสามารถใช้ FTP ในการโอนข้อมูลจำนวนมากจากแหล่งที่อนุญาตให้ใช้ได้ ซึ่งเรียกว่าเป็นแหล่งบริการ FTP ซึ่งมักเป็นศูนย์รวบรวมข้อมูลข่าวสารต่างๆ อยู่มาก และเปิดบริการทั่วไป เพียงแค่ผู้ใช้อินเตอร์เน็ตเข้าไปใช้บริการคัดลอกแฟ้มข้อมูลต่างๆ มาใช้งาน

วิธีการทำงานของ FTP

FTP ทำงานในแบบไคลเอนต์เซิร์ฟเวอร์ โดยพัฒนาขึ้นตามโปรโตคอลพื้นฐาน TCP ซึ่งจะต้องมีการติดต่อเพื่อจองช่องสื่อสาร (Connection Establishment) ก่อนทำการสื่อสารจริง ซึ่งเรียกว่าเป็นการติดต่อแบบที่ต้องขอเชื่อมต่อก่อน (Connection - Oriented) ในการใช้งาน FTP เพื่อเริ่มการติดต่อสื่อสารนั้น จะต้องระบุหมายเลข IP ปลายทาง และต้องผ่านการแจ้งรหัส Login และ Password ของเซิร์ฟเวอร์ที่จะติดต่อก่อนจึงจะเข้าใช้งานได้

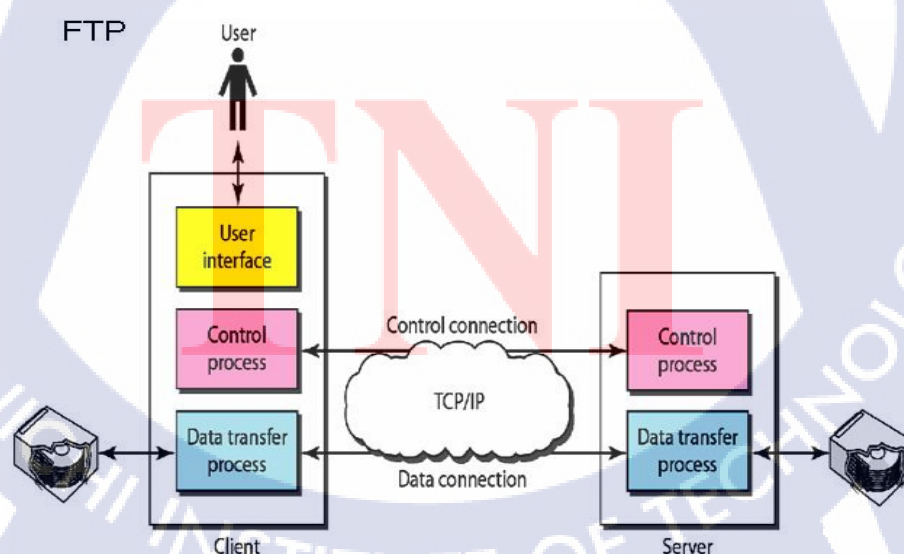
ข้อมูลของ FTP ที่สื่อสารระหว่างกันมี 2 ประเภทคือ

- ข้อมูล(Data) หมายถึงข้อมูลต่างๆ ที่ต้องการรับส่ง รวมทั้งไฟล์ที่รับมาจากเซิร์ฟเวอร์ หรือส่งมาจาก ไคลเอนต์แล้วไปเก็บที่เซิร์ฟเวอร์
- ข้อมูลคำสั่ง (Command) FTP จะมีคำสั่งที่ใช้สั่งงานต่างๆ เช่น Dir เป็นคำสั่งที่ใช้แสดงชื่อไฟล์หรือไดเรกทอรีในเครื่องเซิร์ฟเวอร์หรือ Get ใช้โหลดไฟล์มาที่เครื่องไคลเอนต์ผ่านโปรแกรม FTP แล้วโปรแกรมจะส่งคำสั่งไปยังเซิร์ฟเวอร์เพื่อทำงาน และแจ้งผลการทำงานกลับมา

ยังไคลเอนต์ ซึ่งผลการทำงานนี้จะนำหน้าด้วยตัวเลข 3 หลัก เป็นรหัสที่ใช้แสดงสถานะการทำงานภายในของ FTP และต่อด้วยข้อความที่เป็นเท็กซ์ต่อท้าย ซึ่งก็คือผลการทำงานหรือคำอธิบายต่างๆ โดยที่ FTP มีกระบวนการภายในที่จะตรวจสอบได้ว่าข้อมูลที่จะรับส่งนี้เป็นประเภทคำสั่งไม่ใช่ตัวข้อมูลที่ต้องการจะโอนย้าย การที่ FTP สามารถแยกแยะข้อมูลจริงออกจากข้อมูลที่เป็นคำสั่งได้นั้นถือว่าเป็นหน้าที่การทำงานของโมดูลใน FTP ที่เรียกว่าโปรโตคอล (Protocol Interpreter Module หรือ PI) ซึ่งทำหน้าที่รองรับการทำงานคำสั่งต่างๆของ FTP และในส่วนของข้อมูลที่รับส่งนั้นจะเป็นหน้าที่ของโมดูลโอนข้อมูล (Data Transfer หรือ DT) ซึ่งโมดูลทั้งสองนี้จะต้องทำงานอยู่ทั้งในเครื่องที่เป็นเซิร์ฟเวอร์ และไคลเอนต์

ส่วน Shareware หมายถึง ซอฟต์แวร์ที่ผู้ผลิตแจกให้ลองไปใช้ดูก่อน และเมื่อใช้แล้วพอใจจะนำไปใช้จริงก็ค่อยส่งเงินมาชำระทีหลัง ถ้าไม่นำไปใช้จริงก็ไม่ต้องส่งเงินมาชำระ

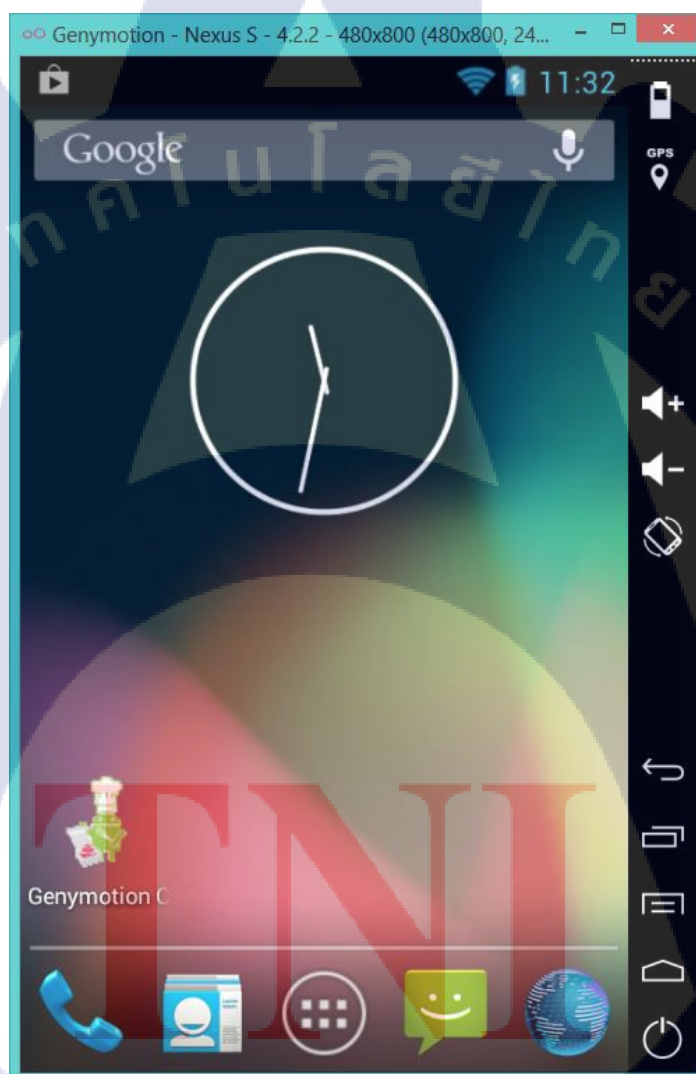
ผู้ผลิต Freeware และ ผู้ผลิต Shareware จะทำการส่งซอฟต์แวร์ของตนเองที่ต้องการแจกจ่ายไปไว้ที่คอมพิวเตอร์ที่เป็น FTP Server และใครก็ตามที่สนใจจะลงนำไปซอฟต์แวร์ของผู้ผลิตไปใช้ดูก็ให้ไปทำการ Download จากคอมพิวเตอร์ที่เป็น FTP Server เครื่องนั้นมายังเครื่องคอมพิวเตอร์ของตนเอง ในบางกรณีถ้าท่านมีข้อมูลที่น่าสนใจและต้องการเผยแพร่ ท่านก็สามารถส่งข้อมูลนั้นไปไว้ที่ FTP Server ได้ ตัวอย่าง FTP Server เช่น ftp.chula.ac.th



ภาพที่ 2.8 ลักษณะการทำงานของ FTP : File Transfer Protocol

2.2.8 GenyMotion

GenyMotion เป็น Emulator ที่สร้างขึ้นมาเพื่อนักพัฒนาซึ่งมีความเร็วกว่า Default Emulator บน Eclipse อย่างมาก ข้อดีของ GenyMotion คือสามารถ Login Google Account เพื่อใช้งาน Sync กับ Gmail ได้ และสามารถเข้า Play Store เพื่อดาวน์โหลดแอปพลิเคชันได้และรองรับ Google Maps Android API v2 ด้วย



ภาพที่ 2.9 Interface ของ GenyMotion

บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนการปฏิบัติงาน

ตารางที่ 3.1 แผนการปฏิบัติงานสหกิจศึกษา

หัวข้องาน	เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	เดือนที่ 4
ศึกษาการใช้งาน JSON / ทำ Application Weather Forecast				
ศึกษาการสร้าง Widget / ทำ Widget ของ App ที่ทำได้				
ศึกษาการใช้งาน SQLite				
ทำ Application ดึง feed จาก Twitter				
ออกแบบ Layout Web หลังบ้านของ Application เกมเวทีทอง				
ออกแบบ Layout Application เกมเวทีทอง				
ศึกษาการเขียน JSON Encode จาก PHP				
เขียนCode Web หลังบ้านของApplication เกมเวทีทอง				
เขียนCode Application เกมเวทีทอง				
ทดสอบ Application เพื่อหาข้อผิดพลาด				
แก้ไขข้อผิดพลาดของ Web หลังบ้าน และ App				
Demo Application เกมเวทีทอง				
จัดทำรูปเล่มรายงาน				
ตรวจสอบและแก้ไขข้อผิดพลาดของรูปเล่มรายงาน				

3.2 รายละเอียดโครงการงาน

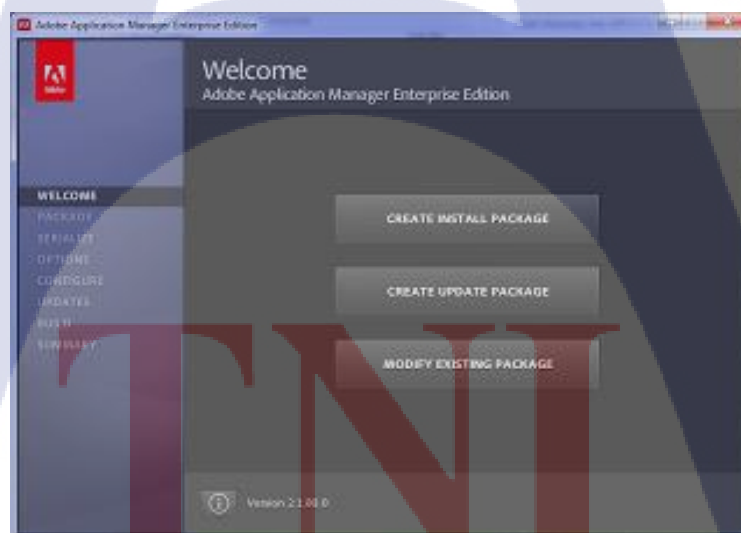
3.2.1 ประชากรและกลุ่มตัวอย่าง

โครงการแอปพลิเคชันเกมเวทีทองมีกลุ่มเป้าหมายคือประชาชนชาวไทยที่ใช้โทรศัพท์มือถือและ SmartPhone ระบบปฏิบัติการAndroid ซึ่งรองรับฟังก์ชันการ TouchScreen และมีเวอร์ชันของระบบปฏิบัติการ Android เป็นเวอร์ชัน 2.3.3 (Gingerbread) ขึ้นไป

3.2.2 การสร้างเครื่องมือที่ใช้ในโครงการงาน

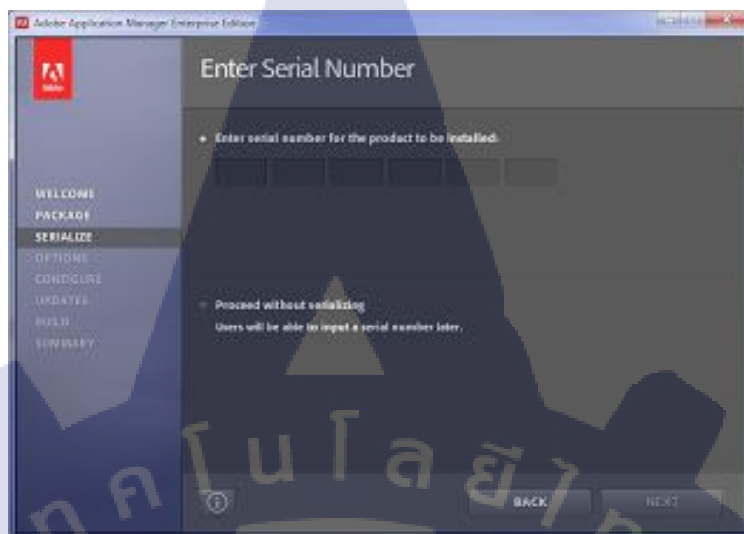
3.2.2.1 วิธีการติดตั้งโปรแกรม Adobe Master Collection CS6

1) ดาวน์โหลด Adobe CS6 ที่ <http://www.adobe.com/downloads> เมื่อดาวน์โหลดโปรแกรมเสร็จแล้ว ให้เปิดโปรแกรมแล้วคลิกที่ CREATE INSTALL PACKAGE เพื่อทำการติดตั้ง



ภาพที่ 3.1 ขั้นตอนการลงโปรแกรม Adobe Master Collection CS6 (1)

2) กรอก Serial Number ของโปรแกรมให้เรียบร้อย



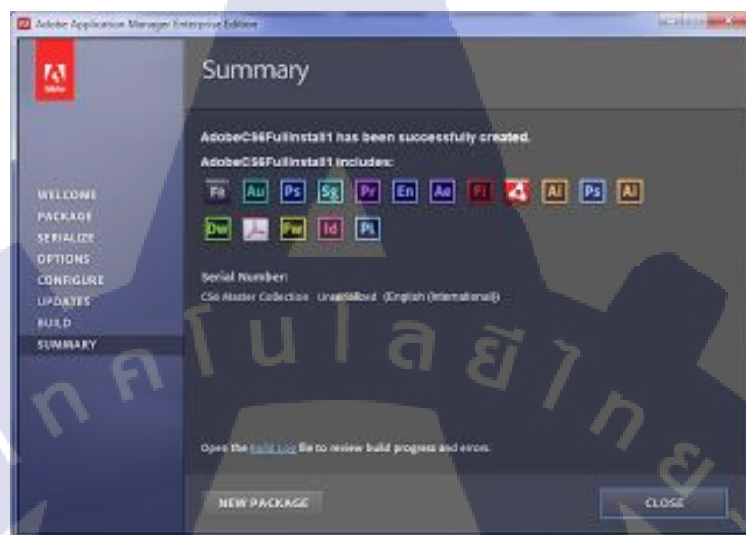
ภาพที่ 3.2 ขั้นตอนการลงโปรแกรม Adobe Master Collection CS6 (2)

3) เลือกโปรแกรมที่ต้องการติดตั้ง โดยคลิกถูกหน้าโปรแกรมที่ต้องการ สำหรับโครงการชิ้นนี้เลือก Adobe Photoshop และ Adobe DreamWeaver เมื่อเลือกโปรแกรมที่ต้องการเสร็จแล้ว ให้กดที่ปุ่ม Next



ภาพที่ 3.3 ขั้นตอนการลงโปรแกรม Adobe Master Collection CS6 (3)

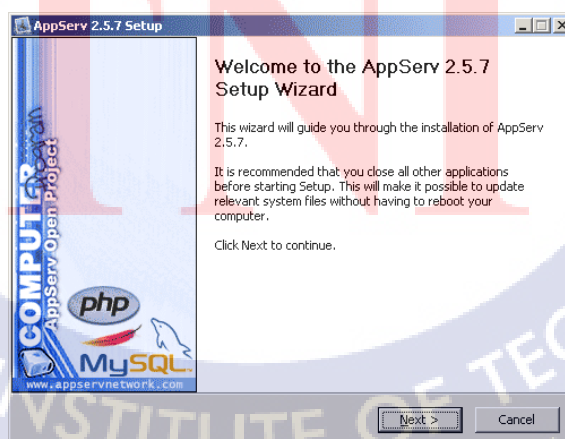
4) ระบบจะทำการติดตั้งโปรแกรมลงในเครื่องคอมพิวเตอร์ หากโปรแกรมทั้งหมดถูกติดตั้งเสร็จสมบูรณ์แล้วจะปรากฏดังภาพที่ 3.4 จากนั้นให้กดปุ่ม Close เพื่อเริ่มต้นการใช้งานโปรแกรม



ภาพที่ 3.4 ขั้นตอนการลงโปรแกรม Adobe Master Collection CS6 (4)

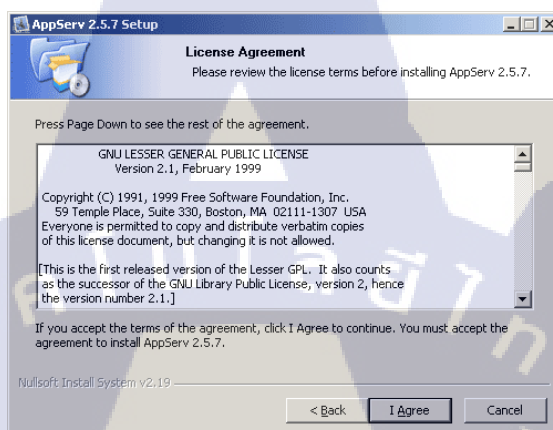
3.2.2.2 วิธีการติดตั้งโปรแกรม AppServ

1) ดาวน์โหลดโปรแกรม AppServ จาก <http://www.appservnetwork.com> เมื่อดาวน์โหลดโปรแกรมเสร็จ ให้ดับเบิลคลิกไฟล์ appserv-win32-x.x.x.exe เพื่อทำการติดตั้ง



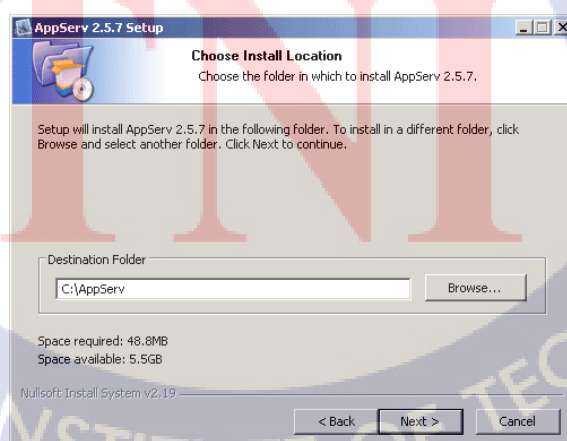
ภาพที่ 3.5 ขั้นตอนการติดตั้งโปรแกรม AppServ

2) เข้าสู่ขั้นตอนเงื่อนไขการใช้งานโปรแกรม โดยโปรแกรม AppServ ได้แจกจ่ายในรูปแบบ GNU License หากผู้ติดตั้งอ่านเงื่อนไขต่างๆ เสร็จสิ้นแล้ว หากยอมรับเงื่อนไขให้กด Next เพื่อเข้าสู่การติดตั้งในขั้นต่อไป แต่หากว่าไม่ยอมรับเงื่อนไขให้กด Cancel เพื่อออกจาก การติดตั้งโปรแกรม AppServ ดังรูป



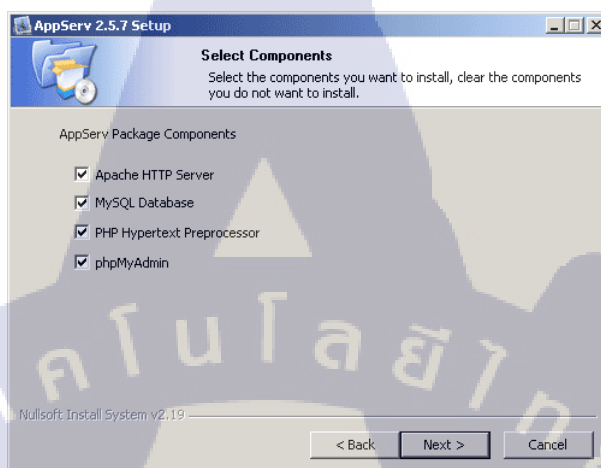
ภาพที่ 3.6 แสดงรายละเอียดเงื่อนไขการ GNU License

3) เข้าสู่ขั้นตอนการเลือกปลายทางที่ต้องการติดตั้ง โดยค่าเริ่มต้นปลายทางที่ติดตั้งจะเป็น C:\AppServ หากต้องการเปลี่ยนปลายทางที่ติดตั้ง ให้กด Browse แล้วเลือกปลายทางที่ต้องการ ตามรูปที่ 3 เมื่อเลือกปลายทางเสร็จสิ้นให้กดปุ่ม Next เพื่อเข้าสู่ขั้นตอนการติดตั้งขั้นต่อไป



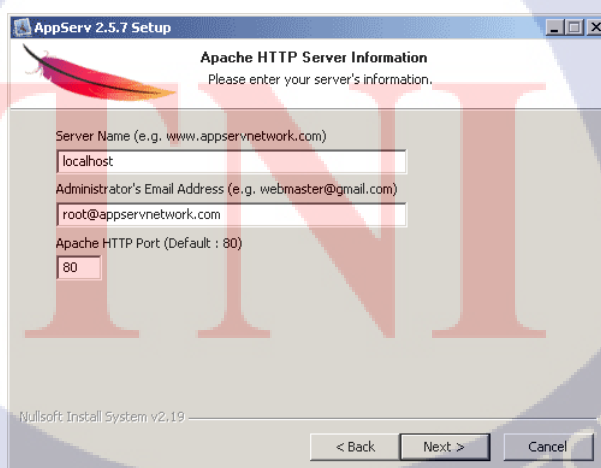
ภาพที่ 3.7 เลือกปลายทางการติดตั้งโปรแกรม AppServ

4) เลือก Package Components ที่ต้องการติดตั้ง โดยค่าเริ่มต้นนั้นจะให้เลือกลงทุก Package แต่หากว่าผู้ใช้งานต้องการเลือกเฉพาะบาง Package ก็สามารถเลือกตามข้อที่ต้องการออก



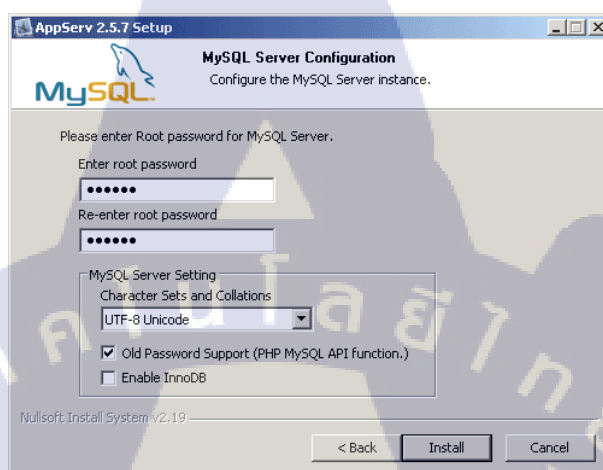
ภาพที่ 3.8 เลือก Package Components ที่ต้องการติดตั้ง

5) กำหนดค่าคอนฟิกของ Apache Web Server มีอยู่ด้วยกันทั้งหมด 3 ส่วน ได้แก่ Server Name คือช่องสำหรับป้อนข้อมูลชื่อ Web Server ของท่าน, Admin Email คือช่องสำหรับป้อนข้อมูล อีเมลล์ผู้ดูแลระบบ, HTTP Port คือช่องสำหรับระบุ Port ที่จะเรียกใช้งาน Apache Web Server



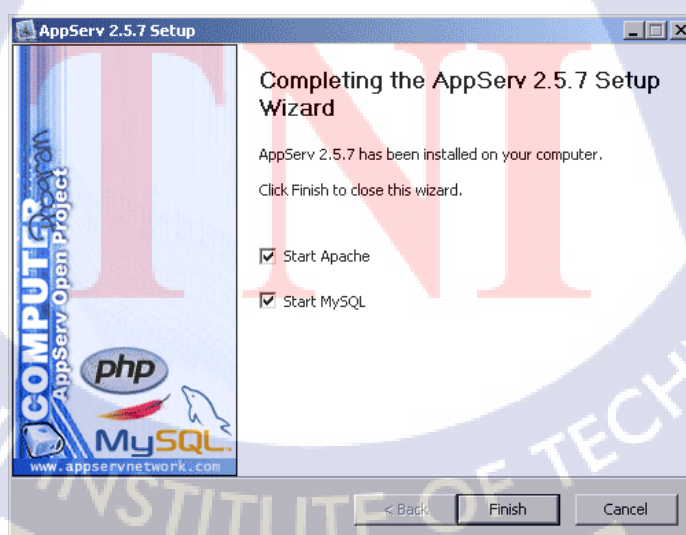
ภาพที่ 3.9 แสดงการกำหนดค่าคอนฟิกค่า Apache Web Server

6) กำหนดค่า Root Password คือช่องสำหรับป้อน รหัสผ่านการใช้งานฐานข้อมูลของ Root Character Sets ใช้ในการกำหนดค่าระบบภาษาที่ใช้ในการจัดเก็บฐานข้อมูล



ภาพที่ 3.10 แสดงการกำหนดค่าคอนฟิกของ MySQL Database

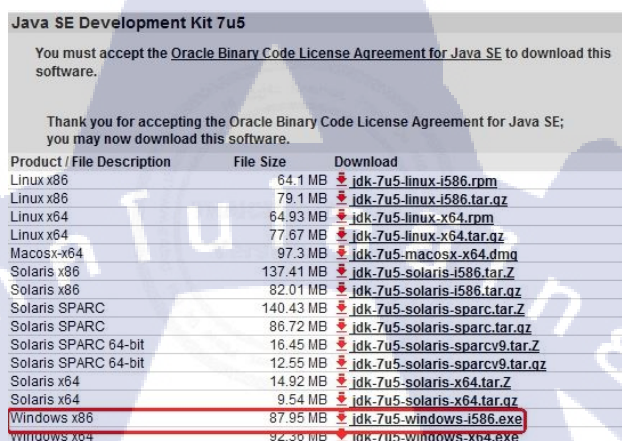
7) สิ้นสุดขั้นตอนการติดตั้งโปรแกรม AppServ สำหรับขั้นตอนสุดท้ายนี้ จะมีให้เลือกว่าต้องการสั่งให้มีการรัน Apache และ MySQL ทันทีหรือไม่ จากนั้นกดปุ่ม Finish เพื่อเสร็จสิ้นการติดตั้งโปรแกรม AppServ



ภาพที่ 3.11 แสดงหน้าจอขั้นตอนสิ้นสุดการติดตั้งโปรแกรม AppServ

3.2.2.3 วิธีการติดตั้ง Java Development Kit : JDK

1) ดาวน์โหลดโปรแกรมได้ที่ <http://www.oracle.com/technetwork/java/javase/downloads/index.html> โดยเลือกดาวน์โหลดตัวที่เป็น JDK เมื่อดาวน์โหลดเสร็จแล้ว ให้ดับเบิลคลิกเพื่อทำการติดตั้งโปรแกรม



Product / File Description	File Size	Download
Linux x86	64.1 MB	jdk-7u5-linux-i586.rpm
Linux x86	79.1 MB	jdk-7u5-linux-i586.tar.gz
Linux x64	64.93 MB	jdk-7u5-linux-x64.rpm
Linux x64	77.67 MB	jdk-7u5-linux-x64.tar.gz
Macosx-x64	97.3 MB	jdk-7u5-macosx-x64.dmg
Solaris x86	137.41 MB	jdk-7u5-solaris-i586.tar.Z
Solaris x86	82.01 MB	jdk-7u5-solaris-i586.tar.gz
Solaris SPARC	140.43 MB	jdk-7u5-solaris-sparc.tar.Z
Solaris SPARC	86.72 MB	jdk-7u5-solaris-sparc.tar.gz
Solaris SPARC 64-bit	16.45 MB	jdk-7u5-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	12.55 MB	jdk-7u5-solaris-sparcv9.tar.gz
Solaris x64	14.92 MB	jdk-7u5-solaris-x64.tar.Z
Solaris x64	9.54 MB	jdk-7u5-solaris-x64.tar.gz
Windows x86	87.95 MB	jdk-7u5-windows-i586.exe
Windows x64	92.36 MB	jdk-7u5-windows-x64.exe

ภาพที่ 3.12 ขั้นตอนการติดตั้ง JDK (1)

2) คลิก Next



ภาพที่ 3.13 ขั้นตอนการติดตั้ง JDK (2)

3) คลิก Next อีกครั้ง



ภาพที่ 3.14 ขั้นตอนการติดตั้ง JDK (3)

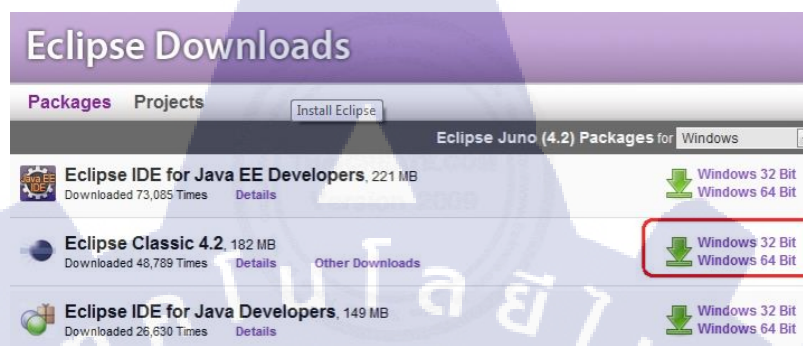
4) เลือก Destination ที่ต้องการจะติดตั้งโปรแกรมลงไป เสร็จแล้วกด Next โปรแกรมจะเริ่มทำการติดตั้ง



ภาพที่ 3.15 ขั้นตอนการติดตั้ง JDK (4)

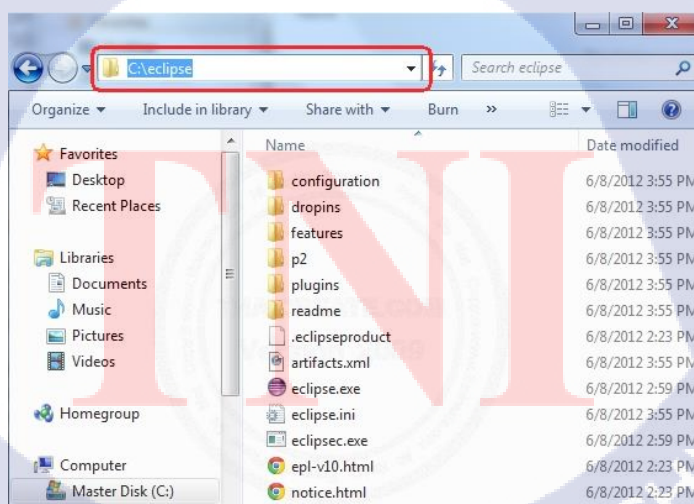
3.2.2.4 วิธีการติดตั้งโปรแกรม Eclipse

1) ดาวน์โหลดโปรแกรมที่ <http://www.eclipse.org/downloads/> โดยเลือกเวอร์ชันที่เป็น Eclipse Classic



ภาพที่ 3.16 ขั้นตอนการติดตั้งโปรแกรม Eclipse (1)

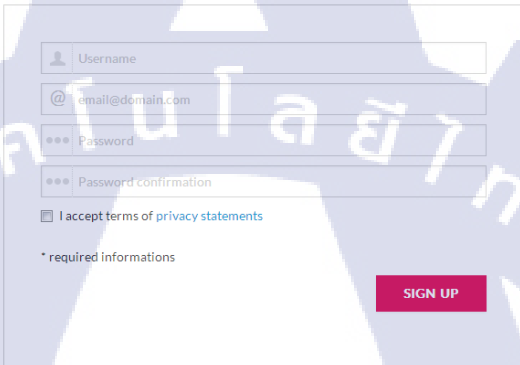
2) หลังจากดาวน์โหลดโปรแกรมเสร็จแล้ว จะได้ไฟล์ Zip มา ให้นำไฟล์นี้ไปแตกไฟล์ในที่ที่ต้องการติดตั้งโปรแกรมไว้ได้เลย (กรณีนี้เลือกติดตั้งไว้ที่ C:\Eclipse) เมื่อแตกไฟล์เสร็จแล้ว สามารถเปิดโปรแกรมใช้งานได้เลยโดยคลิกที่ Eclipse.exe



ภาพที่ 3.17 ขั้นตอนการติดตั้งโปรแกรม Eclipse (2)

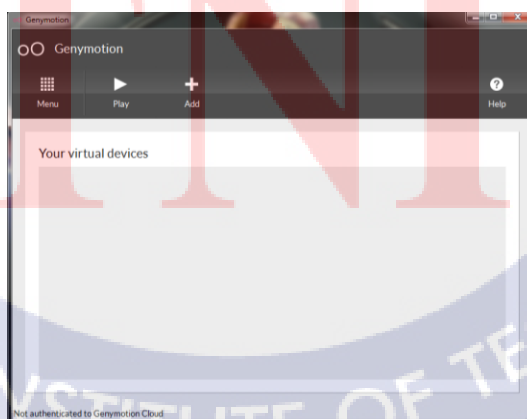
3.2.2.5 วิธีการติดตั้งโปรแกรม GenyMotion

1) สมัครสมาชิกกับทาง GenyMotion ที่ <https://cloud.genymotion.com/page/customer/login/?next=/page/launchpad/download/> โดยกรอก Username, E-mail, Password , และ Confirmation Password ในช่อง Sign Up เมื่อกรอกเสร็จแล้วให้กดปุ่ม Sign Up ระบบจะส่ง E-mail ยืนยันมาให้ใน E-mail กรอกลงไป



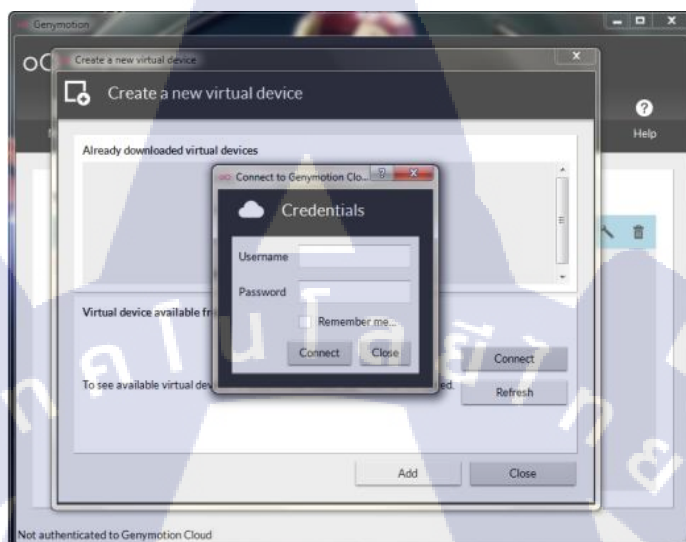
ภาพที่ 3.18 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (1)

2) ดาวน์โหลดโปรแกรมได้ที่ <https://cloud.genymotion.com/page/launchpad/download/> เมื่อดาวน์โหลดโปรแกรมและติดตั้งโปรแกรมเสร็จแล้ว ให้คลิกไปที่ไอคอนของ GenyMotion เพื่อทำการ SetUp Emulator



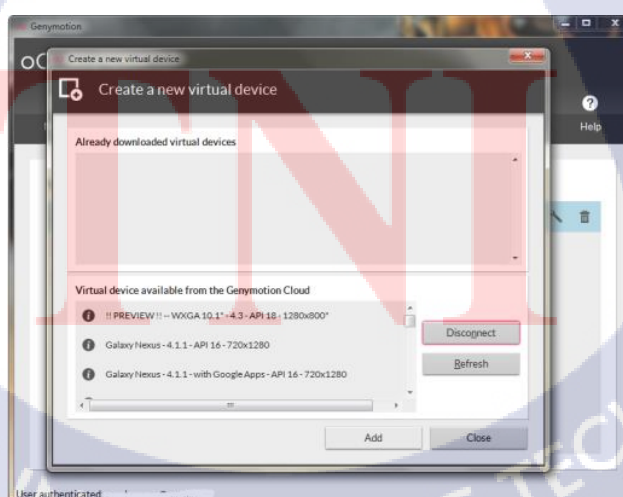
ภาพที่ 3.19 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (2)

3) จากนั้นคลิกไปที่ปุ่ม Add เพื่อทำการดาวน์โหลด Emulator ของโทรศัพท์มือถือที่ท่านต้องการ โดยก่อนจะดาวน์โหลดจะต้องกรอก Username และ password ที่ได้สมัครในขั้นตอนแรก



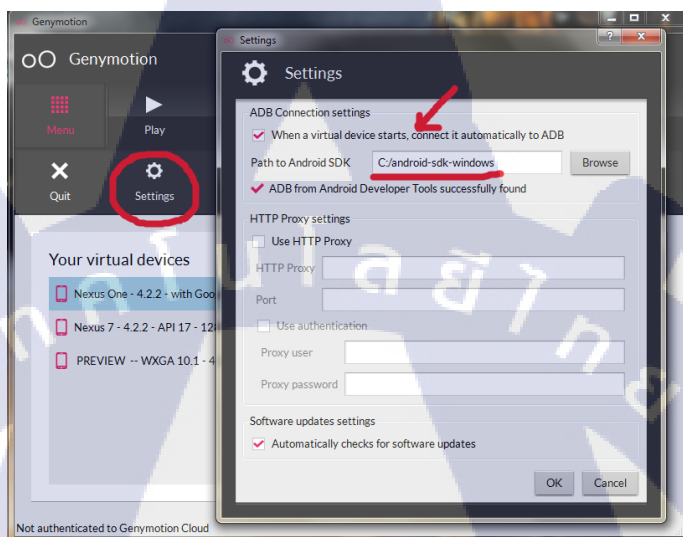
ภาพที่ 3.20 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (3)

4) จากนั้นเลือกรุ่นของ Emulator ของโทรศัพท์มือถือที่ต้องการ แล้วคลิกปุ่ม Add โปรแกรมจะทำการดาวน์โหลด Emulator



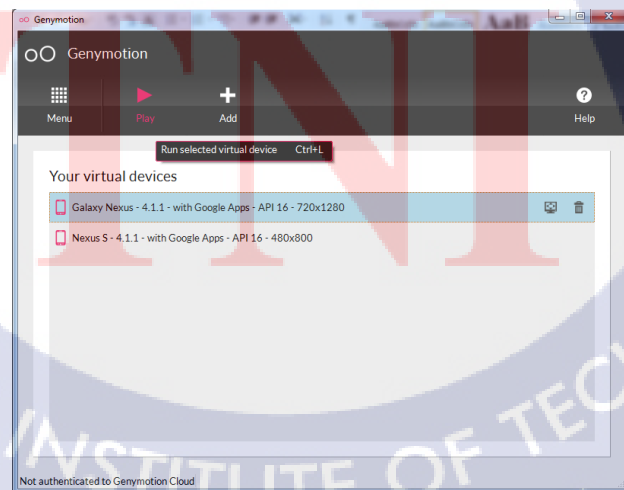
ภาพที่ 3.21 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (4)

5) เมื่อดาวน์โหลด Emulator เสร็จแล้ว ชื่อของ Emulator จะปรากฏขึ้นมา แต่ก่อนจะใช้งาน Emulator ต้องตั้งค่าให้ Emulator เชื่อมต่อกับ Eclipse โดยคลิกที่ Menu > Settings แล้วเปลี่ยน Path ให้ตรงกับตำแหน่งที่เก็บไฟล์ Android-sdk-windows ไว้ (กรณีนี้เป็น C:\ android-sdk-windows)



ภาพที่ 3.22 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (5)

6) เปิดการใช้งาน Emulator โดยคลิกเลือก Emulator ที่ต้องการ จากนั้นคลิกปุ่ม Play



ภาพที่ 3.23 ขั้นตอนการติดตั้งโปรแกรม GenyMotion (6)

3.2.3 การเก็บรวบรวมข้อมูล

เก็บและรวบรวมความต้องการของผู้ใช้จากนักศึกษาคณะเทคโนโลยีสารสนเทศ
 ชั้นปีที่ 3 และ 4 ของสถาบันเทคโนโลยีไทย-ญี่ปุ่นที่ใช้โทรศัพท์มือถือหรือ SmartPhone
 ระบบปฏิบัติการ Android และค้นคว้ารวบรวมข้อมูลในการดำเนินการโครงการแอปพลิเคชันเกม
 เวทีทองในส่วนของเว็บไซต์จาก Internet และค้นคว้ารวบรวมข้อมูลในส่วนของแอปพลิเคชันจาก
 Internet กับ หนังสือคู่มือเขียนแอป Android ฉบับรวมโค้ด ของ พร้อมเลิศ หล่อวิจิตร



ภาพที่ 3.24 หนังสือคู่มือเขียนแอป Android ฉบับรวมโค้ด

3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงานหรือโครงการ

3.3.1 งานที่ปฏิบัติ

งานที่ปฏิบัติในระหว่างการสหกิจศึกษาทุกชิ้นงานจะเป็นงานที่พี่ที่ปรึกษา มอบหมายมาให้ เพื่อฝึกฝนทักษะในด้านต่างๆให้เพิ่มพูนมากขึ้น, รู้จักเครื่องมือและเทคนิคใหม่ๆ เพิ่มมากขึ้น, และสร้างไอเดียในการทำโครงการโปรเจก (ขั้นตอนในการดำเนินงานแต่ละขั้น ข้าพเจ้าจะขอละไว้ในหัวข้อนี้ แต่จะนำไปกล่าวในบทที่ 4 แทน) โดยแอปพลิเคชันที่ข้าพเจ้าได้รับ มอบหมายให้ฝึกฝนมีทั้งหมด 3 ชิ้น คือ Sanook Weather Application, SQLite Test Application, และ jsTwit Application

3.3.1.1) Sanook Weather Application

แอปพลิเคชันนี้เป็นแอปพลิเคชันที่ฝึกฝนการดึงข้อมูล JSON มากใช้งานใน Android, ฝึกการทำ SharedPreferences, ฝึกการทำ Custom ListView, ฝึกการแสดงผลจาก URL, และฝึกการทำ Animation ให้กับ ListView โดยจะแสดงผลพยากรณ์อากาศให้กับผู้ใช้งาน ซึ่งจะมีทั้งส่วนที่เป็นแอปพลิเคชันและส่วนที่เป็นWidget โดยข้อมูลพยากรณ์อากาศที่แสดงจะแบ่ง ออกเป็น 2 หมวด คือ สภาพอากาศของวันปัจจุบันและสภาพอากาศ ในสัปดาห์นี้ซึ่งผู้ใช้งานสามารถ เลือกประเภทของหน่วยอุณหภูมิได้ว่าจะดูได้ องศาเซลเซียสหรือองศาฟาเรนไฮด์



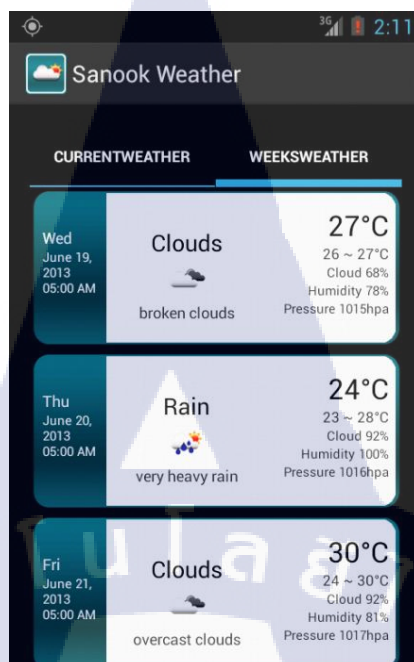
ภาพที่ 3.25 ไอคอนของแอปพลิเคชัน Sanook Weather



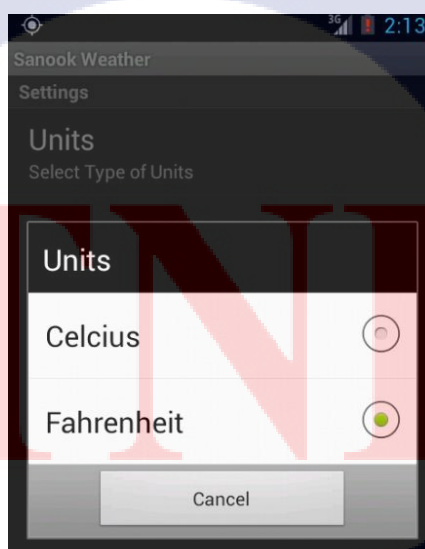
ภาพที่ 3.26 Widget ของแอปพลิเคชัน Sanook Weather



ภาพที่ 3.27 Interface ของแอปพลิเคชัน Sanook Weather
ในหน้าแสดงสภาพอากาศวันปัจจุบัน



ภาพที่ 3.28 Interface ของแอปพลิเคชัน Sanook Weather
ในหน้าแสดงสภาพอากาศในสัปดาห์นี้



ภาพที่ 3.29 Interface ของแอปพลิเคชัน Sanook Weather
ในหน้าตั้งค่าหน่วยของอุณหภูมิที่จะให้แสดง

3.3.1.2) Test SQLite Application

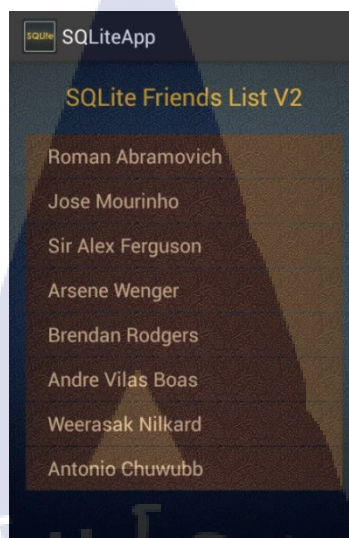
แอปพลิเคชันนี้เป็นแอปพลิเคชันที่ทำขึ้นเพื่อฝึกฝนการใช้งาน SQLite, การจัดการ Database ใน Android, และการทำ SplashScreen ให้กับแอปพลิเคชัน โดยแอปพลิเคชันนี้จะให้ผู้ใช้บันทึกข้อมูลของเพื่อนเข้าไป จากนั้นแอปจะแสดงรายชื่อของเพื่อนออกมาในรูปแบบของ List ซึ่งหากผู้ใช้กดไปที่ชื่อของเพื่อนคนใด ก็จะมี Dialog Box แสดงข้อมูลของเพื่อนคนนั้น และมีปุ่มให้กดเพื่อแก้ไขข้อมูลของเพื่อนคนนั้นและเพื่อลบ ข้อมูลของเพื่อนคนนั้นออกจาก List



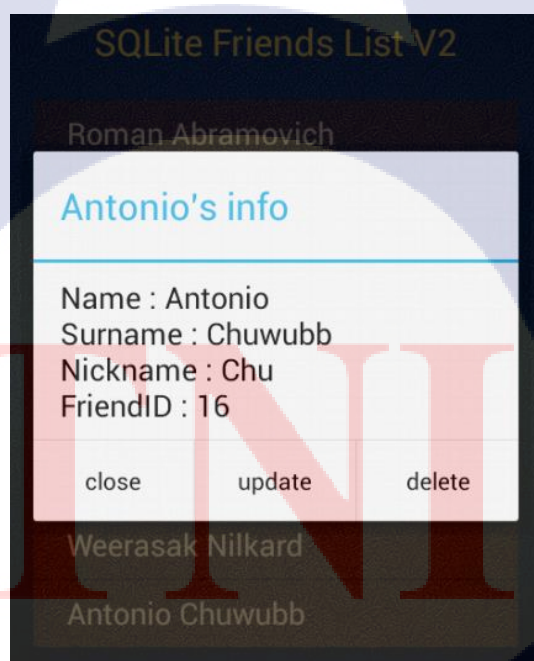
ภาพที่ 3.30 ไอคอนของแอปพลิเคชัน SQLite Test



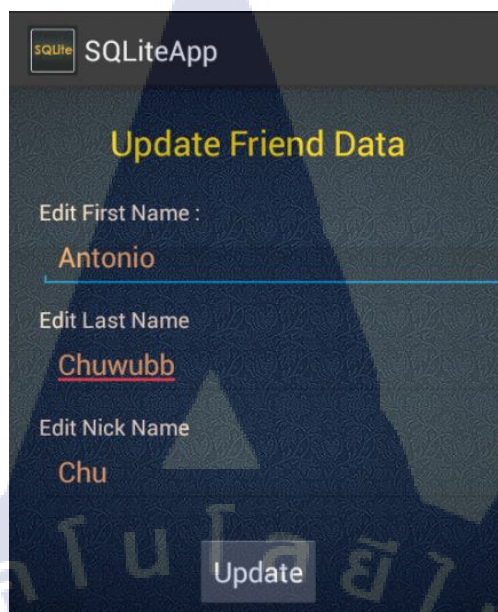
ภาพที่ 3.31 SplashScreen ของแอปพลิเคชัน SQLite



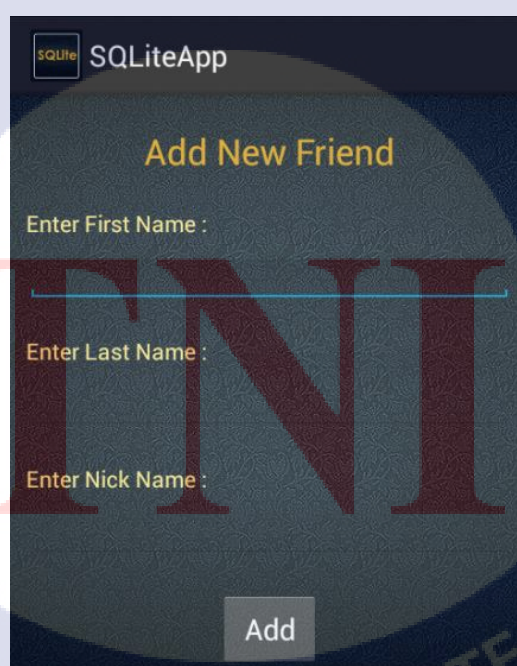
ภาพที่ 3.32 Interface หน้าแรก (แสดงรายชื่อเพื่อน) ของแอปพลิเคชัน SQLite



ภาพที่ 3.33 Interface เมื่อกดไปที่รายชื่อเพื่อนในแอปพลิเคชัน SQLite



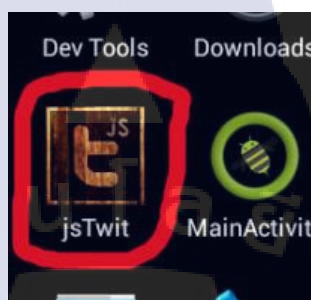
ภาพที่ 3.34 Interface หน้าแก้ไขข้อมูลของเพื่อนในแอปพลิเคชัน SQLite



ภาพที่ 3.35 Interface หน้าเพิ่มเพื่อนใหม่ในแอปพลิเคชัน SQLite

3.3.1.3) jsTwit Application

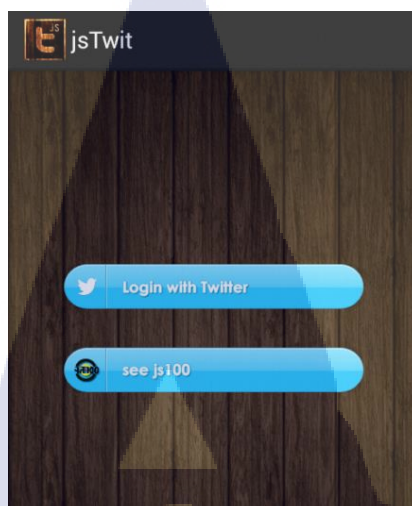
แอปพลิเคชันนี้เป็นแอปพลิเคชันที่ฝึกฝนการใช้งาน Twitter Api 1.1, การตรวจสอบ Internet Connection, การทำ Pull to Refresh, การทำ Pull to Load More, และการเก็บข้อมูลที่ดาวน์โหลดมาจาก Internet ไปไว้ใน SQLite โดยในหน้าแรกของแอปพลิเคชันนี้จะมี 2 ปุ่มให้เลือก คือ Login เข้าสู่ Twitter และดูข้อมูลจส.100



ภาพที่ 3.36 ไอคอนของแอปพลิเคชัน jsTwit



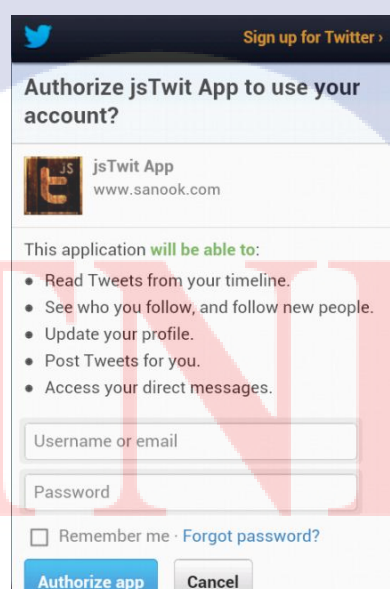
ภาพที่ 3.37 SplashScreen ของแอปพลิเคชัน jsTwit



ภาพที่ 3.38 Interface หน้าแรกของแอปพลิเคชัน jsTwit

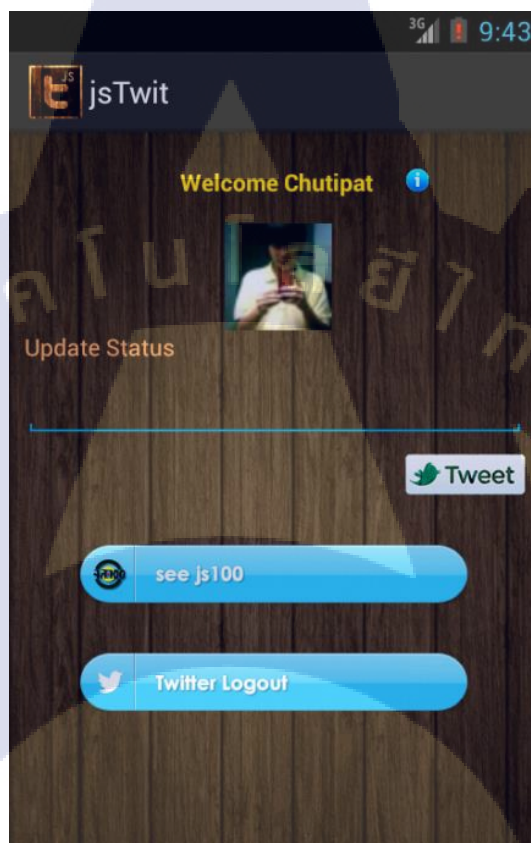
1) login เข้าสู่ twitter

เมื่อผู้ใช้กดปุ่มเข้ามาแล้ว จะมีช่องให้ผู้ใช้อกรอก Username และ Password ของ Twitter เพื่อเข้าใช้งาน



ภาพที่ 3.39 หน้าจอ login เข้าสู่ twitter ของแอปพลิเคชัน jsTwit

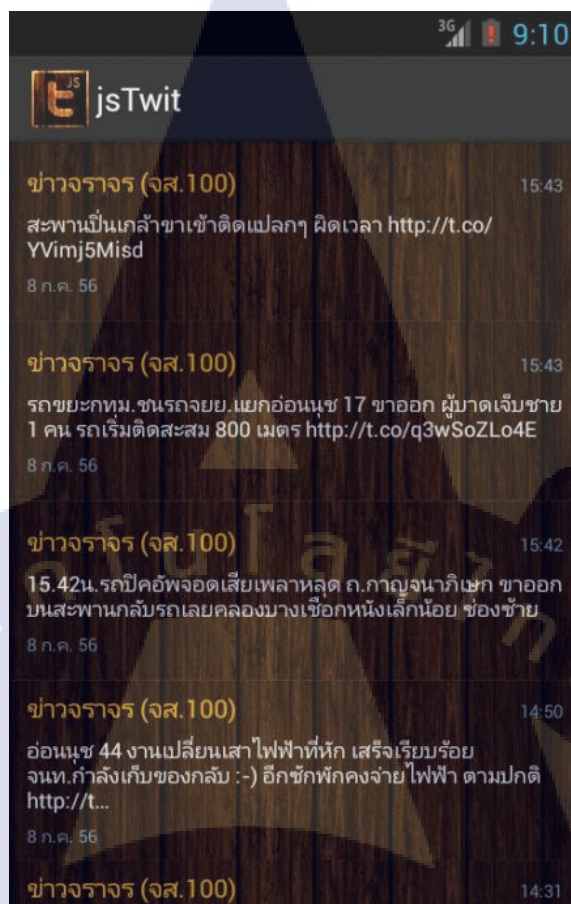
หลังจากกรอกข้อมูลแล้วจะพบกับ Interface ดังภาพที่ 3.16 ซึ่งจะมีข้อความต้อนรับและรูปโปรไฟล์ของผู้ใช้ รวมถึงสามารถ Tweet ข้อความไปยัง Twitter ผ่านแอปพลิเคชัน jsTwit, ดูข้อมูลจส.100, logout ออกจาก twitter, และ, ดูข้อมูลใน Twitter ของผู้ใช้ เช่น ScreenName , จำนวนข้อความที่ Tweet, จำนวนคน Follow, ข้อความล่าสุดที่ Tweet เป็นต้น



ภาพที่ 3.40 Interface หลังจาก login

2) See js100

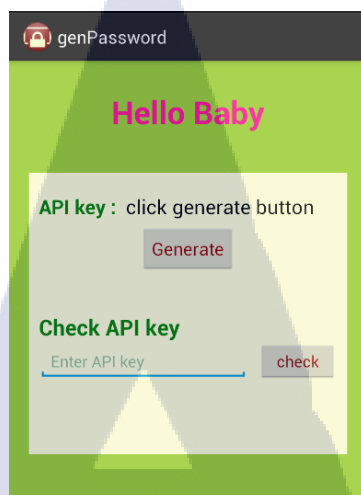
หน้านี้เป็นหน้าที่แสดงข้อมูลข่าวจส.100ที่ได้รับมาจาก Feed API ของ Sanook! ข่าวที่รับมาจะแสดงในรูปแบบของ List ซึ่งเมื่อได้รับข่าวมาแล้ว ระบบจะเก็บข้อมูลข่าวลงใน SQLite หากโทรศัพท์มือถือไม่ได้เชื่อมต่อ Internet ระบบจะดึงข้อมูลข่าวที่เก็บไว้ใน SQLite ออกมาแสดง แต่ถ้าโทรศัพท์มือถือได้เชื่อมต่อ Internet ระบบก็จะไปดาวน์โหลดข่าวจาก Feed มาแสดง



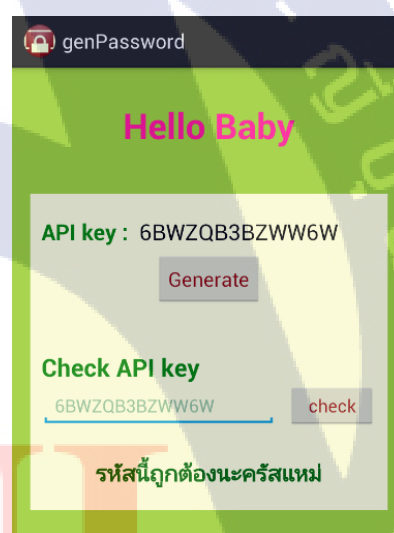
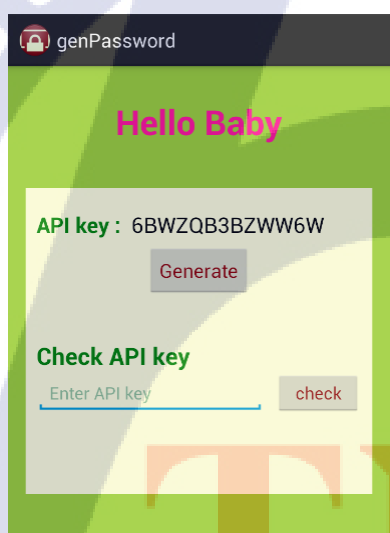
ภาพที่ 3.41 Interface หน้าแสดงข้อมูลข่าว จส.100

3.3.1.4) genPassword Application

แอปพลิเคชันนี้เป็นแอปพลิเคชันที่ฝึกฝนการ generate API Key ให้กับระบบ ซึ่งการทำงานของแอปพลิเคชันนี้คือสามารถ generate API Key ขึ้นมาอัตโนมัติ และสามารถนำ API Key ไปตรวจสอบได้ว่าเป็น API Key ที่ถูกต้องหรือไม่ และ API Key นี้ หมดอายุการใช้งานหรือยัง โดย API Key ที่ถูก generate ขึ้นมา จะถูกเข้ารหัสก่อน เพื่อป้องกันการ Hack จากภายนอก



ภาพที่ 3.42 Interface ของโปรแกรม genPassword



ภาพที่ 3.43 Interface เมื่อกดปุ่ม generate ภาพที่ 3.44 Interface เมื่อนำAPI Key ไปตรวจสอบ

3.3.2 โครงการแอปพลิเคชันเกมเวทีทอง

แอปพลิเคชันนี้ได้นำช่วงปริศนาจิ๊กมกในรายการเวทีทองมาทำให้เป็นแอปพลิเคชัน ลักษณะของช่วงปริศนาจิ๊กมกคือผู้เข้าแข่งขันจะต้องทายคำจากภาพให้ถูกต้องภายในระยะเวลาที่กำหนด ซึ่งข้าพเจ้าได้ปรับแต่งวิธีการเล่นบางอย่างให้เหมาะสมกับการเป็นแอปพลิเคชันบนโทรศัพท์มือถือและเพื่อเพิ่มความสนุกสนานให้มากยิ่งขึ้น โดยโครงการแอปพลิเคชันเกมเวทีทองนี้ประกอบด้วย 2 ส่วน คือ ส่วนของเว็บหลังบ้าน และส่วนของแอปพลิเคชัน

ในส่วนของเว็บหลังบ้านจัดทำขึ้นเพื่อป้อนข้อมูลในรูปแบบ Feed JSON ให้กับแอปพลิเคชันได้นำเอาข้อมูลไปใช้ได้แก่ ภาพด่านเกม, ชื่อของภาพนั้น, และคำใบ้ของภาพนั้น โดยสามารถเพิ่มด่านใหม่, ลบด่านออก, และแก้ไขข้อมูลของด่านภายในเว็บหลังบ้านได้ เมื่อมีการเพิ่มด่านใหม่ในเว็บหลังบ้าน จะมีNotification แสดงขึ้นมาในโทรศัพท์มือถือของผู้ใช้ เพื่อบอกว่าขณะนี้มีการเพิ่มด่านเกมด่านใหม่เข้ามาแล้ว

ในส่วนของแอปพลิเคชันเกมเวทีทองจะประกอบด้วยเกม 2 หมวด คือ โหมดคลาสสิกเกม และ โหมดโกสต์มาสเตอร์ ซึ่งทั้ง 2 หมวด มีความแตกต่างกันดังนี้

- โหมดคลาสสิกเกม
 - 1) คำตอบของโหมดนี้จะมี 2 คำ
 - 2) หากคิดคำตอบไม่ออก สามารถข้ามไปภาพถัดไปได้เลย เพียงแค่สไลด์ภาพจากด้านขวาไปด้านซ้าย หรือหากอยากย้อนกลับไปใหม่ก็เพียงแค่สไลด์ภาพจากด้านซ้ายไปด้านขวา
 - 3) มีตัวจับเวลา 1 ตัว คือ ตัวจับเวลารวม
 - 4) มีตัวช่วย 1 ตัว คือ คำใบ้ ซึ่งมีการกำหนดจำนวนครั้งในการใช้



ภาพที่ 3.45 รูปภาพปุ่มของโหมดคลาสสิกเกม

• โหมดโกสต์มาสเตอร์ 1) คำตอบของโหมดนี้จะมี 3 คำ

2) ไม่สามารถสไลด์ข้าม/ย้อนกลับได้เหมือนกับโหมดคลาสสิก

เกม

3) มีตัวจับเวลา 2 ตัว คือ ตัวจับเวลารวม และตัวจับเวลา 10 วินาที

4) ภาพในโหมดนี้จะมีการจับเวลา 10 วินาที หากยังทายคำไม่ถูกใน 10 วินาที ภาพจะเปลี่ยนเป็นภาพถัดไปอัตโนมัติ

5) ตัวช่วยในโหมดนี้มี 2 ตัว คือ คำใบ้ และ ปุ่มข้าม ซึ่งมีการกำหนดจำนวนครั้งในการใช้ตัวช่วย

6) ไม่มีการเรียงลำดับของภาพในเกม คือจะRandomภาพทุกครั้งที่เข้ามาเล่นใหม่

7) เมื่อทายภาพถูก สามารถแชร์ให้เพื่อนในFacebookลองทายภาพนั้นดูได้โดยกดปุ่มแชร์ ซึ่งภาพที่แชร์นั้นจะไปแสดงอยู่ในหน้าWallของผู้ใช้



ภาพที่ 3.46 รูปภาพปุ่มของโหมดโกสต์มาสเตอร์

วิธีการเล่นเกมนี้ ผู้ใช้จะต้องทายคำจากภาพที่ปรากฏขึ้นมาให้ถูกต้องภายในระยะเวลาที่กำหนด โดยจำนวนคำของคำตอบของภาพจะขึ้นอยู่กับโหมดซึ่งได้กล่าวไปแล้ว การทายคำตอบนั้นให้ผู้ใช้กดปุ่มคำตอบที่ปรากฏอยู่ทางด้านขวาของจอ โดยกดปุ่มแรกเป็นคำตอบของคำแรก กดปุ่มที่ 2 เป็นคำตอบของคำที่สอง เมื่อปุ่มคำตอบถูกกดแล้ว ปุ่มคำตอบนั้นจะหายไปและคำที่อยู่ในปุ่มคำตอบนั้นจะปรากฏอยู่ในช่องคำตอบบริเวณด้านบนของภาพ หากผู้ใช้กดปุ่มคำตอบไปแล้วแต่เกิดอยากเปลี่ยนคำตอบนั้นก็สามารถทำได้โดยกดไปที่ช่องคำตอบของคำที่ต้องการจะเปลี่ยน เมื่อกดไปที่ช่องคำตอบแล้วปุ่มคำตอบนั้นจะปรากฏกลับมาให้เห็นอีกครั้งหนึ่ง ซึ่งผู้ใช้สามารถอ่านวิธีการเล่นได้ในหน้าวิธีการเล่นของแอปพลิเคชัน

ในแอปพลิเคชันเกมเวทีทองนี้ มี Feature ให้ผู้ใช้สามารถดาวน์โหลดด่านเกมใหม่ๆ ได้ โดยจะมีปุ่มดาวน์โหลดอยู่ในหน้าแรกของแอปพลิเคชัน เมื่อผู้ใช้กดเข้าไปจะเข้าสู่หน้าดาวน์โหลดด่านซึ่งหากมีด่านใหม่เข้ามา Interface จะแสดงด่านใหม่ในรูปแบบของ List รายการ หากไม่มีด่านใหม่ Interface จะแสดงข้อความว่าข้อมูลของคุณอัปเดตล่าสุดแล้ว โดยในการดาวน์โหลดด่านใหม่นั้น ผู้ใช้จำเป็นต้องเชื่อมต่อ Internet ก่อนจึงจะสามารถดาวน์โหลดด่านใหม่ได้ และหากมีด่านใหม่อัปเดตเข้ามาในระบบจะมี Notification แสดงขึ้นมาในโทรศัพท์มือถือของผู้ใช้ว่าขณะนี้มีการดาวน์โหลดด่านใหม่เข้ามาแล้ว ซึ่ง Notification แจ้งเตือนนี้จะแสดงขึ้นมาได้ก็ต่อเมื่อโทรศัพท์มือถือของผู้ใช้มีการเชื่อมต่อ Internet ในขณะที่ระบบกำลังอัปเดตด่านใหม่ และผู้ใช้ตั้งค่าให้เปิดการใช้งานการแจ้งเตือนในแอปพลิเคชัน

ผู้ใช้สามารถตั้งค่าเสียงในเกมและตั้งค่าการแจ้งเตือน (Notification) ได้ โดยกดไปที่ปุ่มตั้งค่าเกมในหน้าแรกของแอปพลิเคชัน เมื่อผู้ใช้กดเข้ามาแล้ว หากอยากเปิดการทำงานในส่วนใดให้ติ๊ก ถูกที่ด้านหลังของฟังก์ชันนั้น เช่น อยากให้มีเสียงในขณะที่เล่นเกม ให้ผู้ใช้ติ๊กถูกที่ช่องด้านหลังของฟังก์ชันเสียงดนตรีประกอบ แต่หากผู้ใช้อยากตั้งค่าให้เปิดการแจ้งเตือน ผู้ใช้ต้องเชื่อมต่อ Internet ก่อน จึงสามารถเปิดใช้งานการแจ้งเตือนได้ เนื่องจากระบบจะส่ง Notification แจ้งเตือนมาไม่ได้ หากไม่มีการเชื่อมต่อ Internet

บทที่ 4

ผลการดำเนินงาน การวิเคราะห์และสรุปผลต่างๆ

4.1 ขั้นตอนและผลการดำเนินงาน

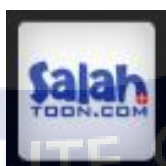
4.1.1) ศึกษาและทำความเข้าใจเกี่ยวกับรูปแบบของช่วงปริศนาจ๊กมกในรายการเวทิตอง

การศึกษาและทำความเข้าใจเกี่ยวกับรูปแบบการดำเนินรายการของช่วงปริศนาจ๊กมกในรายการเวทิตองนั้น ศึกษาเพื่อให้สามารถกำหนดขอบเขตของการทำงาน และทราบถึงขั้นตอนการเล่นเกมนของผู้เข้าแข่งขัน และสามารถออกแบบโครงสร้างของแอปพลิเคชันได้อย่างเหมาะสมซึ่งเป็นส่วนช่วยในการพัฒนาแอปพลิเคชันได้อย่างมีประสิทธิภาพ

ส่วนของข้อมูลจะต้องมีการประเมินข้อมูลที่จะใช้ในการพัฒนาแอปพลิเคชัน เช่น ข้อมูลใดบ้างที่จำเป็นต้องใช้ในการเล่นเกม, ข้อมูลใดเหมาะสมในการนำมาใช้ในแอปพลิเคชัน, และข้อมูลที่น่ามาจะถูกจัดเก็บอยู่ในรูปแบบใดให้สามารถเรียกใช้ข้อมูลได้อย่างถูกต้องรวดเร็วที่สุดเพื่อป้องกัน ไม่ให้ผู้ใดเกิดความรำคาญ รวมถึงต้องศึกษาลักษณะของแอปพลิเคชันเป็นลักษณะของเกมทายคำจากภาพเพื่อนำไอเดียมาประยุกต์ใช้ในการพัฒนาแอปพลิเคชันเกมเวทิตอง

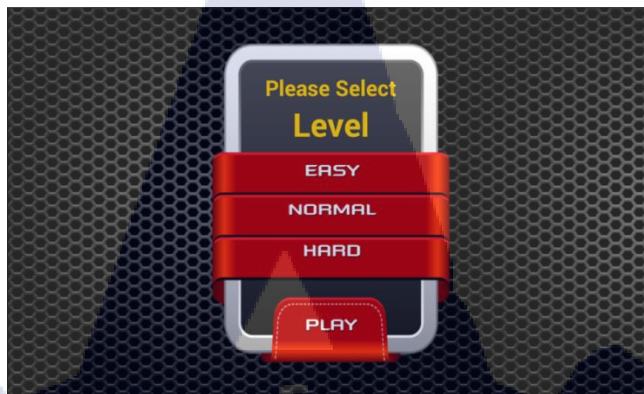
4.1.2) ศึกษาแอปพลิเคชันเกมทายคำจากภาพที่มีอยู่แล้วใน Play Store

การศึกษาแอปพลิเคชันประเภทเกมทายคำศัพท์จากภาพที่มีอยู่แล้วใน Play Store เพื่อให้สามารถกำหนดขอบเขตของการทำงาน, ทราบถึงกระบวนการทำงานของระบบเกม, และทราบถึงสิ่งที่ควรเพิ่มเติมเพื่อความสนุกสนานของระบบเกม ทำให้สามารถออกแบบโครงสร้างและ Interface ของแอปพลิเคชันได้อย่างเหมาะสม ซึ่งเป็นส่วนช่วยในการพัฒนาแอปพลิเคชันให้สามารถใช้งานได้มีประสิทธิภาพ โดยในครั้งนี้ข้าพเจ้าได้นำแอปพลิเคชัน SalahToonLite มาใช้ในการศึกษา



ภาพที่ 4.1 แอปพลิเคชัน SalahToonLite

เมื่อกดเข้ามาแล้ว จะพบกับ Interface ดังภาพที่ 3.19 ซึ่งมี 3 โหมดให้เลือก คือ ง่าย, ปานกลาง, และยาก



ภาพที่ 4.2 Interface หน้าแรก

โดยคำตอบของโหมดง่ายและปานกลางจะมี 2 พยางค์ ส่วนคำตอบของโหมดยากจะมี 3 พยางค์ ในส่วนของระบบเกม เริ่มแรกผู้ใช้งานต้องทายคำศัพท์จากภาพ โดยจะมีปุ่มคำตอบอยู่ด้านข้าง 6 ปุ่ม หากผู้ใช้งานต้องการตอบคำถามก็แค่กดปุ่มคำตอบตามลำดับพยางค์ เช่น คำตอบมี 2 พยางค์ ก็กดพยางค์แรกก่อน แล้วค่อยกดปุ่มพยางค์ที่ 2 ซึ่งเมื่อกดปุ่มใดไปแล้ว ปุ่มคำตอบนั้นก็หายไป และข้อความในปุ่มนั้นจะไปปรากฏอยู่ที่ช่องคำตอบพยางค์นั้นๆ หากผู้ใช้งานกดปุ่มพยางค์แรกไปแล้วแต่ต้องการเปลี่ยนคำตอบใหม่ ก็สามารถทำได้ โดยกดไปที่ช่องคำตอบพยางค์แรก ปุ่มที่กดไปแล้วก็จะปรากฏกลับมาดังเดิม

วิธีเอาชนะเกมนี้ ผู้ใช้จะต้องทายคำศัพท์ให้ถูกต้องทั้งหมดภายในเวลาที่กำหนด ซึ่งตัวเกมจะมีเวลานับถอยหลังอยู่บริเวณมุมด้านซ้ายล่างของจอ หากตอบครบทุกภาพแล้ว

ในส่วนของตัวช่วย ตัวช่วยในเกมนี้มีอยู่ 2 ตัว คือ ข้ามไปรูปถัดไป กับ คำใบ้ ซึ่งจะอยู่บริเวณมุมบนขวาของหน้าจอและสามารถกดกี่ครั้งก็ได้



ภาพที่ 4.3 Interface หน้าเล่นเกม

ข้อเสียของแอปพลิเคชันนี้คือ สามารถกดปุ่มคำใบ้ได้ไม่จำกัดครั้ง, หากทนายภาพ ไม่ถูกสามารถข้ามได้เรื่อยๆ, ด้านเกมไม่มีการ Random ทำให้เกิดความน่าเบื่อ, ไม่มี Feature ให้ คาวนไหลด่านใหม่ๆ, เวลา กด Back 2 ครั้งเพื่อออกจากโปรแกรม ยังมี Force Code อยู่

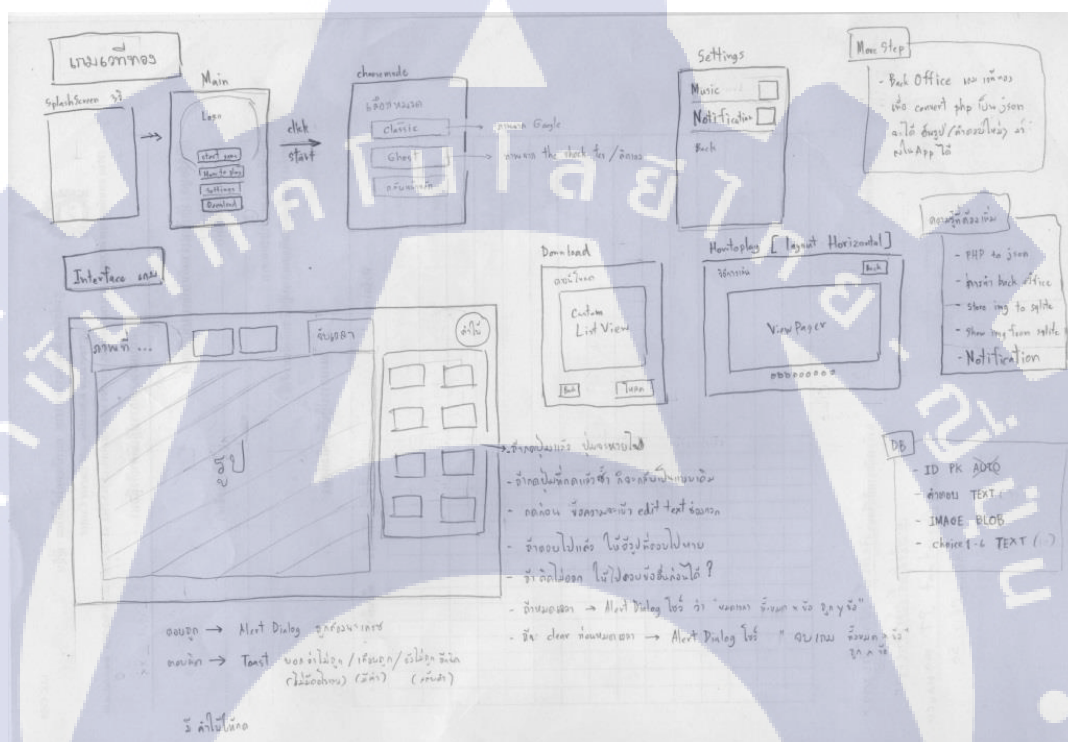
4.1.3) วิเคราะห์ความต้องการของผู้ใช้งาน

การวิเคราะห์ความต้องการของผู้ใช้งานเป็นส่วนหนึ่งของการกำหนดขอบเขตของ ข้อมูลที่ต้องการจะใช้งานและรวบรวมความต้องการของผู้ใช้งานเพื่อนำมาพัฒนาแอปพลิเคชันเกม เวทีทอง ส่งผลให้มีเป้าหมายในการพัฒนาแอปพลิเคชันและได้แอปพลิเคชันที่ตรงตามความ ต้องการของผู้ใช้งาน โดยความต้องการของผู้ใช้งานที่รวบรวมมามีดังนี้

- 1) มีการ Random ด่าน เพื่อให้การเล่นไม่สนุกแค่ครั้งแรกที่ได้เล่น
- 2) กำหนดว่าสามารถข้ามได้กี่ครั้งและสามารถใช้คำใบ้ได้กี่ครั้ง
- 3) มีด่านใหม่ๆให้อัพเดทอยู่เสมอ
- 4) มีหน้าสอนวิธีการเล่นเกม

4.1.4) ออกแบบ MockUp ของแอปพลิเคชัน

การออกแบบ MockUp ของแอปพลิเคชันก่อนเริ่มต้นการพัฒนาแอปพลิเคชัน จะทำให้เห็นภาพของแอปพลิเคชันได้อย่างชัดเจน เช่น มีกี่หน้า อะไรบ้าง, มีปุ่ม กดปุ่มนี้แล้วไปหน้าใด, สิ่งใดที่เราต้องศึกษาและเรียนรู้เพิ่ม, และแต่ละหน้ามีการจัดวาง Layout อย่างไร ซึ่งสิ่งเหล่านี้จะช่วยให้ทีมพัฒนาทั้งทีม โปรแกรมเมอร์และทีมดีไซน์เกิดความเข้าใจที่ตรงกัน



ภาพที่ 4.4 MockUp ของแอปพลิเคชันเกมเวทีทอง

4.1.5) สร้างเว็บหลังบ้านของแอปพลิเคชันเกมเวทีทอง

เนื่องจากแอปพลิเคชันเกมเวทีทองจะมี Feature ในการดาวน์โหลดด้านใหม่ๆ เข้ามา จึงต้องสร้างเว็บขึ้นมาเพื่อใส่ข้อมูลสำหรับเอาไว้ดาวน์โหลดแอปพลิเคชัน เว็บหลังบ้านที่ทำขึ้นมามี 4 ฟังก์ชัน ได้แก่ เพิ่มด้านใหม่, ลบด้าน, แก้ไขด้าน, และแสดง Feed JSON ข้อมูลด้านต่างๆ ของเกม เช่น Link รูปภาพ, คำโปรย, Choice คำตอบ ฯลฯ

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
☐ 1	<u>game_id</u>	int(11)			No	None	AUTO_INCREMENT	Change Drop
☐ 2	<u>mode_id</u>	int(11)			No	None		Change Drop
☐ 3	<u>pic_num</u>	int(11)			No	None		Change Drop
☐ 4	game_answer	text	utf8_general_ci		No	None		Change Drop
☐ 5	game_hint	text	utf8_general_ci		No	None		Change Drop
☐ 6	game_image	text	utf8_general_ci		No	None		Change Drop
☐ 7	game_choice1	text	utf8_general_ci		No	None		Change Drop
☐ 8	game_choice2	text	utf8_general_ci		No	None		Change Drop
☐ 9	game_choice3	text	utf8_general_ci		No	None		Change Drop
☐ 10	game_choice4	text	utf8_general_ci		No	None		Change Drop
☐ 11	game_choice5	text	utf8_general_ci		No	None		Change Drop
☐ 12	game_choice6	text	utf8_general_ci		No	None		Change Drop
☐ 13	game_choice7	text	utf8_general_ci		No	None		Change Drop
☐ 14	game_choice8	text	utf8_general_ci		No	None		Change Drop
☐ 15	game_pic_type	text	utf8_general_ci		No	None		Change Drop

ภาพที่ 4.5 ฐานข้อมูลของตาราง game

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
☐ 1	<u>mode_id</u>	int(11)			No	None		Change
☐ 2	mode_name	text	utf8_general_ci		No	None		Change

ภาพที่ 4.6 ฐานข้อมูลตาราง mode

#	Name	Type	Collation	Attributes	Null	Default	Extra
☐ 1	<u>id</u>	int(11)			No	None	AUTO_INCREMENT
☐ 2	gcm_regid	text	latin1_swedish_ci		Yes	NULL	
☐ 3	status	text	latin1_swedish_ci		No	None	
☐ 4	created_at	timestamp			No	CURRENT_TIMESTAMP	

ภาพที่ 4.7 ฐานข้อมูลตาราง gcm_users

ตารางที่ 4.1 Data Dictionary ของฐานข้อมูล 578032

ชื่อแฟ้ม	ความหมาย	รายละเอียด	ความหมาย
game	ข้อมูลของเกม	game_id mode_id pic_num game_answer game_hint game_image game_pic_type game_choice1 game_choice2 game_choice3 game_choice4 game_choice5 game_choice6 game_choice7 game_choice8	รหัสด้านเกม รหัสโหมดเกม หมายเลขรูปภาพในเกม คำตอบของเกม คำใบ้ของเกม ภาพของเกม นามสกุลของภาพใน เกม ตัวเลือกที่1 ของเกม ตัวเลือกที่2 ของเกม ตัวเลือกที่3 ของเกม ตัวเลือกที่4 ของเกม ตัวเลือกที่5 ของเกม ตัวเลือกที่6 ของเกม ตัวเลือกที่7 ของเกม ตัวเลือกที่8 ของเกม
mode	ข้อมูลของโหมดเกม	mode_id mode_name	รหัสโหมดเกม ชื่อของโหมดเกม

ตารางที่ 4.2 Data Dictionary ของฐานข้อมูล 578032db2

ชื่อแฟ้ม	ความหมาย	รายละเอียด	ความหมาย
gcm_users	ข้อมูลของ user ใน gcm	id gcm_regid status created_at	รหัส user รหัสของ user ใน gcm สถานะของ user วันที่ลงทะเบียนกับ gcm



ภาพที่ 4.8 Interface หน้าแรกของเว็บหลังบ้าน



ภาพที่ 4.9 Interface หน้าเพิ่มด้านใหม่



ภาพที่ 4.10 Interface หน้าเลือก feed

```
{
  mode: 1,
  name: "คลาสสิกเกม",
  feed: [
    {
      id: 38,
      number: 1,
      answer: "ลิง",
      hint: "ลิง",
      link_image: "http://chuwubbandroid.orgfree.com/game_images/1_1.png",
      choice1: "ลิง",
      choice2: "ลิง",
      choice3: "ลิง",
      choice4: "ลิง",
      choice5: "ลิง",
      choice6: "ลิง",
      choice7: "ลิง",
      choice8: "ลิง"
    },
    {
      id: 39,
      number: 2,
      answer: "ลิง",
      hint: "ลิง",
      link_image: "http://chuwubbandroid.orgfree.com/game_images/1_2.png"
    }
  ]
}
```

ภาพที่ 4.11 หน้า feed

*สามารถดู Code ของเว็บหลังบ้านของแอปพลิเคชันเกมเวทิตองได้ในส่วนของภาคผนวก ก

4.1.6) สร้างเดโมของแอปพลิเคชันเกมเวทีทอง

หลังจากสร้างเว็บหลังบ้านเสร็จเรียบร้อยแล้วจึงเริ่มต้นการพัฒนาแอปพลิเคชัน โดยเริ่มต้นจากออกแบบ Interface และ ปุ่มต่างๆ ในครบทุกหน้าก่อน



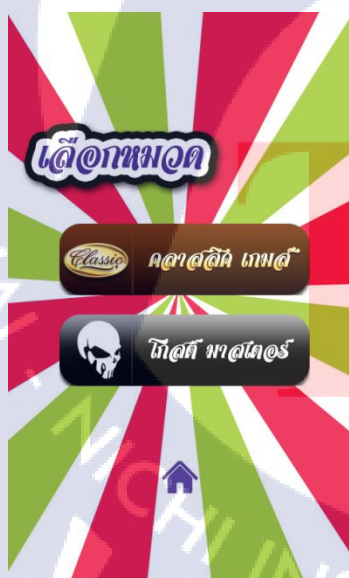
ภาพที่ 4.12 โลโก้แอปพลิเคชัน



ภาพที่ 4.13 หน้า SplashScreen



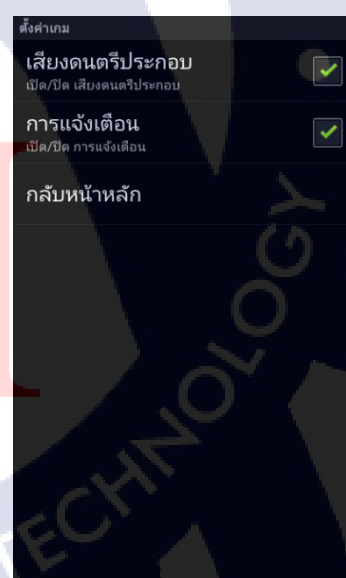
ภาพที่ 4.14 หน้า MainActivity



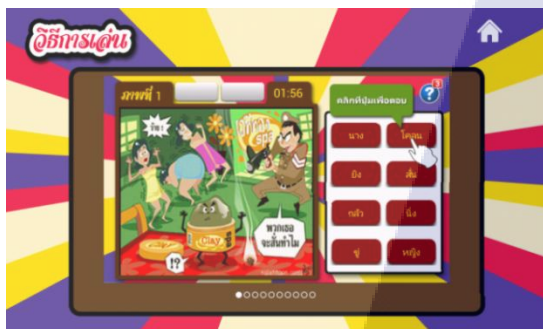
ภาพที่ 4.15 หน้า choosemode



ภาพที่ 4.16 หน้า download



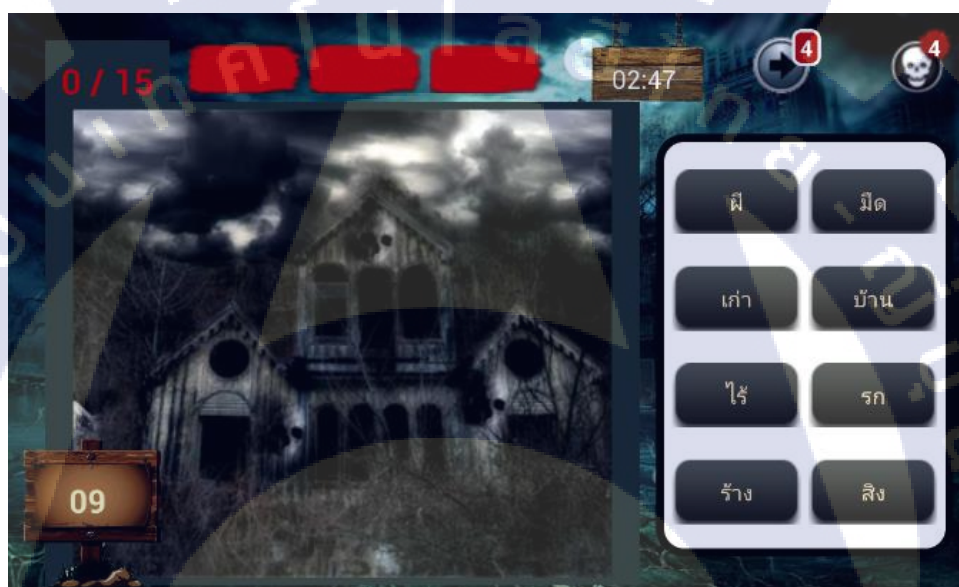
ภาพที่ 4.17 หน้า Settings



ภาพที่ 4.18 หน้า HowToPlay



ภาพที่ 4.19 หน้า classicgame



ภาพที่ 4.20 หน้า ghostmaster

หลังจากออกแบบเสร็จแล้ว จึงเริ่มต้นเขียน Code ของเดโมแอปพลิเคชันเกมเวทิตองด้วยโปรแกรม Eclipse ซึ่งภาษาที่ใช้ในการพัฒนาแอปพลิเคชันประกอบด้วยภาษา Java กับภาษา SQL ตัวเดโมนี้ทำขึ้นมาเพื่อทดสอบ Feature การใช้งานที่จะมีในแอปพลิเคชัน ทำให้ไม่จำเป็นต้องลงรายละเอียดในส่วนของการดีไซน์ เนื่องจากการทดสอบ Feature การใช้งานเท่านั้น

4.1.7) ทดสอบการทำงานของเดโมแอปพลิเคชัน

หลังจากสร้างเดโมของแอปพลิเคชันเสร็จแล้ว ต้องทำการทดสอบเพื่อหาข้อผิดพลาดในการทำงานของเดโมแอปพลิเคชันก่อนที่จะนำไปนำเสนอต่อบริษัท ซึ่งการทดสอบในขั้นตอนนี้ ยังไม่จำเป็นต้องทดสอบกับโทรศัพท์มือถือจริง สามารถทดสอบกับ Emulator เพียงอย่างเดียวได้

4.1.8) นำ Feedback และข้อเสนอแนะที่ได้จากการนำเสนอเดโมแอปพลิเคชันไปปรับปรุง

ขั้นตอนนี้เป็นขั้นตอนที่สำคัญมาก เนื่องจากเราจะได้รับ Feedback และข้อเสนอแนะจากทางบริษัท ทำให้รู้ถึงข้อผิดพลาดของเดโมแอปพลิเคชัน, ได้รับไอเดียใหม่ๆ ในการพัฒนาแอปพลิเคชัน, รวมถึงได้รับความรู้ใหม่ๆ เพื่อนำมาพัฒนาตนเองต่อไปในอนาคต

4.1.9) พัฒนาเดโมแอปพลิเคชันให้เป็นแอปพลิเคชันตัวเต็ม

หลังจากนำ Feedback ที่ได้รับมาปรับปรุงให้กับเดโมแอปพลิเคชันแล้ว ขั้นตอนถัดไปคือพัฒนาเดโมแอปพลิเคชันให้เป็นแอปพลิเคชันตัวเต็ม โดยการลงรายละเอียดในส่วนของการดีไซน์และการจัดวาง Layout ให้สวยงาม รวมถึงทำให้แอปพลิเคชันสามารถใช้งานได้บนโทรศัพท์มือถือระบบปฏิบัติการ Android ให้ครอบคลุมมากที่สุด

*สามารถดู Code ของแอปพลิเคชันเกมเวททองได้ในส่วนของภาคผนวก ข

4.1.10) ทดสอบการทำงานของแอปพลิเคชันเพื่อหาข้อผิดพลาดและนำไปปรับปรุง

หลังจากสร้างแอปพลิเคชันเสร็จแล้วจะต้องนำแอปพลิเคชันไปทดสอบก่อนเนื่องจากหาก Launch แอปพลิเคชันขึ้น Google Play โดยที่ไม่ได้ทดสอบก่อน แล้วเกิดข้อผิดพลาดตามมา จะส่งผลให้ผู้ใช้งานเกิดความรำคาญและลบแอปพลิเคชันของเราทิ้งไปได้ ซึ่งผู้ทำการทดสอบแอปพลิเคชันส่วนใหญ่จะเป็นทีม Test ของบริษัทนั้นหรือนักพัฒนาแอปพลิเคชันของบริษัทนั้น เนื่องจากหากบุคคลภายนอกเป็นผู้ทำการทดสอบ ข้อมูลหรือสิ่งที่เรากำลังพัฒนาอยู่อาจจะรั่วไหลไปยังบริษัทคู่แข่งได้ ทำให้บริษัทของเราสูญเสียรายได้ในอนาคต

การทดสอบการทำงานในขั้นตอนนี้ จะต้องทดสอบกับโทรศัพท์มือถือจริงหลายๆ รุ่น เนื่องจากโทรศัพท์มือถือที่ใช้ระบบปฏิบัติการ Android นั้นมีจำนวนมาก ซึ่งแต่ละรุ่นจะมีขนาดของหน้าจอที่แตกต่างกัน, ขนาดของ Resolution ที่แตกต่างกัน, และเวอร์ชันของระบบปฏิบัติการ Android ที่แตกต่างกัน ส่งผลให้การแสดงผลของแอปพลิเคชันในแต่ละ Devices สามารถผิดเพี้ยนไปได้

4.2 ผลการวิเคราะห์ข้อมูล

4.2.1) วิเคราะห์การทำงานและการตอบสนองของระบบเกม

จากการทดสอบระบบเกมทั้งโหมดคลาสสิกเกมและโหมดโกสต์มาสเตอร์ พบว่าระบบเกมสามารถทำงานได้เป็นอย่างดี และไม่มีข้อผิดพลาดในระบบเกมที่ทำให้ต้องยุติระบบเพื่อแก้ไข แต่การประมวลผลของโหมดโกสต์มาสเตอร์จะเร็วกว่าโหมดคลาสสิกเล็กน้อย เนื่องจากภาพในเกมของโหมดโกสต์มาสเตอร์จะถูกดึงข้อมูลมาจาก Internal Storage ของโทรศัพท์มือถือทั้งหมด ส่วนภาพในเกมของโหมดคลาสสิกเกมจะถูกดึงข้อมูลมาจาก Internal Storage (เฉพาะภาพด่านที่ดาวน์โหลดมาใหม่) และ Folder res/drawable ในแอปพลิเคชัน ซึ่งภาพที่ถูกเก็บไว้ในนี้จะมีการประมวลผลข้อมูลช้ากว่าภาพที่ถูกเก็บเอาไว้ใน SD Card

4.2.2) วิเคราะห์การทำงานและการตอบสนองของ features ต่างๆในแอปพลิเคชัน

จากการทดสอบการทำงานของ Features ทั้งการดาวน์โหลดด่านใหม่, การสอนวิธีการเล่น, การรับ Notification, การตั้งค่าเสียงและ Notification, และการแชร์ภาพบน Facebook พบว่าทุก Features สามารถทำงานได้เป็นอย่างดี และแสดงผลได้ถูกต้องตามที่ต้องการ

4.2.3) วิเคราะห์การทำงานและการตอบสนองของเว็บหลังบ้าน

จากการทดสอบการทำงานของเว็บหลังบ้านพบว่าเว็บหลังบ้านสามารถทำงานได้เป็นอย่างดี โดยสามารถ Insert, Update, Delete ข้อมูลด่านเกมได้ และสามารถให้ข้อมูล Feed JSON ได้อย่างถูกต้อง ซึ่งแอปพลิเคชันสามารถดึงข้อมูลจาก Feed ได้อย่างถูกต้อง

4.3 วิจัยข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายการปฏิบัติงานหรือการจัดทำโครงการ

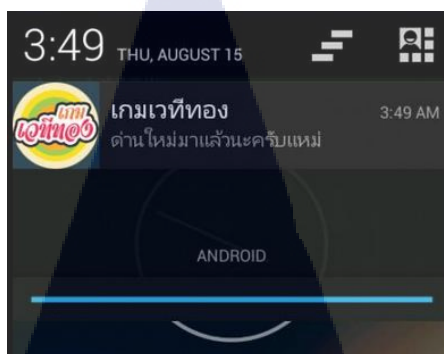
จากวัตถุประสงค์ของการพัฒนาแอปพลิเคชันเกมเวทิตองมีดังต่อไปนี้

- 1) เพื่อศึกษาการใช้งาน Push Notification โดยใช้ GCM
- 2) เพื่อศึกษาการใช้งาน ViewPager ในAndroid
- 3) เพื่อศึกษาการ Get JSON Data จาก feed มาเก็บไว้ใน Database ของโทรศัพท์มือถือ และนำข้อมูลที่เก็บอยู่ใน Database มาแสดงในหน้าจอโทรศัพท์มือถือ
- 4) เพื่อศึกษาการทำงานของ ArrayList ในAndroid
- 5) เพื่อศึกษาการดาวน์โหลดรูปภาพจาก Internet ผ่านURLของรูปภาพนั้น มาเก็บเอาไว้ในหน่วยความจำของโทรศัพท์มือถือ
- 6) เพื่อศึกษาการใช้งาน Custom ListView ในAndroid
- 7) เพื่อศึกษาการใช้งาน SharedPreferences ในAndroid
- 8) เพื่อศึกษาการประยุกต์ใช้ Library ของโปรแกรมที่นักพัฒนาแอปพลิเคชันได้เขียนเอาไว้แล้ว มาใช้กับแอปพลิเคชันของเรา
- 9) เพื่อศึกษาการทำ Animation แบบง่ายๆให้กับ SplashScreen ของแอปพลิเคชัน
- 10) เพื่อฝึกการออกแบบ User Interface และ Layout ในหน้าต่างๆของแอปพลิเคชันให้รองรับกับโทรศัพท์มือถือและแท็บเล็ตที่ใช้ระบบปฏิบัติการAndroid อย่างครอบคลุมมากที่สุด

หลังจากการทำโครงงานแอปพลิเคชันเกมเวทิตองเสร็จสิ้นแล้ว สามารถเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์ในการจัดทำโครงงานได้ ดังต่อไปนี้

4.3.1 ศึกษาการใช้งาน Push Notification โดยใช้ GCM

ระบบสามารถบันทึกข้อมูลผู้ใช้จากแอปพลิเคชันเข้าสู่ฐานข้อมูลของ GCM ในเซิร์ฟเวอร์ได้อย่างถูกต้อง และเว็บหลังบ้านสามารถ Push Notification ไปที่โทรศัพท์มือถือของผู้ใช้ได้



ภาพที่ 4.21 Notification ในโทรศัพท์มือถือ เมื่อมีการเพิ่มด่านใหม่

4.3.2 ศึกษาการใช้งาน ViewPager ใน Android

ViewPager เป็นเครื่องมือที่เอาไว้ใช้ในการสไลด์ภาพไปมา โดยใช้นิ้วสัมผัสที่จอแล้วเลื่อนซ้าย-ขวา ซึ่งได้นำ ViewPager มาใช้ในระบบเกมในโหมดคลาสสิกเกมในส่วนของการสไลด์รูปเพื่อเข้าไปยังรูปถัดไปหรือย้อนกลับมาภาพก่อนหน้า



ภาพที่ 4.22 การใช้งาน ViewPager ในโหมดคลาสสิกเกม

4.3.3 ศึกษาการ get json data จาก feed มาเก็บไว้ใน Database ของโทรศัพท์มือถือ และนำข้อมูลที่เก็บอยู่ใน Database มาแสดงในหน้าจอโทรศัพท์มือถือ

แอปพลิเคชันสามารถดึงข้อมูลด้านใหม่ๆ จากเว็บหลังบ้านซึ่งแสดงข้อมูลใน Format ของ JSON ซึ่งข้อมูลเหล่านั้นมีทั้งส่วนที่เป็นข้อความและส่วนที่เป็น URL ของภาพ เอามาเก็บไว้ใน Database ของโทรศัพท์มือถือได้ และสามารถนำข้อมูลที่ถูกรวบรวมไว้ใน Database มาแสดงในแอปพลิเคชันได้อย่างถูกต้อง โดยนำข้อมูลที่ถูกรวบรวมไว้ใน Database มาแสดงในหน้าเกมทั้งโหมดคลาสสิกเกมและโหมดโกสต์มาสเตอร์

4.3.4 ศึกษาการทำงานของ ArrayList ใน Android

เนื่องจากแอปพลิเคชันเกมเวทีทองมี Feature ในการดาวน์โหลดด่านใหม่ จึงต้องมีการแสดงรายการด่านใหม่ที่มีในระบบเพื่อให้ผู้ใช้ได้ดูก่อนที่จะดาวน์โหลดด่านเกมด่านใหม่ๆ ลงในโทรศัพท์มือถือ ซึ่งได้ใช้ ArrayList ในการเอาข้อมูลด่านใหม่ๆ มาแสดงในแอปพลิเคชัน

4.3.5 ศึกษาการดาวน์โหลดรูปภาพจาก Internet ผ่าน URL ของรูปภาพนั้น มาเก็บไว้ในหน่วยความจำของโทรศัพท์มือถือ

เนื่องจากเว็บหลังบ้านของแอปพลิเคชันเกมเวทีทองให้ข้อมูลของรูปเป็น URL ทำให้ฝั่งแอปพลิเคชันต้องแปลง URL เป็น ภาพ Bitmap ก่อน จากนั้นจึงแปลงภาพ Bitmap เป็น BLOB เพื่อที่จะเก็บภาพลงในหน่วยความจำของโทรศัพท์มือถือ ซึ่งการทำงานของเมธอดในการแปลงจาก URL เป็น Bitmap และแปลง Bitmap เป็น BLOB สามารถทำงานได้เป็นอย่างดี

4.3.6 ศึกษาการใช้งาน Custom ListView ใน Android

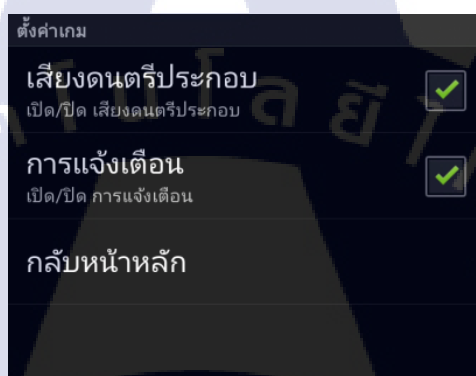
สืบเนื่องจากข้อ 4.3.4 แอปพลิเคชันจะแสดงข้อมูลของด่านใหม่ๆ ที่มีในระบบให้ผู้ใช้ได้ดูก่อนจะทำการดาวน์โหลด ซึ่งข้อมูลด่านใหม่จะถูกแสดงในรูปแบบของ List รายการ โดยใช้ Tools ListView ใน Android มาใช้ ซึ่งแสดงอยู่ในหน้าดาวน์โหลดด่านใหม่ ดังภาพที่ 4.23



ภาพที่ 4.23 การใช้งาน Custom ListView ในหน้า download

4.3.7 ศึกษาการใช้งาน SharedPreferences ใน Android

ในแอปพลิเคชันเกมเวทีทองมี feature ให้ผู้ใช้สามารถตั้งค่าในเกมได้ เช่น การตั้งค่าเสียง ต้องมีการเก็บค่าจากทางแอปพลิเคชันว่าตอนนี้ผู้ใช้เปิด/ปิดการใช้งานเสียงอยู่หรือไม่ ซึ่งใน Eclipse มีฟังก์ชันชื่อ SharedPreferences ซึ่งสามารถเก็บค่าการตั้งค่าได้ และเมื่อผู้ใช้เข้าใช้งานแอปพลิเคชันในครั้งถัดไป ค่าที่ตั้งค่าไว้จะยังคงอยู่เหมือนเดิม หลังจากการนำ SharedPreferences มาเก็บการตั้งค่าในแอปพลิเคชัน ปรากฏว่าสามารถเก็บค่าการตั้งค่าต่างๆได้อย่างถูกต้อง



ภาพที่ 4.24 การใช้ SharedPreferences ในหน้า Settings

4.3.8 ศึกษาการประยุกต์ใช้ Library ของโปรแกรมที่มีนักพัฒนาแอปพลิเคชันได้เขียนเอาไว้แล้ว มาใช้กับแอปพลิเคชันของเรา

เนื่องจากในแอปพลิเคชันเกมเวทีทองมีหลาย Features ซึ่งในบาง Feature จำเป็นต้องเรียกใช้ Library จากไฟล์อื่นๆซึ่งได้มีผู้เขียนไว้แล้ว เช่น ในการแชร์ภาพลง Facebook ต้องเรียกใช้ Library Facebook - SDK จากทาง Facebook, การแสดงจุดเคลื่อนไหวที่ในขณะสไลด์ภาพในหน้าวิธีการเล่น ต้องเรียกใช้ Library ViewPagerIndicator ของ Jake Wharton, และแสดงภาพที่เก็บไว้ในหน่วยความจำของโทรศัพท์มือถือในด่านเกม ต้องเรียกใช้ Library Universal ImageLoader ของ Sergey Tarasevich มาใช้ซึ่งหลังจากเรียกใช้งาน Library ทั้งหมดที่กล่าวมาใช้งานแล้ว แอปพลิเคชันก็สามารถทำงานได้ถูกต้องและแสดงผลได้ถูกต้องตามที่ต้องการ

4.3.9 ศึกษาการทำ Animation แบบง่ายๆให้กับ SplashScreen ของแอปพลิเคชัน

การใส่ Animation ให้กับ SplashScreen ทำเพื่อเพิ่มความน่าสนใจให้กับแอปพลิเคชัน ซึ่ง SplashScreen ของแอปพลิเคชันเกมเวทีทอง สามารถเล่น Animation ได้ถูกต้องตามที่ได้ออกแบบไว้ในตอนแรก



ภาพที่ 4.25 Splash Screen Animation ของแอปพลิเคชันเกมเวทีทอง

4.3.10 ฝึกการออกแบบ User Interface และ Layout ในหน้าต่างๆของแอปพลิเคชันให้รองรับกับโทรศัพท์มือถือและแท็บเล็ตที่ใช้ระบบปฏิบัติการAndroid อย่างครอบคลุมมากที่สุด

เนื่องจาก Devices ที่ใช้ระบบปฏิบัติการ Android มีขนาดของหน้าจอที่ต่างกัน, ขนาดของ Resolution ของจอภาพที่แตกต่างกัน, และเวอร์ชันของระบบปฏิบัติการ Android ที่แตกต่างกัน ทำให้การแสดงผลในแต่ละ Devices อาจเกิดการผิดเพี้ยนได้ ทำให้ต้องออกแบบ Layout ให้รองรับกับ Devices ให้ได้มากที่สุด ซึ่งหลังจากการแก้ไข ส่งผลให้แอปพลิเคชันสามารถแสดงผลกับ Devices ส่วนใหญ่ได้ถูกต้องตามที่ต้องการ

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินโครงการ

แอปพลิเคชันเกมเวทีทองสามารถตอบสนองการทำงานตามความต้องการในการใช้งานได้ดีระดับหนึ่ง สามารถนำมาใช้งานได้จริงและไม่เกิดข้อผิดพลาดใดๆ ซึ่งยังสามารถนำไปปรับปรุงและพัฒนาต่อไปในอนาคตได้ต่อไป โดย Features ที่มีในเวอร์ชันปัจจุบัน มีดังต่อไปนี้

- 1) สามารถดาวน์โหลดด่านเกมใหม่ๆ ได้
- 2) สามารถตั้งค่าการเปิด/ปิดเสียง และ เปิด/ปิด Notification ได้
- 3) สามารถแชร์ภาพลงใน Facebook ได้ (ในโหมดโกสต์มาสเตอร์)
- 4) เมื่อมีด่านใหม่เข้ามาในระบบ จะมี Notification แจ้งเตือนให้ผู้ใช้ทราบ

นอกจากนี้ได้มีการพัฒนาระบบเกมให้สามารถ Random ด่านได้ และพัฒนาระบบเกมของโหมดทั้ง 2 โหมดให้แตกต่างกัน เพื่อเพิ่มความตื่นเต้นและความสนุกสนานให้กับผู้ใช้

ในส่วนของการพัฒนาแอปพลิเคชันนั้น มีอุปสรรคและปัญหาพอสมควร เนื่องจากความรู้ที่จำเป็นต้องใช้ในการพัฒนาแอปพลิเคชันเกมเวทีทองนี้ เป็นความรู้ใหม่เกือบทั้งหมด จึงต้องมีการศึกษาหาความรู้จากทาง Internet, จากหนังสือ ,และจากการสอบถามผู้ที่ปรึกษา ซึ่งในบางครั้งต้องมีการลองผิดลองถูก เพื่อความเข้าใจที่มากขึ้นในการเรียนรู้ และต้องวางแผนในการทำงานให้รอบคอบ เพื่อไม่ให้ใช้ระยะเวลาในการพัฒนาแอปพลิเคชันเกินกว่าที่กำหนดไว้

อีกปัญหาหนึ่งคือจำนวนรุ่นของ Devices ที่ใช้ระบบปฏิบัติการ Android เนื่องจาก Devices ที่ใช้ระบบปฏิบัติการ Android มีขนาดของหน้าจอที่ต่างกัน, ขนาดของ Resolution ของจอภาพที่แตกต่างกัน, และเวอร์ชันของระบบปฏิบัติการ Android ที่แตกต่างกัน ทำให้การแสดงผลในแต่ละ Devices อาจเกิดการผิดเพี้ยนได้

5.2 แนวทางการแก้ไข้ปัญหา

ในการพัฒนาแอปพลิเคชันในเวอร์ชันต่อไป ควรพัฒนาให้มี Features ที่มากขึ้น เช่น สามารถแชร์ภาพลงใน Twitter ได้, มีการเก็บคะแนนเพื่อแข่งกับเพื่อนใน Facebook, และพัฒนาระบบเกมให้มีความสนุกมากยิ่งขึ้น นอกจากนี้ควรพัฒนาให้แอปพลิเคชันรองรับกับ Devices รุ่นใหม่ๆ ซึ่งผลิตออกมาอย่างต่อเนื่องและเวอร์ชันของระบบปฏิบัติการ Android รุ่นใหม่ๆ

5.3 ข้อเสนอแนะจากการดำเนินงาน

ในการสหกิจศึกษาตลอดระยะเวลา 4 เดือนที่ผ่านมาพบว่า การทำงานของ Developer ในการพัฒนาแอปพลิเคชันขึ้นมา 1 แอปพลิเคชัน จะต้องร่วมมือกับฝ่ายต่างๆ เช่น Developer ในบริษัท สนุกออนไลน์จำกัด จะต้องประสานงานกับฝ่ายดีไซน์กับฝ่าย Business ซึ่ง Developer จะต้องคอยแนะนำหรือแสดงความคิดเห็นในแอปพลิเคชันที่ทางฝ่ายดีไซน์กับฝ่าย Business คิดขึ้นมา ว่ามีส่วนไหนที่สามารถทำได้ส่วนไหนที่ไม่ควรทำ ส่วนไหนที่ทำได้แต่ขัดกับ Experience ของผู้ใช้งาน เป็นต้น

การวางแผนในการทำงานเป็นสิ่งที่สำคัญมากสำหรับผู้ที่ต้องการจะเป็น Developer เนื่องจาก Developer แต่ละคนไม่ได้เขียนแอปพลิเคชันเดียว บางคนอาจจะต้องเขียนถึง 2 – 3 แอปพร้อมๆกัน หากไม่มีการวางแผนการทำงานอาจทำให้งานทุกส่วน Delay ได้

การปฏิบัติสหกิจศึกษาที่บริษัท สนุก ออนไลน์ จำกัด เหมาะกับผู้ที่ต้องการเรียนรู้การเขียนเว็บไซต์, การจัดการเว็บไซต์, และการทำแอปพลิเคชันในระบบปฏิบัติการ Android / iOS / WindowsPhone ซึ่งหากนักศึกษาได้ศึกษาสิ่งเหล่านั้นมาก่อน จะทำให้สามารถเรียนรู้ได้รวดเร็วมากขึ้นและได้รับความรู้ใหม่ๆเป็นจำนวนมาก แต่เนื่องจากบริษัท สนุก ออนไลน์ จำกัด เป็นบริษัทที่มีชื่อเสียง พนักงานแต่ละคนจึงมีงานที่เยอะตลอดเวลา ทำให้ทักษะในการเรียนรู้สิ่งใหม่ๆด้วยตัวเองจึงมีความจำเป็นมาก

เอกสารอ้างอิง

1. Ravi Tamada, 2012, **AndroidHive** [Online], Available :
<http://www.androidhive.info/2012/01/android-json-parsing-tutorial> [2013, June 3].
2. Ravi Tamada, 2012, **AndroidHive** [Online], Available :
<http://www.androidhive.info/2012/07/android-detect-internet-connection-status>
 [2013, June 5].
3. ณัฐพล ภิณโย, 2013, สอน **Android** สไลด์ไทยๆ [Online], Available :
<http://www.youtube.com/watch?v=QNa8iO4bpOc&list=PL3D46C3E9831EE466&index=19> [2013, June 6].
4. ณัฐพล ภิณโย, 2012, สอน **Android** สไลด์ไทยๆ [Online], Available :
<http://www.youtube.com/watch?v=Q8PrOEewuJI&list=PL3D46C3E9831EE466&index=6> [2013, June 10].
5. ณัฐพล ภิณโย, 2013, สอน **Android** สไลด์ไทยๆ [Online], Available :
<http://www.youtube.com/watch?v=AfiBo5cmRro&list=PL3D46C3E9831EE466&index=16> [2013, June 10].
6. ณัฐพล ภิณโย, 2012, สอน **Android** สไลด์ไทยๆ [Online], Available :
<http://www.youtube.com/watch?v=fghB6EdrCyw&list=PL3D46C3E9831EE466&index=12> [2013, June 17].
7. ณัฐพล ภิณโย, 2012, สอน **Android** สไลด์ไทยๆ [Online], Available :
http://www.youtube.com/watch?v=Y1k_ggWeSxw&list=PL3D46C3E9831EE466&index=13 [2013, June 20].

8. Ravi Tamada, 2012, **AndroidHive** [Online], Available :
<http://www.androidhive.info/2012/03/android-listview-with-load-more-button>
[2013, June 24].
9. Twitter4J, 2013, **Code Examples** [Online], Available : <http://twitter4j.org/en/code-examples.html> [2013, June 27].
10. ThaiCreate.com Team, 2012, **Android Pull Down to Refresh And Release to Refresh or Update (Part 1)** [Online], Available : <http://www.thaicreate.com/mobile/android-pull-down-refresh.html> [2013, July 4] .
11. 2013, **json_encode** [Online], Available : <http://www.php.net/manual/en/function.json-encode.php>[2013, July 17].
12. 2013, **CountDownTimer** [Online], Available :
<http://developer.android.com/reference/android/os/CountDownTimer.html> [2013, July 23].
13. 2013, **Using ViewPager for Screen Slides** [Online], Available :
<http://developer.android.com/training/animation/screen-slide.html> [2013, July 25].
14. Chris Banes, 2013, **chrisbanes/PhotoView** [Online], Available :
<https://github.com/chrisbanes/PhotoView> [2013, July 25].
15. Jake Wharton, 2012, **Android ViewPagerIndicator** [Online], Available :
<https://github.com/JakeWharton/Android-ViewPagerIndicator> [2013, August 1].
16. พร้อมเลิศ หล่อวิจิตร, 2013, **คู่มือเขียนแอปAndroidฉบับรวมโค้ด**, พิมพ์ครั้งที่ 1, บริษัท Provision จำกัด, บริษัท พิมพ์ดี จำกัด, หน้า 58-61 .

17. Sergey Tarasevich, 2013, **Universal Image Loader For Android** [Online], Available :
<https://github.com/nostra13/Android-Universal-Image-Loader> [2013, August 7].
18. Ravi Tamada, 2012, **AndroidHive** [Online], Available :
<http://www.androidhive.info/2012/10/android-push-notifications-using-google-cloud-messaging-gcm-php-and-mysql> [2013, August 13].
19. 2012, **Sleeping For Less** [Online], Available :
<http://www.akexorcist.com/2012/06/android-code-supporting-multiple.html>
[2013, August 13].
20. Prajak Boonjaratcha, 2013, **MobileDep** [Online], Available :
<http://mobiledep.blogspot.com/2011/09/activity.html> [2013, August 26].
21. PHP MindPHP.com, 2012, **Java คืออะไร** [Online], Available : <http://www.mindphp.com/คู่มือ/73-คืออะไร/2185-java-คืออะไร.html> [2013, August 26].
22. iTCyBER, 2010, **ทำความรู้จักกับ JSON** [Online], Available : <http://www.itcyber.com/intr-json> [2013, August 26].
23. JoomlaHitZ, 2013, **MySQL คืออะไร** [Online], Available :
<http://www.joomlahitz.com/index.php/basic-website/46-what-is-my-sql.html#.UhsnLRvp2So> [2013, August 26].
24. 2013, **Google Cloud Messaging For Android** [Online], Available :
<http://developer.android.com/google/gcm/index.html> [2013, August 27].

ภาคผนวก

TNI

ภาคผนวก ก

Code Web หลังบ้าน ของ Application เกมเวทีทอง

TNI

1. หน้า connect.php

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<?php
    $host = "localhost";
    $user = "578032";
    $pass = "Fvhc,jg0hk";
    $conn = @mysql_connect($host,$user,$pass) or die("cannot connect Database");

    $dbname = "578032";
    $db = mysql_select_db($dbname,$conn) or die("cannot select Database");
    mysql_query("SET NAMES utf8");

?>
```

2. หน้า db_connect.php

```
<?php
class DB_Connect {
    // constructor
    function __construct() {
    }
    // destructor
    function __destruct() {
        // $this->close();
    }
    // Connecting to database
    public function connect() {
        require_once 'config.php';
        // connecting to mysql
        $con = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
        // selecting database
        mysql_select_db(DB_DATABASE);
        // return database handler
        return $con;
    }
    // Closing database connection
    public function close() {
        mysql_close();
    }
}
?>
```

3. หน้า db_functions.php

```
<?php
class DB_Functions {

    private $db;

    //put your code here
    // constructor
    function __construct() {
        include_once("db_connect.php");
        // connecting to database
        $this->db = new DB_Connect();
        $this->db->connect();
    }
    // destructor
    function __destruct(){
    }
    /**
     * Storing new user
     * returns user details
     */
    public function storeUser($gcm_regid,$status) {
        // insert user into database
        $sql = "SELECT * FROM gcm_users WHERE gcm_regid = '$gcm_regid'";
        $query = mysql_query($sql) or die(mysql_error());
        $row = mysql_num_rows($query);
        if($row == 0){
            $result = mysql_query("INSERT INTO gcm_users(gcm_regid,status,created_at)
                                VALUES ('$gcm_regid','$status',NOW())");
        }
        else{
            //skip registration
        }
        // check for successful store
        if ($result) {
            // get user details
            $id = mysql_insert_id(); // last inserted id
            $result = mysql_query("SELECT * FROM gcm_users WHERE id = $id")
                        or die(mysql_error());
            // return user details
            if (mysql_num_rows($result) > 0) {
                return mysql_fetch_array($result);
            } else {
                return false;
            }
        } else {
            return false;
        }
    }
}
```

```

public function changeStatus($regId, $status) {
    $result = mysql_query("UPDATE gcm_users
                           SET status = '$status'
                           WHERE gcm_regid = '$regId'")
        or die(mysql_error());
    return $result;
}

/* Getting all users*/
public function getAllUsers() {
    $result = mysql_query("SELECT gcm_regid FROM gcm_users");
    return $result;
}
}
?>

```

4. หน้า config.php

```

<?php
/**
 * Database config variables
 */
define("DB_HOST", "localhost");
define("DB_USER", "578032");
define("DB_PASSWORD", "Fvhc,jg0hk");
define("DB_DATABASE", "578032db2");

/*
 * Google API Key
 */
// Place your Google API Key
define("GOOGLE_API_KEY", "AIzaSyBg7p9UQgJonAyX4t7ScmBdtzJg3urdrw");
?>

```

5. หน้า send_message.php

```

<?php
if (isset($_GET["regId"])) {
    $regId = $_GET["regId"];
    $message = $_GET["message"];
    $status = $_GET["status"];

    include_once("GCM.php");
    $gcm = new GCM();
    $registration_ids = array($regId);
    $message = array("message" => $message);

    if($status == "1"){
        $result = $gcm->send_notification($registration_ids, $message);
        echo $result;
    }
}
?>

```

6. หน้า gcm.php

```
<?php
class GCM {
    //put your code here
    // constructor
    function __construct() {

    }

    /**
     * Sending Push Notification
     */
    public function send_notification($registration_ids, $message) {
        // include config
        include_once './config.php';

        // Set POST variables
        $url = 'https://android.googleapis.com/gcm/send';

        $fields = array(
            'registration_ids' => $registration_ids,
            'data' => $message,
        );

        $headers = array(
            'Authorization: key=' . GOOGLE_API_KEY,
            'Content-Type: application/json'
        );

        // Open connection
        $ch = curl_init();

        // Set the url, number of POST vars, POST data
        curl_setopt($ch, CURLOPT_URL, $url);

        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

        // Disabling SSL Certificate support temporarily
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

        curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));

        // Execute post
        $result = curl_exec($ch);
        if ($result === FALSE) {
            die('Curl failed: ' . curl_error($ch));
        }
    }
}
```

```

        // Close connection
        curl_close($ch);
        echo $result;
    }
}
?>

```

7. หน้า register.php

```

<?php
// response json
$json = array();
/*Registering a user device Store reg id in users table */
if (isset($_POST["regId"])) {
    $gcm_regid = $_POST["regId"]; // GCM Registration ID
    // Store user details in db
    include_once './db_functions.php';
    include_once './GCM.php';
    $db = new DB_Functions();
    $gcm = new GCM();
    $res = $db->storeUser($gcm_regid);
    $registration_ids = array($gcm_regid);
    $message = array("Notification" => "เปิดใช้งานการแจ้งเตือนเรียบร้อยแล้ว");
    $result = $gcm->send_notification($registration_ids, $message);
    echo $result;
} else {
    // user details missing
}
?>

```

8. หน้า index.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>เวทีเพลง BackStage</title>

<style type="text/css">
head table {
    font-family: "Century Gothic";
}

.tableHead {
    font-family: "Century Gothic";
    font-weight: bold;
    font-size: 14px;
    color: #FF6;
}
.editDelete {
    font-family: "Century Gothic";
    font-size: 12px;
}
.previousNext {
    font-family: "Century Gothic";
    font-size: 14px;
    font-weight: bold;
    color: #36F;
}
.content {
    font-family: supermarket;
    font-size: 14px;
}
</style>
```

```

<script type="text/javascript">
function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
      if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}

function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
    if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
</script>
</head>

<?php
include ('connect.php');
$sql = "SELECT * FROM game g , mode m
        WHERE g.mode_id = m.mode_id
        ORDER BY g.game_id DESC";
$query = mysql_query($sql) or die(mysql_error());
$row = mysql_num_rows($query);
$page = 10;
$totalpage = ceil($row/$page);

//Split Page
$page = $_GET['page'];
if($page=="")
{
    $page = 1;
}
$startpoint = ($page-1)*$page;
$sql = $sql." limit $startpoint,$page ";
$query = mysql_query( $sql ) or die(mysql_error());
$rowpage = mysql_num_rows( $query );
?>

```

```

<body onload="MM_preloadImages('images/menu_home_active.gif',
                                'images/menu_feed_active.gif',
                                'images/menu_add_active.gif')">
<form id="form1" name="form1" method="post" action="">
  <table bgcolor="#ffdb9" width="960" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
      <td></td>
    </tr>
    <tr>
      <td><table width="960" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td><a href="index.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('menu_home','','images/menu_home_active.gif',1)">
            </a></td>
          <td><a href="Feed.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('menu_feed','','images/menu_feed_active.gif',1)">
            </a></td>
          <td><a href="InsertForm.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('menu_add','','images/menu_add_active.gif',1)">
            </a></td>
          <td></td>
        </tr>
      </table></td>
    </tr>
    <tr>
      <td><table width="960" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td>&nbsp;</td>
        </tr>
        <tr>
          <td><table width="960" border="0" cellspacing="0" cellpadding="0">
            <tr>
              <td width="50">&nbsp;</td>
              <td><table width="860" border="1" cellspacing="1" cellpadding="0">
                <tr class="tableHead">
                  <td align="center" bgcolor="#333333">ID</td>
                  <td align="center" bgcolor="#333333" class="tableHead">MODE_NAME</td>
                  <td align="center" bgcolor="#333333">PIC_NUM</td>
                  <td align="center" bgcolor="#333333">ANSWER</td>
                  <td align="center" bgcolor="#333333">&nbsp;</td>
                  <td align="center" bgcolor="#333333">&nbsp;</td>
                </tr>

```

```

<?php
for($i=1;$i<=$rowperpage;$i++)
{
    $result = mysql_fetch_array($query);
?>

|  |  |  |  |
| --- | --- | --- | --- |
|<table width="960" border="0" cellspacing="0" cellpadding="0">
<tr>
 <table width="960" border="0" cellspacing="0" cellpadding="0"> <tr>  &nbsp;</td> </tr> <tr>  <table width="960" border="0" cellspacing="0" cellpadding="0"> <tr>   | | | |

```

```

<?php
for ( $p=1;$p<=$totalpage;$p++ )
{
    if( $p != $page ){
        echo "<a href='?page=$p'> $p </a> &nbsp;" ;
    }
    else{
        echo "<b> [$p] </b> &nbsp;";
    }
}
?>
</td>
<td width="100" align="center" class="previousNext" style="color: #60C">
<?php if($page<$totalpage) { ?>
<a href="?page=<?php echo $page+1; ?>">next</a>
<?php } ?>
</td>
<td width="300"><a href="testnoti.php">test noti</a></td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
<tr>
<td>&nbsp;</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
<tr>
<td></td>
</tr>
</table>
</form>
</body>
</html>

```

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

9. หน้า InsertForm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>หน้า BackStage</title>

<style type="text/css">
head table {
    font-family: "Century Gothic";
}
.tableHead {
    font-family: "Century Gothic";
    font-weight: bold;
    font-size: 14px;
    color: #FF6;
}
.editDelete {
    font-family: "Century Gothic";
    font-size: 12px;
}

.previousNext {
    font-family: "Century Gothic";
    font-size: 14px;
    font-weight: bold;
    color: #36F;
}
.content {
    font-family: supermarket;
    font-size: 14px;
}
.AddNewHeader {
    font-family: "Century Gothic";
    font-size: 16px;
    font-weight: bold;
    color: #FFF;
}
</style>
</head>
<?php
    include ('connect.php');
?>

<body onload="MM_preloadImages('images/menu_home_active.gif',
                                'images/menu_feed_active.gif',
                                'images/menu_add_active.gif')">
<form action="InsertSave.php" method="post" enctype="multipart/form-data"
name="form1" id="form1" >
```

```

<table width="960" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td></td>
</tr>
<tr>
<td><table width="960" border="0" cellspacing="0" cellpadding="0">
<tr>
<td><a href="index.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('Image4','','images/menu_home_active.gif',1)">
</a></td>
<td><a href="Feed.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('Image5','','images/menu_feed_active.gif',1)">
</a></td>
<td><a href="InsertForm.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('Image6','','images/menu_add_active.gif',1)">
</a></td>
<td></td>
</tr>
</table></td>
</tr>

<tr>
<td><table width="960" border="0" cellspacing="0" cellpadding="0" bgcolor="#ffdab9">
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td><table width="960" border="0" cellspacing="0" cellpadding="0">
<tr>
<td><table width="960" border="0" cellspacing="0" cellpadding="0">
<tr>
<td><table width="960" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="100">&nbsp;</td>
<td><table width="760" border="1" cellspacing="1" cellpadding="0">
<tr>
<td height="30" align="center" bgcolor="#333333"
class="AddNewHeader">Add New</td>
</tr>

<tr bgcolor="#DFDFDF">
<td><table width="760" border="1" cellspacing="0" cellpadding="0">
<tr>
<td width="200" height="30" align="right">mode : </td>
<td height="30"><label for="mode"></label>
<select name="mode" id="mode">
<?php
$sql = "SELECT * FROM mode";
$query = mysql_query($sql) or die(mysql_error());
$row = mysql_num_rows($query);
for($i=1;$i<=$row;$i++)
{
$result = mysql_fetch_array($query);
?> |

```

```

        <option value="<?php echo $result['mode_id']; ?>" selected="selected">
            <?php echo $result['mode_name']; ?>
        </option>
    <?php } ?>
</select></td>
</tr>
<tr>
    <td width="200" height="30" align="right">pic_num : </td>
    <td height="30"><label for="pic_num"></label>
        <input name="pic_num" type="text" id="pic_num" size="10" /></td>
</tr>
<tr>
    <td width="200" height="30" align="right">image : </td>
    <td height="30"><label for="img"></label>
        <input name="img" type="file" id="img" size="50" /></td>
</tr>
<tr>
    <td width="200" height="30" align="right">game_answer : </td>
    <td height="30"><label for="game_hint"></label>
        <input name="game_answer" type="text" id="game_answer" size="20" /></td>
</tr>
<tr>
    <td width="200" height="30" align="right">game_hint : </td>
    <td height="30"><label for="game_hint"></label>
        <input name="game_hint" type="text" id="game_hint" size="10" /></td>
</tr>
<tr>
    <td width="200" height="30" align="right">game_choice1 : </td>
    <td height="30"><label for="game_choice1"></label>
        <input name="game_choice1" type="text" id="game_choice1" size="10" /></td>
</tr>
<tr>
    <td width="200" height="30" align="right">game_choice2 : </td>
    <td height="30"><label for="game_choice2"></label>
        <input name="game_choice2" type="text" id="game_choice2" size="10" /></td>
</tr>
<tr>
    <td width="200" height="30" align="right">game_choice3 : </td>
    <td height="30"><label for="game_choice3"></label>
        <input name="game_choice3" type="text" id="game_choice3" size="10" /></td>
</tr>
<tr>
    <td width="200" height="30" align="right">game_choice4 : </td>
    <td height="30"><label for="game_choice4"></label>
        <input name="game_choice4" type="text" id="game_choice4" size="10" /></td>
</tr>
<tr>
    <td width="200" height="30" align="right">game_choice5 : </td>
    <td height="30"><label for="game_choice5"></label>
        <input name="game_choice5" type="text" id="game_choice5" size="10" /></td>
</tr>

```


10. หน้า InsertSave.php

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<?php
    if($_POST['game_answer']=="")
    {
        echo "<script>>window.location='index.php';</script>";
        exit();
    }
    $mode = $_POST['mode'];
    $pic_num = $_POST['pic_num'];
    $game_answer = $_POST['game_answer'];
    $game_hint = $_POST['game_hint'];
    $game_choice1 = $_POST['game_choice1'];
    $game_choice2 = $_POST['game_choice2'];
    $game_choice3 = $_POST['game_choice3'];
    $game_choice4 = $_POST['game_choice4'];
    $game_choice5 = $_POST['game_choice5'];
    $game_choice6 = $_POST['game_choice6'];
    $game_choice7 = $_POST['game_choice7'];
    $game_choice8 = $_POST['game_choice8'];

    $file = $_FILES['img']['name'];
    $typefile = $_FILES['img']['type'];
    $sizefile = $_FILES['img']['size'];

    $game_pic_type = substr($typefile,6);
    $Nfilename = $mode."_".$pic_num."_".$game_pic_type;
    $linkIMG = "http://chuwubbandroid.orgfree.com/game_images/" . $Nfilename;

    //จะเก็บข้อมูลเป็น array แสดงว่าในตัวแปร img ข้อมูลจะเก็บเป็น array
    ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    if($sizefile>0)
    {
        if($sizefile>300*1024) //300*1024 = 300kb
        {
            echo "<script>alert('ไฟล์รูปใหญ่เกินไป กรุณาใส่ไฟล์รูปที่มีขนาดไม่เกิน 300kb ด้วยครับ');
            history.back();</script>";
            exit();
        }
    }
    else
    {
        echo "<script>alert('กรุณารับรูปภาพด้วยครับ');
        history.back();</script>";
        exit();
    }

    move_uploaded_file($_FILES['img']['tmp_name'], "game_images/" . $Nfilename);
    //upload ไฟล์แล้ว ไฟล์จะไปวางไว้ตรงไหน
```

```

include('connect.php');
$sql = "INSERT INTO game(mode_id,pic_num,game_image,game_answer,game_hint,
                        game_choice1,game_choice2,game_choice3,game_choice4,
                        game_choice5,game_choice6,game_choice7,game_choice8,
                        game_pic_type)
VALUES ('$mode','$pic_num','$linkIMG','$game_answer','$game_hint',
        '$game_choice1','$game_choice2','$game_choice3','$game_choice4',
        '$game_choice5','$game_choice6','$game_choice7','$game_choice8',
        '$game_pic_type')";

$query = mysql_query($sql) or die( mysql_error() );
$id = mysql_insert_id();
mysql_close($conn);

$host = "localhost";
$user = "578032";
$pass = "Fvhc,jg0hk";
$conn = @mysql_connect($host,$user,$pass) or die("cannot connect Database");

$dbname = "578032db2";
$db = mysql_select_db($dbname,$conn) or die("cannot select Database");
$sql1 = "SELECT gcm_regid FROM gcm_users";
$query1 = mysql_query($sql1) or die(mysql_error());
$row1 = mysql_num_rows($query1);

include_once './GCM.php';
$gcm = new GCM();
$message = array("message" => "ด่านใหม่มาน่าเล่นครับหะ");
for($i=0;$i<$row1;$i++){
    $result1 = mysql_fetch_array($query1);
    $registration_ids = array($result1['gcm_regid']);
    $result = $gcm->send_notification($registration_ids, $message);
    echo $result;
}
mysql_close($conn);

?>

<script>
    alert('เพิ่มข้อมูลเรียบร้อยแล้วครับ');
    window.location='index.php';
</script>

```

11. หน้า EditForm.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>เวทีทอง BackStage</title>

<style type="text/css">
head table {
    font-family: "Century Gothic";
}
.tableHead {
    font-family: "Century Gothic";
    font-weight: bold;
    font-size: 14px;
    color: #FF6;
}
.editDelete {
    font-family: "Century Gothic";
    font-size: 12px;
}
.previousNext {
    font-family: "Century Gothic";
    font-size: 14px;
    font-weight: bold;
    color: #36F;
}
.content {
    font-family: supermarket;
    font-size: 14px;
}
.AddNewHeader {
    font-family: "Century Gothic";
    font-size: 16px;
    font-weight: bold;
    color: #FFF;
}
</style>
</head>
<?php
    if($_GET['id']== "")
    {
        echo "<script>window.location='index.php'; </script>";
        exit();
    }

    $id = $_GET['id'];

    include ('connect.php');
    $sql = "SELECT * FROM game
        WHERE game_id = $id";
    $query = mysql_query($sql) or die( mysql_error() );
    $result = mysql_fetch_array($query);
?>

```

```

<body onload="MM_preloadImages('images/menu_home_active.gif',
                                'images/menu_feed_active.gif',
                                'images/menu_add_active.gif')">
<form action="EditSave.php" method="post" enctype="multipart/form-data" name="form1" id="form1">
  <table width="960" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
      <td></td>
    </tr>
    <tr>
      <td><table width="960" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td><a href="index.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('Image5','','images/menu_home_active.gif',1)">
            </a></td>
          <td><a href="Feed.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('Image6','','images/menu_feed_active.gif',1)">
            </a></td>
          <td><a href="InsertForm.php" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('Image7','','images/menu_add_active.gif',1)">
            </a></td>
          <td></td>
        </tr>
      </table></td>
    </tr>
    <tr>
      <td><table width="960" border="0" cellspacing="0" cellpadding="0" bgcolor="#ffdab9">
        <tr>
          <td>&nbsp;</td>
        </tr>
        <tr>
          <td><table width="960" border="0" cellspacing="0" cellpadding="0">
            <tr>
              <td><table width="960" border="0" cellspacing="0" cellpadding="0">
                <tr>
                  <td width="100">&nbsp;</td>
                  <td><table width="760" border="1" cellspacing="1" cellpadding="0">
                    <tr>
                      <td height="30" align="center" bgcolor="#333333" class="AddNewHeader">
                        Edit Form</td>
                    </tr>
                    <tr bgcolor="#DFDFDF">
                      <td><table width="760" border="1" cellspacing="0" cellpadding="0">
                        <tr>
                          <td width="200" height="30" align="right">mode : </td>
                          <td height="30"><label for="mode"></label>
                          <select name="mode" id="mode">
                            <?php
                                $sql2 = "SELECT * FROM mode";
                                $query2 = mysql_query($sql2) or die(mysql_error());
                                $row2 = mysql_num_rows($query2);
                                for($i=1;$i<=$row2;$i++)
                                {
                                    $result2 = mysql_fetch_array($query2);
                                }
                                ?>

```

```
<option>  
    <?php  
        if($result2['mode_id']==$result['mode_id']){  
            echo "selected"; //ถ้าตัวที่ดึงจาก result2 กับ result มันตรงกัน  
        }  
        ?>  
        value="<?php echo $result2['mode_id']; ?>"  
        <?php echo $result2['mode_name']; ?>  
    </option>  
    <?php } ?>  
</select></td>  
</tr>  
<tr>  
    <td width="200" height="30" align="right">pic_num : </td>  
    <td height="30"><label for="pic_num"></label>  
        <input name="pic_num" type="text" id="pic_num"  
            value="<?php echo $result['pic_num']; ?>"  
            size="10" /></td>  
</tr>  
<tr>  
    <td width="200" height="30" align="right">image : </td>  
    <td height="30">  
  
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
  
          
        <p>  
            <input name="img" type="file" id="img" size="50" />  
        </p></td>  
</tr>  
<tr>  
    <td width="200" height="30" align="right">game_answer : </td>  
    <td height="30"><label for="game_hint"></label>  
        <input name="game_answer" type="text" id="game_answer"  
            value="<?php echo $result['game_answer']; ?>" size="20" /></td>  
</tr>  
<tr>  
    <td width="200" height="30" align="right">game_hint : </td>  
    <td height="30"><label for="game_hint"></label>  
        <input name="game_hint" type="text" id="game_hint"  
            value="<?php echo $result['game_hint']; ?>" size="20" /></td>  
</tr>  
<tr>  
    <td width="200" height="30" align="right">game_choice1 : </td>  
    <td height="30"><label for="game_choice1"></label>  
        <input name="game_choice1" type="text" id="game_choice1"  
            value="<?php echo $result['game_choice1']; ?>" size="10" /></td>  
</tr>  
<tr>  
    <td width="200" height="30" align="right">game_choice2 : </td>  
    <td height="30"><label for="game_choice2"></label>  
        <input name="game_choice2" type="text" id="game_choice2"  
            value="<?php echo $result['game_choice2']; ?>" size="10" /></td>  
</tr>
```



```

        </tr>
    </table></td>
</tr>
<tr>
    <td>&nbsp;</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
<tr>
    <td></td>
</tr>
</table>
</form>
</body>
</html>

```

12. หน้า EditSave.php

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<?php
    include('connect.php');
    if($_POST['game_answer']=="")
    {
        echo "<script>window.location='index.php';</script>";
        exit();
    }
    $game_id = $_POST['id'];
    $type = $_POST['type'];
    $mode = $_POST['mode'];
    $pic_num = $_POST['pic_num'];
    $game_answer = $_POST['game_answer'];
    $game_hint = $_POST['game_hint'];
    $game_choice1 = $_POST['game_choice1'];
    $game_choice2 = $_POST['game_choice2'];
    $game_choice3 = $_POST['game_choice3'];
    $game_choice4 = $_POST['game_choice4'];
    $game_choice5 = $_POST['game_choice5'];
    $game_choice6 = $_POST['game_choice6'];
    $game_choice7 = $_POST['game_choice7'];
    $game_choice8 = $_POST['game_choice8'];

    $file = $_FILES['img']['name'];
    $typefile = $_FILES['img']['type'];
    $sizefile = $_FILES['img']['size'];
    if ( strcmp ( $type, substr($typefile,6)) != 0 ){
        $game_pic_type = $type;
    }

```

```

else{
    $game_pic_type = substr($typefile,6);
}
$Nfilename = $mode."_".$pic_num."_".$game_pic_type;
$linkIMG = "http://chuwubbandroid.orgfree.com/game_images/" . $Nfilename;
//จะเก็บข้อมูลเป็น array แสดงว่าในตัวแปร img ข้อมูลจะเก็บเป็น array
////////////////////////////////////
if($sizefile>0)
{
    if($sizefile>300*1024) //300*1024 = 300kb
    {
        echo "<script>alert('ไฟล์รูปใหญ่เกินไป กรุณาใส่ไฟล์รูปที่มีขนาดไม่เกิน 300kb ด้วยครับ');";
        history.back();</script>";
        exit();
    }
}
move_uploaded_file($_FILES['img']['tmp_name'], "game_images/" . $Nfilename);
//uploadไฟล์แล้ว ให้เอาไปวางไว้ตรงไหน
////////////////////////////////////

$sql = "UPDATE game
        SET mode_id = '$mode', pic_num = '$pic_num', game_image = '$linkIMG',
            game_answer = '$game_answer', game_hint = '$game_hint',
            game_choice1 = '$game_choice1', game_choice2 = '$game_choice2',
            game_choice3 = '$game_choice3', game_choice4 = '$game_choice4',
            game_choice5 = '$game_choice5', game_choice6 = '$game_choice6',
            game_choice7 = '$game_choice7', game_choice8 = '$game_choice8',
            game_pic_type = '$game_pic_type'
        WHERE game_id = '$game_id'";

$query = mysql_query($sql) or die( mysql_error() );

mysql_close($conn);
?>

<script>
    alert('แก้ไขข้อมูลเรียบร้อยแล้วครับ');
    window.location='EditForm.php?id=<?php echo $game_id; ?>
    &type=<?php echo $game_pic_type; ?>';
</script>

```

13. หน้า Feed.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>หน้า BackStage</title>

<style type="text/css">
head table {
    font-family: "Century Gothic";
}
.tableHead {
    font-family: "Century Gothic";
    font-weight: bold;
    font-size: 14px;
    color: #FF6;
}
.editDelete {
    font-family: "Century Gothic";
    font-size: 12px;
}
.previousNext {
    font-family: "Century Gothic";
    font-size: 14px;
    font-weight: bold;
    color: #36F;
}
.content {
    font-family: supermarket;
    font-size: 14px;
}
</style>
</head>
<body onload=
"MM_preloadImages('images/feed_classicbtn_active.gif','images/feed_ghostmasterbtn_active.gif',
'images/menu_home_active.gif','images/menu_feed_active.gif',
'images/menu_add_active.gif')">
<form id="form1" name="form1" method="post" action="">
<table bgcolor="#ffdb9" width="960" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td></td>
</tr>
<tr>
<td><table width="960" border="0" cellspacing="0" cellpadding="0">
<tr>
<td><a href="index.php" onmouseout="MM_swapImgRestore()" onmouseover=
"MM_swapImage('Image6','','images/menu_home_active.gif',1)">
</a></td>
<td><a href="Feed.php" onmouseout="MM_swapImgRestore()" onmouseover=
"MM_swapImage('menu_feed','','images/menu_feed_active.gif',1)">
</a></td>
<td><a href="InsertForm.php" onmouseout="MM_swapImgRestore()" onmouseover=
"MM_swapImage('menu_add','','images/menu_add_active.gif',1)">
</a></td>

```

```

        <td></td>
    </tr>
</table></td>
</tr>
<tr>
    <td><table width="960" border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td><table width="960" border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td width="330">&nbsp;</td>
                    <td width="350"><table width="350" border="0" cellspacing="0" cellpadding="0">
                        <tr>
                            <td><a href="feed_classicgame.php" onmouseout="MM_swapImgRestore()"
                                onmouseover="MM_swapImage('Image7','','images/feed_classicbtn_active.gif',1)">
                                </a></td>
                        </tr>
                        <tr>
                            <td>&nbsp;</td>
                        </tr>
                        <tr>
                            <td><a href="feed_ghostmaster.php" onmouseout="MM_swapImgRestore()"
                                onmouseover="MM_swapImage('Image8','','images/feed_ghostmasterbtn_active.gif',1)">
                                </a>
                            </td>
                        </tr>
                    </table></td>
                    <td>&nbsp;</td>
                </tr>
            </table></td>
        </tr>
        <tr>
            <td><table width="960" border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td><table width="960" border="0" cellspacing="0" cellpadding="0">
                        <tr>
                            <td>&nbsp;</td>
                        </tr>
                    </table></td>
                </tr>
                <tr>
                    <td>&nbsp;</td>
                </tr>
            </table></td>
        </tr>
        <tr>
            <td><table width="960" border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td><table width="960" border="0" cellspacing="0" cellpadding="0">
                        <tr>
                            <td></td>
                        </tr>
                    </table>
                </tr>
            </table>
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

14. หน้า Delete.php

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<?php
    if($_GET['id']=="")
    {
        echo "<script>window.location='index.php'; </script>";
        exit();
    }
    $id = $_GET['id'];
    $mode = $_GET['mode'];
    $pic_num = $_GET['pic_num'];
    $type = $_GET['type'];

    include('connect.php');
    $myFile = "game_images/".$mode."_".$pic_num."_".$type;

    $sql1 = "DELETE FROM game
            WHERE game_id = $id";

    $query1 = mysql_query($sql1) or die ( mysql_error() );

    unlink($myFile);
    mysql_close($conn);
?>

<script>
    alert('ลบข้อมูลเรียบร้อยแล้วครับ');
    window.location='index.php';
</script>
```

15. changeStatus.php

```
<?php
if (isset($_GET["regId"])) {
    $regId = $_GET["regId"];
    $status = $_GET["status"];

    include_once("GCM.php");
    $db = new DB_Functions();
    $gcm = new GCM();

    $result = $db->changeStatus($regId, $status);
    echo $result;
}
?>
```

16. หน้า feed_all.php

```
<?php
    $host = "localhost";
    $user = "578032";
    $pass = "Fvhc,jg0hk";
    $conn = @mysql_connect($host,$user,$pass) or die("cannot connect Database");

    $dbname = "578032";
    $db = mysql_select_db($dbname,$conn) or die("cannot select Database");
    mysql_query("SET NAMES utf8");

    $sql = "SELECT * FROM game g , mode m
            WHERE g.mode_id = m.mode_id and game_id > 42";
    $query = mysql_query($sql) or die(mysql_error());
    $json = array();

    while ($result = mysql_fetch_array($query, MYSQL_ASSOC)){
        $json['feed'][] = array(
            'mode' => (int)$result['mode_id'],
            'name' => $result['mode_name'],
            'id' => (int)$result['game_id'],
            'number' => (int)$result['pic_num'],
            'answer' => $result['game_answer'],
            'hint' => $result['game_hint'],
            'link_image' => $result['game_image'],
            'choice1' => $result['game_choice1'],
            'choice2' => $result['game_choice2'],
            'choice3' => $result['game_choice3'],
            'choice4' => $result['game_choice4'],
            'choice5' => $result['game_choice5'],
            'choice6' => $result['game_choice6'],
            'choice7' => $result['game_choice7'],
            'choice8' => $result['game_choice8']
        );
    }
    echo json_encode($json);
?>
```

17. หน้า feed_classicgame.php

```
<?php
    $host = "localhost";
    $user = "578032";
    $pass = "Fvhc,jg0hk";
    $conn = @mysql_connect($host,$user,$pass) or die("cannot connect Database");

    $dbname = "578032";
    $db = mysql_select_db($dbname,$conn) or die("cannot select Database");
    mysql_query("SET NAMES utf8");
    $sql = "SELECT * FROM game
            WHERE mode_id = 1";
    $query = mysql_query($sql) or die(mysql_error());
    $json = array(
        'mode' => 1,
        'name' => "คลาสสิกเกมส์"
    );

    while ($result = mysql_fetch_array($query, MYSQL_ASSOC))
    {
        $json['feed'][] = array(
            'id' => (int)$result['game_id'],
            'number' => (int)$result['pic_num'],
            'answer' => $result['game_answer'],
            'hint' => $result['game_hint'],
            'link_image' => $result['game_image'],
            'choice1' => $result['game_choice1'],
            'choice2' => $result['game_choice2'],
            'choice3' => $result['game_choice3'],
            'choice4' => $result['game_choice4'],
            'choice5' => $result['game_choice5'],
            'choice6' => $result['game_choice6'],
            'choice7' => $result['game_choice7'],
            'choice8' => $result['game_choice8']
        );
    }
    echo json_encode($json);
?>
```

18. หน้า feed_ghostmaster.php

```
<?php
    $host = "localhost";
    $user = "578032";
    $pass = "Fvhc,jg0hk";
    $conn = @mysql_connect($host,$user,$pass) or die("cannot connect Database");
    $dbname = "578032";
    $db = mysql_select_db($dbname,$conn) or die("cannot select Database");
    mysql_query("SET NAMES utf8");
    $sql = "SELECT * FROM game
            WHERE mode_id = 2";
    $query = mysql_query($sql) or die(mysql_error());
    $json = array(
        'mode' => 2,
        'name' => "โกสต์มาสเตอร์"
    );

    while ($result = mysql_fetch_array($query, MYSQL_ASSOC))
    {
        $json['feed'][] = array(
            'id' => (int)$result['game_id'],
            'number' => (int)$result['pic_num'],
            'answer' => $result['game_answer'],
            'hint' => $result['game_hint'],
            'link_image' => $result['game_image'],
            'choice1' => $result['game_choice1'],
            'choice2' => $result['game_choice2'],
            'choice3' => $result['game_choice3'],
            'choice4' => $result['game_choice4'],
            'choice5' => $result['game_choice5'],
            'choice6' => $result['game_choice6'],
            'choice7' => $result['game_choice7'],
            'choice8' => $result['game_choice8']
        );
    }
    echo json_encode($json);
?>
```

ภาคผนวก ข

Code Android ของ Application เกมเวทีทอง

TNI

1. หน้า SplashScreen.java

```
package com.example.gamewaeteethong;

import android.app.Activity;
import android.content.Intent;
import android.graphics.drawable.AnimationDrawable;
import android.os.Bundle;
import android.os.Handler;
import android.widget.ImageView;

public class SplashScreen extends Activity {

    ImageView bg_pic;
    AnimationDrawable splash;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splashscreen);
        bg_pic = (ImageView) findViewById(R.id.bg_pic);
        splash = (AnimationDrawable) bg_pic.getBackground();
        bg_pic.post(new Starter());

        new Handler().postDelayed(new Runnable() {
            public void run() {
                Intent mainIntent = new Intent(SplashScreen.this, MainActivity.class);
                startActivity(mainIntent);
                SplashScreen.this.finish();
            }
        }, 3000);
    }

    class Starter implements Runnable {
        public void run() {
            splash.start();
        }
    }
}
```

2. หน้า MainActivity.java

```
package com.example.gamewaeteethong;

import static com.example.gamewaeteethong.CommonUtilities.DISPLAY_MESSAGE_ACTION;
import static com.example.gamewaeteethong.CommonUtilities.EXTRA_MESSAGE;
import static com.example.gamewaeteethong.CommonUtilities.SENDER_ID;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import com.google.android.gms.gcm.GCMRegistrar;
import android.app.Activity;
import android.app.AlertDialog;
```

```

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.SharedPreferences;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.content.pm.PackageManager.NameNotFoundException;
import android.content.pm.Signature;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends Activity {
    public static SharedPreferences sharedPreferences;
    ImageView button_start, button_howtoplay,
        button_settings, button_download;
    Boolean sound;
    Context context;
    // Connection detector
    ConnectionDetector cd;
    // AsyncTask
    AsyncTask<Void, Void, Void> mRegisterTask;

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

        context = this;

        // View Matching
        button_start = (ImageView) findViewById(R.id.button_start);
        button_howtoplay = (ImageView) findViewById(R.id.button_howtoplay);
        button_settings = (ImageView) findViewById(R.id.button_setting);
        button_download = (ImageView) findViewById(R.id.button_download);

        // logKeyHash();
        button_start.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent = new Intent(MainActivity.this, choosemode.class);
                startActivity(intent);
                finish();
                overridePendingTransition(R.layout.mainfadein,
                    R.layout.mainfadeout);
            }
        });
    }
}

```

```

button_howtoplay.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent intent = new Intent(MainActivity.this, HowToPlay.class);
        startActivity(intent);
        finish();
        overridePendingTransition(R.layout.mainfadein,
            R.layout.mainfadeout);
    }
});

button_settings.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        Intent intent = new Intent(MainActivity.this, Settings.class);
        startActivity(intent);
        finish();
        overridePendingTransition(R.layout.mainfadein,
            R.layout.mainfadeout);
    }
});

button_download.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        Intent intent = new Intent(MainActivity.this, download.class);
        startActivity(intent);
        finish();
        overridePendingTransition(R.layout.mainfadein,
            R.layout.mainfadeout);
    }
});

// Check if Internet present
if (!lcd.isConnectingToInternet()) {
    // If not connected
    // skip gcm registration
} else {
    // Make sure the device has the proper dependencies.
    GCMRegistrar.checkDevice(this);
    // Make sure the manifest was properly set - comment out this line
    // while developing the app, then uncomment it when it's ready.
    GCMRegistrar.checkManifest(this);
    registerReceiver(mHandleMessageReceiver, new IntentFilter(
        DISPLAY_MESSAGE_ACTION));
    // Get GCM registration id
    final String regId = GCMRegistrar.getRegistrationId(this);
    Log.d("regId", "regId : " + regId);

    // Check if regid already presents
    if (regId.equals("")) {
        // Registration is not present, register now with GCM
        GCMRegistrar.register(this, SENDER_ID);
    }
}

```

```

else {
    // Device is already registered on GCM
    if (GCMRegistrar.isRegisteredOnServer(this)) {
        // Skip
    } else {
        // Try to register again, but not in the UI thread.
        // It's also necessary to cancel the thread onDestroy(),
        // hence the use of AsyncTask instead of a raw thread.
        final Context context = this;
        mRegisterTask = new AsyncTask<Void, Void, Void>() {

            @Override
            protected Void doInBackground(Void... params) {
                // Register on our server
                // On server creates a new user
                ServerUtilities.register(context, regId);
                return null;
            }

            @Override
            protected void onPostExecute(Void result) {
                mRegisterTask = null;
            }
        };
        mRegisterTask.execute(null, null, null);
    }
}

public void logKeyHash() {
    try {
        PackageInfo info = getPackageManager().getPackageInfo(
            getPackageName(), PackageManager.GET_SIGNATURES);
        for (Signature signature : info.signatures) {
            MessageDigest md = MessageDigest.getInstance("SHA");
            md.update(signature.toByteArray());
            Log.d("KeyHash",
                Base64.encodeToString(md.digest(), Base64.DEFAULT));
        }
    } catch (NameNotFoundException e) {
        e.printStackTrace();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
}

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
    builder.setTitle("เกมเวทย์ธอง")
        .setIcon(getResources().getDrawable(
            R.drawable.Logo_gamewaeteethong))
        .setMessage("คุณต้องการจะออกจากโปรแกรม")
        .setPositiveButton("ยกเลิก",
            new DialogInterface.OnClickListener() {

                public void onClick(DialogInterface dialog,
                    int which) {
                    // TODO Auto-generated method stub
                    dialog.dismiss();
                }
            })
        .setNegativeButton("ตกลง", null);
    builder.create().show();
}
}

```

```

        .setNegativeButton("%s",
            new DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog,
                    int which) {
                    // TODO Auto-generated method stub
                    MainActivity.this.finish();
                }
            });

        AlertDialog alert = builder.create();
        alert.show();
    }

    public static void androidOS() {
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
                .permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
    }

    private final BroadcastReceiver mHandleMessageReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String newMessage = intent.getExtras().getString(EXTRA_MESSAGE);
            // Waking up mobile if it is sleeping
            WakeLocker.acquire(context);
            // Take appropriate action on this message
            // depending upon your app requirement
            // For now i am just displaying it on the screen
            Toast.makeText(context, "New Message: " + newMessage,
                Toast.LENGTH_LONG).show();
            // Releasing wake lock
            WakeLocker.release();
        }
    };

    @Override
    protected void onDestroy() {
        if (mRegisterTask != null) {
            mRegisterTask.cancel(true);
        }
        try {
            unregisterReceiver(mHandleMessageReceiver);
            GCMRegistrar.onDestroy(this);
        } catch (Exception e) {
            Log.e("UnRegister Receiver Error", "> " + e.getMessage());
        }
        super.onDestroy();
    }
}

```

3. หน้า AlertDialogManager.java

```
package com.example.gamewaeteethong;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;

public class AlertDialogManager {

    @SuppressWarnings("deprecation")
    public void showAlertDialog(Context context, String title, String message, Boolean status) {
        AlertDialog alertDialog = new AlertDialog.Builder(context).create();

        // Setting Dialog Title
        alertDialog.setTitle(title);

        // Setting Dialog Message
        alertDialog.setMessage(message);

        // Setting OK Button
        alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
            }
        });

        // Showing Alert Message
        alertDialog.show();
    }
}
```

4. WakeLocker.java

```
package com.example.gamewaeteethong;

import android.content.Context;
import android.os.PowerManager;

public abstract class WakeLocker {
    private static PowerManager.WakeLock wakeLock;

    public static void acquire(Context context) {
        if (wakeLock != null) wakeLock.release();

        PowerManager pm = (PowerManager) context.getSystemService(Context.POWER_SERVICE);
        wakeLock = pm.newWakeLock(PowerManager.FULL_WAKE_LOCK |
            PowerManager.ACQUIRE_CAUSES_WAKEUP |
            PowerManager.ON_AFTER_RELEASE, "WakeLocker");
        wakeLock.acquire();
    }

    public static void release() {
        if (wakeLock != null) wakeLock.release(); wakeLock = null;
    }
}
```

5. ConnectionDetector.java

```
package com.example.gamewaeteethong;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;

public class ConnectionDetector {
    private Context _context;

    public ConnectionDetector(Context context){
        this._context = context;
    }

    /**Checking for all possible internet providers**/
    public boolean isConnectingToInternet(){
        ConnectivityManager connectivity = (ConnectivityManager) _context
            .getSystemService(Context.CONNECTIVITY_SERVICE);

        if (connectivity != null){
            NetworkInfo[] info = connectivity.getAllNetworkInfo();
            if (info != null)
                for (int i = 0; i < info.length; i++)
                    if (info[i].getState() == NetworkInfo.State.CONNECTED){
                        return true;
                    }
        }
        return false;
    }
}
```

6. CommonUtilities.java

```
package com.example.gamewaeteethong;

import android.content.Context;
import android.content.Intent;

public class CommonUtilities {

    // give your server registration url here
    static final String SERVER_URL = "http://chuwubbandroid.orgfree.com/register.php";
    static final String SERVER_URL_changeStatus = "http://chuwubbandroid.orgfree.com/changeStatus.php";
    // Google project id
    static final String SENDER_ID = "884467770136";
    static final String TAG = "INSTITUTE";
    static final String DISPLAY_MESSAGE_ACTION = "com.example.gamewaeteethong.DISPLAY_MESSAGE";
    static final String EXTRA_MESSAGE = "message";

    static void displayMessage(Context context, String message) {
        Intent intent = new Intent(DISPLAY_MESSAGE_ACTION);
        intent.putExtra(EXTRA_MESSAGE, message);
        context.sendBroadcast(intent);
    }
}
```

7. JSONParser.java

```
package com.example.gamewaeteethong;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;
import android.util.Log;

public class JSONParser {
    InputStream is = null;
    JSONObject jsonObj = null;
    String json = "";
    // constructor
    public JSONParser() {

    }
    public JSONObject getJSONFromUrl(String url) {
        // Making HTTP request
        try {
            // defaultHttpClient
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);
            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            BufferedReader reader = new BufferedReader(new InputStreamReader(
                is, "iso-8859-1"), 8);
            StringBuilder sb = new StringBuilder();
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            is.close();
            json = sb.toString();

            Log.d("json parser string", json);
        } catch (Exception e) {
            Log.e("Buffer Error", "Error converting result " + e.toString());
        }

        // try parse the string to a JSON object
        try {
            jsonObj = new JSONObject(json);
        } catch (JSONException e) {
            Log.e("JSON Parser", "Error parsing data " + e.toString());
        }
        // return JSON String
        return jsonObj;
    }
}
```

8. GCMIntentService.java

```
package com.example.gamewaeteethong;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import android.util.Log;

import com.google.android.gcm.GCMBaseIntentService;

import static com.example.gamewaeteethong.CommonUtilities.SENDER_ID;
import static com.example.gamewaeteethong.CommonUtilities.displayMessage;

public class GCMIntentService extends GCMBaseIntentService {

    private static final String TAG = "GCMIntentService";
    public static SharedPreferences sharedPreferences;
    public static boolean notification = true;

    public GCMIntentService() {
        super(SENDER_ID);
    }

    /**Method called on device registered**/
    @Override
    protected void onRegistered(Context context, String registrationId) {
        Log.i(TAG, "Device registered: regId = " + registrationId);
        // displayMessage(context, "Your device registered with GCM");
        ServerUtilities.register(context, registrationId);
    }

    /**Method called on device un registered**/
    @Override
    protected void onUnregistered(Context context, String registrationId) {
        Log.i(TAG, "Device unregistered");
        ServerUtilities.unregister(context, registrationId);
    }

    /**Method called on Receiving a new message**/
    @Override
    protected void onMessage(Context context, Intent intent) {
        Log.i(TAG, "Received message");
        String message = intent.getExtras().getString("message");
        // check settings6
        sharedPreferences = PreferenceManager
            .getDefaultSharedPreferences(context);
        notification = sharedPreferences.getBoolean("Notification", true);
        // notifies user
        if (notification == true) {
            generateNotification(context, message);
        }
    }

    /**Method called on receiving a deleted message**/
    @Override
    protected void onDeletedMessages(Context context, int total) {
        Log.i(TAG, "Received deleted messages notification");
        String message = getString(R.string.gcm_deleted, total);
        // notifies user
        generateNotification(context, message);
    }
}
```

```

/**
 * Method called on Error
 */
@Override
public void onError(Context context, String errorId) {
    Log.i(TAG, "Received error: " + errorId);
    // displayMessage(context, getString(R.string.gcm_error, errorId));
}

@Override
protected boolean onRecoverableError(Context context, String errorId) {
    // log message
    Log.i(TAG, "Received recoverable error: " + errorId);
    // displayMessage(context, getString(R.string.gcm_recoverable_error,
    // errorId));
    return super.onRecoverableError(context, errorId);
}

/**
 * Issues a notification to inform the user that server has sent a message.
 */
private static void generateNotification(Context context, String message) {
    int icon = R.drawable.logo_gamewaeteethong;
    long when = System.currentTimeMillis();
    NotificationManager notificationManager = (NotificationManager) context
        .getSystemService(Context.NOTIFICATION_SERVICE);
    Notification notification = new Notification(icon, message, when);

    String title = context.getString(R.string.app_name);

    Intent notificationIntent = new Intent(context, download.class);
    // set intent so it does not start a new activity
    notificationIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP
        | Intent.FLAG_ACTIVITY_SINGLE_TOP);

    PendingIntent intent = PendingIntent.getActivity(context, 0,
        notificationIntent, 0);
    notification.setLatestEventInfo(context, title, message, intent);
    notification.flags |= Notification.FLAG_AUTO_CANCEL;

    // Play default notification sound
    notification.defaults |= Notification.DEFAULT_SOUND;

    // Vibrate if vibrate is enabled
    notification.defaults |= Notification.DEFAULT_VIBRATE;
    notificationManager.notify(0, notification);
}
}

```

9. ServerUtilities.java

```
package com.example.gamewaeteethong;

import static com.example.gamewaeteethong.CommonUtilities.SERVER_URL;
import static com.example.gamewaeteethong.CommonUtilities.SERVER_URL_changeStatus;
import static com.example.gamewaeteethong.CommonUtilities.TAG;
import java.io.IOException;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Random;
import android.content.Context;
import android.os.StrictMode;
import android.util.Log;
import com.google.android.gcm.GCMRegistrar;

public final class ServerUtilities {
    private static final int MAX_ATTEMPTS = 5;
    private static final int BACKOFF_MILLI_SECONDS = 2000;
    private static final Random random = new Random();

    /** Register this account/device pair within the server. */
    static void register(final Context context, final String regId) {
        Log.i(TAG, "registering device (regId = " + regId + ")");
        String serverUrl = SERVER_URL;
        Map<String, String> params = new HashMap<String, String>();
        params.put("regId", regId);
        params.put("status", "1");
        long backoff = BACKOFF_MILLI_SECONDS + random.nextInt(1000);
        // Once GCM returns a registration id, we need to register on our server
        // As the server might be down, we will retry it a couple times.
        for (int i = 1; i <= MAX_ATTEMPTS; i++) {
            Log.d(TAG, "Attempt #" + i + " to register");
            try {
                post(serverUrl, params);
                GCMRegistrar.setRegisteredOnServer(context, true);
                return;
            } catch (IOException e) {
                Log.e(TAG, "Failed to register on attempt " + i + ":" + e);
                if (i == MAX_ATTEMPTS) {
                    break;
                }
            }
            try {
                Log.d(TAG, "Sleeping for " + backoff + " ms before retry");
                Thread.sleep(backoff);
            } catch (InterruptedException e1) {
                // Activity finished before we complete - exit.
                Log.d(TAG, "Thread interrupted: abort remaining retries!");
                Thread.currentThread().interrupt();
                return;
            }
        }
    }
}
```

```

        // increase backoff exponentially
        backoff *= 2;
    }
}

/** Unregister this account/device pair within the server.*/
static void unregister(Context context, final String regId) {
    Log.i(TAG, "unregistering device (regId = " + regId + ")");
    String serverUrl = SERVER_URL + "/unregister";
    Map<String, String> params = new HashMap<String, String>();
    params.put("regId", regId);
    try {
        post(serverUrl, params);
        GCMRegistrar.setRegisteredOnServer(context, false);
    } catch (IOException e) {
    }
}

static void changeStatus(final String regId, String status){
    String serverUrl = SERVER_URL_changeStatus;
    Map<String, String> params = new HashMap<String, String>();
    params.put("regId", regId);
    params.put("status", status);
    try {
        post(serverUrl, params);
        return;
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

/**
 * Issue a POST request to the server.
 * @param endpoint POST address.
 * @param params request parameters.
 * @throws IOException propagated from POST.
 */
private static void post(String endpoint, Map<String, String> params)
    throws IOException {

    URL url;
    try {
        url = new URL(endpoint);
    } catch (MalformedURLException e) {
        throw new IllegalArgumentException("invalid url: " + endpoint);
    }
    StringBuilder bodyBuilder = new StringBuilder();
    Iterator<Entry<String, String>> iterator = params.entrySet().iterator();
    // constructs the POST body using the parameters
    while (iterator.hasNext()) {
        Entry<String, String> param = iterator.next();
        bodyBuilder.append(param.getKey()).append('=')
            .append(param.getValue());
        if (iterator.hasNext()) {
            bodyBuilder.append('&');
        }
    }
}

```

```

String body = bodyBuilder.toString();
Log.v(TAG, "Posting '" + body + "' to " + url);
byte[] bytes = body.getBytes();
URLConnection conn = null;
try {
    Log.e("URL", "> " + url);
    conn = (URLConnection) url.openConnection();
    conn.setDoOutput(true);
    conn.setUseCaches(false);
    conn.setFixedLengthStreamingMode(bytes.length);
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Content-Type",
        "application/x-www-form-urlencoded; charset=UTF-8");
    // post the request
    OutputStream out = conn.getOutputStream();
    out.write(bytes);
    out.close();
    // handle the response
    int status = conn.getResponseCode();
    if (status != 200) {
        throw new IOException("Post failed with error code " + status);
    }
} finally {
    if (conn != null) {
        conn.disconnect();
    }
}
}
}

```

10. DataGame.java

```

package com.example.gamewaeteethong;

public class DataGame {

    String choice1,choice2,choice3,choice4,
        choice5,choice6,choice7,choice8,
        answer, hint, image;

    int pic_number;

    public String getHint() {
        return hint;
    }
    public void setHint(String hint) {
        this.hint = hint;
    }
    public String getAnswer() {
        return answer;
    }
    public void setAnswer(String answer) {
        this.answer = answer;
    }
}

```

```

public int getPic_number() {
    return pic_number;
}
public void setPic_number(int pic_number) {
    this.pic_number = pic_number;
}
public String getChoice1() {
    return choice1;
}
public void setChoice1(String choice1) {
    this.choice1 = choice1;
}
public String getChoice2() {
    return choice2;
}
public void setChoice2(String choice2) {
    this.choice2 = choice2;
}
public String getChoice3() {
    return choice3;
}
public void setChoice3(String choice3) {
    this.choice3 = choice3;
}
public String getChoice4() {
    return choice4;
}
public void setChoice4(String choice4) {
    this.choice4 = choice4;
}
public String getChoice5() {
    return choice5;
}
public void setChoice5(String choice5) {
    this.choice5 = choice5;
}
public String getChoice6() {
    return choice6;
}
public void setChoice6(String choice6) {
    this.choice6 = choice6;
}
public String getChoice7() {
    return choice7;
}
public void setChoice7(String choice7) {
    this.choice7 = choice7;
}
public String getChoice8() {
    return choice8;
}
public void setChoice8(String choice8) {
    this.choice8 = choice8;
}
public String getImage() {
    return image;
}
public void setImage(String image) {
    this.image = image;
}
}

```

11. choosemode.java

```
package com.example.gamewaeteethong;

import java.util.ArrayList;
import com.example.gamewaeteethong.db.DBHelper;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageView;
import android.widget.Toast;

public class choosemode extends Activity {

    ImageView button_mode_classicgame, button_mode_ghostmaster, home_purple;
    Context context;
    int size;
    DBHelper helper;
    ClassicGameDataEntry DataEntry;
    ArrayList<ClassicGameDataEntry> DataEntryArray = new ArrayList<ClassicGameDataEntry>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.choosemode_layout);

        context = this;

        button_mode_classicgame = (ImageView) findViewById(R.id.button_mode_classicgame);
        button_mode_ghostmaster = (ImageView) findViewById(R.id.button_mode_ghostmaster);
        home_purple = (ImageView) findViewById(R.id.home_purple);

        button_mode_classicgame.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent = new Intent(choosemode.this, classicgame.class);
                startActivity(intent);
                finish();
                overridePendingTransition(R.layout.mainfadein,
                    R.layout.mainfadeout);
            }
        });

        button_mode_ghostmaster.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                helper = new DBHelper(context);
                helper.getWritableDatabase();
                DataEntryArray = helper.SelectGhost();
                size = DataEntryArray.size();
                if (size == 0) {
                    Toast.makeText(
                        context,
                        "กรุณาทำน้โหมตกานก่อน \n กรุณาระโใช้ฐานโหมทด้โคคช",
                        Toast.LENGTH_LONG).show();
                } else {
                    Intent intent = new Intent(choosemode.this,
                        ghostmaster.class);
                }
            }
        });
    }
}
```

```

        startActivity(intent);
        finish();
        overridePendingTransition(R.layout.mainfadein,
            R.layout.mainfadeout);
    }
}
});

home_purple.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent intent = new Intent(choosemode.this, MainActivity.class);
        startActivity(intent);
        choosemode.this.finish();
        overridePendingTransition(R.layout.mainfadein,
            R.layout.mainfadeout);
    }
});
}

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    Intent intent = new Intent(choosemode.this, MainActivity.class);
    startActivity(intent);
    choosemode.this.finish();
    overridePendingTransition(R.layout.mainfadein, R.layout.mainfadeout);
}
}
}

```

12. HowToPlay.java

```

package com.example.gamewaeteethong;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.view.PagerAdapter;
import android.support.v4.view.ViewPager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.ImageView;
import com.viewpagerindicator.CirclePageIndicator;

public class HowToPlay extends Activity {
    public static int[] howtoplay_image = { R.drawable.howto_pic1,
        R.drawable.howto_pic2, R.drawable.howto_pic3,
        R.drawable.howto_pic4, R.drawable.howto_pic5,
        R.drawable.howto_pic6, R.drawable.howto_pic7,
        R.drawable.howto_pic8, R.drawable.howto_pic9,
        R.drawable.howto_pic10, R.drawable.http_ghost_0,
        R.drawable.http_ghost_1, R.drawable.http_ghost_2,
        R.drawable.http_ghost_3, R.drawable.http_ghost_4,
        R.drawable.http_ghost_5, R.drawable.http_ghost_6 };
    Context context;
    ImageView backpage;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.howtoplay_layout);

    context = this;
    backhome = (ImageView) findViewById(R.id.backhome);

    backhome.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            Intent intent = new Intent(HowToPlay.this, MainActivity.class);
            startActivity(intent);
            finish();
            overridePendingTransition(R.layout.mainfadein,
                R.layout.mainfadeout);
        }
    });

    ViewPager pager = (ViewPager) findViewById(R.id.pager);
    CirclePageIndicator circlePageIndicator = (CirclePageIndicator) findViewById(R.id.indicator);

    ViewPagerAdapter adapter = new ViewPagerAdapter(context,
        howtoplay_image);
    pager.setAdapter(adapter);
    circlePageIndicator.setViewPager(pager);
}

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    Intent intent = new Intent(HowToPlay.this, MainActivity.class);
    startActivity(intent);
    HowToPlay.this.finish();
    overridePendingTransition(R.layout.mainfadein, R.layout.mainfadeout);
}

static class ViewPagerAdapter extends PagerAdapter {

    Context context;
    int[] howtoplay_image;
    ImageView classic_pic;
    LayoutInflater inflater;

    public ViewPagerAdapter(Context context, int[] howtoplay_image) {
        // TODO Auto-generated constructor stub

        this.context = context;
        this.howtoplay_image = howtoplay_image;
    }

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return howtoplay_image.length;
    }
}

```

```

@Override
public boolean isViewFromObject(View view, Object object) {
    // TODO Auto-generated method stub
    return view.equals(object);
}

@Override
public Object instantiateItem(ViewGroup container, int position) {

    // // Slide Pic ////
    ImageView pic = new ImageView(context);
    pic.setImageResource(howtoplay_image[position]);
    ((ViewPager) container).addView(pic, 0);
    return pic;
}

@Override
public void destroyItem(ViewGroup container, int position, Object object) {
    ((ViewPager) container).removeView((View) object);
}
}
}

```

13. Settings.java

```

package com.example.gamewaeteethong;

import com.google.android.gcm.GCMRegistrar;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.OnSharedPreferenceChangeListener;
import android.os.AsyncTask;
import android.os.Bundle;
import android.preference.PreferenceActivity;
import android.preference.PreferenceManager;

public class Settings extends PreferenceActivity implements
    OnSharedPreferenceChangeListener {

    private static final int RESULT_SETTINGS = 1;
    private final String MY_WIDGET = "android.intent.action.CHANGED_SETTINGS";
    public static SharedPreferences sharedPreferences;
    public static boolean Notification = true;
    Context context;

    String status;
    // Connection detector
    ConnectionDetector cd;
    // Alert dialog manager
    AlertDialogManager alert = new AlertDialogManager();
    // AsyncTask
    AsyncTask mRegisterTask;

    String regId;

```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    addPreferencesFromResource(R.xml.preferences);
    context = this;
    cd = new ConnectionDetector(context);
    // check settings
    sharedPreferences = PreferenceManager
        .getDefaultSharedPreferences(context);
    Notification = sharedPreferences.getBoolean("Notification", true);
    if (!cd.isConnectingToInternet()) {
        // Internet Connection is not present
        alert.showAlertDialog(Settings.this, "คำเตือน",
            "หากท่านต้องการที่จะทำการแจ้งเตือน กรุณาเชื่อมต่ออินเทอร์เน็ตก่อนนะคะ", false);
        // stop executing code by return
        return;
    }
}

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    Intent intent = new Intent(Settings.this, MainActivity.class);
    startActivity(intent);
    finish();
    overridePendingTransition(R.layout.mainfadein,
        R.layout.mainfadeout);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case RESULT_SETTINGS:
            showUserSettings();
            break;
    }
}

@Override
public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
    // TODO Auto-generated method stub
    Intent intent = new Intent(MY_WIDGET);
    if (key.equals("Notification")) {
        // Check if Internet present
        if (!cd.isConnectingToInternet()) {
            // Internet Connection is not present
            alert.showAlertDialog(Settings.this, "แจ้งเตือน",
                "กรุณาเชื่อมต่ออินเทอร์เน็ตก่อนนะคะ", false);
            // stop executing code by return
            return;
        } else {
            sendBroadcast(intent);
            regId = GCMRegistrar.getRegistrationId(this);
            new changeStatusAsync().execute();
        }
    } else {
        sendBroadcast(intent);
    }
}
}

```

```

public class changeStatusAsync extends AsyncTask<String, Void, Void> {

    @Override
    protected Void doInBackground(String... params) {
        // TODO Auto-generated method stub
        sharedPreferences = PreferenceManager.getDefaultSharedPreferences(context);
        Notification = sharedPreferences.getBoolean("Notification", true);
        String status;
        if(Notification == true){
            status = "1";
        }else{
            status = "0";
        }
        ServerUtilities.changeStatus(regId, status);
        return null;
    }

    private void showUserSettings() {
        SharedPreferences sharedPreferences = PreferenceManager
            .getDefaultSharedPreferences(this);
        sharedPreferences.registerOnSharedPreferenceChangeListener(this);
    }

    @Override
    protected void onResume() {
        super.onResume();
        getPreferenceScreen().getSharedPreferences()
            .registerOnSharedPreferenceChangeListener(this);
    }

    @Override
    protected void onPause() {
        super.onPause();
        getPreferenceScreen().getSharedPreferences()
            .unregisterOnSharedPreferenceChangeListener(this);
    }
}

```



TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

14. download.java

```
package com.example.gamewaeteethong;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.util.ArrayList;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import com.example.gamewaeteethong.db.DBHelper;
import com.example.gamewaeteethong.db.storeImg;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.util.Log;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class download extends Activity {
    private String url;
    private static final String TAG_feed = "feed";
    JSONArray feed = null;
    private static final String TAG_mode = "mode";
    private static final String TAG_name = "name";
    private static final String TAG_id = "id";
    private static final String TAG_number = "number";
    private static final String TAG_answer = "answer";
    private static final String TAG_hint = "hint";
    private static final String TAG_Link = "link_image";
    private static final String TAG_choice1 = "choice1";
    private static final String TAG_choice2 = "choice2";
    private static final String TAG_choice3 = "choice3";
    private static final String TAG_choice4 = "choice4";
    private static final String TAG_choice5 = "choice5";
    private static final String TAG_choice6 = "choice6";
    private static final String TAG_choice7 = "choice7";
    private static final String TAG_choice8 = "choice8";
```

```

// //////////////////for show data //////////////////
private String mode;
private String name;
private String id;
private String number;
private String answer;
private String hint;
private String link;
private String txt_choice1;
private String txt_choice2;
private String txt_choice3;
private String txt_choice4;
private String txt_choice5;
private String txt_choice6;
private String txt_choice7;
private String txt_choice8;
// //////////////////

DBHelper helper;
storeImg imgHelper;
ClassicGameDataEntry DataEntry;
private ListViewAdapter listViewAdapter;

// Internet Connection detector
private ConnectionDetector cd;

// //Variable
ListView download_listview;
ImageView btn_download, homebtn;
LinearLayout linearLayout;
// Alert Dialog Manager
AlertDialogManager alert = new AlertDialogManager();

private ProgressDialog mProgressDialog;
public static final int DIALOG_DOWNLOAD_JSON_PROGRESS = 0;
public static final int DIALOG_DOWNLOAD_DATATOSQLITE_PROGRESS = 1;

ArrayList<ClassicGameDataEntry> DataEntryArray = new ArrayList<ClassicGameDataEntry>();
private ListViewHolder holder;
String first_picname, last_picname;
int size, size1, pos1;
boolean update = false, cancel;
Context context;

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.download);

    context = this;
    // connect to internet
    androidOS();

    download_listview = (ListView) findViewById(R.id.download_listview);
    btn_download = (ImageView) findViewById(R.id.btn_download);
    linearLayout = (LinearLayout) findViewById(R.id.relativeLayout);
    homebtn = (ImageView) findViewById(R.id.homebtn);

    cd = new ConnectionDetector(getApplicationContext());

```

```

homebtn.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        new Handler().postDelayed(new Runnable() {
            public void run() {
                Intent intent = new Intent(download.this,
                    MainActivity.class);
                startActivity(intent);
                finish();
                overridePendingTransition(R.layout.mainfadein,
                    R.layout.mainfadeout);
            }
        }, 1000);
    }
});

// Check Internet Connection
if (!cd.isConnectingToInternet()) {
    // If not connected
    alert.showAlertDialog(download.this, "Internet Connection",
        "กรุณารอสัญญาณอินเทอร์เน็ตสักครู่", false);
} else {
    // if Internet connected
    new DownloadJSONAsync().execute();
}
}

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    new Handler().postDelayed(new Runnable() {
        public void run() {
            Intent intent = new Intent(download.this,
                MainActivity.class);
            startActivity(intent);
            finish();
            overridePendingTransition(R.layout.mainfadein,
                R.layout.mainfadeout);
        }
    }, 1000);
}

@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case DIALOG_DOWNLOAD_JSON_PROGRESS:
            mProgressDialog = new ProgressDialog(context);
            mProgressDialog.setMessage("กำลังดาวน์โหลด...");
            mProgressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
            mProgressDialog.setCancelable(true);
            mProgressDialog.show();
            return mProgressDialog;
        case DIALOG_DOWNLOAD_DATATOSQLITE_PROGRESS:
            cancel = false;
            mProgressDialog = new ProgressDialog(context);
            mProgressDialog.setMessage("กำลังดาวน์โหลดไฟล์ข้อมูล...");
            mProgressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
            mProgressDialog.setMax(size1 - size);
    }
}

```

```

mProgressDialog.setButton(DialogInterface.BUTTON_NEGATIVE,
    "ยกเลิก", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            cancel = true;
            Toast.makeText(context, "การดาวน์โหลดถูกยกเลิก",
                Toast.LENGTH_LONG).show();
        }
    });
mProgressDialog.setProgress(0);
mProgressDialog.setCancelable(false);
mProgressDialog.show();
return mProgressDialog;
default:
    return null;
}
}

public class DownloadJSONAsync extends AsyncTask<String, Void, Void> {

    protected void onPreExecute() {
        super.onPreExecute();
        showDialog(DIALOG_DOWNLOAD_JSON_PROGRESS);
        setProgressBarIndeterminateVisibility(true);
    }

    @Override
    protected Void doInBackground(String... arg0) {
        // TODO Auto-generated method stub
        try {
            JSONParser jParser = new JSONParser();
            url = "http://chuwubbandroid.orgfree.com/feed_all.php";
            helper = new DBHelper(context);
            helper.getWritableDatabase();
            DataEntryArray = helper.SelectAll();
            imgHelper = new storeImg(context);
            imgHelper.getWritableDatabase();
            size = DataEntryArray.size();
            JSONObject json = jParser.getJSONFromUrl(url);
            feed = json.getJSONArray(TAG_feed);

            if (size < feed.length()) {
                for (int i = size; i < feed.length(); i++) {
                    ClassicGameDataEntry cgde = new ClassicGameDataEntry();
                    JSONObject game_feed = feed.getJSONObject(i);
                    mode = game_feed.getString(TAG_mode);
                    name = game_feed.getString(TAG_name);
                    id = game_feed.getString(TAG_id);
                    number = game_feed.getString(TAG_number);
                    answer = game_feed.getString(TAG_answer);
                    hint = game_feed.getString(TAG_hint);
                    link = game_feed.getString(TAG_link);
                    txt_choice1 = game_feed.getString(TAG_choice1);
                    txt_choice2 = game_feed.getString(TAG_choice2);
                    txt_choice3 = game_feed.getString(TAG_choice3);
                    txt_choice4 = game_feed.getString(TAG_choice4);
                    txt_choice5 = game_feed.getString(TAG_choice5);
                    txt_choice6 = game_feed.getString(TAG_choice6);
                    txt_choice7 = game_feed.getString(TAG_choice7);
                    txt_choice8 = game_feed.getString(TAG_choice8);
                }
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```

        int add_id = Integer.parseInt(id);
        if ((add_id - 42) > size) { // // check if answer is not
            // exist in db
            cgde.setMode_id(mode);
            cgde.setMode_name(name);
            cgde.setId(id);
            cgde.setPic_num(number);
            cgde.setAnswer(answer);
            cgde.setHint(hint);
            cgde.setLink(link);
            cgde.setChoice1(txt_choice1);
            cgde.setChoice2(txt_choice2);
            cgde.setChoice3(txt_choice3);
            cgde.setChoice4(txt_choice4);
            cgde.setChoice5(txt_choice5);
            cgde.setChoice6(txt_choice6);
            cgde.setChoice7(txt_choice7);
            cgde.setChoice8(txt_choice8);
            DataEntryArray.add(cgde);
        }
        imgHelper.InsertImg(DataEntryArray.get(i).getLink(),
            DataEntryArray.get(i).getLink());
    }
    listViewAdapter = new ListViewAdapter(download.this,
        DataEntryArray);

    } else {
        update = true;
    }

    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return null;
}

protected void onPostExecute(Void unused) {
    setProgressBarIndeterminateVisibility(false);

    if (update == true) {
        download_listview.setVisibility(View.GONE);
        TextView text = new TextView(context);
        // text.setGravity(Gravity.CENTER);
        text.setTextColor(Color.BLACK);
        text.setText("ข้อมูลของครูยังไม่ถูกโหลดเสร็จ");
        linearLayout.setGravity(Gravity.CENTER);
        linearLayout.addView(text);
        btn_download.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Toast.makeText(context,
                    "ข้อมูลของครูยังไม่ถูกโหลดเสร็จ",
                    Toast.LENGTH_SHORT).show();
            }
        });
        mProgressDialog.dismiss();
    } else {
        // setup adapter
        download_listview.setAdapter(listViewAdapter);
    }
}

```

```

        btn_download.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                if (size >= feed.length()) {
                    Toast.makeText(context,
                        "ข้อมูลของคอมพิวเตอร์หมดแล้วครับ",
                        Toast.LENGTH_SHORT).show();
                } else {
                    new DownloadDataToSQLite().execute();
                }
            }
        });
        mProgressDialog.dismiss();
    }
    super.onPostExecute(unused);
}

public class DownloadDataToSQLite extends AsyncTask<String, Void, Void> {

    protected void onPreExecute() {
        super.onPreExecute();
        showDialog(DIALOG_DOWNLOAD_DATATOSQLITE_PROGRESS);
        setProgressBarIndeterminateVisibility(true);
        helper = new DBHelper(getApplicationContext());
        helper.getWritableDatabase();
    }

    @Override
    protected Void doInBackground(String... arg0) {
        // TODO Auto-generated method stub
        DataEntry = new ClassicGameDataEntry();
        SQLiteDatabase db = imgHelper.getWritableDatabase();
        imgHelper.Delete(db);
        Log.d("size in OnClick", "size = " + size + "\nsize1 = " + size1);
        for (int pos = size; pos < size1; pos++) {
            // Add into DB/
            DataEntry.mode_id = DataEntryArray.get(pos).getMode_id();
            DataEntry.mode_name = DataEntryArray.get(pos).getMode_name();
            DataEntry.id = DataEntryArray.get(pos).getId();
            DataEntry.pic_num = DataEntryArray.get(pos).getPic_num();
            DataEntry.answer = DataEntryArray.get(pos).getAnswer();
            DataEntry.hint = DataEntryArray.get(pos).getHint();
            DataEntry.link = DataEntryArray.get(pos).getLink();
            DataEntry.choice1 = DataEntryArray.get(pos).getChoice1();
            DataEntry.choice2 = DataEntryArray.get(pos).getChoice2();
            DataEntry.choice3 = DataEntryArray.get(pos).getChoice3();
            DataEntry.choice4 = DataEntryArray.get(pos).getChoice4();
            DataEntry.choice5 = DataEntryArray.get(pos).getChoice5();
            DataEntry.choice6 = DataEntryArray.get(pos).getChoice6();
            DataEntry.choice7 = DataEntryArray.get(pos).getChoice7();
            DataEntry.choice8 = DataEntryArray.get(pos).getChoice8();

            if (DataEntry.link.substring(46, 47).equals("1")) { // mode
                first_picname = "classicgame_pic";
            } else {
                first_picname = "ghostmaster_pic";
            }
            last_picname = DataEntry.link.substring(48);
            String picFileName = first_picname.concat(last_picname);

```

```

        helper.Insert(Integer.parseInt(DataEntry.mode_id),
            DataEntry.mode_name, Integer.parseInt(DataEntry.id),
            Integer.parseInt(DataEntry.pic_num), DataEntry.answer,
            DataEntry.hint, picFileName, DataEntry.choice1,
            DataEntry.choice2, DataEntry.choice3,
            DataEntry.choice4, DataEntry.choice5,
            DataEntry.choice6, DataEntry.choice7, DataEntry.choice8);

        mProgressDialog.incrementProgressBy(1);

        try {
            URL url = new URL(DataEntry.link);
            String storagePath = getApplicationContext().getFilesDir()
                .toString();
            File folder = new File(storagePath + File.separator
                + "GameWaeteetong");
            Log.d("Path", "path = " + storagePath + File.separator
                + "GameWaeteetong");
            folder.mkdir();

            String pic_name = picFileName;
            File file = new File(folder, pic_name);
            if (file.exists()) {
                file.delete();
            }

            /* Open a connection to that URL. */
            URLConnection ucon = url.openConnection();
            InputStream inputStream = null;
            HttpURLConnection httpConn = (HttpURLConnection) ucon;
            httpConn.setRequestMethod("GET");
            httpConn.connect();
            if (httpConn.getResponseCode() == HttpURLConnection.HTTP_OK) {
                inputStream = httpConn.getInputStream();
            }

            FileOutputStream fos = new FileOutputStream(file);
            int size = 1024 * 1024;
            byte[] buf = new byte[size];
            int bytesRead, bytesDownloaded = 0;
            while (((byteRead = inputStream.read(buf)) != -1)) {
                fos.write(buf, 0, byteRead);
                bytesDownloaded += byteRead;
            }
            fos.close();

        } catch (MalformedURLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    return null;
}

```

```

protected void onPostExecute(Void unused) {
    setProgressBarIndeterminateVisibility(false);
    alert.showAlertDialog(download.this, "ดาวน์โหลด",
        "โหลดข้อมูลเรียบร้อยแล้ว.", false);
    size = DataEntryArray.size();
    Log.d("size after store", "size : " + size);
    download_listview.setVisibility(View.GONE);
    TextView text = new TextView(context);
    text.setTextColor(Color.BLACK);
    text.setText("ข้อมูลของคอลเลกชันเกมคลาสสิกแล้วครับ");
    linearLayout.setGravity(Gravity.CENTER);
    linearLayout.addView(text);
    mProgressDialog.dismiss();
    super.onPostExecute(unused);
}

// Online ListViewAdapter
private class ListViewAdapter extends BaseAdapter {

    private Activity activity;
    private ArrayList<ClassicGameDataEntry> data;
    private LayoutInflater inflater = null;
    byte[] imgBlob;
    int i = 0;

    public ListViewAdapter(Activity a, ArrayList<ClassicGameDataEntry> d) {
        activity = a;
        data = d;
    }

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return DataEntryArray.size() - size;
    }

    @Override
    public Object getItem(int position) {
        // TODO Auto-generated method stub
        return position;
    }

    @Override
    public long getItemId(int position) {
        // TODO Auto-generated method stub
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // TODO Auto-generated method stub
        inflater = (LayoutInflater) activity
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        if (convertView == null) {
            convertView = LayoutInflater.from(download.this).inflate(
                R.layout.download_listitem, null);
        }
    }
}

```

```

        if (DataEntryArray.get(position + size).getMode_name() != null) {
            holder.mode_name.setText("ကွက်: "
                + DataEntryArray.get(position + size).getMode_name());
        }
        if (DataEntryArray.get(position + size).getPic_num() != null) {
            holder.pic_number.setText("ပုံ: "
                + DataEntryArray.get(position + size).getPic_num());
        }

        if (DataEntryArray.get(position + size).getLink() != null) {
            holder.pic_img.setImageBitmap(BitmapFactory.decodeByteArray(
                getImg(position), 0, getImg(position).length));
        }
        i++;
        size1 = DataEntryArray.size();
        return convertView;
    }

    public byte[] getImg(int position) {
        SQLiteDatabase db = imgHelper.getWritableDatabase();
        String sql = "SELECT img FROM ImgDatabase WHERE link = " + "'"
            + DataEntryArray.get(position).getLink() + "'";
        Cursor cursor = db.rawQuery(sql, new String[] {});

        if (cursor.moveToFirst()) {
            imgBlob = cursor.getBlob(0);
        }
        if (cursor != null && !cursor.isClosed()) {
            cursor.close();
        }
        db.close();

        if (cursor.getCount() == 0) {
            return null;
        } else {
            return imgBlob;
        }
    }
}

private class ListViewHolder {

    public TextView mode_name;
    public TextView pic_number;
    public ImageView pic_img;
}

public static void androidOS() {
    if (android.os.Build.VERSION.SDK_INT > 9) {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
            .permitAll().build();
        StrictMode.setThreadPolicy(policy);
    }
}

```

```
// receive pic from internet and convert to bitmap
public Bitmap fetchImage(String imageUrl) {
    try {
        URL url = new URL(imageUrl.trim());
        InputStream input = null;
        URLConnection conn = url.openConnection();
        HttpURLConnection httpConn = (HttpURLConnection) conn;
        httpConn.setRequestMethod("GET");
        httpConn.setReadTimeout(40000);
        httpConn.connect();

        if (httpConn.getResponseCode() == HttpURLConnection.HTTP_OK) {
            input = httpConn.getInputStream();
        }
        Bitmap bitmap = BitmapFactory.decodeStream(input);
        input.close();
        httpConn.disconnect();
        return bitmap;
    } catch (MalformedURLException e) {
        Log.d("fetchImage", "MalformedURLException invalid URL: "
            + imageUrl);
    } catch (IOException e) {
        Log.d("fetchImage", "IO exception: " + e);
    } catch (Exception e) {
        Log.d("fetchImage", "Exception: " + e);
    }
    return null;
}
```

15. DBHelper.java

```
package com.example.gamewaeteethong.db;

import java.util.ArrayList;
import com.example.gamewaeteethong.ClassicGameDataEntry;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DBHelper extends SQLiteOpenHelper {

    ArrayList<ClassicGameDataEntry> CGDataEntry;
    SQLiteDatabase db;
    public static String DB_NAME = "database.db";
    public static String TABLE_NAME = "GameDatabase";
    public static String TABLE_KEY_ID = "game_id";
    public static String TABLE_KEY_MODE_ID = "mode_id";
    public static String TABLE_KEY_MODE_NAME = "mode_name";
    public static String TABLE_KEY_PIC_NUMBER = "pic_number";
    public static String TABLE_KEY_ANSWER = "game_answer";
    public static String TABLE_KEY_HINT = "game_hint";
    public static String TABLE_KEY_IMAGE = "game_image";
    public static String TABLE_KEY_CHOICE1 = "game_choice1";
    public static String TABLE_KEY_CHOICE2 = "game_choice2";
    public static String TABLE_KEY_CHOICE3 = "game_choice3";
    public static String TABLE_KEY_CHOICE4 = "game_choice4";
    public static String TABLE_KEY_CHOICES5 = "game_choice5";
    public static String TABLE_KEY_CHOICE6 = "game_choice6";
    public static String TABLE_KEY_CHOICE7 = "game_choice7";
    public static String TABLE_KEY_CHOICE8 = "game_choice8";

    public static int TABLE_VERSION = 1;
    public static String TABLE_CREATE_SQL = "CREATE TABLE " + TABLE_NAME + "("
        + TABLE_KEY_ID + " INTEGER PRIMARY KEY, " + TABLE_KEY_MODE_ID
        + " INTEGER, " + TABLE_KEY_MODE_NAME + " TEXT(30), "
        + TABLE_KEY_PIC_NUMBER + " INTEGER, " + TABLE_KEY_ANSWER
        + " TEXT(30), " + TABLE_KEY_HINT + " TEXT(10), " + TABLE_KEY_IMAGE
        + " TEXT(15), " + TABLE_KEY_CHOICE1 + " TEXT(10), "
        + TABLE_KEY_CHOICE2 + " TEXT(10), " + TABLE_KEY_CHOICES3
        + " TEXT(10), " + TABLE_KEY_CHOICE4 + " TEXT(10), "
        + TABLE_KEY_CHOICES5 + " TEXT(10), " + TABLE_KEY_CHOICE6
        + " TEXT(10), " + TABLE_KEY_CHOICE7 + " TEXT(10), "
        + TABLE_KEY_CHOICE8 + " TEXT(10) )";

    // constructor
    public DBHelper(Context context) {
        super(context, DB_NAME, null, TABLE_VERSION);
        // TODO Auto-generated constructor stub
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // TODO Auto-generated method stub
        db.execSQL(TABLE_CREATE_SQL);
    }
}
```

```

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
    db.execSQL("DROP TABLE " + TABLE_NAME);
    // and create it again
    onCreate(db);
    // check that "Is the table was upgraded?"
    Log.i("DBHelper", "Table has upgraded from " + oldVersion + " to "
        + newVersion);
}

// Insert
public void Insert(int mode_id, String mode_name, int id, int pic_num,
    String answer, String hint, String image, String choice1,
    String choice2, String choice3, String choice4, String choice5,
    String choice6, String choice7, String choice8) {

    db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(TABLE_KEY_ID, Integer.toString(id));
    values.put(TABLE_KEY_MODE_ID, Integer.toString(mode_id));
    values.put(TABLE_KEY_MODE_NAME, mode_name);
    values.put(TABLE_KEY_PIC_NUMBER, Integer.toString(pic_num));
    values.put(TABLE_KEY_ANSWER, answer);
    values.put(TABLE_KEY_HINT, hint);
    values.put(TABLE_KEY_IMAGE, image);
    values.put(TABLE_KEY_CHOICE1, choice1);
    values.put(TABLE_KEY_CHOICE2, choice2);
    values.put(TABLE_KEY_CHOICE3, choice3);
    values.put(TABLE_KEY_CHOICE4, choice4);
    values.put(TABLE_KEY_CHOICES5, choice5);
    values.put(TABLE_KEY_CHOICE6, choice6);
    values.put(TABLE_KEY_CHOICE7, choice7);
    values.put(TABLE_KEY_CHOICE8, choice8);

    db.execSQL("INSERT INTO " + TABLE_NAME + " (" + TABLE_KEY_MODE_ID + ","
        + TABLE_KEY_MODE_NAME + "," + TABLE_KEY_ID + ","
        + TABLE_KEY_PIC_NUMBER + "," + TABLE_KEY_ANSWER + ","
        + TABLE_KEY_HINT + "," + TABLE_KEY_IMAGE + ","
        + TABLE_KEY_CHOICE1 + "," + TABLE_KEY_CHOICE2 + ","
        + TABLE_KEY_CHOICE3 + "," + TABLE_KEY_CHOICE4 + ","
        + TABLE_KEY_CHOICES5 + "," + TABLE_KEY_CHOICE6 + ","
        + TABLE_KEY_CHOICE7 + "," + TABLE_KEY_CHOICE8 + ")")
        + " SELECT " + values.get(TABLE_KEY_MODE_ID) + ","
        + values.get(TABLE_KEY_MODE_NAME) + ","
        + values.get(TABLE_KEY_ID) + ","
        + values.get(TABLE_KEY_PIC_NUMBER) + ","
        + values.get(TABLE_KEY_ANSWER) + ","
        + values.get(TABLE_KEY_HINT) + ","
        + values.get(TABLE_KEY_IMAGE) + ","
        + values.get(TABLE_KEY_CHOICE1) + ","
        + values.get(TABLE_KEY_CHOICE2) + ","
        + values.get(TABLE_KEY_CHOICE3) + ","
        + values.get(TABLE_KEY_CHOICE4) + ","
        + values.get(TABLE_KEY_CHOICES5) + ","
        + values.get(TABLE_KEY_CHOICE6) + ","
        + values.get(TABLE_KEY_CHOICE7) + ","
        + values.get(TABLE_KEY_CHOICE8) + " "
        + " WHERE NOT EXISTS ( SELECT " + TABLE_KEY_ANSWER + " FROM "
        + TABLE_NAME + " WHERE " + TABLE_KEY_ANSWER + " = '" + answer
        + "') LIMIT 1;");
}

```

```

        db.close(); // Closing database connection
    }

    // Select Mode 1
    public ArrayList<ClassicGameDataEntry> SelectClassic() {
        db = this.getWritableDatabase();
        CGDataEntry = new ArrayList<ClassicGameDataEntry>();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME +
                                     " WHERE " + TABLE_KEY_MODE_ID + " = 1", null);

        // check
        if (cursor.getCount() != 0) {
            if (cursor.moveToFirst()) {
                do {
                    ClassicGameDataEntry cg = new ClassicGameDataEntry();
                    cg.setId(cursor.getString(cursor.getColumnIndex(TABLE_KEY_ID)));
                    cg.setMode_id(cursor.getString(cursor.getColumnIndex(TABLE_KEY_MODE_ID)));
                    cg.setMode_name(cursor.getString(cursor.getColumnIndex(TABLE_KEY_MODE_NAME)));
                    cg.setPic_num(cursor.getString(cursor.getColumnIndex(TABLE_KEY_PIC_NUMBER)));
                    cg.setAnswer(cursor.getString(cursor.getColumnIndex(TABLE_KEY_ANSWER)));
                    cg.setHint(cursor.getString(cursor.getColumnIndex(TABLE_KEY_HINT)));
                    cg.setImage_pic(cursor.getString(cursor.getColumnIndex(TABLE_KEY_IMAGE)));
                    cg.setChoice1(cursor.getString(cursor.getColumnIndex(TABLE_KEY_CHOICE1)));
                    cg.setChoice2(cursor.getString(cursor.getColumnIndex(TABLE_KEY_CHOICE2)));
                    cg.setChoice3(cursor.getString(cursor.getColumnIndex(TABLE_KEY_CHOICE3)));
                    cg.setChoice4(cursor.getString(cursor.getColumnIndex(TABLE_KEY_CHOICE4)));
                    cg.setChoice5(cursor.getString(cursor.getColumnIndex(TABLE_KEY_CHOICE5)));
                    cg.setChoice6(cursor.getString(cursor.getColumnIndex(TABLE_KEY_CHOICE6)));
                    cg.setChoice7(cursor.getString(cursor.getColumnIndex(TABLE_KEY_CHOICE7)));
                    cg.setChoice8(cursor.getString(cursor.getColumnIndex(TABLE_KEY_CHOICE8)));

                    CGDataEntry.add(cg);
                } while (cursor.moveToNext());
            }
        }
        cursor.close();
        db.close();
        return CGDataEntry;
    }

    // Select Mode 2
    public ArrayList<ClassicGameDataEntry> SelectGhost() {
        db = this.getWritableDatabase();
        CGDataEntry = new ArrayList<ClassicGameDataEntry>();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME + " WHERE "
                                     + TABLE_KEY_MODE_ID + " = 2", null);

        // check
        if (cursor.getCount() != 0) {
            if (cursor.moveToFirst()) {
                do {
                    ClassicGameDataEntry cg = new ClassicGameDataEntry();
                    cg.setId(cursor.getString(cursor
                        .getColumnIndex(TABLE_KEY_ID)));
                    cg.setMode_id(cursor.getString(cursor
                        .getColumnIndex(TABLE_KEY_MODE_ID)));
                    cg.setMode_name(cursor.getString(cursor
                        .getColumnIndex(TABLE_KEY_MODE_NAME)));
                    cg.setPic_num(cursor.getString(cursor
                        .getColumnIndex(TABLE_KEY_PIC_NUMBER)));
                    cg.setAnswer(cursor.getString(cursor
                        .getColumnIndex(TABLE_KEY_ANSWER)));
                    cg.setHint(cursor.getString(cursor
                        .getColumnIndex(TABLE_KEY_HINT)));
                } while (cursor.moveToNext());
            }
        }
    }

```

```

cg.setImage_pic(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_IMAGE)));
cg.setChoice1(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_CHOICE1)));
cg.setChoice2(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_CHOICE2)));
cg.setChoice3(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_CHOICE3)));
cg.setChoice4(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_CHOICE4)));
cg.setChoice5(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_CHOICE5)));
cg.setChoice6(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_CHOICE6)));
cg.setChoice7(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_CHOICE7)));
cg.setChoice8(cursor.getString(cursor
    .getColumnIndex(TABLE_KEY_CHOICE8)));

CGDataEntry.add(cg);
} while (cursor.moveToNext());
}
}
cursor.close();
db.close();
return CGDataEntry;
}

// Select ALL
public ArrayList<ClassicGameDataEntry> SelectAll() {
    db = this.getWritableDatabase();
    CGDataEntry = new ArrayList<ClassicGameDataEntry>();
    Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
    // check
    if (cursor.getCount() != 0) {
        if (cursor.moveToFirst()) {
            do {
                ClassicGameDataEntry cg = new ClassicGameDataEntry();
                cg.setId(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_ID)));
                cg.setMode_id(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_MODE_ID)));
                cg.setMode_name(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_MODE_NAME)));
                cg.setImage_pic(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_IMAGE)));
                cg.setPic_num(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_PIC_NUMBER)));
                cg.setAnswer(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_ANSWER)));
                cg.setHint(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_HINT)));
                cg.setChoice1(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_CHOICE1)));
                cg.setChoice2(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_CHOICE2)));
                cg.setChoice3(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_CHOICE3)));
                cg.setChoice4(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_CHOICE4)));
            } while (cursor.moveToNext());
        }
    }
}

```

```

        cg.setChoice5(cursor.getString(cursor
            .getColumnIndex(TABLE_KEY_CHOICE5)));
        cg.setChoice6(cursor.getString(cursor
            .getColumnIndex(TABLE_KEY_CHOICE6)));
        cg.setChoice7(cursor.getString(cursor
            .getColumnIndex(TABLE_KEY_CHOICE7)));
        cg.setChoice8(cursor.getString(cursor
            .getColumnIndex(TABLE_KEY_CHOICE8)));

        CGDataEntry.add(cg);
    } while (cursor.moveToNext());
    }
}
cursor.close();
db.close();
return CGDataEntry;
}
}

```

16. storeImg.java

```

package com.example.gamewaeteethong.db;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
import java.util.ArrayList;
import org.apache.http.util.ByteArrayBuffer;
import com.example.gamewaeteethong.ClassicGameDataEntry;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteStatement;
import android.util.Log;

public class storeImg extends SQLiteOpenHelper {

    ArrayList<ClassicGameDataEntry> CGDataEntry;
    SQLiteDatabase db;
    public static int TABLE_VERSION = 1;
    public static String DB_NAME = "storeImg.db";
    public static String TABLE_NAME = "ImgDatabase";
    public static String TABLE_KEY_PICNO = "img_picno";
    public static String TABLE_KEY_IMG = "img";
    public static String TABLE_KEY_LINK = "link";

    public static String TABLE_CREATE_SQL = "CREATE TABLE " + TABLE_NAME + "("
        + TABLE_KEY_IMG + " BLOB," + TABLE_KEY_LINK + " TEXT(255) )";

    public storeImg(Context context) {
        super(context, DB_NAME, null, TABLE_VERSION);
        // TODO Auto-generated constructor stub
    }
}

```

```

// Insert
public void InsertImg(String link, String imageLink) {
    db = this.getWritableDatabase();
    byte[] gameImage = getImage(imageLink);
    String sql = "INSERT INTO " + TABLE_NAME + " (" + TABLE_KEY_IMG + ","
        + TABLE_KEY_LINK + ") " + "VALUES(?,?)";
    SQLiteStatement insertStmt = db.compileStatement(sql);
    insertStmt.clearBindings();
    insertStmt.bindBlob(1, gameImage);
    insertStmt.bindString(2, link);
    insertStmt.executeInsert();
    db.close();
}

private byte[] getImage(String url) {
    try {
        URL imageUrl = new URL(url);
        URLConnection ucon = imageUrl.openConnection();

        InputStream is = ucon.getInputStream();
        BufferedInputStream bis = new BufferedInputStream(is);

        ByteBuffer baf = new ByteBuffer(500);
        int current = 0;
        while ((current = bis.read()) != -1) {
            baf.append((byte) current);
        }
        return baf.toByteArray();
    } catch (Exception e) {
        Log.d("ImageManager", "Error: " + e.toString());
    }
    return null;
}

@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub
    db.execSQL(TABLE_CREATE_SQL);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
    db.execSQL("DROP TABLE " + TABLE_NAME);
    // and create it again
    onCreate(db);
    // check that "Is the table was upgraded?"
    Log.i("DBHelper", "Table has upgraded from " + oldVersion + " to "
        + newVersion);
}

public void Delete(SQLiteDatabase db) {
    // TODO Auto-generated method stub
    db.execSQL("DROP TABLE " + TABLE_NAME);
}

```

```

// Select ALL
public ArrayList<ClassicGameDataEntry> SelectAll() {
    db = this.getWritableDatabase();
    CGDataEntry = new ArrayList<ClassicGameDataEntry>();
    Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
    // check
    if (cursor.getCount() != 0) {
        if (cursor.moveToFirst()) {
            do {
                ClassicGameDataEntry cg = new ClassicGameDataEntry();
                cg.setBlob(cursor.getBlob(cursor
                    .getColumnIndex(TABLE_KEY_IMG)));
                cg.setLink(cursor.getString(cursor
                    .getColumnIndex(TABLE_KEY_LINK)));
                CGDataEntry.add(cg);
            } while (cursor.moveToNext());
        }
    }
    cursor.close();
    db.close();
    return CGDataEntry;
}
}

```

17. ClassicGameDataEntry.java

```

package com.example.gamewaeteethong;

public class ClassicGameDataEntry {

    public String mode_name,id,mode_id,pic_num,hint;

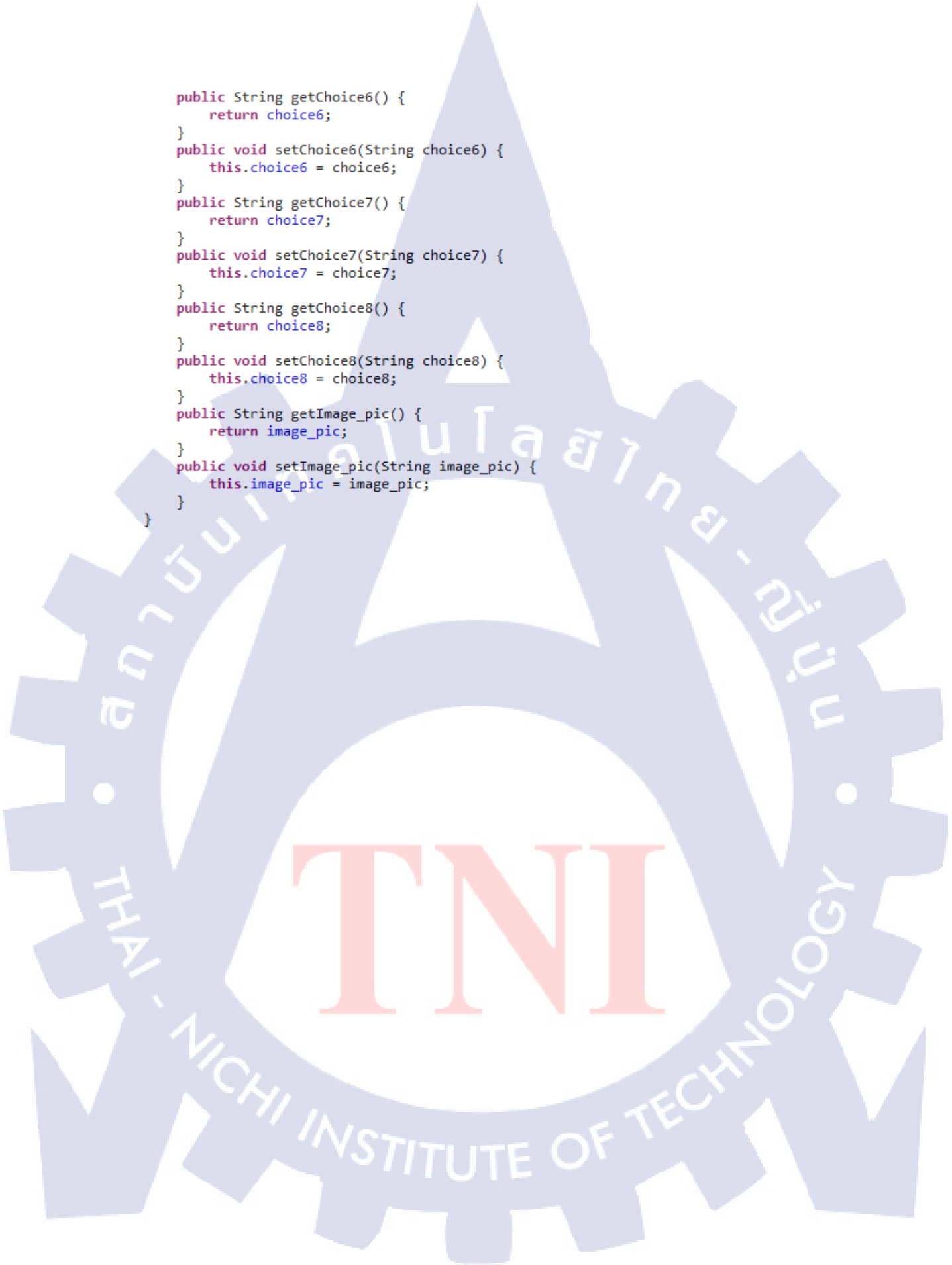
    public String answer;
    public String link;
    public String choice1,choice2,choice3,choice4,
        choice5,choice6,choice7,choice8;
    public String image_pic;
    public byte[] blob;

    public byte[] getBlob() {
        return blob;
    }
    public void setBlob(byte[] blob) {
        this.blob = blob;
    }
    public String getLink() {
        return link;
    }
    public void setLink(String link) {
        this.link = link;
    }
    public String getHint() {
        return hint;
    }
    public void setHint(String hint) {
        this.hint = hint;
    }
}

```

```
public String getMode_name() {  
    return mode_name;  
}  
public void setMode_name(String mode_name) {  
    this.mode_name = mode_name;  
}  
public String getId() {  
    return id;  
}  
public void setId(String id) {  
    this.id = id;  
}  
public String getMode_id() {  
    return mode_id;  
}  
public void setMode_id(String mode_id) {  
    this.mode_id = mode_id;  
}  
public String getPic_num() {  
    return pic_num;  
}  
public void setPic_num(String pic_num) {  
    this.pic_num = pic_num;  
}  
public String getAnswer() {  
    return answer;  
}  
public void setAnswer(String answer) {  
    this.answer = answer;  
}  
public String getChoice1() {  
    return choice1;  
}  
public void setChoice1(String choice1) {  
    this.choice1 = choice1;  
}  
public String getChoice2() {  
    return choice2;  
}  
public void setChoice2(String choice2) {  
    this.choice2 = choice2;  
}  
public String getChoice3() {  
    return choice3;  
}  
public void setChoice3(String choice3) {  
    this.choice3 = choice3;  
}  
public String getChoice4() {  
    return choice4;  
}  
public void setChoice4(String choice4) {  
    this.choice4 = choice4;  
}  
public String getChoice5() {  
    return choice5;  
}  
public void setChoice5(String choice5) {  
    this.choice5 = choice5;  
}
```

```
public String getChoice6() {  
    return choice6;  
}  
public void setChoice6(String choice6) {  
    this.choice6 = choice6;  
}  
public String getChoice7() {  
    return choice7;  
}  
public void setChoice7(String choice7) {  
    this.choice7 = choice7;  
}  
public String getChoice8() {  
    return choice8;  
}  
public void setChoice8(String choice8) {  
    this.choice8 = choice8;  
}  
public String getImage_pic() {  
    return image_pic;  
}  
public void setImage_pic(String image_pic) {  
    this.image_pic = image_pic;  
}  
}
```



THAI-NICHI INSTITUTE OF TECHNOLOGY

TNI

18. classicgame.java

```
package com.example.gamewaeteethong;

import java.util.ArrayList;
import java.util.Arrays;
import uk.co.senab.photoview.PhotoViewAttacher;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.Typeface;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.preference.PreferenceManager;
import android.support.v4.view.PagerAdapter;
import android.support.v4.view.ViewPager;
import android.support.v4.view.ViewPager.OnPageChangeListener;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.example.gamewaeteethong.db.DBHelper;
import com.nostra13.universalimageloader.core.DisplayImageOptions;
import com.nostra13.universalimageloader.core.ImageLoader;
import com.nostra13.universalimageloader.core.ImageLoaderConfiguration;
import com.nostra13.universalimageloader.core.assist.ImageScaleType;
import com.nostra13.universalimageloader.core.display.FadeInBitmapDisplayer;

public class classicgame extends Activity {

    //////////////////////////////////////
    DBHelper helper;
    ClassicGameDataEntry DataEntry;
    ArrayList<ClassicGameDataEntry> DataEntryArray = new ArrayList<ClassicGameDataEntry>();
```

```

public static SharedPreferences sharedPreferences ;
public static boolean Sound = true;

private CountDownTimer countDownTimer;
Button ans1, ans2, btn_choice1, btn_choice2, btn_choice3, btn_choice4,
    btn_choice5, btn_choice6, btn_choice7, btn_choice8;
TextView pic_number, countdown_time, hint_count;
ImageView hint_button;
int i, count = 1;
public long startTime;
private final long interval = 1 * 1000;
public long onPauseTime ;
Context context;
int btn, btn2;
LinearLayout hintbox;
String answer1, answer2, userAnswer, gameAnswer;
int picNo, size;
int pos = 0;
int score = 0;
int posInbtn;

public String[] sDrawables = { "drawable://" + R.drawable.classic_pic1,
    "drawable://" + R.drawable.classic_pic2,
    "drawable://" + R.drawable.classic_pic3,
    "drawable://" + R.drawable.classic_pic4,
    "drawable://" + R.drawable.classic_pic5 };

String[] choice1 = { "ง", "ด", "ค", "ข", "ช" };
String[] choice2 = { "ไค", "น", "ธ", "ร", "ช" };
String[] choice3 = { "ง", "า", "เห", "ไ", "ง" };
String[] choice4 = { "น", "า", "น", "ร", "อ" };
String[] choice5 = { "ค", "น", "น", "ร", "อ" };
String[] choice6 = { "ง", "า", "ค", "น", "อ" };
String[] choice8 = { "อ", "น", "น", "น", "อ" };
String[] choice7 = { "อ", "น", "น", "อ", "น" };
String[] answer = { "ไค", "ธ", "น", "ร", "อ", "อ" };
String[] hint = { "ไค", "ธ", "น", "อ", "อ" };

MediaPlayer soundEffect;
public ViewPager _viewPager;
PagerAdapter adapter;
ArrayList<DataGame> arraylist = new ArrayList<DataGame>();
boolean[] pic;
static int lastpic;
int hintCount = 1;
ImageLoader imageLoader;
DisplayImageOptions options;

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.classicgame_layout);

```

```

context = this;
// View Matching
_viewPager = (ViewPager) findViewById(R.id._viewPager);
hint_button = (ImageView) findViewById(R.id.btn_hint);
pic_number = (TextView) findViewById(R.id.pic_number);
countdown_time = (TextView) findViewById(R.id.countdown_txt);
ans1 = (Button) findViewById(R.id.btn_ansLog1);
ans2 = (Button) findViewById(R.id.btn_ansLog2);
btn_choice1 = (Button) findViewById(R.id.btn_choice1);
btn_choice2 = (Button) findViewById(R.id.btn_choice2);
btn_choice3 = (Button) findViewById(R.id.btn_choice3);
btn_choice4 = (Button) findViewById(R.id.btn_choice4);
btn_choice5 = (Button) findViewById(R.id.btn_choice5);
btn_choice6 = (Button) findViewById(R.id.btn_choice6);
btn_choice7 = (Button) findViewById(R.id.btn_choice7);
btn_choice8 = (Button) findViewById(R.id.btn_choice8);
hint_count = (TextView) findViewById(R.id.hint_count);
hintbox = (LinearLayout) findViewById(R.id.hintbox);

ImageLoaderConfiguration config = new ImageLoaderConfiguration
    .Builder(getApplicationContext()).build();
ImageLoader.getInstance().init(config);
options = new DisplayImageOptions.Builder()
    .resetViewBeforeLoading(true)
    .cacheOnDisc(true)
    .imageScaleType(ImageScaleType.EXACTLY)
    .bitmapConfig(Bitmap.Config.RGB_565)
    .displayer(new FadeInBitmapDisplayer(300))
    .build();

for (int i = 0; i < sDrawables.length; i++) {
    DataGame dg = new DataGame();
    dg.setPic_number(i+1);
    dg.setImage(sDrawables[i]);
    dg.setChoice1(choice1[i]);
    dg.setChoice2(choice2[i]);
    dg.setChoice3(choice3[i]);
    dg.setChoice4(choice4[i]);
    dg.setChoice5(choice5[i]);
    dg.setChoice6(choice6[i]);
    dg.setChoice7(choice7[i]);
    dg.setChoice8(choice8[i]);
    dg.setAnswer(answer[i]);
    dg.setHint(hint[i]);
    arraylist.add(dg);
}

helper = new DBHelper(context);
helper.getWritableDatabase();
DataEntryArray = helper.SelectClassic();
helper.close();
size = DataEntryArray.size();

```

```

if(size > 0){
    for(int i=0;i<size;i++){
        String img_name = DataEntryArray.get(i).getImage_pic();
        String path = "file:///data/data/com.example.gamewaeteethong/files"
            + "/GameWaeteetong/" +img_name;

        DataGame dg1 = new DataGame();
        dg1.setPic_number(Integer.parseInt(DataEntryArray.get(i).getPic_num()));
        dg1.setImage(path);
        dg1.setChoice1(DataEntryArray.get(i).getChoice1());
        dg1.setChoice2(DataEntryArray.get(i).getChoice2());
        dg1.setChoice3(DataEntryArray.get(i).getChoice3());
        dg1.setChoice4(DataEntryArray.get(i).getChoice4());
        dg1.setChoice5(DataEntryArray.get(i).getChoice5());
        dg1.setChoice6(DataEntryArray.get(i).getChoice6());
        dg1.setChoice7(DataEntryArray.get(i).getChoice7());
        dg1.setChoice8(DataEntryArray.get(i).getChoice8());
        dg1.setAnswer(DataEntryArray.get(i).getAnswer());
        dg1.setHint(DataEntryArray.get(i).getHint());
        arraylist.add(dg1);
    }
}

long s = arraylist.size();
startTime = ( 12 * 1000 ) * s;
hintCount = hintCount+(arraylist.size()/5);
pic = new boolean[arraylist.size()];
for(int i = 0; i<arraylist.size() ;i++){
    Arrays.fill(pic, Boolean.FALSE);
}

lastpic = arraylist.size() - 1;
// set first
pic_number.setText("ภาพ"+Integer.toString(arraylist.get(0).getPic_number()));
btn_choice1.setText(arraylist.get(0).getChoice1());
btn_choice2.setText(arraylist.get(0).getChoice2());
btn_choice3.setText(arraylist.get(0).getChoice3());
btn_choice4.setText(arraylist.get(0).getChoice4());
btn_choice5.setText(arraylist.get(0).getChoice5());
btn_choice6.setText(arraylist.get(0).getChoice6());
btn_choice7.setText(arraylist.get(0).getChoice7());
btn_choice8.setText(arraylist.get(0).getChoice8());
hint_count.setText(String.valueOf(hintCount));
hint_count.setTypeface(null, Typeface.BOLD);

// set countdown time
countDownTimer = new MyCountDownTimer(startTime, interval);
countdown_time.setText(String.valueOf(startTime / 1000));

adapter = new ViewPagerAdapter(this, arraylist);
_viewPager.setAdapter(adapter);

_viewPager.setOnPageChangeListener(new OnPageChangeListener() {

    @Override
    public void onPageSelected(int positionInViewPager) {
        // TODO Auto-generated method stub

        if (positionInViewPager == arraylist.size() && pic[arraylist.size()] == true) {
            positionInViewPager = 0;
        }
    }
}

```

```

        pos = positionInViewPager;
        changePic(positionInViewPager);
    }

    @Override
    public void onPageScrollStateChanged(int state) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onPageScrolled(int position, float positionOffset,
        int positionOffsetPixels) {
        // TODO Auto-generated method stub
    }
});

countDownTimer.start();

hint_button.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        String firstWord = arraylist.get(pos).getHint();
        if (hintCount == 0) {

            Toast.makeText(context, "คุณใช้คำไปไม่หมดคะนะคุณเล่น",
                Toast.LENGTH_SHORT).show();
        } else {
            hintCount--;
            hint_count.setText(String.valueOf(hintCount));
            AlertDialog.Builder builder = new Builder(classicgame.this);
            builder.setTitle("คำไป");
            builder.setMessage("คำแรกคือ..." + firstWord
                + " นะคุณเล่น\ถ้านวนคำไม่หมดละ : " + hintCount);
            builder.setNegativeButton("ผิด",
                new DialogInterface.OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog,
                        int arg1) {
                        // TODO Auto-generated method stub
                        dialog.dismiss();
                    }
                });
            AlertDialog alert = builder.create();
            alert.setCanceledOnTouchOutside(false);
            alert.show();
            if(hintCount==0){
                hintbox.setVisibility(View.INVISIBLE);
                hint_count.setVisibility(View.GONE);
            }
        }
    }
});

```

```

btn_choice1.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice1.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(0).getChoice1());
            btn = 1;
        } else {
            ans2.setText(arraylist.get(0).getChoice1());
            btn2 = 1;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice2.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice2.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(0).getChoice2());
            btn = 2;
        } else {
            ans2.setText(arraylist.get(0).getChoice2());
            btn2 = 2;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice3.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice3.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(0).getChoice3());
            btn = 3;
        } else {
            ans2.setText(arraylist.get(0).getChoice3());
            btn2 = 3;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice4.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice4.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(0).getChoice4());
            btn = 4;
        }
    }
});

```

```

        else {
            ans2.setText(arraylist.get(0).getChoice4());
            btn2 = 4;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice5.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice5.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(0).getChoice5());
            btn = 5;
        } else {
            ans2.setText(arraylist.get(0).getChoice5());
            btn2 = 5;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice6.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice6.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(0).getChoice6());
            btn = 6;
        }
        else {
            ans2.setText(arraylist.get(0).getChoice6());
            btn2 = 6;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice7.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice7.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(0).getChoice7());
            btn = 7;
        } else {
            ans2.setText(arraylist.get(0).getChoice7());
            btn2 = 7;
            i = 0;
            checkAnswer();
        }
    }
});

```

```

btn_choice8.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice8.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(0).getChoice8());
            btn = 8;
        } else {
            ans2.setText(arraylist.get(0).getChoice8());
            btn2 = 8;
            i = 0;
            checkAnswer();
        }
    }
});

ans1.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if (!ans1.getText().equals("")) {
            ans1.setText("");
            onClickAnsButton();
        }
    }
});

//check settings
sharedPreferences = PreferenceManager.getDefaultSharedPreferences(context);
Sound = sharedPreferences.getBoolean("Sound", true);
}

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    Intent intent = new Intent(classicgame.this,
        choosemode.class);
    startActivity(intent);
    finish();
    overridePendingTransition(R.layout.mainfadein,
        R.layout.mainfadeout);
}

public void playSound(int i){
    switch(i) {
        case 0: soundEffect = MediaPlayer.create(this, R.raw.ooooh);
                break;
        case 1: soundEffect = MediaPlayer.create(this, R.raw.yahoo);
                break;
        case 2: soundEffect = MediaPlayer.create(this, R.raw.yippee);
                break;
        case 3: soundEffect = MediaPlayer.create(this, R.raw.wild_laugh);
                break;
    }
    soundEffect.start();
    while (soundEffect.isPlaying()) {
        // do nothing
    }
}

```

```

        soundEffect.stop();
        soundEffect.release();
    }

    public void OnClickAnsButton() {
        if (btn == 1) {
            btn_choice1.setVisibility(View.VISIBLE);
        } else if (btn == 2) {
            btn_choice2.setVisibility(View.VISIBLE);
        } else if (btn == 3) {
            btn_choice3.setVisibility(View.VISIBLE);
        } else if (btn == 4) {
            btn_choice4.setVisibility(View.VISIBLE);
        } else if (btn == 5) {
            btn_choice5.setVisibility(View.VISIBLE);
        } else if (btn == 6) {
            btn_choice6.setVisibility(View.VISIBLE);
        } else if (btn == 7) {
            btn_choice7.setVisibility(View.VISIBLE);
        } else if (btn == 8) {
            btn_choice8.setVisibility(View.VISIBLE);
        }
    }

    public void CorrectDialogButton() {
        ans1.setText("");
        ans2.setText("");
        btn_choice1.setVisibility(View.VISIBLE);
        btn_choice2.setVisibility(View.VISIBLE);
        btn_choice3.setVisibility(View.VISIBLE);
        btn_choice4.setVisibility(View.VISIBLE);
        btn_choice5.setVisibility(View.VISIBLE);
        btn_choice6.setVisibility(View.VISIBLE);
        btn_choice7.setVisibility(View.VISIBLE);
        btn_choice8.setVisibility(View.VISIBLE);
    }

    public void WrongAnswer() {
        if (Sound == true) {
            playSound(0);
        }
        if (count == 1) {
            Toast.makeText(context, "ผิดแล้วสิคุณหนู", Toast.LENGTH_SHORT).show();
            count = 2;
        } else if (count == 2) {
            Toast.makeText(context, "ไม่ถูกแล้วละ ฮึฮึฮึ", Toast.LENGTH_SHORT)
                .show();
            count = 3;
        } else {
            Toast.makeText(context, "ถูก!!...หรรษาแล้วสินะ", Toast.LENGTH_SHORT).show();
            count = 1;
        }
        CorrectDialogButton();
    }

    public void checkAnswer() {
        answer1 = (String) ans1.getText();
        answer2 = (String) ans2.getText();
        userAnswer = answer1.concat(answer2);
        AlertDialog.Builder builder = new Builder(classicgame.this);
        if (arraylist.get(pos).getAnswer().equals(userAnswer)) {

```



```

        startActivity(intent);
        finish();
        overridePendingTransition(R.layout.mainfadein,
            R.layout.mainfadeout);
    }
});
builder2.setNegativeButton("กลับหน้าหลัก",
    new DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            Intent intent = new Intent(classicgame.this,
                MainActivity.class);
            startActivity(intent);
            finish();
            overridePendingTransition(R.layout.mainfadein,
                R.layout.mainfadeout);
        }
    });
AlertDialog alert = builder2.create();
alert.setCanceledOnTouchOutside(false);
alert.show();
if(Sound == true){
    playSound(2);
}
}

public View removePage(int position) {
    if ((position < 0) || (position >= arraylist.size())
        || (arraylist.size() <= 1)) {
        return null;
    } else {
        if (position == arraylist.size() - 1) {
            Log.d("On removePage", "in if");
            arraylist.remove(position);
            _viewPager.setAdapter(adapter);
            _viewPager.setCurrentItem(0);
            changePic(0);
            pos = 0;

        } else {
            Log.d("In RemovePage", "position = " + position);
            arraylist.remove(position);
            _viewPager.setAdapter(adapter);
            _viewPager.setCurrentItem(position);
            changePic(position);

        }
        return _viewPager;
    }
}

public void changePic(int position) {
    onClickAnsButton();
    ans1.setText("");
    ans2.setText("");
    pic_number.setText("ภาพ"+Integer.toString(arraylist.get(position).getPic_number()));
    btn_choice1.setText(arraylist.get(position).getChoice1());
    btn_choice2.setText(arraylist.get(position).getChoice2());
    btn_choice3.setText(arraylist.get(position).getChoice3());
    btn_choice4.setText(arraylist.get(position).getChoice4());
    btn_choice5.setText(arraylist.get(position).getChoice5());
}

```

```

btn_choice6.setText(arraylist.get(position).getChoice6());
btn_choice7.setText(arraylist.get(position).getChoice7());
btn_choice8.setText(arraylist.get(position).getChoice8());
posInbtn = position;
btn_choice1.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice1.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice1());
            btn = 1;
        } else {
            ans2.setText(arraylist.get(posInbtn).getChoice1());
            btn2 = 1;
            checkAnswer();
        }
    }
});

btn_choice2.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice2.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice2());
            btn = 2;
        } else {
            ans2.setText(arraylist.get(posInbtn).getChoice2());
            btn2 = 2;
            checkAnswer();
        }
    }
});

btn_choice3.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice3.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice3());
            btn = 3;
        } else {
            ans2.setText(arraylist.get(posInbtn).getChoice3());
            btn2 = 3;
            checkAnswer();
        }
    }
});

btn_choice4.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice4.setVisibility(View.GONE);

```

```

        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice4());
            btn = 4;
        } else {
            ans2.setText(arraylist.get(posInbtn).getChoice4());
            btn2 = 4;
            checkAnswer();
        }
    }
});
btn_choice5.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        btn_choice5.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice5());
            btn = 5;
        } else {
            ans2.setText(arraylist.get(posInbtn).getChoice5());
            btn2 = 5;
            checkAnswer();
        }
    }
});

btn_choice6.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        btn_choice6.setVisibility(View.GONE);

        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice6());
            btn = 6;
        } else {
            ans2.setText(arraylist.get(posInbtn).getChoice6());
            btn2 = 6;
            checkAnswer();
        }
    }
});

btn_choice7.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        btn_choice7.setVisibility(View.GONE);

        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice7());
            btn = 7;
        }
    }
});

```

```

        else {
            ans2.setText(arraylist.get(posInbtn).getChoice7());
            btn2 = 7;
            checkAnswer();
        }
    }
});

btn_choice8.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        btn_choice8.setVisibility(View.GONE);

        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice8());
            btn = 8;
        } else {
            ans2.setText(arraylist.get(posInbtn).getChoice8());
            btn2 = 8;
            checkAnswer();
        }
    }
});
}

static class ViewPagerAdapter extends PagerAdapter {
    Context context;
    ArrayList<DataGame> arraylist;
    ImageView classic_pic;
    LayoutInflater inflater;
    private PhotoViewAttacher mAttacher;
    static final String PHOTO_TAP_TOAST_STRING = "Photo Tap! X: %.2f %% Y: %.2f %%";
    DisplayImageOptions options;
    ImageLoader imageLoader;

    public ViewPagerAdapter(Context context, ArrayList<DataGame> arraylist) {
        // TODO Auto-generated constructor stub
        this.context = context;
        this.arraylist = arraylist;
        this.imageLoader = imageLoader;
        this.options = options;
    }

    @Override
    public int getCount() {
        return arraylist.size();
    }

    @Override
    public View instantiateItem(ViewGroup container, int position) {
        // // Slide Pic ////
        inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View itemView = inflater.inflate(R.layout.viewpager_item,
            container, false);
        classic_pic = (ImageView) itemView.findViewById(R.id.classic_pic);
        imageLoader.getInstance().displayImage(arraylist.get(position).getImage()
            , classic_pic);
    }
}

```

```

        container.addView(itemView);
        return itemView;
    }

    @Override
    public void destroyItem(ViewGroup container, int position, Object object) {
        ((ViewPager) container).removeView((View) object);
    }

    @Override
    public boolean isViewFromObject(View view, Object object) {
        return view.equals(object);
    }
}

public class MyCountDownTimer extends CountDownTimer {
    public MyCountDownTimer(long startTime, long interval) {
        super(startTime, interval);
    }

    @Override
    public void onFinish() {
        // TODO Auto-generated method stub
        countdown_time.setText("00:00");

        lastpic = lastpic+1;

        AlertDialog.Builder builder = new Builder(classicgame.this);
        builder.setTitle("จบเกม");
        builder.setMessage("เวลาหมดแล้วคะแนนของคุณคือ " + score
            + " คะแนน "+lastpic+" คะแนน");
        builder.setPositiveButton("เล่นใหม่",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub
                    Intent intent = new Intent(classicgame.this,
                        classicgame.class);
                    startActivity(intent);
                    finish();
                    overridePendingTransition(R.layout.mainfadein,
                        R.layout.mainfadeout);
                }
            });
        builder.setNegativeButton("กลับหน้าจอหลัก",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub
                    Intent intent = new Intent(classicgame.this,
                        MainActivity.class);
                    startActivity(intent);
                    finish();
                    overridePendingTransition(R.layout.mainfadein,
                        R.layout.mainfadeout);
                }
            });
        AlertDialog alert = builder.create();
        alert.setCanceledOnTouchOutside(false);
        alert.show();
        if(Sound == true){
            playSound(3);
        }
    }
}

```

```

}

// ////////////////////////////////// Count Down Timer //////////////////////////////////

public String formatTime(long millis) {
    String output = "00:00";
    long seconds = millis / 1000;
    long minutes = seconds / 60;
    seconds = seconds % 60;
    minutes = minutes % 60;
    String sec = String.valueOf(seconds);
    String min = String.valueOf(minutes);
    if (seconds < 10)
        sec = "0" + seconds;
    if (minutes < 10)
        min = "0" + minutes;
    output = min + ":" + sec;
    return output;
} // formatTime

@Override
public void onTick(long millisUntilFinished) {
    // TODO Auto-generated method stub
    onPauseTime = millisUntilFinished;
    countdown_time.setText(formatTime(millisUntilFinished));
}
}
// ////////////////////////////////// //////////////////////////////////

```



TNI

19. ghostmaster.java

```

package com.example.gamewaeteethong;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Random;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.content.pm.Signature;
import android.content.pm.PackageManager.NameNotFoundException;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Typeface;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.preference.PreferenceManager;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.example.gamewaeteethong.db.DBHelper;
import com.facebook.Request;
import com.facebook.Response;
import com.facebook.Session;
import com.facebook.Session.StatusCallback;
import com.facebook.SessionState;
import com.facebook.model.GraphObject;
import com.facebook.model.GraphUser;
import com.nostra13.universalimageloader.core.DisplayImageOptions;
import com.nostra13.universalimageloader.core.ImageLoader;
import com.nostra13.universalimageloader.core.ImageLoaderConfiguration;
import com.nostra13.universalimageloader.core.assist.ImageScaleType;
import com.nostra13.universalimageloader.core.display.FadeInBitmapDisplayer;

public class ghostmaster extends Activity {
    DBHelper helper;
    ClassicGameDataEntry DataEntry;
    ArrayList<ClassicGameDataEntry> DataEntryArray = new ArrayList<ClassicGameDataEntry>();

```

```

public static SharedPreferences sharedPreferences;
public static boolean Sound = true;
FrameLayout passbox;
private CountdownTimer countDownTimer;
private CountdownTimer tenCountDownTimer;
Button btn_choice1, btn_choice2, btn_choice3, btn_choice4, btn_choice5,
        btn_choice6, btn_choice7, btn_choice8, ans1, ans2, ans3;
TextView pic_number, countdown_time, hint_count, tenCountTxt, passtxt;
ImageView hint_button;
int i, count = 1;
public long startTime;
public long tenSec = 11 * 1000;
private final long interval = 1 * 1000;
public long onPauseTime;
Context context;
int btn, btn2, btn3;
String answer1, answer2, answer3, userAnswer, gameAnswer;
int picNo, size, previousRandom, currentRandom, pos = 0, score = 0,
    posInbtn, arraylistSize, prevIndex;
MediaPlayer soundEffect;
ArrayList<DataGame> arraylist = new ArrayList<DataGame>();
boolean[] pic;
boolean correct = false;
static int Lastpic;
int hintCount = 1, passCount = 1, index;
ImageLoader imageLoader;
DisplayImageOptions options;
ImageView gamepic, passbtn;
String nameApp = "เกมปริศนา";
private Random randomGenerator = new Random();
// Connection detector
ConnectionDetector cd;
// Alert dialog manager
AlertDialogManager alert = new AlertDialogManager();
private PendingIntent pendingAction = PendingIntent.NONE;
private enum PendingAction {
    NONE, POST_PHOTO
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.ghostmaster_layout);
    context = this;

    // View Matching
    passbox = (FrameLayout) findViewById(R.id.passbox);
    hint_button = (ImageView) findViewById(R.id.btn_hint);
    pic_number = (TextView) findViewById(R.id.pic_number);
    countdown_time = (TextView) findViewById(R.id.countdown_txt);
    ans1 = (Button) findViewById(R.id.ans1);
    ans2 = (Button) findViewById(R.id.ans2);
    ans3 = (Button) findViewById(R.id.ans3);
    btn_choice1 = (Button) findViewById(R.id.btn_choice1);
    btn_choice2 = (Button) findViewById(R.id.btn_choice2);
    btn_choice3 = (Button) findViewById(R.id.btn_choice3);
    btn_choice4 = (Button) findViewById(R.id.btn_choice4);
    btn_choice5 = (Button) findViewById(R.id.btn_choice5);
    btn_choice6 = (Button) findViewById(R.id.btn_choice6);

```

```

btn_choice7 = (Button) findViewById(R.id.btn_choice7);
btn_choice8 = (Button) findViewById(R.id.btn_choice8);
hint_count = (TextView) findViewById(R.id.hint_count);
tenCountTxt = (TextView) findViewById(R.id.tenCountTxt);
gamepic = (ImageView) findViewById(R.id.gamepic);
passtxt = (TextView) findViewById(R.id.passtxt);
passbtn = (ImageView) findViewById(R.id.passbtn);

ImageLoaderConfiguration config = new ImageLoaderConfiguration.Builder(
    getApplicationContext()).build();
ImageLoader.getInstance().init(config);
options = new DisplayImageOptions.Builder()
    .resetViewBeforeLoading(true).cacheOnDisc(true)
    .imageScaleType(ImageScaleType.EXACTLY)
    .bitmapConfig(Bitmap.Config.RGB_565)
    .displayer(new FadeInBitmapDisplayer(300)).build();

helper = new DBHelper(context);
helper.getWritableDatabase();
DataEntryArray = helper.SelectGhost();
helper.close();
size = DataEntryArray.size();
logKeyHash();
for (int i = 0; i < size; i++) {
    String img_name = DataEntryArray.get(i).getImage_pic();
    String path = "file:///data/data/com.example.gamewaeteethong/files/GameWaeteetong/"
        + img_name;
    DataGame dg1 = new DataGame();
    dg1.setImage(path);
    dg1.setChoice1(DataEntryArray.get(i).getChoice1());
    dg1.setChoice2(DataEntryArray.get(i).getChoice2());
    dg1.setChoice3(DataEntryArray.get(i).getChoice3());
    dg1.setChoice4(DataEntryArray.get(i).getChoice4());
    dg1.setChoice5(DataEntryArray.get(i).getChoice5());
    dg1.setChoice6(DataEntryArray.get(i).getChoice6());
    dg1.setChoice7(DataEntryArray.get(i).getChoice7());
    dg1.setChoice8(DataEntryArray.get(i).getChoice8());
    dg1.setAnswer(DataEntryArray.get(i).getAnswer());
    dg1.setHint(DataEntryArray.get(i).getHint());
    arraylist.add(dg1);
}
arraylistSize = arraylist.size();
long s = arraylist.size();
startTime = (12 * 1000) * s;
hintCount = hintCount + (arraylist.size() / 5);
passCount = passCount + (arraylist.size() / 5);
pic = new boolean[arraylist.size()];
for (int i = 0; i < arraylist.size(); i++) {
    Arrays.fill(pic, Boolean.FALSE);
}

index = randomGenerator.nextInt(arraylist.size());
currentRandom = index;
lastpic = arraylist.size() - 1;
pic_number.setText("0 / " + Integer.toString(arraylist.size()));
pic_number.setTypeface(null, Typeface.BOLD);
btn_choice1.setText(arraylist.get(index).getChoice1());
btn_choice2.setText(arraylist.get(index).getChoice2());
btn_choice3.setText(arraylist.get(index).getChoice3());
btn_choice4.setText(arraylist.get(index).getChoice4());

```

```

btn_choice5.setText(arraylist.get(index).getChoice5());
btn_choice6.setText(arraylist.get(index).getChoice6());
btn_choice7.setText(arraylist.get(index).getChoice7());
btn_choice8.setText(arraylist.get(index).getChoice8());
imageLoader.getInstance().displayImage(arraylist.get(index).getImage(),
    gamepic);
hint_count.setText(String.valueOf(hintCount));
hint_count.setTypeface(null, Typeface.BOLD);
passtxt.setText(String.valueOf(passCount));

// set countdown time
tenCountDownTimer = new TenCountDownTimer(tenSec, interval);
tenCountTxt.setText(String.valueOf(tenSec / 1000));
tenCountDownTimer.start();
// set countdown time
countDownTimer = new MyCountDownTimer(startTime, interval);
countdown_time.setText(String.valueOf(startTime / 1000));
countDownTimer.start();

passbtn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        tenCountDownTimer.cancel();
        if (passCount == 0) {
            Toast.makeText(context,
                "คุณไม่สามารถทานได้เพราะระดับเกม",
                Toast.LENGTH_SHORT).show();
        } else {
            passCount--;
            passtxt.setText(String.valueOf(passCount));
            changePic();
            if (passCount == 0) {
                passbox.setVisibility(View.INVISIBLE);
            }
        }
    }
});

hint_button.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        String firstWord = arraylist.get(index).getHint();
        if (hintCount == 0) {
            Toast.makeText(context, "คุณใช้คำไปหมดแล้วระดับเกม",
                Toast.LENGTH_SHORT).show();
        } else {
            hintCount--;
            hint_count.setText(String.valueOf(hintCount));
            AlertDialog.Builder builder = new Builder(ghostmaster.this);
            builder.setTitle("คำใบ้");
            builder.setMessage("คำแรกคือ..." + firstWord
                + " ระดับเกม/จำนวนคำใบ้ที่เหลือ : " + hintCount);
            builder.setNegativeButton("ปิด",
                new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog,
                        int arg1) {
                        // TODO Auto-generated method stub
                        dialog.dismiss();
                    }
                }
            );
        }
    }
});

```

```

        AlertDialog alert = builder.create();
        alert.setCanceledOnTouchOutside(false);
        alert.show();
        if (hintCount == 0) {
            hint_button
                .setImageResource(R.drawable.ghost_hint_icon_default);
            hint_count.setVisibility(View.GONE);
        }
    }
});

btn_choice1.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice1.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(index).getChoice1());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 1;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(index).getChoice1());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 1;
        } else {
            ans3.setText(arraylist.get(index).getChoice1());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 1;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice2.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice2.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(index).getChoice2());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 2;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(index).getChoice2());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 2;
        } else {
            ans3.setText(arraylist.get(index).getChoice2());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 2;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice3.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice3.setVisibility(View.GONE);

```

```

        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(index).getChoice3());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 3;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(index).getChoice3());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 3;
        } else {
            ans3.setText(arraylist.get(index).getChoice3());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 3;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice4.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice4.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(index).getChoice4());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 4;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(index).getChoice4());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 4;
        } else {
            ans3.setText(arraylist.get(index).getChoice4());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 4;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice5.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice5.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(index).getChoice5());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 5;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(index).getChoice5());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 5;
        } else {
            ans3.setText(arraylist.get(index).getChoice5());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 5;
            i = 0;
            checkAnswer();
        }
    }
});

```

```

btn_choice6.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice6.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(index).getChoice6());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 6;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(index).getChoice6());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 6;
        } else {
            ans3.setText(arraylist.get(index).getChoice6());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 6;
            i = 0;
            checkAnswer();
        }
    }
});
btn_choice7.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice7.setVisibility(View.INVISIBLE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(index).getChoice7());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 7;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(index).getChoice7());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 7;
        } else {
            ans3.setText(arraylist.get(index).getChoice7());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 7;
            i = 0;
            checkAnswer();
        }
    }
});
btn_choice8.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice8.setVisibility(View.INVISIBLE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(index).getChoice8());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 8;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(index).getChoice8());

```

```

        ans2.setTypeface(null, Typeface.BOLD);
        btn2 = 8;
    } else {
        ans3.setText(arraylist.get(index).getChoice8());
        ans3.setTypeface(null, Typeface.BOLD);
        btn3 = 8;
        i = 0;
        checkAnswer();
    }
}
});
ans1.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if (!ans1.getText().equals("")) {
            ans1.setText("");
            onClickAnsButton();
        }
    }
});
ans2.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if (!ans2.getText().equals("")) {
            ans2.setText("");
            onClickAnsButtonTwo();
        }
    }
});
// check settings
sharedPreferences = PreferenceManager
    .getDefaultSharedPreferences(context);
Sound = sharedPreferences.getBoolean("Sound", true);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    Session.getActiveSession().onActivityResult(ghostmaster.this,
        requestCode, resultCode, data);
}

@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    countdownTimer.cancel();
    tenCountDownTimer.cancel();
}

@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
}
}

```

```

@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    // TODO Auto-generated method stub
    super.onSaveInstanceState(outState);
}

public void playSound(int i) {
    switch (i) {
        case 0:
            soundEffect = MediaPlayer.create(this, R.raw.ooooh);
            break;
        case 1:
            soundEffect = MediaPlayer.create(this, R.raw.oooo);
            break;
        case 2:
            soundEffect = MediaPlayer.create(this, R.raw.yahoo);
            break;
        case 3:
            soundEffect = MediaPlayer.create(this, R.raw.ghostlaugh);
            break;
    }
    soundEffect.start();
    while (soundEffect.isPlaying()) {
        // do nothing
    }
    soundEffect.stop();
    soundEffect.release();
}

public void OnClickAnsButton() {
    if (btn == 1) {
        btn_choice1.setVisibility(View.VISIBLE);
    } else if (btn == 2) {
        btn_choice2.setVisibility(View.VISIBLE);
    } else if (btn == 3) {
        btn_choice3.setVisibility(View.VISIBLE);
    } else if (btn == 4) {
        btn_choice4.setVisibility(View.VISIBLE);
    } else if (btn == 5) {
        btn_choice5.setVisibility(View.VISIBLE);
    } else if (btn == 6) {
        btn_choice6.setVisibility(View.VISIBLE);
    } else if (btn == 7) {
        btn_choice7.setVisibility(View.VISIBLE);
    } else if (btn == 8) {
        btn_choice8.setVisibility(View.VISIBLE);
    }
}

public void OnClickAnsButtonTwo() {
    if (btn2 == 1) {
        btn_choice1.setVisibility(View.VISIBLE);
    } else if (btn2 == 2) {
        btn_choice2.setVisibility(View.VISIBLE);
    } else if (btn2 == 3) {
        btn_choice3.setVisibility(View.VISIBLE);
    }
}

```

```

    } else if (btn2 == 4) {
        btn_choice4.setVisibility(View.VISIBLE);
    } else if (btn2 == 5) {
        btn_choice5.setVisibility(View.VISIBLE);
    } else if (btn2 == 6) {
        btn_choice6.setVisibility(View.VISIBLE);
    } else if (btn2 == 7) {
        btn_choice7.setVisibility(View.VISIBLE);
    } else if (btn2 == 8) {
        btn_choice8.setVisibility(View.VISIBLE);
    }
}

public void checkAnswer() {
    answer1 = (String) ans1.getText();
    answer2 = (String) ans2.getText();
    answer3 = (String) ans3.getText();
    userAnswer = answer1.concat(answer2);
    userAnswer = userAnswer.concat(answer3);
    AlertDialog.Builder builder = new Builder(ghostmaster.this);
    if (arraylist.get(index).getAnswer().equals(userAnswer)) {
        btn_choice1.setVisibility(View.VISIBLE);
        btn_choice2.setVisibility(View.VISIBLE);
        btn_choice3.setVisibility(View.VISIBLE);
        btn_choice4.setVisibility(View.VISIBLE);
        btn_choice5.setVisibility(View.VISIBLE);
        btn_choice6.setVisibility(View.VISIBLE);
        btn_choice7.setVisibility(View.VISIBLE);
        btn_choice8.setVisibility(View.VISIBLE);
        score++;
        countdownTimer.cancel();
        tenCountDownTimer.cancel();
        if (score == lastpic + 1) {
            Winner();
        } else {
            correct = true;
            prevIndex = index;
            builder.setTitle("คำตอบ");
            builder.setMessage(arraylist.get(index).getAnswer()
                + " ถูกต้องแล้วครับ");
            builder.setPositiveButton("ใช่",
                new DialogInterface.OnClickListener() {
                    @SuppressWarnings("deprecation")
                    @Override
                    public void onClick(DialogInterface dialog,
                        int which) {
                        // TODO Auto-generated method stub
                        tenCountDownTimer.cancel();
                        countdownTimer.cancel();
                        cd = new ConnectionDetector(context);

                        // check Internet Connect
                        if (!cd.isConnectingToInternet()) {
                            // Internet Connection is not present
                            alert.showAlertDialog(context, "การเชื่อมต่ออินเทอร์เน็ต",
                                "คุณไม่ได้เชื่อมต่ออินเทอร์เน็ต", false);
                        }
                    }
                }
            );
        }
    }
}

```

```

    } else {
        Session.openActiveSession(ghostmaster.this,
            true, new StatusCallback() {

            @Override
            public void call(Session session,
                SessionState state,
                Exception exception) {

                if (session.isOpened()) {
                    Request.executeRequestAsync(session,
                        new Request.GraphUserCallback() {

                        @Override
                        public void onCompleted(
                            GraphUser user,
                            Response response) {

                            if (user != null) {
                                postPhoto();
                            }
                        }
                    });
                }
            }
        });

        countDownTimer = new MyCountDownTimer(
            onPauseTime, interval);
        countDownTimer.start();
        tenCountDownTimer = new TenCountDownTimer(
            tenSec, interval);
        tenCountDownTimer.start();
    }
});

builder.setNegativeButton("❌",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int arg1) {
            // TODO Auto-generated method stub
            dialog.dismiss();
            countDownTimer = new MyCountDownTimer(
                onPauseTime, interval);
            countDownTimer.start();
            tenCountDownTimer.cancel();
            tenCountDownTimer = new TenCountDownTimer(
                tenSec, interval);
            tenCountDownTimer.start();
            CorrectDialogButton();
        }
    });

AlertDialog alert = builder.create();
alert.setCanceledOnTouchOutside(false);
alert.show();

if (Sound == true) {
    playSound(1);
}
pic[pos] = true;
removePage(pos);
}
} else {
    WrongAnswer();
}
}
}

```

```

private void postPhoto() {
    if (hasPublishPermission()) {
        String img_name = DataEntryArray.get(prevIndex).getImage_pic();
        Log.d("Log", "img_name = " + img_name);
        String path = new File(getFilesDir(), img_name).getAbsolutePath();
        Log.d("Log", "path = " + path);

        File location = new File(
            "/data/data/com.example.gamewaeteethong/files/GameWaeteetong/");
        File dest = new File(location, img_name);
        FileInputStream fis;
        try {
            fis = new FileInputStream(dest);
            Bitmap bitmap = BitmapFactory.decodeStream(fis);
            com.facebook.Request request = com.facebook.Request
                .newUploadPhotoRequest(Session.getActiveSession(),
                    bitmap, new Request.Callback() {
                        @Override
                        public void onCompleted(Response response) {
                            if (response.getError() == null) {
                                Toast.makeText(context,
                                    "อัพโหลดภาพเรียบร้อยแล้ว",
                                    Toast.LENGTH_LONG).show();
                            } else {
                                Toast.makeText(context, response.getError()
                                    .getErrorMessage(),
                                    Toast.LENGTH_LONG).show();
                            }
                        }
                    });
            Bundle params = request.getParameters();
            params.putString("name", "รูปภาพที่อัพโหลด");
            request.setParameters(params);
            Request.executeBatchAsync(request);
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    } else {
        pendingAction = PendingAction.POST_PHOTO;
    }
}

private boolean hasPublishPermission() {
    Session session = Session.getActiveSession();
    return session != null
        && session.getPermissions().contains("publish_actions");
}

@SuppressWarnings("incomplete-switch")
private void handlePendingAction() {
    PendingAction previouslyPendingAction = pendingAction;
    // These actions may re-set pendingAction if they are still pending, but
    // we assume they
    // will succeed.
    pendingAction = PendingAction.NONE;

    switch (previouslyPendingAction) {
        case POST_PHOTO:
            postPhoto();
            break;
    }
}

```

```

private interface GraphObjectWithId extends GraphObject {
    String getId();
}

public void WrongAnswer() {
    if (Sound == true) {
        playSound(0);
    }
    if (count == 1) {
        Toast.makeText(context, "คำตอบที่ถูกต้อง", Toast.LENGTH_SHORT).show();
        count = 2;
    } else if (count == 2) {
        Toast.makeText(context, "ไม่ถูกต้อง ข้อที่ 2", Toast.LENGTH_SHORT)
            .show();
        count = 3;
    } else {
        Toast.makeText(context, "ถูกต้อง... ข้อที่ 3", Toast.LENGTH_SHORT)
            .show();
        count = 1;
    }
    CorrectDialogButton();
}

public void CorrectDialogButton() {
    ans1.setText("");
    ans2.setText("");
    ans3.setText("");
    btn_choice1.setVisibility(View.VISIBLE);
    btn_choice2.setVisibility(View.VISIBLE);
    btn_choice3.setVisibility(View.VISIBLE);
    btn_choice4.setVisibility(View.VISIBLE);
    btn_choice5.setVisibility(View.VISIBLE);
    btn_choice6.setVisibility(View.VISIBLE);
    btn_choice7.setVisibility(View.VISIBLE);
    btn_choice8.setVisibility(View.VISIBLE);
}

public void logKeyHash() {
    try {
        PackageInfo info = getPackageManager().getPackageInfo(
            getPackageName(), PackageManager.GET_SIGNATURES);
        for (Signature signature : info.signatures) {
            MessageDigest md = MessageDigest.getInstance("SHA");
            md.update(signature.toByteArray());
            Log.d("KeyHash",
                Base64.encodeToString(md.digest(), Base64.DEFAULT));
        }
    } catch (NameNotFoundException e) {
        e.printStackTrace();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
}

public void changePic() {
    if (correct == false) {
        tenCountDownTimer.cancel();
        tenCountDownTimer = new TenCountDownTimer(tenSec, interval);
        tenCountDownTimer.start();
    }
    correct = false;
}

```

```

if(arraylist.size()>1){
    do {
        previousRandom = currentRandom;
        index = randomGenerator.nextInt(arraylist.size());
        currentRandom = index;
    } while (currentRandom == previousRandom);
}
else{
    previousRandom = currentRandom;
    index = randomGenerator.nextInt(arraylist.size());
    currentRandom = index;
}
index = currentRandom;
OnClickAnsButton();
OnClickAnsButtonTwo();
ans1.setText("");
ans2.setText("");
ans3.setText("");
pic_number.setText(Integer.toString(arraylistSize - arraylist.size())
    + " / " + Integer.toString(arraylistSize));
pic_number.setTypeface(null, Typeface.BOLD);
btn_choice1.setText(arraylist.get(index).getChoice1());
btn_choice2.setText(arraylist.get(index).getChoice2());
btn_choice3.setText(arraylist.get(index).getChoice3());
btn_choice4.setText(arraylist.get(index).getChoice4());
btn_choice5.setText(arraylist.get(index).getChoice5());
btn_choice6.setText(arraylist.get(index).getChoice6());
btn_choice7.setText(arraylist.get(index).getChoice7());
btn_choice8.setText(arraylist.get(index).getChoice8());
imageLoader.getInstance().displayImage(arraylist.get(index).getImage(),
    gamepic);

posInbtn = index;

btn_choice1.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        btn_choice1.setVisibility(View.GONE);

        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice1());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 1;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(posInbtn).getChoice1());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 1;
        } else {
            ans3.setText(arraylist.get(posInbtn).getChoice1());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 1;
            i = 0;
            checkAnswer();
        }
    }
});

```

```

btn_choice2.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice2.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice2());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 2;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(posInbtn).getChoice2());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 2;
        } else {
            ans3.setText(arraylist.get(posInbtn).getChoice2());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 2;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice3.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice3.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice3());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 3;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(posInbtn).getChoice3());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 3;
        } else {
            ans3.setText(arraylist.get(posInbtn).getChoice3());
            ans3.setTypeface(null, Typeface.BOLD);
            btn2 = 3;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice4.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice4.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice4());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 4;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(posInbtn).getChoice4());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 4;
        } else {
            ans3.setText(arraylist.get(posInbtn).getChoice4());

```

```

        ans3.setTypeface(null, Typeface.BOLD);
        btn3 = 4;
        i = 0;
        checkAnswer();
    }
}
});

btn_choice5.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice5.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice5());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 5;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(posInbtn).getChoice5());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 5;
        } else {
            ans3.setText(arraylist.get(posInbtn).getChoice5());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 5;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice6.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice6.setVisibility(View.GONE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice6());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 6;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(posInbtn).getChoice6());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 6;
        } else {
            ans3.setText(arraylist.get(posInbtn).getChoice6());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 6;
            i = 0;
            checkAnswer();
        }
    }
});

btn_choice7.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice7.setVisibility(View.INVISIBLE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice7());
            ans1.setTypeface(null, Typeface.BOLD);

```

```

        btn = 7;
    } else if (ans2.getText().equals("")) {
        ans2.setText(arraylist.get(posInbtn).getChoice7());
        ans2.setTypeface(null, Typeface.BOLD);
        btn2 = 7;
    } else {
        ans3.setText(arraylist.get(posInbtn).getChoice7());
        ans3.setTypeface(null, Typeface.BOLD);
        btn3 = 7;
        i = 0;
        checkAnswer();
    }
}
});

btn_choice8.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        btn_choice8.setVisibility(View.INVISIBLE);
        if (ans1.getText().equals("")) {
            ans1.setText(arraylist.get(posInbtn).getChoice8());
            ans1.setTypeface(null, Typeface.BOLD);
            btn = 8;
        } else if (ans2.getText().equals("")) {
            ans2.setText(arraylist.get(posInbtn).getChoice8());
            ans2.setTypeface(null, Typeface.BOLD);
            btn2 = 8;
        } else {
            ans3.setText(arraylist.get(posInbtn).getChoice8());
            ans3.setTypeface(null, Typeface.BOLD);
            btn3 = 8;
            i = 0;
            checkAnswer();
        }
    }
});

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    Intent intent = new Intent(ghostmaster.this,
        choosemode.class);
    startActivity(intent);
    finish();
    overridePendingTransition(R.layout.mainfadein,
        R.layout.mainfadeout);
}

public void Winner() {
    countdownTimer.cancel();
    tenCountDownTimer.cancel();
    AlertDialog.Builder builder2 = new Builder(ghostmaster.this);
    builder2.setTitle("จบเกม");
    builder2.setMessage("ขอบคุณมากครับ/ค่ะที่ร่วมสนุกกับเกม");
    builder2.setPositiveButton("โอเค",
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                // TODO Auto-generated method stub
                Intent intent = new Intent(ghostmaster.this,
                    ghostmaster.class);

```

```

        startActivity(intent);
        finish();
        overridePendingTransition(R.layout.mainfadein,
            R.layout.mainfadeout);
    }
});

builder2.setNegativeButton("กลับหน้าหลัก",
    new DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            Intent intent = new Intent(ghostmaster.this,
                MainActivity.class);
            startActivity(intent);
            finish();
            overridePendingTransition(R.layout.mainfadein,
                R.layout.mainfadeout);
        }
    });
AlertDialog alert = builder2.create();
alert.setCanceledOnTouchOutside(false);
alert.show();
if (Sound == true) {
    playSound(2);
}

public void removePage(int position) {
    if ((position < 0) || (position >= arraylist.size())
        || (arraylist.size() <= 1)) {
    } else {
        if (position == arraylist.size() - 1) {
            arraylist.remove(index);
            changePic();
            pos = 0;
        } else {
            arraylist.remove(index);
            changePic();
        }
    }
}

public class MyCountDownTimer extends CountDownTimer {
    public MyCountDownTimer(long startTime, long interval) {
        super(startTime, interval);
    }

    @Override
    public void onFinish() {
        // TODO Auto-generated method stub
        countdown_time.setText("00:00");
        lastpic = lastpic + 1;
        AlertDialog.Builder builder = new Builder(ghostmaster.this);
        builder.setTitle("จบเกม");
        builder.setMessage("หมดเวลาแล้วระดับไหน\nภาคต่อของคุณ " + score
            + " จาก " + lastpic + " รอบ");
        builder.setPositiveButton("เล่นใหม่",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub
                    Intent intent = new Intent(ghostmaster.this,ghostmaster.class);

```

```

        startActivity(intent);
        finish();
        overridePendingTransition(R.layout.mainfadein,
            R.layout.mainfadeout);
    }
});
builder.setNegativeButton("กลับหน้าจอหลัก",
    new DialogInterface.OnClickListener() {

        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            Intent intent = new Intent(ghostmaster.this,
                MainActivity.class);
            startActivity(intent);
            finish();
            overridePendingTransition(R.layout.mainfadein,
                R.layout.mainfadeout);
        }
    });
AlertDialog alert = builder.create();
alert.setCanceledOnTouchOutside(false);
alert.show();
if (Sound == true) {
    playSound(3);
}
}

public String formatTime(long millis) {
    String output = "00:00";
    long seconds = millis / 1000;
    long minutes = seconds / 60;
    seconds = seconds % 60;
    minutes = minutes % 60;
    String sec = String.valueOf(seconds);
    String min = String.valueOf(minutes);
    if (seconds < 10)
        sec = "0" + seconds;
    if (minutes < 10)
        min = "0" + minutes;

    output = min + ":" + sec;
    return output;
} // formatTime

@Override
public void onTick(long millisUntilFinished) {
    // TODO Auto-generated method stub
    onPauseTime = millisUntilFinished;
    countdown_time.setText(formatTime(millisUntilFinished));
}

}

public class TenCountDownTimer extends CountDownTimer {
    private Random randomGenerator = new Random();

    public TenCountDownTimer(long startTime, long interval) {
        super(startTime, interval);
    }
}

```

```
@Override
public void onFinish() {
    // TODO Auto-generated method stub
    changePic();
}

public String formatTime(long millis) {
    String output = "00";
    long seconds = millis / 1000;
    seconds = seconds % 60;
    String sec = String.valueOf(seconds);
    if (seconds < 10)
        sec = "0" + seconds;
    output = "" + sec;
    return output;
} // formatTime

@Override
public void onTick(long millisUntilFinished) {
    // TODO Auto-generated method stub
    tenCountTxt.setText(formatTime(millisUntilFinished));
}
}
}
```



THAI-NICHI INSTITUTE OF TECHNOLOGY

TNI

ภาคผนวก ค

รายงานประจำสัปดาห์

TNI

ประวัติผู้จัดทำโครงการ

ชื่อ – สกุล	นายชุตติภัทร์ โชคพิพัฒน์พร
วัน เดือน ปีเกิด	7 เมษายน 2535
ประวัติการศึกษา	
ระดับประถมศึกษา	ประถมศึกษาตอนปลาย พ.ศ. 2547 โรงเรียนแม่พระฟาติมา
ระดับมัธยมศึกษา	มัธยมศึกษาตอนปลาย พ.ศ. 2553 โรงเรียนสาธิตแห่งมหาวิทยาลัยรามคำแหง (ฝ่ายมัธยมฯ)
ระดับอุดมศึกษา	คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีมัลติมีเดีย พ.ศ. 2556 สถาบันเทคโนโลยีไทย – ญี่ปุ่น
ทุนการศึกษา	ทุนการศึกษากายใต้กองทุน TNI (ทุนประเภทที่ 3)
ประวัติการฝึกอบรม	1. Computer Art & Graphic Design ณ โรงเรียนสอนคอมพิวเตอร์ NetDesign 2. ธุรกิจAnimationในประเทศไทย ณ สถาบันเทคโนโลยีไทย – ญี่ปุ่น 3. โครงการ Kizuna Bond Project ครั้งที่ 3 ณ เมืองโตเกียว, เมืองเคเซนนุมะ, และเมืองโตโยฮาชิ ประเทศญี่ปุ่น
ผลงานที่ได้รับการตีพิมพ์	- ไม่มี -

TNI