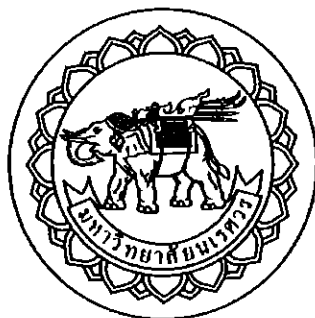




การติดตามตำแหน่งแบบเรียลไทม์บนแอนดรอยด์

REAL TIME LOCATION-BASED TRACKING ON ANDROID OS

นายทรงพล

ยลพันธุ์

รหัส 51364781

นายพัลลภ

อุดมสุข

รหัส 51364880

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 2 ก.ค. 2556
เลขทะเบียน..... 16279946
เลขเรียกหนังสือ..... ผร.
มหาวิทยาลัยนครสวรรค์ 1419

2554

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์

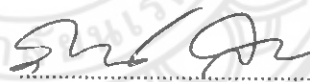
ปีการศึกษา 2554



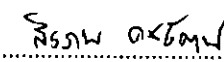
ใบรับรองปริญญาโท

หัวข้อโครงการ	การติดตามตำแหน่งแบบเรียลไทม์บนแอนดรอยด์	
ผู้ดำเนินโครงการ	นายทรงพล ชลพันธุ์	รหัสประจำตัว 51364781
	นายหัตถ์ถ อุดมสุข	รหัสประจำตัว 51364880
อาจารย์ที่ปรึกษา	ดร.พงศ์พันธ์ กิจสนาโยธิน	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2554	

มหาวิทยาลัยนเรศวร อนุมัติโครงการฉบับนี้เป็นส่วนหนึ่งของ คณะวิศวกรรมศาสตร์
การศึกษาดำเนินการหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะกรรมการสอบโครงการวิศวกรรม


.....ประธานกรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)


.....กรรมการ
(อาจารย์จิราพร พุกสุข)


.....กรรมการ
(อาจารย์สิริภพ ละครรัตน์)


.....กรรมการ
(ดร.พงศ์พันธ์ กิจสนาโยธิน)

หัวข้อโครงการ	การติดตามตำแหน่งแบบเรียลไทม์บนแอนดรอยด์		
ผู้ดำเนินโครงการ	นายทรงพล	ชลพันธ์	รหัส 51364781
	นายพัลลภ	อุดมสุข	รหัส 51364880
อาจารย์ที่ปรึกษา	อาจารย์พงศ์พันธ์	กิจสนา โยธิน	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2554		

บทคัดย่อ

โครงการนี้เป็นการออกแบบและสร้างโปรแกรมบนระบบปฏิบัติการแอนดรอยด์ วัตถุประสงค์หลักของโครงการคือการติดตามตำแหน่งของผู้ใช้งานโปรแกรมที่อยู่ในระบบแบบเรียลไทม์ คณะผู้จัดทำโครงการได้ออกแบบโปรแกรมให้เข้าถึงตำแหน่งปัจจุบันของผู้ใช้งานที่อยู่ในระบบจากฐานข้อมูล และส่งต่อข้อมูลเหล่านั้นไปยังผู้ใช้งานอื่นๆที่อยู่ในระบบในรูปแบบ JSON ผ่านทางโปรแกรมเว็บเซอร์วิส โปรแกรมสามารถค้นหาตำแหน่งปัจจุบันของผู้ใช้งานที่อยู่ในระบบจาก GPS หรือเรดาร์เตอร์ (ผ่านทาง Google Android Library) จากนั้น โปรแกรมจะรวบรวมตำแหน่งที่ได้แล้วส่งกลับไปยังเซิร์ฟเวอร์ฐานข้อมูลในรูปแบบ JSON ผ่านทางเว็บเซอร์วิส โดยเครื่องมือพัฒนาซอฟต์แวร์ที่ถูกนำมาประยุกต์ใช้ได้แก่ Eclipse Android SDK-Platform และ Google map

Project Title	Real time Location-Based Tracking on Android		
Name	Mr. Songpon Yonpan	ID.	51364781
	Mr. Panlop Udomsook	ID.	51364880
Project Advisor	Phongphun Kijsanayothin , Ph.D.		
Major	Computer Engineering.		
Department	Electrical and Computer Engineering.		
Academic year	2011		

.....

Abstracts

This project designed and implemented a software program on Android operating system. The main objective of the program is to track the locations of online users in real time mode. We designed the program to access the current location of online users from database and broadcast those information to other online users in JSON format via our simple web-service application. The current location of online users can be gathered either GPS or Router (via Google Android Library). The gathered location is also sent back to the database server in JSON format via web-service. The software development tools such as Eclipse, Android SDK-Platform, and Google Map have been applied for this project.

กิตติกรรมประกาศ

ทางคณะผู้จัดทำโครงการการติดตามตำแหน่งแบบเรียลไทม์บนแอนดรอยด์ขอขอบคุณ
อาจารย์พงศ์พันธ์ กิจสนาโยธิน ที่ให้คำแนะนำและความช่วยเหลือในโครงการนี้ให้ดำเนินการไป
ได้ด้วยดีตลอดจนอาจารย์ทุกท่านในภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่คอยดูแลและให้
คำปรึกษาด้วยความเต็มใจเสมอมาจนโครงการสำเร็จลุล่วงไปได้ด้วยดี สุดท้ายขอขอบพระคุณบิดา
และมารดาของคณะผู้จัดทำที่เป็นผู้สนับสนุนในทุกๆด้าน โดยเฉพาะด้านการศึกษาลดลงมา คณะ
ผู้จัดทำขอขอบพระทัยในพระคุณ และขอกราบขอบพระคุณมา ณ ที่นี้



นายทรงพล ชลพันธ์
นายพัลลภ อุดมสุข

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญรูป.....	ช
บทที่ 1 บทนำ	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
1.4 ขอบเขตการทำโครงการ	2
1.5 ขั้นตอนการดำเนินงาน	2
1.6 แผนการดำเนินงาน	3
1.7 รายละเอียดการประมาณตลอดโครงการ.....	4
บทที่ 2 หลักการและทฤษฎี.....	5
2.1 โปรแกรมอีคลิปส์.....	5
2.1.1 ความหมาย	6
2.1.2 คุณสมบัติที่น่าสนใจของโปรแกรม Eclipse Platform.....	6
2.1.3 สถาปัตยกรรมของโปรแกรม Eclipse Platform และ Plug-in.....	7
2.2 ระบบปฏิบัติการแอนดรอยด์.....	10
2.3 พีเอชพี, มาย เอสคิวแอล และองค์ประกอบที่เกี่ยวข้อง	10
2.3.1 พีเอชพี.....	11
2.3.2 เอชทีทีพี	11
2.3.3 มายเอสคิวแอล	11
2.4 สัญกรณ์วัตถุจาวาสคริปต์.....	12
2.5 จีพีเอส	15
2.5.1 จีพีเอส คืออะไร.....	15
2.5.2 การทำงานของจีพีเอส	16
2.5.3 หน้าที่สำคัญของดาวเทียมจีพีเอส.....	16

สารบัญ (ต่อ)

หน้า

2.6 แผนที่ภูเก็ล.....	17
บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน	18
3.1 แผนผังการออกแบบการทำงานโดยรวมของโปรแกรม	18
3.2 การออกแบบหน้าต่าง GUI สำหรับการใช้งาน.....	19
3.2.1 หน้าลงชื่อเข้าใช้งาน (Login Page).....	19
3.2.2 หน้าลงทะเบียน (Register Page)	20
3.2.3 หน้าแสดงผู้ใช้งาน (Online Page)	21
3.2.4 หน้าแสดงแผนที่ (Map Page).....	22
3.3 ระบบลงทะเบียน (Register).....	23
3.4 ระบบเข้าใช้งาน (Log in)	24
3.5 การรับพิกัด	25
3.6 การขอเอพีไอคีย์เพื่อเรียกใช้งานแผนที่ของกูเกิล	26
บทที่ 4 ผลการดำเนินโครงการ	29
4.1 ผลการทดลองเข้าใช้งานโปรแกรม	29
4.1.1 ผลการทดลองส่วนของการลงทะเบียน	30
4.1.2 ผลการทดลองส่วนของการลงชื่อเข้าใช้งาน.....	31
4.2 ผลการทดลองโปรแกรมในกรณีผู้ใช้งานอยู่ในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร).....	33
4.2.1 กรณีผู้ใช้งาน 2 ผู้ใช้งาน	33
4.2.2 กรณีผู้ใช้งาน 3 ผู้ใช้งาน	34
4.3 ผลการทดลองโปรแกรมในกรณีผู้ใช้งานอยู่ในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร).....	35
4.3.1 กรณีผู้ใช้งาน 2 ผู้ใช้งาน	36
4.3.2 กรณีผู้ใช้งาน 3 ผู้ใช้งาน	37
บทที่ 5 สรุปและแนวทางการพัฒนา	43
5.1 สรุปผลการดำเนินงาน	43
5.2 ปัญหาและแนวทางการแก้ไข.....	43
5.3 แนวทางการพัฒนาและข้อเสนอแนะ	43

สารบัญ (ต่อ)

	หน้า
เอกสารอ้างอิง	43
ภาคผนวก ก.	44
ประวัติผู้เขียน โครงการ.....	71



สารบัญรูป

	หน้า
รูปที่ 2.1 สัญลักษณ์ Eclipse Platform	5
รูปที่ 2.2 หน้าต่างของโปรแกรม Eclipse Platform	6
รูปที่ 2.3 หน้าต่างแสดงสถาปัตยกรรมของโปรแกรม Eclipse Platform และ Plug-in	7
รูปที่ 2.4 หน้าต่างแสดงการจำลอง Eclipse Platform ที่มีหลาย Plug-in ประกอบกันขึ้น	8
รูปที่ 2.5 หน้าต่างแสดงขั้นตอนการทำงานของโปรแกรม Eclipse Platform	9
รูปที่ 2.6 แสดงตัวอย่างโปรแกรมเจชัน	13
รูปที่ 2.7 แสดงการเปรียบเทียบรูปแบบของ เอกซ์เอ็มแอล กับ เจชัน	13
รูปที่ 2.8 แสดงรูปแบบโครงสร้างเจชัน(1).....	14
รูปที่ 2.9 แสดงรูปแบบโครงสร้างเจชัน(2).....	15
รูปที่ 3.1 การออกแบบการทำงานโดยรวมของโปรแกรม	18
รูปที่ 3.2 แสดงหน้าลงชื่อเข้าใช้งาน (Login Page).....	19
รูปที่ 3.3 แสดงหน้าลงทะเบียน (Register Page)	20
รูปที่ 3.4 แสดงหน้าแสดงผู้ใช้งานปัจจุบัน (Online Page).....	21
รูปที่ 3.5 แสดงหน้าแสดงแผนที่ (Map Page).....	22
รูปที่ 3.6 แสดงทำงานของระบบลงทะเบียน (Register)	23
รูปที่ 3.7 แสดงการทำงานของระบบเข้าใช้งาน (Login).....	24
รูปที่ 3.8 แสดงการทำงานของระบบการรับพัสดุ.....	25
รูปที่ 3.9 แสดงการใส่ที่อยู่ของไฟล์ชื่อ Keytool.exe บน Command Prompt	26
รูปที่ 3.10 แสดงการใส่คำสั่งการรันไฟล์ debug.keystore บน Command Prompt	27
รูปที่ 3.11 แสดงผลการรันไฟล์ debug.keystore บน Command Prompt.....	27
รูปที่ 3.12 แสดงการขอ API Key	28
รูปที่ 4.1 การกรอกข้อมูลในการลงทะเบียน.....	30
รูปที่ 4.2 การบันทึกข้อมูลผู้ใช้งานในฐานข้อมูล	30
รูปที่ 4.3 การลงชื่อเข้าใช้งาน	31
รูปที่ 4.4 การลงชื่อเข้าใช้งานโดยกรอกข้อมูลผิด	32
รูปที่ 4.5 หน้าต่างแสดงผู้ใช้งานที่ออนไลน์	32
รูปที่ 4.6 แสดงความเปลี่ยนแปลงคอลัมน์ Active ในฐานข้อมูล.....	33

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.7 แสดงฐานข้อมูลกรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะใกล้เคียงกัน 33 (น้อยกว่า 1.5x2.5 กิโลเมตร)..... 33	33
รูปที่ 4.8 แสดงแผนที่กรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร)..... 34	34
รูปที่ 4.9 แสดงฐานข้อมูลกรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะใกล้เคียงกัน 34 (น้อยกว่า 1.5x2.5 กิโลเมตร)..... 34	34
รูปที่ 4.10 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร).... 35	35
รูปที่ 4.11 แสดงฐานข้อมูลกรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร) 36	36
รูปที่ 4.12 แสดงแผนที่กรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)..... 36	36
รูปที่ 4.13 แสดงแผนที่กรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)..... 37	37
รูปที่ 4.14 แสดงฐานข้อมูลกรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร) 37	37
รูปที่ 4.15 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)..... 38	38
รูปที่ 4.16 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)..... 39	39
รูปที่ 4.17 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)..... 40	40
รูป ก-1 แสดงการเรียกใช้งาน Google Map..... 44	44
รูป ก-2 แสดงหน้า GUI โดยรวม 45	45
รูป ก-3 แสดงหน้าการทำ Login Page 46	46
รูป ก-4 แสดงหน้าการทำ Register Page 47	47
รูป ก-5 แสดงหน้าการทำ Online Page..... 48	48
รูป ก-6 แสดงหน้าการทำ Map Page 49	49
รูป ก-7 แสดงหน้าต้อนรับของโปรแกรม AppServ 51	51
รูป ก-8 แสดงหน้าข้อตกลงของโปรแกรม AppServ 52	52
รูป ก-9 แสดงหน้าเลือกพื้นที่สำหรับเก็บข้อมูลของโปรแกรม AppServ 53	53
รูป ก-10 แสดงหน้าเลือก โปรแกรมที่เกี่ยวข้องของ โปรแกรม AppServ 54	54
รูป ก-11 แสดงหน้าสำหรับกรอกรายละเอียดสำหรับ Apache server ของ โปรแกรม AppServ 55	55
รูป ก-12 แสดงหน้าสำหรับกรอกรายละเอียดสำหรับมายเอสคิวแอลของ โปรแกรม AppServ 56	56
รูป ก-13 แสดงหน้าหลังจากติดตั้งเสร็จสมบูรณ์ของ โปรแกรม AppServ 57	57
รูป ก-14 แสดงหน้าหลักหลังจากติดตั้ง โปรแกรม AppServ 58	58
รูป ก-15 แสดงหน้า localhost และการสร้างฐานข้อมูล 59	59

สารบัญรูป (ต่อ)

	หน้า
รูป ก-16 แสดงหน้าการสร้างตารางและกำหนดเขตของข้อมูล	59
รูป ก-17 แสดงหน้าการใส่รายละเอียดของเขตข้อมูล(ส่วนที่ 1)	60
รูป ก-18 แสดงหน้าการใส่รายละเอียดของเขตข้อมูล(ส่วนที่ 2)	61
รูป ก-19 การบันทึกและการเพิ่ม Field	61
รูป ก-20 แสดงหน้าการใส่รายละเอียดของเขตข้อมูล	62



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

กูเกิลแอนดรอยด์ (Google Android) คือ ระบบปฏิบัติการอย่างหนึ่งที่เป็นซอฟต์แวร์แพลตฟอร์มบนอุปกรณ์ไร้สาย โดยสร้างขึ้นมาจากระบบปฏิบัติการลินุกซ์ ถูกพัฒนาโดย Google ทาง Google ได้เปิดให้นักพัฒนาเข้ามามีส่วนร่วมในการจัดการโค้ดต่างๆได้ด้วยภาษาจาวา (Java programming language) และสามารถเขียนโค้ดเพื่อควบคุมส่วนต่างๆผ่านทาง Google-Developed Java Libraries ที่ทาง Google มีไว้ให้ สามารถเขียนโปรแกรมเพื่อรันบนแอนดรอยด์ได้หลายภาษา

ทางผู้จัดทำโครงการจึงเห็นถึงยุคสมัยในอนาคตที่ระบบปฏิบัติการแอนดรอยด์มีอัตราการเติบโตที่เพิ่มมากขึ้น จึงเกิดแรงบันดาลใจที่จะศึกษา โดยใช้เครื่องมือในการพัฒนาคือ Eclipse , Android SDK-Platform , Android Database และ API สำหรับ Android Version นำมาเขียนโปรแกรมเพื่อพัฒนาต่อไป

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 สร้างโปรแกรมที่สามารถค้นหาตำแหน่งอุปกรณ์ที่ติดตั้งโปรแกรมที่พัฒนาขึ้น
- 1.2.2 สร้างเว็บเซอร์วิสเพื่อติดต่อระหว่างแอนดรอยด์กับฐานข้อมูลได้
- 1.2.3 สามารถนำพิกัดจาก GPS และ Internet มาใช้งานเพื่อแสดงตำแหน่งบนแผนที่กูเกิลได้
- 1.2.4 สร้างฐานข้อมูลเพื่อเก็บข้อมูลของผู้ใช้งานโปรแกรม
- 1.2.5 สามารถนำโปรแกรมที่พัฒนาไปใช้กับอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ได้
- 1.2.6 โปรแกรมที่พัฒนาสามารถนำไปประยุกต์และพัฒนาต่อไปได้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 สามารถติดตั้งโปรแกรม Eclipse และเข้าใจการทำงานของโปรแกรม
- 1.3.2 สามารถทำให้โปรแกรมที่พัฒนาขึ้นติดต่อกับ GPS และเรียกตำแหน่งจาก GPS มาแสดงตำแหน่งบนแผนที่กูเกิลและรันบนอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ได้

- 1.3.3 สามารถเข้าใจการใช้ภาษาจาวา (Java) เขียนโปรแกรมค้นหาตำแหน่งโทรศัพท์
- 1.3.4 สามารถสร้างและเข้าใจหลักการทำงานของเว็บเซิร์ฟเวอร์ที่เชื่อมต่อกับโปรแกรม
- 1.3.5 สามารถสร้างฐานข้อมูลเพื่อเก็บรายละเอียดข้อมูลของผู้ใช้งานโปรแกรม
- 1.3.6 สามารถสร้างโปรแกรมค้นหาตำแหน่งโทรศัพท์ที่ติดตั้งโปรแกรมเกมที่พัฒนาขึ้น 1 โปรแกรม

1.4 ขอบเขตการทำโครงการ

- 1.4.1 ใช้โปรแกรม Eclipse เวอร์ชัน Classic ในการพัฒนาโปรแกรม
- 1.4.2 ใช้ Appserv เวอร์ชัน 2.5.10 ในการสร้างเว็บเซิร์ฟเวอร์และฐานข้อมูล
- 1.4.3 ใช้ GPS ในการเรียกพิกัดเมื่อเชื่อมต่อ GPS
- 1.4.4 ใช้ Network ในการเรียกพิกัดเมื่อเชื่อมต่อ Internet
- 1.4.5 โปรแกรมทำงานได้กับอุปกรณ์ที่เป็นระบบปฏิบัติการแอนดรอยด์
- 1.4.6 โปรแกรมสามารถเรียกใช้งานแผนที่ของกูเกิลได้
- 1.4.7 โปรแกรมสามารถแสดงตำแหน่งบนแผนที่กูเกิลแบบ Real Time ได้

1.5 ขั้นตอนการดำเนินงาน

- 1.5.1 ศึกษาขั้นตอนการติดตั้งโปรแกรม Eclipse
- 1.5.2 ศึกษาการติดตั้ง SDK Platform ของ Android แต่ละ Version
- 1.5.3 ศึกษาการใช้ภาษาจาวา (Java) ที่ใช้ในการพัฒนาโปรแกรม
- 1.5.4 ศึกษาการสร้างโปรแกรม Android บนโปรแกรม Eclipse เพื่อนำมาติดตั้งในโทรศัพท์
- 1.5.5 ศึกษาการสร้างเว็บเซิร์ฟเวอร์เพื่อติดต่อกับโปรแกรม
- 1.5.6 ศึกษาการสร้างฐานข้อมูลเพื่อเก็บข้อมูลของผู้ใช้งานโปรแกรมศึกษาการใช้งาน GPS

บนระบบปฏิบัติการ Android

- 1.5.7 ศึกษาการรับตำแหน่ง GPS มาแสดงบนแผนที่
- 1.5.8 ทดสอบการแสดงผลตำแหน่งโทรศัพท์เครื่องที่ติดตั้งจากโปรแกรมที่พัฒนาขึ้น
- 1.5.9 ทดสอบการรับตำแหน่งจากโทรศัพท์เครื่องอื่นที่ได้ติดตั้งโปรแกรมที่พัฒนาขึ้นแล้ว

บันทึกผลการทดสอบ

- 1.5.10 หาข้อเสนอแนะและข้อแก้ไขพร้อมทั้งสรุปผลการทดสอบ

	เดียว) จากโปรแกรมที่พัฒนาขึ้น										
10.	ทดสอบการรับตำแหน่งจากโทรศัพท์เครื่องอื่นที่ได้ติดตั้งโปรแกรมที่พัฒนาขึ้นแล้วบันทึกผลการทดสอบ									←→	
11.	หาข้อเสนอแนะและข้อแก้ไขพร้อมทั้งสรุปผลการทดสอบ									←→	
12.	จัดทำรูปเล่มโครงการ							←→			←→

1.7 รายละเอียดการประมาณตลอดโครงการ

1. ค่าใช้จ่ายในการจัดทำสำเนาเอกสาร 1,000 บาท
 2. ค่าใช้จ่ายในการจัดทำรูปเล่มโครงการ 1,000 บาท
- รวม 2,000 บาท (สองพันบาทถ้วน)

บทที่ 2

หลักการและทฤษฎี

ในการพัฒนา Application บน Android ต้องมีการศึกษาโปรแกรมและส่วนต่างๆที่ใช้สำหรับพัฒนาคือ โปรแกรม Eclipse Platform ,Android SDK-Platform ,Database และ API สำหรับ Android Version ต่างๆรวมทั้งการศึกษาการทฤษฎีต่างๆเพื่อนำมาประยุกต์เพื่อให้ Application ที่พัฒนาขึ้นทำงานได้ตรงตามวัตถุประสงค์ ดังนี้

2.1 โปรแกรมอีคลิปส์

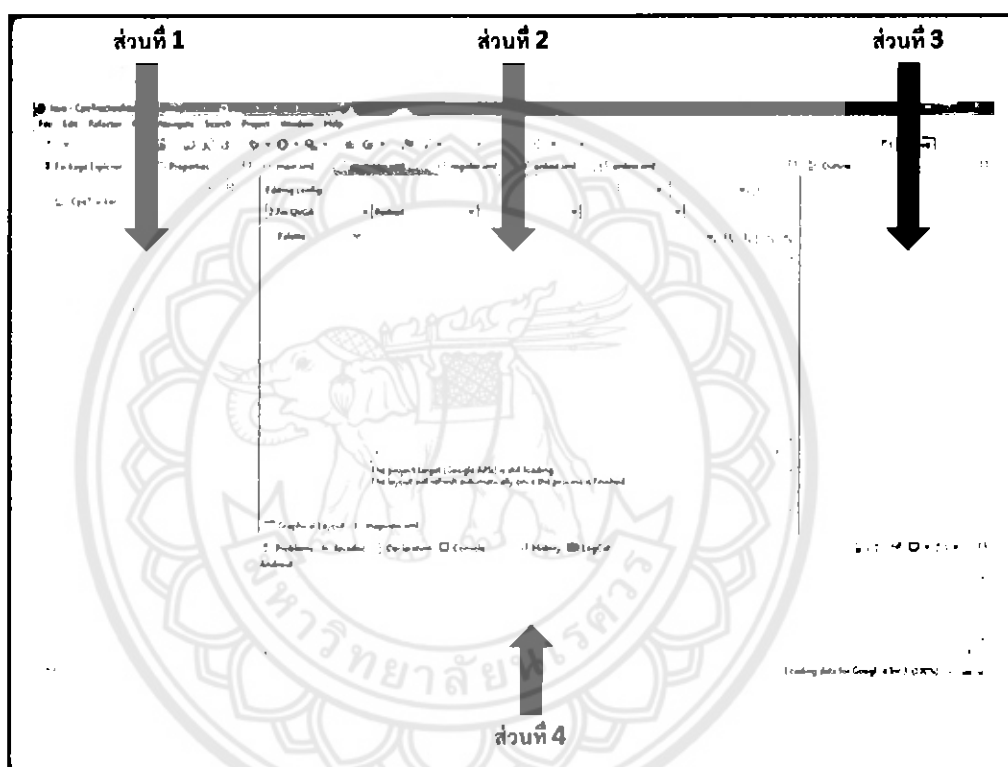
Eclipse [1] เป็นเครื่องมืออีกตัวหนึ่งที่กำลังได้รับความนิยมเป็นอย่างมาก จากผู้ที่ต้องการพัฒนาโปรแกรมด้วยภาษาจาวา โดยมีคุณสมบัติติดตั้งง่าย มีโปรแกรมเสริม (Plug-in) ที่ใช้เพิ่มประสิทธิภาพของโปรแกรมและที่สำคัญเป็นฟรีแวร์อีกด้วย สำหรับแนวโน้มในอนาคตของการใช้ภาษาจาวาที่ควบคู่กับการออกแบบเชิงวัตถุจะมีมากขึ้น เป็นเหตุผลให้ตัวอีคลิปส์เองจำเป็นต้องมีการพัฒนามากขึ้น โดยมีสัญลักษณ์ที่โดดเด่น ดังรูปที่ 2.1



รูปที่ 2.1 สัญลักษณ์ Eclipse Platform

2.1.1 ความหมาย

Eclipse Platform คือ โปรแกรมประเภทไอดีอี (IDE : Integrated Development Environment) เพื่อใช้เป็นเครื่องมือในการนำไปใช้พัฒนาซึ่งช่วยในการสร้างโปรแกรมประยุกต์ (Application), เว็บไซต์, โปรแกรมภาษาซีพลัสพลัส (C++) และอื่นๆ อีกทั้งเป็นโปรแกรมฟรีแวร์, มีโปรแกรมเสริม เพื่อใช้เสริมประสิทธิภาพการทำงานมากมาย ใช้ได้ทั้งระบบปฏิบัติการวินโดวส์ และ ลินุกซ์ โดยมีรูปร่างหน้าตาโปรแกรมดังนี้



รูปที่ 2.2 หน้าต่างของโปรแกรม Eclipse Platform

จากรูปที่ 2.2 แสดงแต่ละส่วนของหน้าต่างโปรแกรม Eclipse Platform โดยสามารถแบ่งได้ 4 ส่วน คือ

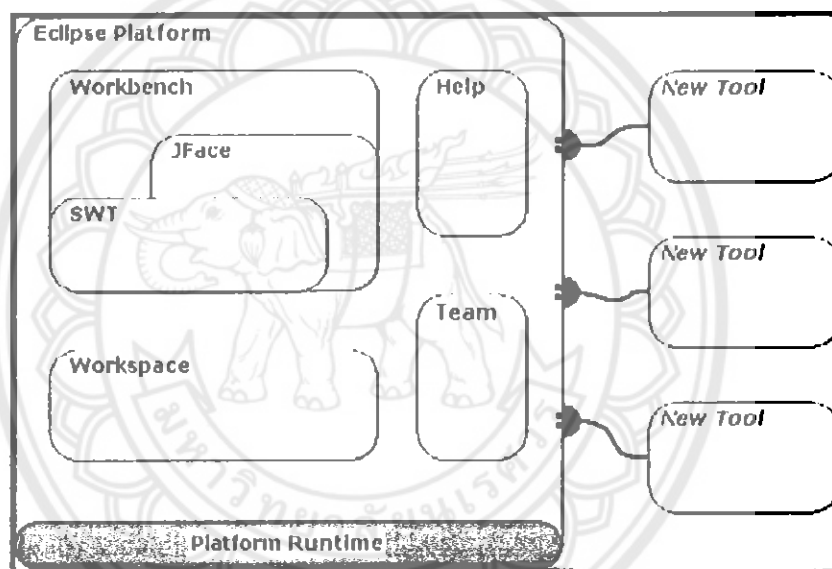
- ส่วนที่ 1 คือ ส่วน Navigator view ส่วนนี้เป็นส่วนที่ใช้ดูไฟล์ที่จะใช้งาน
- ส่วนที่ 2 คือ ส่วน Text view ส่วนที่ใช้ออกแบบและเขียนโค้ด
- ส่วนที่ 3 คือ ส่วนแสดงโครงสร้างโดยรวมของงาน
- ส่วนที่ 4 คือ ส่วนแสดงผลของการทำงาน

2.1.2 คุณสมบัติที่น่าสนใจของโปรแกรม Eclipse Platform

- สนับสนุนโครงสร้างของโปรแกรมที่หลากหลาย

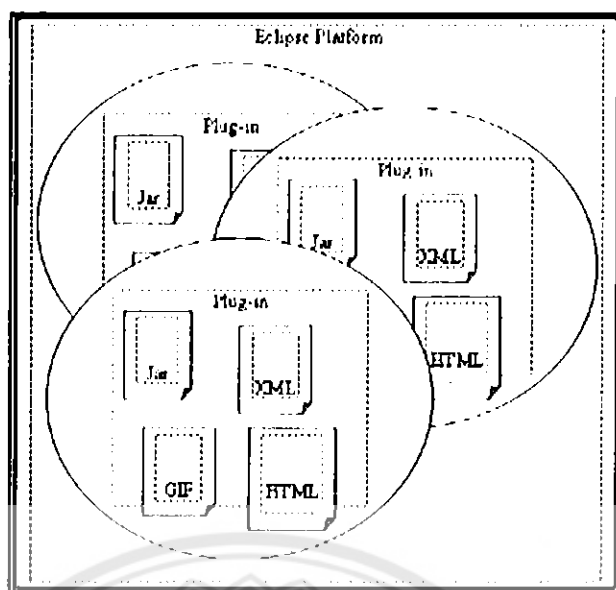
- เป็นโปรแกรมที่ใช้งานได้ฟรี
- เป็นโปรแกรมที่ทำงานร่วมกับไฟล์ได้หลายชนิด เช่น HTML, java, C, JSP, EJB, XML, GIF เป็นต้น
- สนับสนุนทั้งแบบส่วนต่อประสานกราฟฟิค (GUI : Graphical User Interface) และแบบไม่เป็นส่วนต่อประสานกราฟฟิค (non-GUI : non-Graphical User Interface) เป็นส่วนของสภาพแวดล้อมในการพัฒนาโปรแกรม
- ใช้งานได้ทั้งระบบปฏิบัติการวินโดวส์ และ ลินุกซ์

2.1.3 สถาปัตยกรรมของโปรแกรม Eclipse Platform และ Plug-in



รูปที่ 2.3 หน้าต่างแสดงสถาปัตยกรรมของโปรแกรม Eclipse Platform และ Plug-in
ที่มา: พนมพร (2548)

จากรูปที่ 2.3 แสดงสถาปัตยกรรมของโปรแกรม Eclipse Platform และ Plug-in โดยโปรแกรมประกอบไปด้วย Plug-in หลายๆตัว ซึ่ง Plug-in แต่ละตัวจะถูกเขียนด้วยภาษาจาวาและในรูปแบบไฟล์รวมไว้ด้วยกันคือไฟล์คอทจา (.jar : java Archive) บางครั้งอาจเป็นไฟล์ที่ใช้ทำงานเพียงไฟล์เดียว หรืออาจเกี่ยวข้องกับทรัพยากรอื่น เช่น รูปภาพ, Web Templates , Massage Catalogs และอื่นๆ Plug-in บางประเภทอาจจะอยู่ในรูปแบบของ HTML

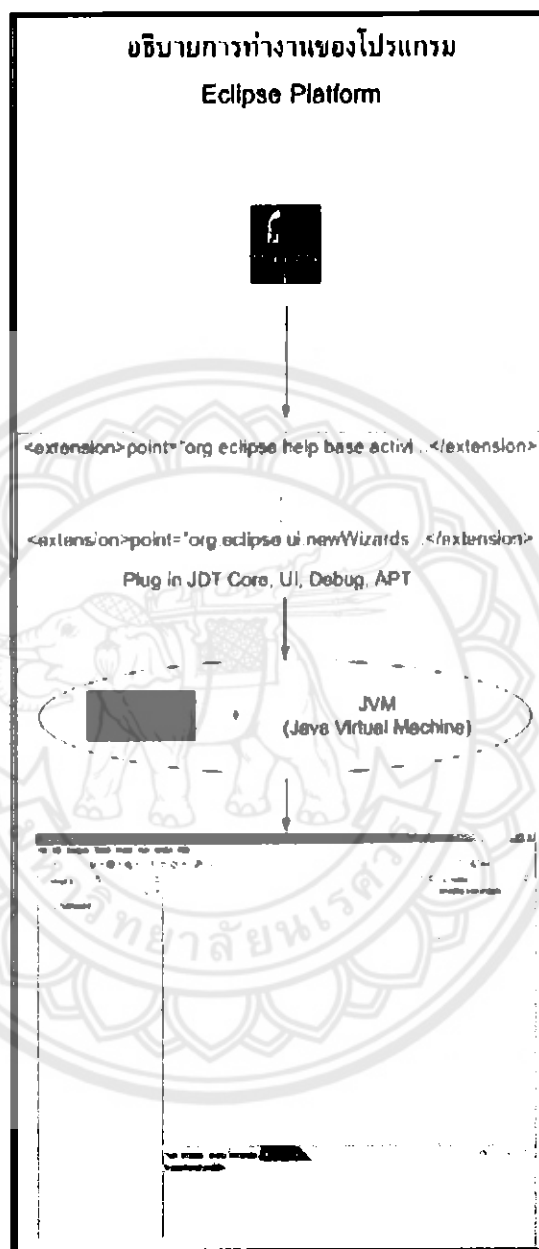


รูปที่ 2.4 หน้าต่างแสดงการจำลอง Eclipse Platform ที่มีหลาย Plug-in ประกอบกันขึ้น
ที่มา: พนมพร (2548)

จากรูปที่ 2.4 แสดงการจำลอง Eclipse Platform ที่มีหลาย Plug-in ประกอบกันขึ้น ผู้ใช้สามารถกำหนดการใช้งาน Plug-in ต่างๆ ได้โดยตั้งค่าในเมนูแทนที่จะต้องระบุชื่อ Plug-in ที่ใช้งานด้วยตนเอง

เมื่อมีการเปิดใช้งานโปรแกรม ขั้นตอนจะเริ่มจากตัวแพลตฟอร์มจะเข้าไปอ่านไฟล์แฟ้มที่ประกอบด้วยข้อมูลเกี่ยวกับไฟล์ที่แนบมาในการเก็บเดียวกัน (Manifest) ที่อยู่ใน Plug-in การอ่านแต่ละครั้งจะถูกบันทึกไปสู่หน่วยความจำโดยอ่านที่ส่วนต่อขยาย (Extension) โดยส่วนต่อขยายใดไม่มีการระบุชื่อ Plug-in นั้นก็จะไม่ถูกส่งต่อเข้าสู่หน่วยความจำทำให้ Plug-in ดังกล่าวไม่ถูกแสดงในหน่วยความจำที่โหลด Plug-in เข้ามาจะทำงานร่วมกับจาวา เวอร์ชวลแมชชีน (JVM : Java Virtual Machine)

การใช้งาน Plug-in บางชนิดจะต้องเรียกใช้โค้ดคำสั่งของ Plug-in ทำให้การเปิดใช้งาน Eclipse Platform ช้าเล็กน้อย โดยมีขั้นตอนการทำงานของโปรแกรม Eclipse Platform ดังรูปที่ 2.5



รูปที่ 2.5 หน้าต่างแสดงขั้นตอนการทำงานของโปรแกรม Eclipse Platform

ที่มา: พนมพร (2548)

2.2 ระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์ (Android) [2] คือ ระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ, แท็บเล็ตคอมพิวเตอร์, เน็ตบุ๊ก, ทำงานบนลินุกซ์เคอร์เนล (Linux kernel) เริ่มพัฒนาโดยบริษัทแอนดรอยด์ (Android Inc.) จากนั้น บริษัทแอนดรอยด์ถูกซื้อโดยกูเกิล และนำแอนดรอยด์ไปพัฒนาต่อภายหลังถูกพัฒนาในนามของกลุ่มพันธมิตรทางธุรกิจสำหรับพัฒนาอุปกรณ์พกพาโอเพ่นแฮนด์เซตอัลไลแอนซ์ (Open Handset Alliance) โดยทางกูเกิลได้เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่างๆ ด้วยภาษาจาวา และควบคุมอุปกรณ์ผ่านทางชุดจาวา ไลบรารี (Java libraries) ที่กูเกิลพัฒนาขึ้น

แอนดรอยด์ได้เป็นที่รู้จักต่อสาธารณชนเมื่อวันที่ 5 พฤศจิกายน ค.ศ. 2007 โดยทางกูเกิลได้ประกาศก่อตั้ง โอเพ่นแฮนด์เซตอัลไลแอนซ์ กลุ่มบริษัทฮาร์ดแวร์, ซอฟต์แวร์ และการสื่อสาร 48 แห่ง ที่ร่วมมือกันเพื่อพัฒนามาตรฐานเปิดสำหรับอุปกรณ์มือถือ ลิขสิทธิ์ของโค้ดแอนดรอยด์นี้จะใช้ในลักษณะของซอฟต์แวร์เสรี

จุดเด่นที่แอนดรอยด์เป็นระบบปฏิบัติการที่ได้รับความนิยมแพร่หลายอย่างรวดเร็วนี้ หลายๆปัจจัยที่เป็นแรงจูงใจให้ผู้ใช้งานอุปกรณ์พกพาหันมาสนใจและให้ความสำคัญ เพราะแอนดรอยด์เป็นระบบปฏิบัติการที่ยืดหยุ่น ใช้งานได้โดยไม่เสียค่าใช้จ่ายเพิ่มเติม มีเครื่องมือสำหรับพัฒนาอยู่เสมอ เปิดโอกาสกว้างสำหรับบุคคลที่สนใจพัฒนาต่อยอด ทำให้ทุกวันนี้ ระบบปฏิบัติการแอนดรอยด์มีผู้ใช้นับล้านทั่วโลก

2.3 พีเอชที, มาย เอสคิวแอล และองค์ประกอบที่เกี่ยวข้อง

อินเทอร์เน็ตเป็นเครือข่ายที่ประกอบไปด้วยคอมพิวเตอร์จำนวนมากเชื่อมต่อกัน และกำลังได้รับความนิยมเป็นอย่างสูง โดยการติดต่อสื่อสารจะใช้เอชทีทีพี (HTTP : Hyper Text Transfer Protocol) [3] และมีภาษา Markup ที่เรียกว่าเอชทีเอ็มแอล (HTML : Hyper Text Markup Language) เพื่อใช้สร้างเว็บเพจสำหรับแสดงบน Web browser ในปัจจุบันการให้บริการเว็บไซต์ถือเป็นบริการหลักบนเครือข่ายอินเทอร์เน็ต จึงมีการพัฒนารูปแบบเว็บไซต์อยู่ตลอดเวลา จากเดิมที่เนื้อหาบนเว็บเพจจะเป็นแบบตายตัว (Static Web Content) ซึ่งไม่สามารถโต้ตอบกับผู้ใช้ได้ ก็ได้รับการพัฒนาให้สามารถโต้ตอบกับผู้ใช้ได้ (Dynamic Web Content)

2.3.1 พีเอชพี

พีเอชพี (PHP: Hypertext Preprocessor) เป็นภาษา Script ที่นิยมนำมาใช้สร้างเว็บเพจที่สามารถโต้ตอบกับผู้ใช้ได้ (Dynamic Web Page) เนื่องจากมีรูปแบบที่ง่าย และถูกพัฒนามาเพื่อใช้ร่วมกับฐานข้อมูลมายเอสคิวแอล ได้อย่างมีประสิทธิภาพ นอกจากนี้ยังเป็น Open Source ซึ่งสามารถนำไปใช้งานได้โดยไม่เสียค่าใช้จ่าย

ความสามารถของพีเอชพีสามารถจัดการกับตัวแปรหลายๆประเภท เช่น เลขจำนวนเต็ม (Integer), เลขทศนิยมทศนิยม (float), สตริง (string) และ อาร์เรย์ (array) เป็นต้น สามารถรับข้อมูลแบบเอชทีเอ็มแอล มีความสามารถในการติดต่อและจัดการฐานข้อมูล การสร้างภาพกราฟฟิก และความสามารถอีกหลายด้านที่พีเอชพีทำได้

2.3.2 เอชทีทีพี

เอชทีทีพี (HTTP : Hyper Text Transfer Protocol) เป็นมาตรฐานที่ใช้ติดต่อสื่อสารในรูปแบบของการร้องขอ (Request) และการตอบสนอง (Response) ระหว่าง Browser ที่ทำงานอยู่บนเครื่องคอมพิวเตอร์ของผู้ใช้ (Client) กับ Web Server ที่ทำหน้าที่ให้บริการ หรือตอบสนองต่อคำร้องต่างๆ ที่ส่งมาจากคอมพิวเตอร์ที่ร้องขอ โดยทั่วไปจะเป็นการร้องขอเว็บเพจ โดยระหว่างการติดต่อสื่อสารกับ Server กับ Client อาจต้องผ่านอุปกรณ์เป็นจำนวนมาก เช่น Router เป็นต้น ส่วนการรับส่งข้อมูลจะเป็นหน้าที่ของเครือข่ายอินเทอร์เน็ต โดย Web Server จะรองรับการเชื่อมต่อจาก Client ได้พร้อมกันมากกว่า 1 การเชื่อมต่อ เมื่อ Web Server ให้บริการแก่ทุกคำร้องขอครบแล้ว จะหยุดรอและตรวจสอบว่ามีการร้องขอบริการเข้ามาใหม่หรือไม่ หากมีการร้องขอเข้ามา จึงจะทำการตอบกลับเพื่อแจ้งให้ Client ทราบว่าได้รับการร้องขอแล้ว จากนั้นจึงจะให้บริการตามที่ Client ต้องการ

2.3.3 มายเอสคิวแอล

มายเอสคิวแอล (MySQL) เป็นโปรแกรมบริหารจัดการในฐานข้อมูล โดยเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relational Database Management System) โดยการเข้าไปจัดการกับข้อมูล ต้องอาศัยภาษาคอมพิวเตอร์ที่เรียกว่า ภาษาเอสคิวแอล (SQL : Structured Query Language) โดยมายเอสคิวแอลเป็นระบบฐานข้อมูลที่มีรูปแบบเรียบง่าย และมีการเตรียมเครื่องมือต่างๆที่จำเป็นสำหรับสร้างและจัดการฐานข้อมูลบนเว็บไว้ให้ ทำให้ Web Server สามารถค้นหาและส่งข้อมูลที่ Browser ร้องขอกลับไปได้อย่างมีประสิทธิภาพ

จากความนิยม สังเกตได้ในหลายๆคู่มือคอมพิวเตอร์ต่างๆจะมีการสอนการใช้งานพีเอชพี และ มายเอสคิวแอลควบคู่กัน นอกจากนี้ หลายๆภาษาที่สามารถทำงานร่วมกับมายเอสคิวแอล เช่น ภาษาซี, ภาษาซีพลัสพลัส, ภาษาปาสคาล, ภาษาซีชาร์ป, ภาษาจาวา, ภาษาเพิร์ล, ภาษาไพทอน, ภาษารูบี และภาษาอื่นๆ

2.4 สัญกรณ์วัตถุจาวาสคริปต์

สัญกรณ์วัตถุจาวาสคริปต์ (JSON : JavaScript object notation) [4] หรือเรียกโดยย่อว่า "เจชัน" เป็นฟอร์แมต (Format) สำหรับแลกเปลี่ยนข้อมูลคอมพิวเตอร์ ฟอร์แมต JSON นั้นอยู่ในรูปข้อความธรรมดา (plain text) ที่ทั้งมนุษย์และโปรแกรมคอมพิวเตอร์สามารถอ่านเข้าใจได้

มาตรฐานของฟอร์แมตเจชัน คือ RFC 4627 มีประเภทของสื่ออินเทอร์เน็ต (Internet media type) เป็นโปรแกรมประยุกต์ (Application) และมีนามสกุลของไฟล์เป็นคอตเจชัน (.json)

ในปัจจุบัน JSON นิยมใช้ในเว็บแอปพลิเคชัน โดยเฉพาะเอเจ็กซ์ (AJAX : Asynchronous JavaScript And XML) โดยเจชันเป็นฟอร์แมตทางเลือกในการส่งข้อมูลนอกเหนือไปจากเอกซ์เอ็มแอล (XML : Extensible Markup Language) ซึ่งนิยมใช้กันอยู่แต่เดิมแล้ว โดยสาเหตุที่เจชันเริ่มได้รับความนิยมเป็นเพราะมีความกระชับและสามารถเข้าใจได้ง่ายกว่าเอกซ์เอ็มแอล โดยเจชันเป็นรูปแบบที่ใช้ในการส่งข้อมูลขนาดเล็กระหว่าง เครื่องแม่ข่ายโปรแกรมประยุกต์บนเว็บ (Web Application Server) กับเครื่องลูกข่ายโปรแกรมประยุกต์บนเว็บ (Web Application Client) โดยเจชันจะใช้วากยสัมพันธ์ (syntax) ของภาษาจาวาสคริปต์ แต่ไม่ถูกมองว่าเป็นภาษาโปรแกรม จะถูกมองว่าเป็นภาษาในการแลกเปลี่ยนข้อมูลมากกว่า ในปัจจุบันมีไลบรารีของภาษาโปรแกรมอื่นๆ ที่ใช้ประมวลผลข้อมูลในรูปแบบเจชันมากมาย

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

รูปที่ 2.6 แสดงตัวอย่างโปรแกรมเจสัน

ที่มา: <http://lh.wikipedia.org/wiki/JSON>

จากรูปที่ 2.6 แสดงตัวอย่างของโปรแกรมเจสันที่มีฟอร์แมตในรูปแบบเฉพาะ สามารถอ่านเข้าใจได้ทั้งคอมพิวเตอร์และมนุษย์ โดยจุดประสงค์ในการนำเจสันมาใช้งานจะคล้ายกับการที่ใช้เอกซ์เอ็มแอลในการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ที่พัฒนามาจากโปรแกรมประยุกต์อื่นๆ ลองเปรียบเทียบโครงสร้างของเอกซ์เอ็มแอลกับเจสันที่มีข้อมูลแบบเดียวกันจะได้ดังรูปที่ 2.7

XML	JSON
<pre><Employee> <ID>001</ID> <Name>Banpote</Name> <Email>banpote_tt@hotmail.com</Email> </Employee></pre>	<pre>{ "Employee": { "ID": "001", "Name": "Banpote", "Email": "banpote_tt@hotmail.com" } }</pre>

รูปที่ 2.7 แสดงการเปรียบเทียบรูปแบบของ เอกซ์เอ็มแอล กับ เจสัน

ที่มา: <http://lh.wikipedia.org/wiki/JSON>

รูปแบบโครงสร้างของเจสันมีรูปแบบเบื้องต้นดังนี้

```

"Employee":
{
    "ID": "001",
    "Name": "Baupote",
    "Email": "baupote_tt@hotmail.com"
}
}

```

รูปที่ 2.8 แสดงรูปแบบโครงสร้างเจสัน(1)

ที่มา: <http://th.wikipedia.org/wiki/JSON>

จากรูปที่ 2.8 แสดงลักษณะโครงสร้างของเจสัน จะประกอบไปด้วย Name/Value คู่กันซึ่ง Name และ Value จะต้องอยู่ในเครื่องหมาย " (double quote) และ Name กับ Value จะถูกคั่นด้วยเครื่องหมาย : (colon) ในชุดข้อมูลแต่ละคู่ของ Name/Value ก็คือ Property 1 Property ในแต่ละ Property หนึ่งๆ ก็จะถูกแบ่งเป็นแต่ละ Property ด้วยการคั่นด้วยเครื่องหมาย , (comma) และสุดท้ายก็ใช้เครื่องหมาย { (left brace) และเครื่องหมาย } (right brace) ครอบ Property ทั้งหมดอีกทีให้เป็น 1 Object อย่างเช่นตัวอย่างดังรูปที่ 2.9

```
{
  "ID" : "001",
  "Name" : "Banpote",
  "Email" : "banpote_tt@hotmail.com"
}
```

รูปที่ 2.9 แสดงรูปแบบโครงสร้างเจสัน(2)

ที่มา: <http://th.wikipedia.org/wiki/JSON>

จากรูปที่ 2.9 โครงสร้างเจสันนี้มีอยู่ 3 Property คือ ID, Name, Email ที่อยู่ทางฝั่งซ้ายของเครื่องหมาย colon และทางฝั่งขวาของเครื่องหมาย colon ก็คือค่าของแต่ละ Property ใน JSON. ตัวเจสันเองเป็นส่วนหนึ่งของภาษา JavaScript ที่อยู่ในลักษณะของ Plain Text ทำให้เวลาที่ส่งข้อมูลนี้ไปยัง JavaScript มันสามารถใช้ Function Eval ในการแปลงเจสัน กลับออกมาเป็น Object ได้

ประโยชน์ของเจสัน คือ การที่สามารถแปลง Object ที่ฝั่งของ เครื่องแม่ข่ายโปรแกรม ประยุกต์บนเว็บไปเป็นเจสัน แล้วส่งเจสันนั้นลงไปยัง เครื่องลูกข่ายโปรแกรมประยุกต์บนเว็บในรูปแบบที่เป็น Object แทนการส่งข้อมูลในรูปแบบสตริง ที่คั่นด้วยตัวแบ่งแยก)Separator(ต่างๆ แล้วมาทำการ Split ออกในภายหลัง

2.5 จีพีเอส

2.5.1 จีพีเอส คืออะไร

จีพีเอส (GPS : Global Positioning System) [5] คือ ระบบระบุตำแหน่งบนพื้นโลก ซึ่งระบบจีพีเอส ประกอบไปด้วย 3 ส่วนหลัก คือ

1. ส่วนอวกาศ (Space Segment) ประกอบด้วยเครือข่ายดาวเทียม 3 ค่า คือ ค่าของอเมริกา, รัสเซีย, และยุโรป ในส่วนของอเมริกา ชื่อ NAVSTAR (Navigation Satellite Timing and Ranging GPS) มีดาวเทียม 28 ดวง ใช้งานจริง 24 ดวง อีก 4 ดวงเป็นค่าสำรอง บริหารงานโดย Department of Defences ของค่ายยุโรปชื่อ Galileo มี 27 ดวง บริหารงานโดย ESA หรือ European Satellite Agency และค่ายรัสเซียชื่อ GLONASS หรือ Global Navigation Satellite บริหารโดย Russia VKS (Russia Military Space Force) ในขณะนี้ภาคประชาชนทั่วโลกสามารถใช้ข้อมูลจากดาวเทียมของทางอเมริกา (NAVSTAR) ได้ฟรี เนื่องจากนโยบายสิทธิการเข้าถึงข้อมูลและข่าวสาร

สำหรับประชาชนของรัฐบาลสหรัฐ จึงเปิดให้ประชาชนทั่วไปสามารถใช้ข้อมูลดังกล่าวในระดับความแม่นยำที่ไม่เป็นภัยต่อความมั่นคงของรัฐ กล่าวคือมีความแม่นยำในระดับบวก / ลบ 10 เมตร

2. ส่วนควบคุม (Ground Segment) ประกอบด้วยสถานีภาคพื้นดิน สถานีใหญ่อยู่ที่ Falcon Air Force Bas ประเทศสหรัฐอเมริกา และศูนย์ควบคุมย่อยอีก 5 จุด กระจายไปยังภูมิภาคต่าง ๆ ทั่วโลก

3. ส่วนผู้ใช้งาน (Users Segment) ผู้ใช้งานต้องมีเครื่องรับสัญญาณที่สามารถรับคลื่นและแปรรหัสจากดาวเทียมเพื่อนำมาประมวลผลให้เหมาะสมกับการใช้งานในรูปแบบต่าง ๆ

2.5.2 การทำงานของจีพีเอส

จีพีเอสทำงานโดยการรับสัญญาณจากดาวเทียมแต่ละดวง โดยสัญญาณดาวเทียมนี้ประกอบไปด้วยข้อมูลที่ระบุตำแหน่งและเวลาขณะส่งสัญญาณไว้ โดยตัวเครื่องรับสัญญาณจีพีเอส จะต้องประมวลผลความแตกต่างของเวลาในการรับสัญญาณเทียบกับเวลาจริง ณ เวลาปัจจุบัน เพื่อแปรเป็นระยะทางระหว่างเครื่องรับสัญญาณกับดาวเทียมแต่ละดวง ซึ่งได้ระบุตำแหน่งของมันมา กับสัญญาณดังกล่าว เพื่อให้เกิดความแม่นยำในการค้นหาตำแหน่งด้วยดาวเทียม ต้องมีดาวเทียมอย่างน้อย 4 ดวง เพื่อบอกตำแหน่งบนผิวโลก ซึ่งระยะห่างจากดาวเทียมทั้ง 3 กับภาคพื้นดิน จะสามารถระบุตำแหน่งบนผิวโลกได้หากพื้นโลกอยู่ในแนวระนาบ แต่ในความเป็นจริงพื้นโลกมีความโค้งเนื่องจากสัณฐานของโลกมีลักษณะกลม ดังนั้นดาวเทียมดวงที่ 4 จะทำให้สามารถคำนวณเรื่องความสูงเพื่อทำให้ได้ตำแหน่งที่ถูกต้องมากขึ้น

2.5.3 หน้าที่สำคัญของดาวเทียมจีพีเอส

1. รับข้อมูลวงโคจรที่ถูกต้องของดาวเทียม (Ephemeris Data) ที่ส่งมาจากสถานีควบคุมดาวเทียมหลัก (Master Control Station) เพื่อส่งกระจายสัญญาณข้อมูลนี้ ลงไปยังพื้นโลก สำหรับเครื่องรับข้อมูลจีพีเอส (GPS Receiver) ใช้ในการคำนวณ ระยะห่าง (Range) ระหว่างดาวเทียมดวงนั้น กับ GPS Receiver และตำแหน่งของดาวเทียมบนท้องฟ้า เพื่อใช้คำนวณหา ตำแหน่งพิกัด ของตัวเครื่อง GPS Receiver เอง

2. ส่งรหัส (Code) และข้อมูล Carrier Phase ไปกับคลื่นวิทยุ ลงไปยังพื้นโลก สำหรับ GPS Receiver ใช้ในการคำนวณ ระยะห่าง (Range) ระหว่างดาวเทียมดวงนั้นกับ GPS Receiver

3. ส่งข้อมูลตำแหน่งโดยประมาณของดาวเทียมทั้งหมด (Almanac Information) และข้อมูลสุขภาพ ของดาวเทียม ลงไปยังพื้นโลก สำหรับ GPS Receiver ใช้ในการกำหนดดาวเทียม ที่จะสามารถรับสัญญาณได้

2.6 แผนที่กูเกิล

Google Maps คือ บริการของกูเกิลที่ให้บริการเทคโนโลยีด้านแผนที่ประสิทธิภาพสูง ใช้งานง่าย และข้อมูลของธุรกิจในห้องดิน ได้แก่ ที่ตั้งของธุรกิจ รายละเอียดการติดต่อ และเส้นทางการขับขี่ ด้วย Google Maps โดยคุณสมบัติที่เป็นเอกลักษณ์เฉพาะมีดังต่อไปนี้

- ผลการสืบค้นที่ผสมรวมข้อมูลของธุรกิจ ค้นพบข้อมูลที่ตั้งและรายละเอียดการติดต่อของธุรกิจทั้งหมดที่แสดงผลรวมไว้ ในที่เดียวบนแผนที่
- แผนที่ที่สามารถลากได้ คลิกและลากแผนที่เพื่อดูส่วนที่อยู่ติดกันได้ทันที
- ภาพถ่ายจากดาวเทียม ดูภาพถ่ายจากดาวเทียมหรือภาพถ่ายจากดาวเทียมพร้อมกับข้อมูลแผนที่วางซ้อนทับกันของที่ตั้งที่ต้องการ ซึ่งสามารถขยายและเลื่อนดูในมุมมองกว้างได้
- มุมมองระดับถนน ดูและค้นหาเส้นทางภายในภาพถ่ายระดับถนน
- เส้นทางโดยละเอียด ป้อนที่อยู่และให้ Google Maps วาดแผนผังที่ตั้งและหรือเส้นทางการขับขี่ วางแผนการเดินทางโดยเพิ่มปลายทางหลายแห่งลงในเส้นทางของผู้ใช้งาน

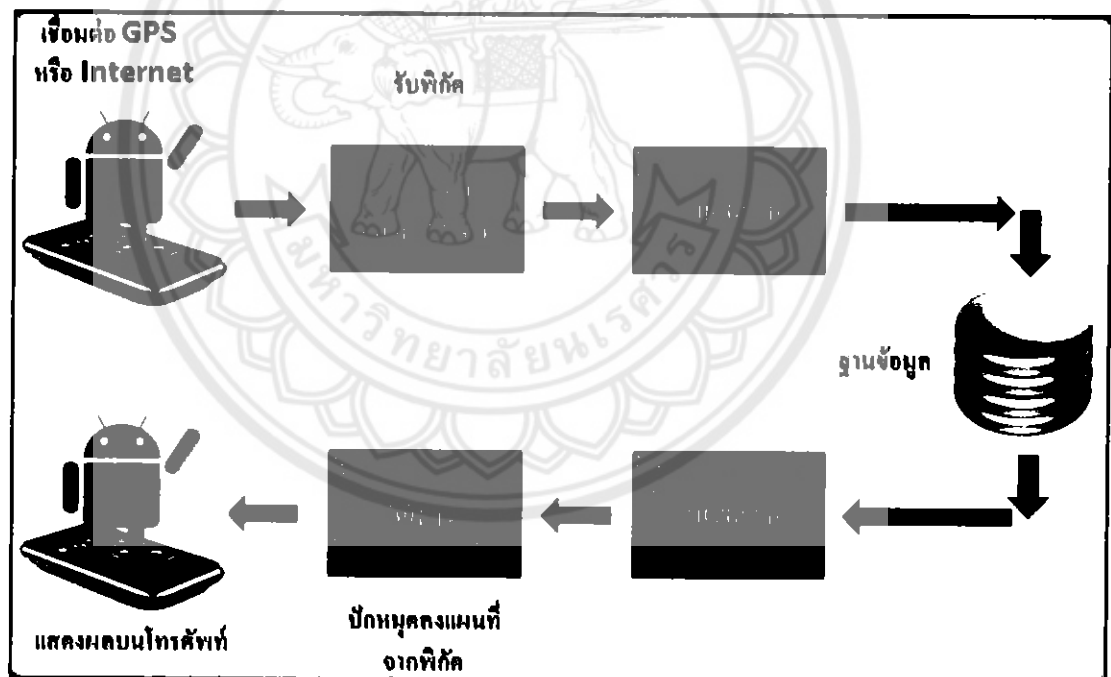
บทที่ 3

ขั้นตอนและวิธีการดำเนินงาน

จากการศึกษาค้นคว้าทฤษฎีบทที่ผ่านมาเบื้องต้น จึงสามารถทำให้เข้าใจถึงหลักการและวิธีการดำเนินงาน ในบทนี้จะนำความรู้ที่ได้ศึกษามาประยุกต์ใช้

3.1 แผนผังการออกแบบการทำงานของโปรแกรม

ในการทำงานของโปรแกรมนั้น จะคำนึงถึงการรับส่งข้อมูลระหว่างอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์กับฐานข้อมูลเป็นหลัก โดยจะแสดงให้เห็นถึงการนำข้อมูลในฐานข้อมูลบนเว็บไซต์มาใช้งานร่วมกัน โดยแสดงผลผ่านทางแผนที่ที่ถูกระบุบนอุปกรณ์นั้นๆ โดยมีการออกแบบหลักการการทำงานของโปรแกรมโดยรวม ดังรูปที่ 3.1

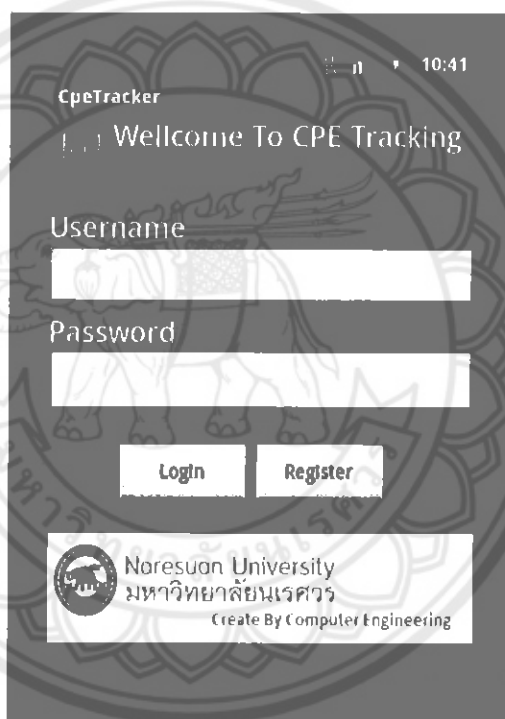


รูปที่ 3.1 การออกแบบการทำงานของโปรแกรม

3.2 การออกแบบหน้าต่าง GUI สำหรับการใช้งาน

GUI (Graphic User Interface) คือ ส่วนที่เครื่องใช้ติดต่อกับผู้ใช้โดยมีรูปแบบเป็นกราฟฟิก โดยการออกแบบในส่วนนี้ได้ทำการออกแบบและสร้างหน้าต่าง GUI สำหรับให้ผู้ใช้งานโปรแกรมใช้งานได้ง่ายขึ้น มีรูปลักษณ์ที่สวยงามมากขึ้น โดยได้สร้างหน้าต่าง GUI ไว้ 4 หน้า คือ หน้าลงชื่อเข้าใช้งาน (Login Page), หน้าลงทะเบียน (Register Page), หน้าแสดงผู้ใช้งาน (Online Page) และ หน้าแสดงแผนที่ (Map Page)

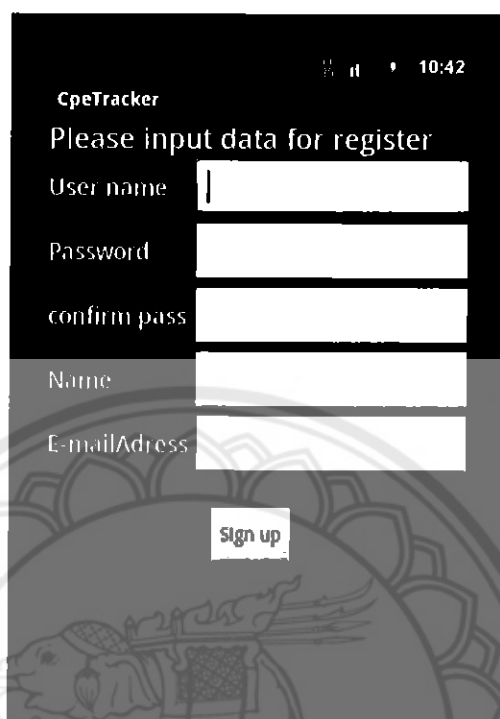
3.2.1 หน้าลงชื่อเข้าใช้งาน (Login Page)



รูปที่ 3.2 แสดงหน้าลงชื่อเข้าใช้งาน (Login Page)

จากรูปที่ 3.2 เป็นหน้าลงชื่อเข้าใช้งานซึ่งมีการออกแบบมาให้สามารถใส่ข้อมูลชื่อผู้ใช้งาน และรหัสผ่านประจำตัวได้ทันทีสำหรับผู้ใช้งานที่ได้ลงทะเบียนไว้แล้ว หากผู้ใช้งานเข้าใช้งานเป็นครั้งแรก สามารถกดที่ปุ่มลงทะเบียน (Register) เพื่อทำการลงทะเบียนก่อนที่จะใช้งานโปรแกรม

3.2.2 หน้าลงทะเบียน (Register Page)

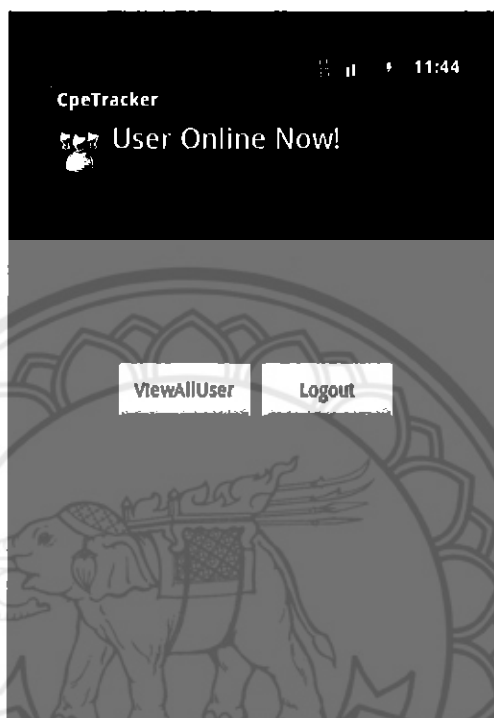


รูปที่ 3.3 แสดงหน้าลงทะเบียน (Register Page)

จากรูปที่ 3.3 เป็นหน้าลงทะเบียนสำหรับผู้ใช้งานใหม่ที่ต้องการใช้โปรแกรม สามารถกรอกข้อมูลเบื้องต้นทั้งหมด 5 หัวข้อ คือ ชื่อเข้าใช้งาน (User Name), รหัสผ่าน (Password), ยืนยันรหัสผ่าน (Confirm Password), ชื่อผู้ใช้งาน (Name), อีเมล (E-mail) โดยต้องกรอกข้อมูลให้ครบทุกหัวข้อ เพื่อใช้เป็นรายละเอียดของผู้ใช้งานเอง ที่จะนำส่วนต่างๆ ไปใช้งานและแสดงผล

3.2.3 หน้าแสดงผู้ใช้งาน (Online Page)

เมื่อผู้ใช้งานได้ทำการลงทะเบียนเสร็จเรียบร้อยแล้ว ทำการเข้าสู่ระบบในหน้าเข้าสู่ระบบก็จะเข้ามาพบกับหน้าแสดงผู้ใช้งานที่เปิดใช้งานอยู่ในขณะนั้นทั้งหมด



รูปที่ 3.4 แสดงหน้าแสดงผู้ใช้งานปัจจุบัน (Online Page)

จากรูปที่ 3.4 เป็นหน้าแสดงชื่อผู้ใช้งานทั้งหมดที่ได้เปิดใช้งานโปรแกรมอยู่ในขณะนั้น โดยชื่อผู้ใช้งานทั้งหมดจะนำไปแสดงในหน้าแผนที่ด้วย ฉะนั้น ผู้ใช้งานที่ไม่ได้ทำการเข้าสู่ระบบไว้ก็จะไม่แสดงให้เห็นในหน้านี้และหน้าแผนที่ด้วย ผู้ใช้งานสามารถกดปุ่ม ViewAllUser เพื่อเรียกดูหน้าแผนที่ได้ และสามารถกดปุ่ม Logout เพื่อทำการลงชื่อออกจากโปรแกรม

3.2.4 หน้าแสดงแผนที่ (Map Page)

หลังจากกดปุ่ม ViewAllUser ในหน้า Online Page แล้ว ก็จะเข้าสู่หน้าแผนที่ดังนี้

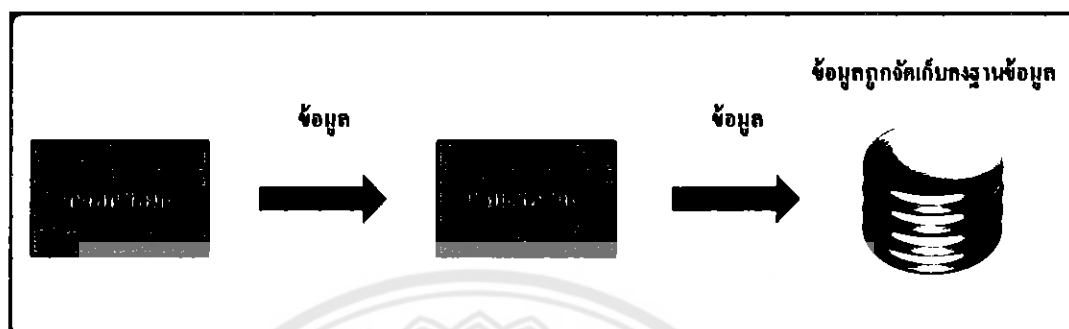


รูปที่ 3.5 แสดงหน้าแสดงแผนที่ (Map Page)

จากรูปที่ 3.5 หน้าแสดงแผนที่จะแสดงภาพของแผนที่กูเกิล โดยจุดมุ่งหมายจะทำให้แสดงตำแหน่งของผู้ใช้งานที่เปิดใช้งานทุกคน แสดงอยู่บนแผนที่นี้ เพื่อให้ทราบถึงตำแหน่งที่อยู่ที่สามารถรู้ได้โดยทันที และให้สามารถเลื่อนหาดำแหน่งของผู้ใช้งานคนอื่นได้กว้างทั่วโลก มีความละเอียดในการค้นหามากที่สุด

3.3 ระบบลงทะเบียน (Register)

ในระบบลงทะเบียนนี้มีความต้องการองค์ประกอบอื่นๆเข้ามาช่วยในการทำงาน นั่นคือระบบของเว็บเซิร์ฟเวอร์ และ ระบบฐานข้อมูล ดังนี้

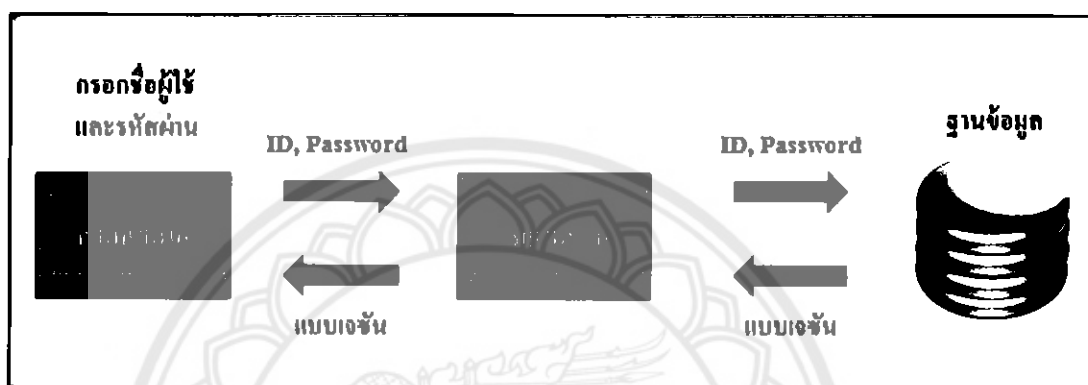


รูปที่ 3.6 แสดงทำงานของระบบลงทะเบียน (Register)

จากรูปที่ 3.6 เป็นหลักการทำงานของระบบการลงทะเบียน คือ การรับข้อมูลที่ผู้ใช้งานกรอกข้อมูลในหน้าลงทะเบียน (Register Page) ไว้ แอนดรอยด์จะทำการส่งคำสั่งไปยังเว็บเซิร์ฟเวอร์ โดยเว็บเซิร์ฟเวอร์ทำหน้าที่เป็นส่วนกลางในการติดต่อกันระหว่างผู้ใช้งานกับระบบฐานข้อมูล โดยสามารถส่งคำสั่งในรูปแบบ String เพื่อทำการจัดเก็บข้อมูลของผู้ใช้งาน เพื่อไว้สำหรับเรียกใช้งานในส่วนถัดไปได้

3.4 ระบบเข้าใช้งาน (Log in)

ในส่วนระบบเข้าใช้งานของผู้ใช้งานจะยังคงใช้องค์ประกอบเหมือนระบบลงทะเบียนที่ต้องอาศัยการรับส่งข้อมูลกับเว็บเซิร์ฟเวอร์ และ ฐานข้อมูล แต่การลงชื่อเข้าใช้นั้นเป็นเหมือนขั้นตอนถัดจากขั้นตอนการลงทะเบียน ที่ต้องการข้อมูลที่เก็บไว้ในส่วนแรก นำมาใช้งาน โดยมีการทำงานดังนี้

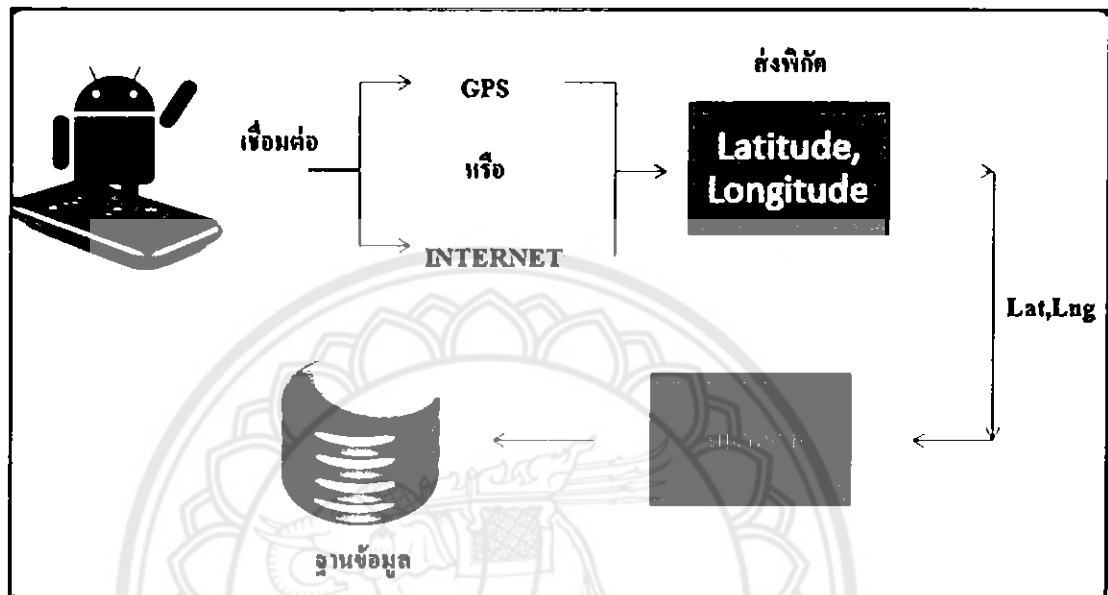


รูปที่ 3.7 แสดงการทำงานของระบบเข้าใช้งาน (Login)

จากรูปที่ 3.7 เป็นขั้นตอนการดำเนินงานในระบบเข้าใช้งานนี้จะแบ่งออกเป็น 2 ส่วน คือ ส่วนของการติดต่อกับเว็บเซิร์ฟเวอร์เพื่อเรียกข้อมูลผู้ใช้งานจากฐานข้อมูลและ ส่วนของการนำข้อมูลผู้ใช้งานมาตรวจสอบความถูกต้อง โดยในส่วนแรก การทำงานจะเริ่มต้นจากระบบ Android จะทำการส่งคำสั่งในรูปแบบ String ให้เว็บเซิร์ฟเวอร์ โดยให้เว็บเซิร์ฟเวอร์ทำการ Query ข้อมูลที่บันทึกในฐานข้อมูลส่งกลับมายัง Android ในรูปแบบเจสัน จากนั้น Android จะทำการตรวจสอบชื่อเข้าใช้งานและรหัสผ่านที่ผู้ใช้งานกรอกกับข้อมูลที่ Query มาจากฐานข้อมูลว่ามีความถูกต้องหรือไม่ ถ้าพบว่า มีความถูกต้อง Android จะแสดงหน้าต่างแสดงผู้ใช้งานต่อไป แต่ถ้าตรวจสอบแล้วข้อมูลไม่ตรงกัน Android จะแสดงข้อความบ่งบอกว่ากรอกข้อมูลผิดนั่นเองและจะกลับมาหน้าต่างลงทะเบียนใหม่ เพื่อกรอกข้อมูลใหม่หรือทำการลงทะเบียน

3.5 การรับพิกัด

พิกัดเป็นสิ่งที่เราต้องการ เพื่อนำมาเป็นตัวระบุถึงสถานที่อยู่ที่เป็นสำหรับโปรแกรม ที่ต้องการระบุตำแหน่งให้ผู้ใช้ทราบ โดยมีขั้นตอนการทำงานดังนี้



รูปที่ 3.8 แสดงการทำงานของ การรับพิกัด

จากรูปที่ 3.8 แสดงขั้นตอนการทำงานในส่วนการรับพิกัด สามารถรับได้จาก โทรศัพท์เคลื่อนที่ที่ใช้ระบบปฏิบัติการ Android ซึ่งสามารถแสดงพิกัดได้จากสองทาง คือ GPS และ Internet โดยจะให้ความสำคัญกับ GPS เป็นหลัก นั่นคือ ถ้าผู้ใช้โทรศัพท์เปิดการใช้งาน GPS อยู่ โปรแกรมจะทำการเรียกพิกัดจาก GPS แต่ถ้าผู้ใช้งานไม่ได้เปิดการใช้งาน GPS โปรแกรมจะทำการเรียกพิกัดจากการเชื่อมต่อ Internet แทน โดยพิกัดที่เรียกมาจะเป็นแบบ Latitude, Longitude โดยจะถูกส่งไปเก็บไว้ในฐานข้อมูลผ่านทางเว็บเซิร์ฟเวอร์ พิกัดที่ถูกเก็บไว้ในฐานข้อมูลเพื่อการเรียกใช้งานข้อมูลต่อไป

3.6 การขอเอพีไอคีย์เพื่อเรียกใช้งานแผนที่ของกูเกิล

ในปัจจุบัน การเขียน โปรแกรมส่วนใหญ่ นิยม นำแผนที่เข้ามา มีส่วนร่วมกับตัวโปรแกรม เพราะเป็นองค์ประกอบที่น่าสนใจที่จะนำมาประยุกต์ใช้ และสามารถทำได้ไม่ยาก ซึ่งแผนที่ของกูเกิล (Google Map) สามารถเปิดใช้งาน ได้โดยไม่เสียค่าใช้จ่าย และมีประสิทธิภาพค่อนข้างสูง ทำให้ เป็นที่น่าสนใจที่จะนำมาใช้งานร่วมกับโปรแกรมต่างๆ รวมถึงโครงการนี้ ที่นำแผนที่ของกูเกิลมาใช้งานเช่นกัน

การขอใช้งานแผนที่ของกูเกิลนั้น มีขั้นตอนการขอใช้งานค่อนข้างไม่ซับซ้อน ในที่นี้จะใช้โปรแกรม Eclipse ที่ผู้จัดทำใช้เป็นโปรแกรมสำหรับพัฒนานำมาขอเอพีไอคีย์ของกูเกิล ความหมายโดยรวมของเอพีไอคีย์ คือ เปรียบเสมือนกุญแจสำหรับใช้งานแผนที่ของกูเกิล ผู้ที่จะใช้งานแผนที่ของกูเกิลได้ก็จำเป็นต้องขอเอพีไอคีย์จากกูเกิลเพื่อนำไปใช้งาน โดยการขอเอพีไอคีย์มีขั้นตอนดังนี้

1. หาที่อยู่ของไฟล์ชื่อว่า debug.keystore ที่มาจากโปรแกรม Eclipse ซึ่งโดยปกติจะอยู่ใน "C:\Documents and Settings\<username>\Local Settings\Application Data\Android" แต่เพื่อให้ง่ายต่อการนำไปใช้ ทางผู้จัดทำได้ย้ายมาไฟล์ไว้ที่ "C:\android\" ในคอมพิวเตอร์ที่ใช้งาน
2. ขั้นตอนนี้จะทำการรัน Keytool.exe ซึ่งไฟล์นี้ติดมาพร้อมกับชุดพัฒนาจาวา (JDK : Java Development Kit) ที่ติดตั้งไว้ในส่วนแรกของการเตรียมโปรแกรม ตามที่อยู่ไฟล์ที่ติดตั้งไว้ ในที่นี้คือ "C:\Program Files\Java\jdk1.7.0_02\bin" โดยมีการรันผ่านทาง Command Prompt มีขั้นตอนดังนี้



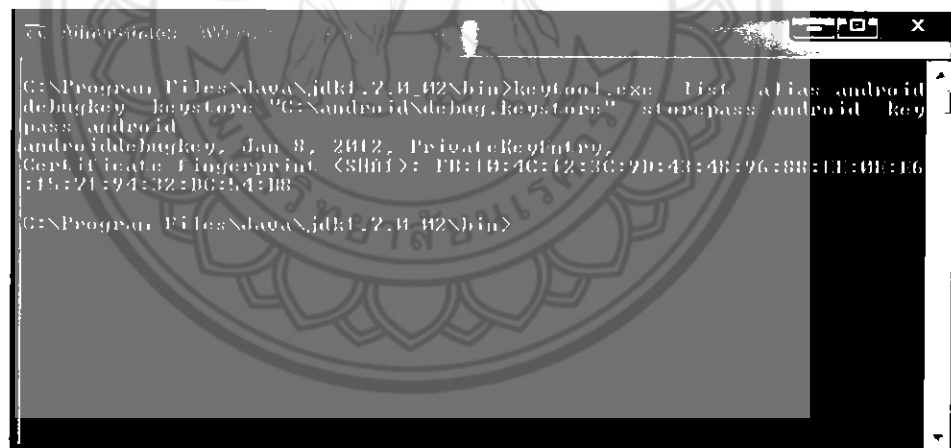
รูปที่ 3.9 แสดงการใส่ที่อยู่ของไฟล์ชื่อ Keytool.exe บน Command Prompt

จากรูปที่ 3.9 หลังจากเปิดหน้าต่าง Command Prompt แล้ว ให้เปลี่ยนที่อยู่ (location) ไปยังที่อยู่ของไฟล์ Keytool.exe ดังนี้



รูปที่ 3.10 แสดงการใส่คำสั่งการรันไฟล์ debug.keystore บน Command Prompt

จากรูปที่ 3.10 แสดงการใส่คำสั่งเพื่อใช้รันไฟล์ debug.keystore คือ “keytool.exe -list -alias androiddebugkey -keystore "C:\android\debug.keystore" -storepass android -keypass android” สังเกตการใส่ที่อยู่ไฟล์ debug.keystore ควรใส่ให้ถูกต้อง มิฉะนั้นจะไม่สามารถรันได้ หลังจากนั้นกด Enter หนึ่งครั้ง



รูปที่ 3.11 แสดงผลการรันไฟล์ debug.keystore บน Command Prompt

จากรูปที่ 3.11 แสดงการรันไฟล์ debug.keystore จะเห็นว่าจะได้ข้อมูลที่เรียกว่า Certificate fingerprint เป็น FB:10:4C:12:3C:9D::43:48:96:88EE:0E:E6 :15:71:94:32:DC::54B8 สำหรับนำไปใช้งานต่อไป

3. ขั้นตอนนี้จะเป็นการขอเอพีไอคีย์จากหน้าเว็บไซต์ โดยเข้าไปที่

<http://code.google.com/intl/th/android/maps-api-signup.html> ในเว็บไซต์นี้จะมีรายละเอียด

ข้อตกลงที่ควรทำความเข้าใจ หลังจากทำความเข้าใจแล้วให้ใส่เครื่องหมายถูกเพื่อรับทราบข้อตกลง

จากนั้นนำ Certificate fingerprint ที่ได้จากขั้นตอนที่แล้วไปใส่ในช่อง My certificate's MD5 fingerprint จากนั้นกด Generate API Key ดังนี้

principal place of business is at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States ("Google"). This legal agreement is referred to as the "Terms."

1.2. Unless otherwise agreed in writing with Google, the Terms will include the following: 1) the terms and conditions set forth in this document (the "Maps APIs Terms"); 2) the Legal Notices (http://www.google.com/intl/en-us/help/legalnotices_maps.html); and 3) the Privacy Policy (<http://www.google.com/privacy.html>). Before you use the Maps APIs, you should read each of the documents comprising the Terms, and print or save a local copy for your records.

1.3. If you use the Maps APIs in conjunction with any other Google products or services, including any other Google API, (collectively, the "Services"), your agreement with Google will also include the terms applicable to those Services. All of these are referred to as the "Additional Terms." If Additional Terms apply,

☒ I have read and agree with the terms and conditions ([printable version](#))

My certificate's MD5 fingerprint: 23C:9D:43:48:96:88:EE:0E:E6:15:71:94:32:DC:54:B8

รูปที่ 3.12 แสดงการขอ API Key

จากรูปที่ 3.12 แสดงหน้าของข้อตกลงและการนำ Certificate fingerprint ประจำเครื่องมาขอรับเอพีไอคีย์ของกูเกิล ผ่านทาง <http://code.google.com/intl/th/android/maps-api-signup.html> จากนั้นกดที่ Generate API Key ก็จะได้รับเอพีไอคีย์สำหรับนำไปใช้เรียบร้อยแล้ว

บทที่ 4

ผลการดำเนินโครงการ

หลังจากที่ได้ศึกษาองค์ประกอบในส่วนต่างๆและผ่านขั้นตอนการออกแบบในแต่ละส่วนที่จะนำมาใช้ในโปรแกรมไปแล้ว ในบทนี้จะเป็นการแสดงผลการทดลองในรูปแบบต่างๆ ในหลายๆกรณีและหลายลักษณะที่แตกต่างกัน เพื่อให้เห็นถึงคุณสมบัติของโปรแกรมว่ามีคุณสมบัติมากน้อยเพียงใด ยังมีจุดบกพร่องอะไรที่สามารถพัฒนาต่อในภายหน้าได้ โดยจะยกผลการทดลองในหลายๆรูปแบบ ดังนี้

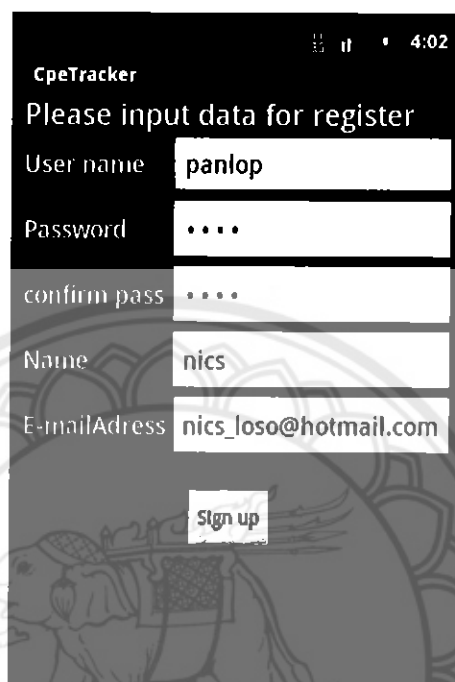
4.1 ผลการทดลองเข้าใช้งานโปรแกรม

ในการใช้งานโปรแกรมนี้นี้ จะกล่าวถึงส่วนของการลงทะเบียนและการลงชื่อเข้าใช้งานว่าผลที่ออกมาเป็นอย่างไร และตรงกับวัตถุประสงค์มากน้อยเพียงไร และมีการเปลี่ยนแปลงอย่างไรในระบบฐานข้อมูล โดยมีผลการทดลองดังนี้



4.1.1 ผลการทดลองส่วนของการลงทะเบียน

ทำการกรอกข้อมูลผู้ใช้งานในหน้าลงทะเบียนให้ครบถ้วน ดังภาพที่ 4.1



รูปที่ 4.1 การกรอกข้อมูลในการลงทะเบียน

หลังจากกรอกข้อมูลและกดที่ "Sign up" แล้ว ในระบบฐานข้อมูลจะทำการเก็บค่าผู้ใช้งานไว้ในฐานข้อมูลดังภาพที่ 4.2

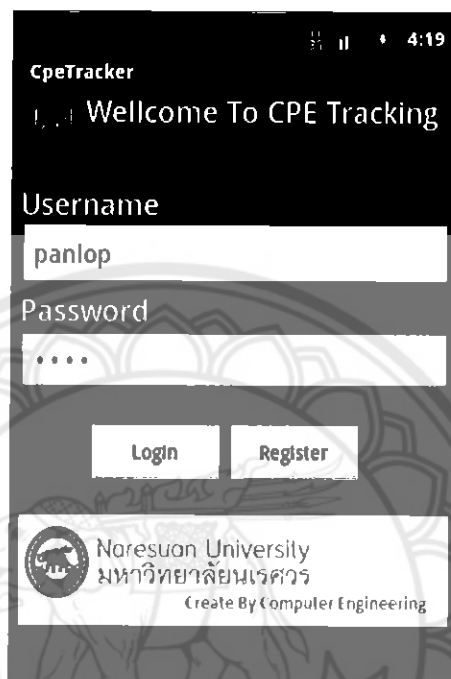
	UserID	Username	Password	Name	Email	Status	SID	Active	Lat	Lng
	24	panlop	1150	nics	nics_loso@hotmail.com	USER	601457f34f0c48f793cfa7b754c7107f	No	16.7531272	100.19846893

รูปที่ 4.2 การบันทึกข้อมูลผู้ใช้งานในฐานข้อมูล

จากขั้นตอนดังกล่าว ในกรณีผู้ใช้อย่างอื่นมาทำการลงทะเบียน โปรแกรมก็จะทำการจัดเก็บข้อมูลลักษณะเดียวกัน เพื่อใช้สำหรับการตรวจสอบและเรียกใช้งานในขั้นตอนการเข้าสู่ระบบต่อไป

4.1.2 ผลการทดสอบส่วนของการลงชื่อเข้าใช้งาน

ส่วนนี้จะเป็นการลงชื่อเข้าใช้งาน โดยจะใช้ข้อมูลจากหัวข้อที่ 4.2.1 นำมาทดลอง โดยในขั้นแรกจะเป็นการลงชื่อเข้าใช้งาน ดังภาพที่ 4.3



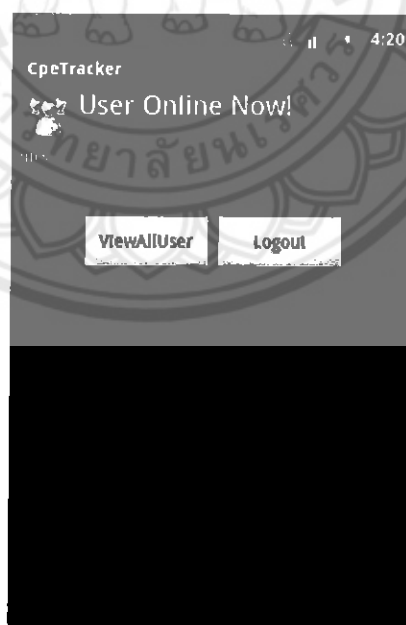
รูปที่ 4.3 การลงชื่อเข้าใช้งาน

หลังจากการกรอกข้อมูลในส่วนการลงชื่อเข้าใช้งานแล้ว โปรแกรมจะทำการตรวจสอบความถูกต้องของข้อมูล โดยถ้าข้อมูลที่ใส่ผิดจากฐานข้อมูล โปรแกรมจะแสดงข้อความ "Don't have Username or Password" ดังภาพที่ 4.4



รูปที่ 4.4 การลงชื่อเข้าใช้งานโดยกรอกข้อมูลผิด

ในกรณีที่กรอกข้อมูลถูก โดยมีข้อมูลตรงกับในฐานข้อมูล โปรแกรมจะทำการเรียกหน้าต่างออนไลน์ขึ้นมา ดังรูปที่ 4.5



รูปที่ 4.5 หน้าต่างแสดงผู้ใช้งานที่ออนไลน์

ในหน้าจอออนไลน์นี้ผู้ใช้ที่แสดงได้จะต้องทำการลงชื่อเข้าใช้งานโปรแกรมอยู่ โดยในฐานข้อมูลจะมีการเปลี่ยนแปลงเกิดขึ้น คือ ผู้ใช้งานที่ลงชื่อเข้าใช้งานได้ถูกต้อง ในคอลัมน์ Active ในฐานข้อมูลจะเปลี่ยนเป็น “Yes” ดังภาพที่ 4.6

	UserID	Username	Password	Name	Email	Status	SID	Active	Lat	Lng
	24	panlop	1150	nics	nics_joso@hotmail.com	USER	601457f34f0c48793cfa7b754c7107f	Yes	16.7531272	100.19846893

รูปที่ 4.6 แสดงความเปลี่ยนแปลงคอลัมน์ Active ในฐานข้อมูล

4.2 ผลการทดลองโปรแกรมในกรณีผู้ใช้งานอยู่ในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร)

การทดลองที่ผู้ใช้งานเปิดใช้งาน โปรแกรมในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร) นั้น รูปแบบของการแสดงผลจะสามารถเห็นได้ในอุปกรณ์ที่ใช้ได้ในทันที โดยไม่ต้องขยายแผนที่เพื่อหาตำแหน่งผู้ใช้อื่น โดยจะมีตำแหน่งของผู้ใช้เองเป็นจุดศูนย์กลาง จะแสดงผลให้เห็นได้ ดังนี้

4.2.1 กรณีผู้ใช้งาน 2 ผู้ใช้งาน

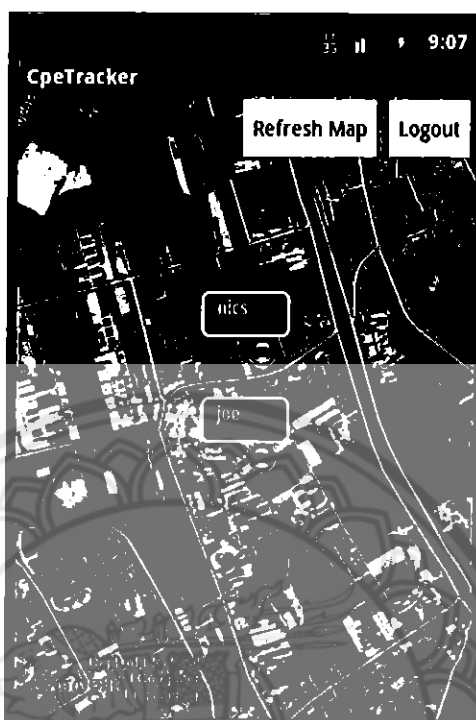
ยกตัวอย่างรายละเอียดในฐานข้อมูล ดังนี้

	UserID	Username	Password	Name	Email	Status	SID	Active	Lat	Lng
	24	panlop	1150	nics	nics_joso@hotmail.com	USER	601457f34f0c48793cfa7b754c7107f	Yes	16.7531272	100.19846893
	25	songpon	1112	joe	za_idmnoon@hotmail.com	USER	d74921fac0f1c4e7a21c3d8e0b3377cd	Yes	0	0

รูปที่ 4.7 แสดงฐานข้อมูลกรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร)

จากรูปที่ 4.7 แสดงถึงรายละเอียดของผู้ใช้งานที่ได้ลงทะเบียนไว้ โดยกรณีนี้เป็นข้อมูลของผู้ใช้งาน 2 ผู้ใช้งาน โดยได้สมมติพิกัดขึ้นมาเพื่อใช้ในการทดลอง

แสดงหน้าแผนที่ผู้ใช้งาน 2 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร)



รูปที่ 4.8 แสดงแผนที่กรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร)

จากรูปที่ 4.8 แสดงแผนที่กรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร) ซึ่งมีระยะห่างไม่ไกลกันมาก ทำให้การแสดงผลบนหน้าแผนที่สามารถแสดงให้เห็นได้ โดยทันที ไม่จำเป็นต้องขยายแผนที่เพื่อค้นหาคำแหน่งผู้ใช้รายอื่น และถ้าต้องการความละเอียดของแผนที่ก็สามารถขยายแผนที่ได้

4.2.2 กรณีผู้ใช้งาน 3 ผู้ใช้งาน

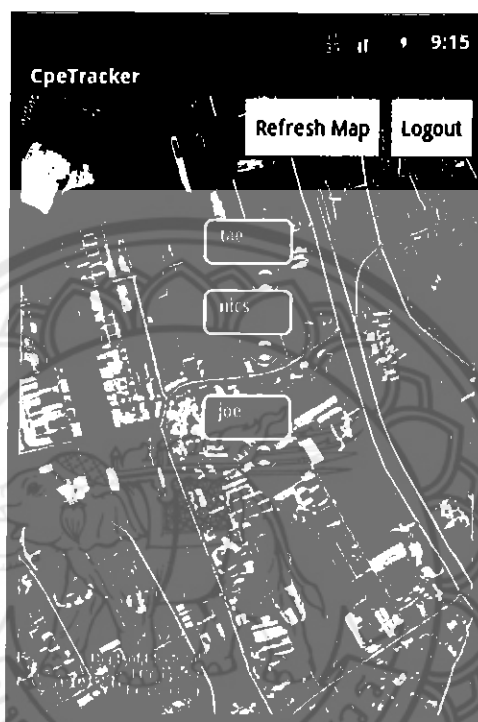
ยกตัวอย่างรายละเอียดในฐานข้อมูล ดังนี้

	UID	Username	Password	Name	Email	Status	SID	Active	Lat	Long
☒	26	aisak	1234	lae	laelaee555@hotmail.com	USER	b0cb433ba02763f589c3cd856ac63c52	Yes	16.7551271	100.19856895
☒	24	panlop	1150	nics	nics_joso@hotmail.com	USER	601457f34f0c48793da7b754c7107f	Yes	16.7531272	100.19856893
☒	25	songpon	1112	joe	za_kimnoon@hotmail.com	USER	d74921fac0f1c4e7a21c3d8edb3377cd	Yes	16.7501274	100.19856894

รูปที่ 4.9 แสดงฐานข้อมูลกรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร)

จากรูปที่ 4.9 แสดงถึงรายละเอียดของผู้ใช้งานที่ได้ลงทะเบียนไว้ โดยกรณีนี้เป็นข้อมูลของผู้ใช้งาน 3 ผู้ใช้งาน โดยได้สมมติพิกัดขึ้นมาเพื่อใช้ในการทดลอง

แสดงหน้าจอแผนที่ผู้ใช้งาน 3 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร)



รูปที่ 4.10 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะใกล้เคียงกัน (น้อยกว่า 1.5x2.5 กิโลเมตร)

จากรูปที่ 4.10 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะใกล้เคียงกัน (3 กิโลเมตร) ซึ่งมีระยะห่างไม่ไกลกันมาก ทำให้การแสดงผลบนหน้าจอที่สามารถแสดงให้เห็นได้โดยทันที ไม่จำเป็นต้องขยายแผนที่เพื่อค้นหาตำแหน่งผู้ใช้งานอื่น และถ้าต้องการความละเอียดของแผนที่ก็สามารถขยายแผนที่ได้

4.3 ผลการทดลองโปรแกรมในกรณีผู้ใช้งานอยู่ในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)

ในรูปแบบที่ผู้ใช้งานเปิดใช้งานโปรแกรมในระยะไกล (มากกว่า 3 กิโลเมตร)นั้น รูปแบบของการแสดงผลนั้น จะไม่สามารถเห็นได้ในอุปกรณ์ที่ใช้ได้โดยทันที เพราะผู้ใช้งานมีระยะทางห่างกันเกินขอบเขตที่ได้กำหนดไว้ในโปรแกรม จึงจำเป็นที่จะต้องขยายแผนที่เพื่อค้นหาตำแหน่งของผู้ใช้งานนั่นเอง โดยแผนที่ของกูเกิลจะแสดงตำแหน่งได้ทั่วโลก จะแสดงผลให้เห็นได้ ดังนี้

4.3.1 กรณีผู้ใช้งาน 2 ผู้ใช้งาน

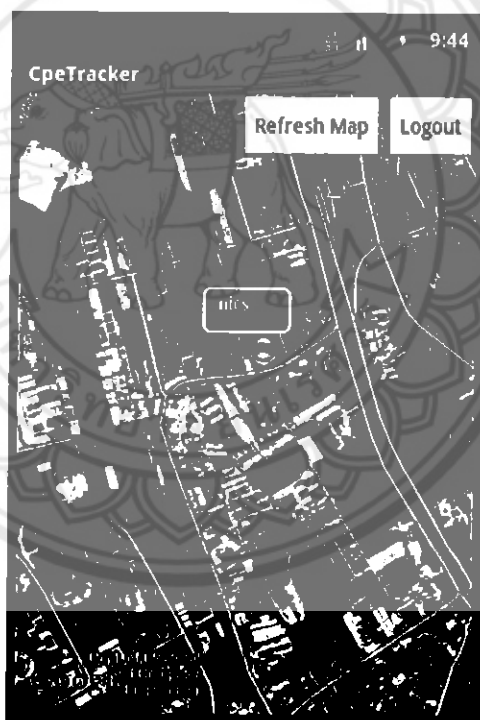
จะยกตัวอย่างรายละเอียดในฐานะข้อมูล ดังนี้

24	songpon	1150	noa	noa_jomchoon@hotmail.com	USER	001457c3400c487030310754271071	Yes	10.7531272	100.19830893
25	songpon	1112	joe	za_idmnoon@hotmail.com	USER	d74921fac01c4e7a21c3d8edb3377cd	Yes	10.7591274	100.15856894

รูปที่ 4.11 แสดงฐานข้อมูลกรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)

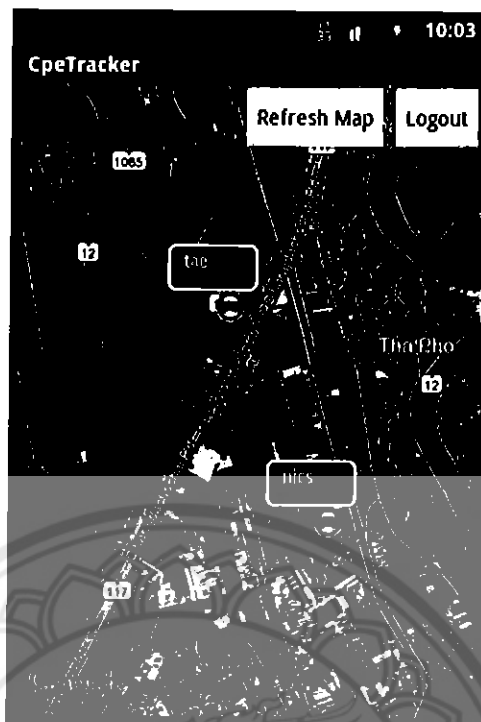
จากรูปที่ 4.11 แสดงถึงรายละเอียดของผู้ใช้งานที่ได้ลงทะเบียนไว้ โดยกรณีนี้เป็นข้อมูลของผู้ใช้งาน 2 ผู้ใช้งาน โดยได้สมบัติพิกัดขึ้นมาเพื่อใช้ในการทดลอง

แสดงหน้าแผนที่ผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)



รูปที่ 4.12 แสดงแผนที่กรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)

จากรูปที่ 4.12 แสดงแผนที่เริ่มต้นกรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร) โดยจะสังเกตได้ว่าจะยังไม่สามารถเห็นผู้ใช้คนอื่นได้ เพราะผู้ใช้คนอื่นอยู่ในระยะห่างมากกว่า 1.5x2.5 กิโลเมตร โดยสามารถขยายแผนที่ออกเพื่อค้นหาผู้ใช้คนอื่นได้ ดังนี้



รูปที่ 4.13 แสดงแผนที่กรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)
เมื่อขยายแผนที่ออก

จากรูปที่ 4.13 แสดงแผนที่เริ่มต้นกรณีผู้ใช้งาน 2 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร) หลังจากการขยายแผนที่ออก จะสังเกตเห็นว่าแผนที่ที่มีลักษณะที่กว้างขึ้น ทำให้สามารถคาดการณ์ตำแหน่งของผู้ใช้คนอื่นได้ไม่ละเอียดมากนัก แต่สามารถขยายแผนที่เข้าเพื่อตรวจสอบตำแหน่งที่แน่ชัดของผู้ใช้คนอื่นได้

4.3.2 กรณีผู้ใช้งาน 3 ผู้ใช้งาน

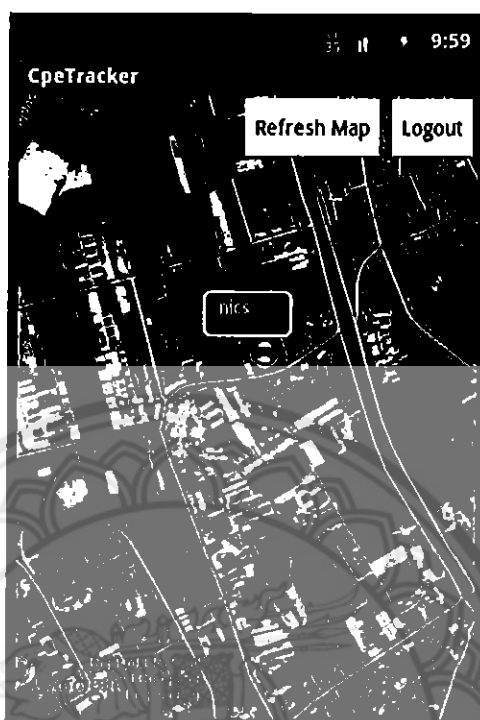
จะยกตัวอย่างรายละเอียดในฐานข้อมูล ดังนี้

		UserID	Username	Password	Name	Email	Status	SD	Active	Lat	Long
	X	26	aksak	1234	tae	taetae555@hotmail.com	USER	b0cb433ba02753f589c3c6f56ac63c52	Yes	16.7551271	100.19856895
	X	24	panlop	1150	nics	nics_fosoo@hotmail.com	USER	601457f34f0c48783dfa7b754c7107f	Yes	16.7531272	100.19856893
	X	25	songpon	1112	joe	za_himnoon@hotmail.com	USER	d74921fac0f1c4e7a21c3d8edb3377cd	Yes	16.7501274	100.19856894

รูปที่ 4.14 แสดงฐานข้อมูลกรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)

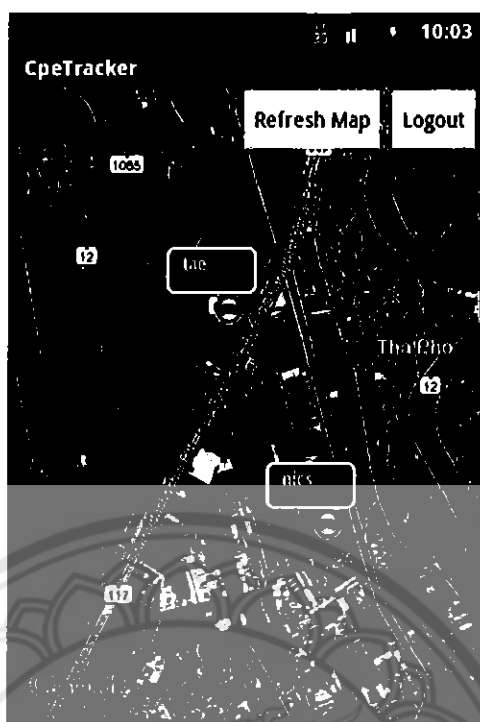
จากรูปที่ 4.14 แสดงถึงรายละเอียดของผู้ใช้งานที่ได้ลงทะเบียนไว้ โดยกรณีนี้เป็นข้อมูลของผู้ใช้งาน 3 ผู้ใช้งาน โดยได้สมมติพิกัดขึ้นมาเพื่อใช้ในการทดลอง

แสดงหน้าแผนที่ผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 3 กิโลเมตร)



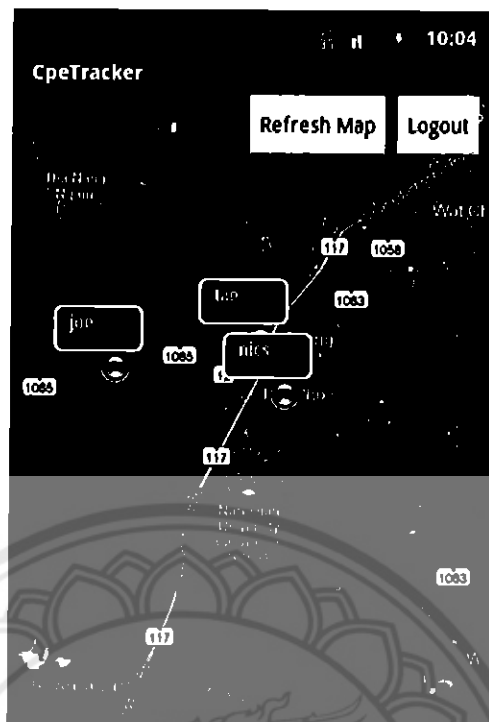
รูปที่ 4.15 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)

จากรูปที่ 4.15 แสดงแผนที่เริ่มต้นกรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร) โดยจะสังเกตได้ว่าจะยังไม่สามารถเห็นผู้ใช้คนอื่นได้ เพราะผู้ใช้คนอื่นสองคนอยู่ในระยะห่างมากกว่า 1.5x2.5 กิโลเมตร โดยสามารถขยายแผนที่ออกเพื่อค้นหาผู้ใช้คนอื่นสองคนได้ ดังนี้



รูปที่ 4.16 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)
เมื่อขยายแผนที่ออก พบผู้คนที่ 2

รูปที่ 4.16 จะสังเกตได้ว่าหลังจากที่ขยายแผนที่ออกแล้วพบกับผู้คนที่สองอยู่บริเวณไม่ไกลมากนัก แต่ยังไม่พบผู้ใช้งานอีกคนที่อาจอยู่ไกลกว่าผู้คนที่สอง สามารถขยายแผนที่ออกได้อีกเรื่อยๆจนกว่าจะพบผู้คนที่สุดท้าย



รูปที่ 4.17 แสดงแผนที่กรณีผู้ใช้งาน 3 ผู้ใช้งานในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร)
เมื่อขยายแผนที่ออก พบผู้ใช้ทั้งหมด

จากรูปที่ 4.17 เป็นรูปที่แสดงแผนที่ในกรณีที่พบผู้ใช้งานครบทั้งสามคนในระยะไกล (มากกว่า 1.5x2.5 กิโลเมตร) แล้ว โดยสังเกตได้ว่าแผนที่มีความกว้างมาก เพราะผู้ใช้แต่ละคนอยู่ไกลกัน ทำให้รายละเอียดในการหาดำแหน่งค่อนข้างน้อย วิธีแก้สามารถทำได้โดยการขยายแผนที่เข้าเพื่อหาดำแหน่งที่แน่นอนของผู้ใช้แต่ละคนที่ละคน ด้วยแผนที่ของกูเกิลที่มีความละเอียดค่อนข้างสูง

จากตัวอย่างข้างต้นเป็นการรันโปรแกรมบนโปรแกรม Eclipse ที่ใช้การจำลองการแสดงผลขึ้นมา สามารถแสดงแผนที่ได้อย่างถูกต้องแม่นยำ โดยในตัวโปรแกรมเองจะทำการ Refresh หน้าจอแผนที่ทุกๆ 1 นาที เพราะในกรณีที่ผู้ใช้งานเคลื่อนที่ไปยังตำแหน่งอื่น โปรแกรมก็จะ Refresh ให้ ทำให้การแสดงผลออกทางแผนที่เปลี่ยน ตำแหน่งที่แสดงเป็นปัจจุบันมากขึ้น

บทที่ 5

สรุปและแนวทางการพัฒนา

5.1 สรุปผลการดำเนินงาน

ในโครงการนี้ เป็นการออกแบบและศึกษาองค์ประกอบต่างเพื่อพัฒนาโปรแกรมสำหรับใช้งานบนระบบปฏิบัติการแอนดรอยด์ เพื่อช่วยเพิ่มขีดจำกัดความสามารถและความหลากหลายของโปรแกรมที่ทำงานบนอุปกรณ์เคลื่อนที่ต่างๆ โดยใช้โปรแกรม Eclipse เป็นเครื่องมือสำหรับปฏิบัติงาน ผลที่ได้จากโปรแกรมนี้อาจสามารถทำงานติดต่อกับระบบฐานข้อมูลผู้ใช้ได้ ตรวจสอบและติดตามตำแหน่ง ผู้ใช้งานบนแผนที่อย่างมีประสิทธิภาพ และยังสามารถนำไปใช้กับอุปกรณ์เคลื่อนที่ที่ใช้ระบบปฏิบัติการแอนดรอยด์ได้หลายชนิด ไม่ว่าจะเป็น โทรศัพท์เคลื่อนที่, แท็บเล็ต เป็นต้น

5.2 ปัญหาและแนวทางการแก้ไข

ปัญหาที่พบและแนวทางการแก้ไขในการทำโครงการ ดังนี้

1. โปรแกรมจำลองใน Eclipse ไม่สามารถรับตำแหน่งจาก GPS หรือ Internet ได้ ดังนั้นจึงต้องทดสอบกับอุปกรณ์จริงเท่านั้น
2. การสร้างโปรแกรมจำลองใน Eclipse ใหม่ จะต้องนำ MD5 ไปทำการขอ API KEY ใหม่ทุกครั้งเพื่อเรียกใช้แผนที่กูเกิล
3. การติด Firewall ของ Apache HTTP Server ทำให้อุปกรณ์ที่ใช้ทดสอบไม่สามารถติดต่อกับฐานข้อมูลได้ ดังนั้น จึงตั้งค่า Firewall ของเครื่อง Local-Host ให้อนุญาตใช้โปรแกรมนี้อย่างสาธารณะ

5.3 แนวทางการพัฒนาและข้อเสนอแนะ

ในการริเริ่มแนวคิดสำหรับการพัฒนา สามารถเพิ่มประสิทธิภาพของโปรแกรมได้หลายทาง ไม่ว่าจะเป็นการพัฒนาประสิทธิภาพให้ดีขึ้นหรือการเพิ่มเติมส่วนประกอบใหม่ๆเข้าไป โดยในโครงการนี้ยังมีหลายจุดที่ยังสามารถพัฒนาต่อได้ ไม่ว่าจะเป็น การทำระบบลงทะเบียนให้มีความสมบูรณ์และปลอดภัยมากยิ่งขึ้น คือ มีการจัดเก็บข้อมูลของระบบฐานข้อมูลที่มีข้อมูลรายละเอียดของผู้ลงทะเบียนมากขึ้น เพื่อป้องกันการซ้ำซ้อนข้อมูลผู้ใช้งาน มีการแจ้งเตือนสำหรับข้อผิดพลาดที่เกิดขึ้นจากการลงทะเบียนและส่วนอื่นๆในทุกๆกรณี เพื่อให้เป็นมาตรฐานมากขึ้น และในตัวโปรแกรมยังสามารถเพิ่มองค์ประกอบอื่นๆลงไปได้ เช่น โฆษณาแนะนำสินค้า, ข่าวประกาศต่างๆ เป็นต้น โดยโปรแกรมสามารถเสริมการทำงานของระบบในด้านสมรรถนะ การ

ประสิทธิภาพของอุปกรณ์ที่ใช้ สามารถปรับเปลี่ยนการรับส่งข้อมูลของโปรแกรมในรูปแบบเรียลไทม์ในช่วงเวลาอื่นที่เหมาะสมกับการใช้งาน เช่น การหยุดการรับส่งข้อมูลในกรณีที่ผู้ใช้งานไม่ได้เคลื่อนที่ ตำแหน่งพิกัดยังอยู่ที่เดิม ก็จะไม่มีการรับส่งข้อมูลไปยังระบบ ทำให้ไม่เกิดการเสียพลังงานโดยใช่เหตุ และ การปรับการรับส่งข้อมูลให้ใช้เวลามากขึ้น อาจเป็นหนึ่งชั่วโมงหรือมากกว่า แล้วแต่พฤติกรรมการใช้งานของผู้ใช้เอง ซึ่งสามารถพัฒนาต่อยอดได้อีกมากมาย ซึ่งทางคณะผู้จัดทำไม่ได้เน้นหนักในการพัฒนาในส่วนดังกล่าว โดยมุ่งเป้าไปในส่วนของการเรียกใช้งานระบบให้ทำงานร่วมกันได้ เพื่อสามารถเรียนรู้และทำความเข้าใจในระบบโดยรวมได้อย่างต่อเนื่องทั้งมีส่วนที่สามารถสร้างเพิ่มเติมและพัฒนาต่อยอดอยู่ในตัวโปรแกรมอีกมากมาย



เอกสารอ้างอิง

- [1] นายพนมพร ตระกูลศิษฐ์. “โปรแกรม Eclipse”. [Online]. Available :
http://race.nstn.ac.th/home/e-weblog/member/4511425009/document/Eclipse_Platform.pdf
- [2] Wikimedia Foundation, Inc. “ระบบปฏิบัติการแอนดรอยด์” . [Online]. Available :
[th.wikipedia.org/wiki/แอนดรอยด์_\(ระบบปฏิบัติการ\)](http://th.wikipedia.org/wiki/แอนดรอยด์_(ระบบปฏิบัติการ))
- [3] สุธี พงศาสกุลชัย.(2553).มือใหม่หัดใช้ PHP . (พิมพ์ครั้งที่ 1). กรุงเทพฯ: สำนักพิมพ์ เคทีพี
- [4] Wikimedia Foundation, Inc. “เจสัน” . [Online]. Available :
<http://th.wikipedia.org/wiki/JSON>
- [5] Grobal5 co., LTD. “ความรู้เรื่อง GPS” . [Online]. Available :
<http://www.global5thailand.com/thai/gps.htm>
- [6] Geekgok. “การใช้งาน Google map บน Android” . [Online]. Available :
<http://www.geekgok.com/2011/10/การใช้งาน-google-map-บน-android-ตอนที่-1/>

ภาคผนวก ก.

การเรียกใช้ Google Map

แผนที่ที่เราเลือกมาแสดงผลจะใช้แผนที่ของ Google ที่เป็นแผนที่คุณภาพสูง ใช้งานง่าย เหมาะสำหรับนำมาใช้งาน มีขั้นตอนการเรียกใช้งานดังรูป ก-1



รูป ก-1 แสดงการเรียกใช้งาน Google Map

การเรียกใช้งาน Google Map เริ่มแรกจะต้องทำการ ขอ API KEY จากทาง Google เพื่อทำการขอการใช้งานแผนที่ หลังจากได้ API KEY แล้ว นำมาใช้ในส่วนโปรแกรม ก็จะสามารถเรียกใช้งานแผนที่ของทาง Google ได้

การออกแบบส่วนต่อประสานกราฟฟิค (GUI)

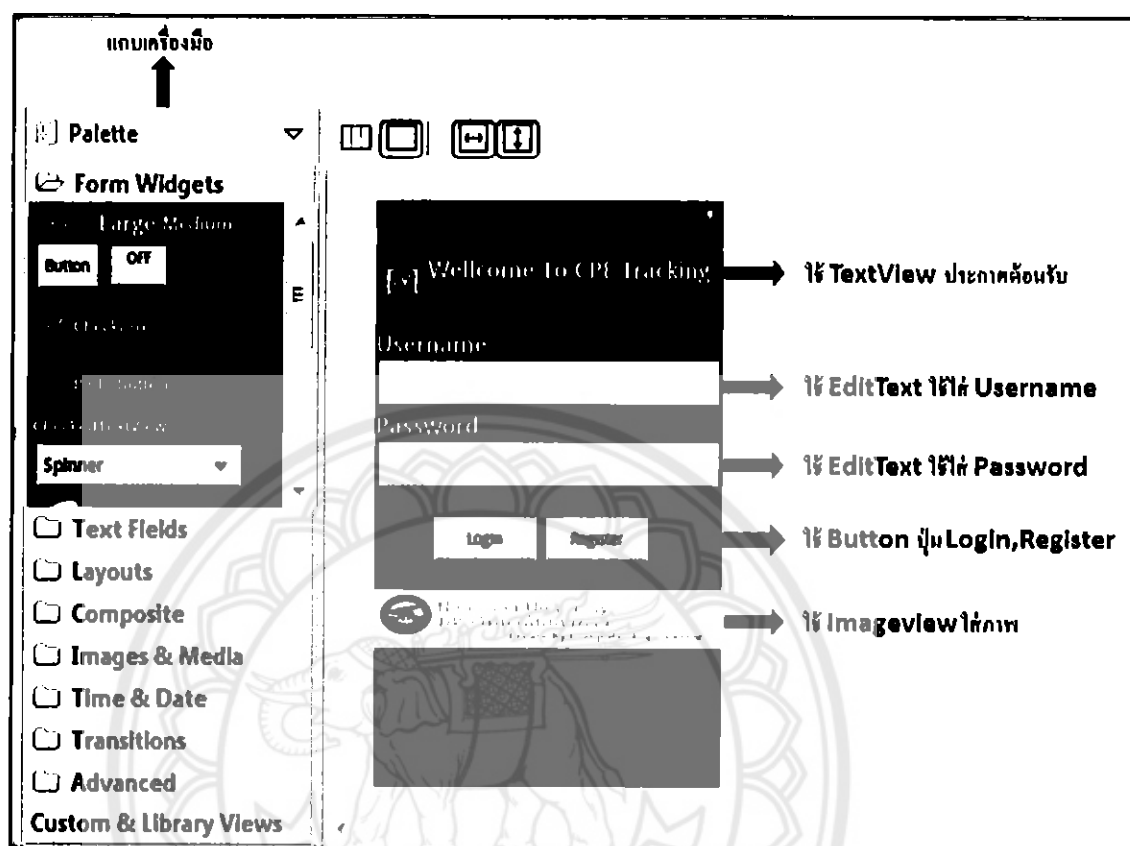
ในส่วนการออกแบบส่วนต่อประสานกราฟฟิคนั้น ผู้ทำโครงการได้ออกแบบไว้ 4 ส่วนด้วยกัน คือ ส่วน Login ส่วน Register ส่วน Online และส่วน Map ดังนี้



รูป ก-2 แสดงหน้า GUI โดยรวม

จากรูป ก-2 แสดงการออกแบบ GUI โดยรวม โดยจัดแบ่งเป็นสี่ส่วนดังกล่าวข้างต้น โดยมีรายละเอียดการทำงานดังต่อไปนี้

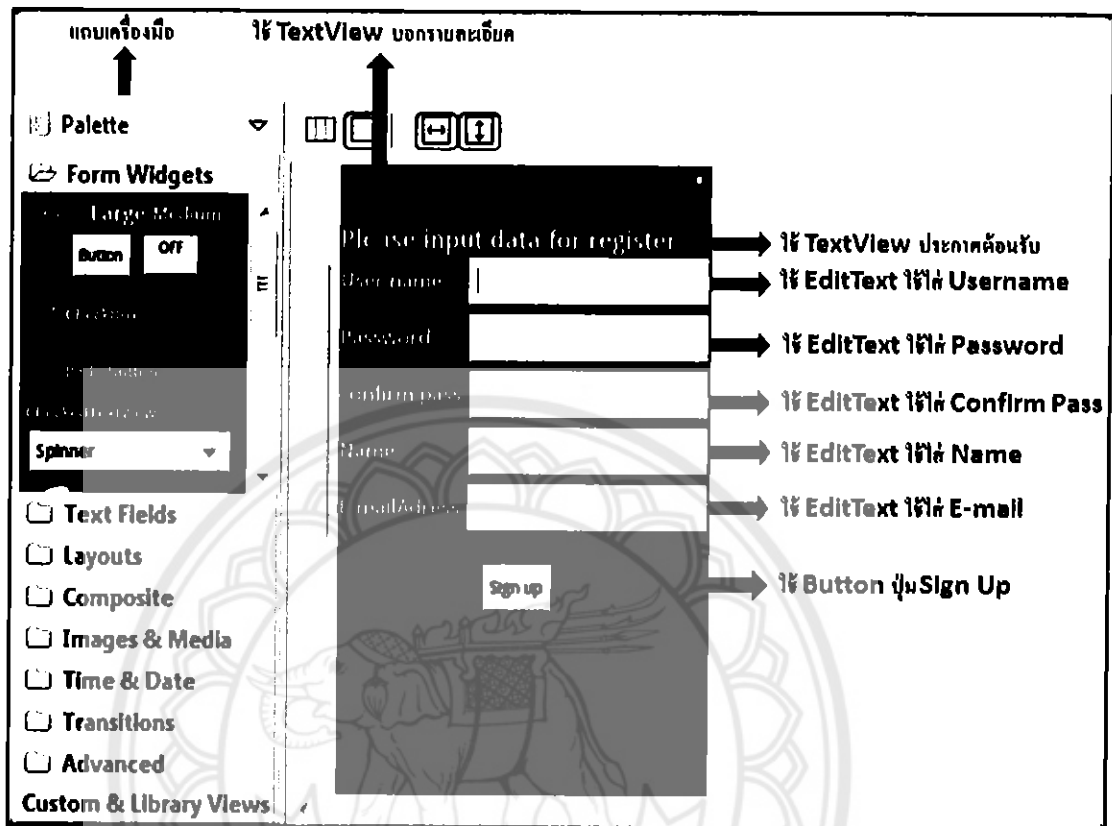
Login Page



รูป ก-3 แสดงหน้าการทำ Login Page

จากรูป ก-3 แสดงการทำ GUI ของหน้า Login ที่ใช้เครื่องมือการสร้างในโปรแกรม Eclipse โดยทำตามที่ได้ออกแบบไว้ในบทที่ 3 โดยในส่วนประกาศต้อนรับเราจะใช้เครื่องมือ "TextView" ใส่ตัวอักษร "Welcome To CPE Tracking" ไว้ในส่วนบน และอักษร "Username" และ "Password" แสดงให้ผู้ใช้งานได้รับรู้ว่าจะกรอกข้อมูลอะไรที่ส่วนไหน โดยช่องที่ให้กรอกข้อมูลนั้นจะใช้เครื่องมือ "EditText" สองช่องสำหรับให้ผู้ใช้กรอก Username และ Password หลังจากนั้นจะนำเครื่องมือ "Button" มาใช้เป็นปุ่มกดสำหรับ "Login" และ "Register" และในส่วนท้ายได้เสริมรูปภาพโดยใช้เครื่องมือ "ImageView" เพื่อเรียกรูปจากเครื่องคอมพิวเตอร์นำมาใช้ สามารถปรับขนาดได้ตามต้องการ

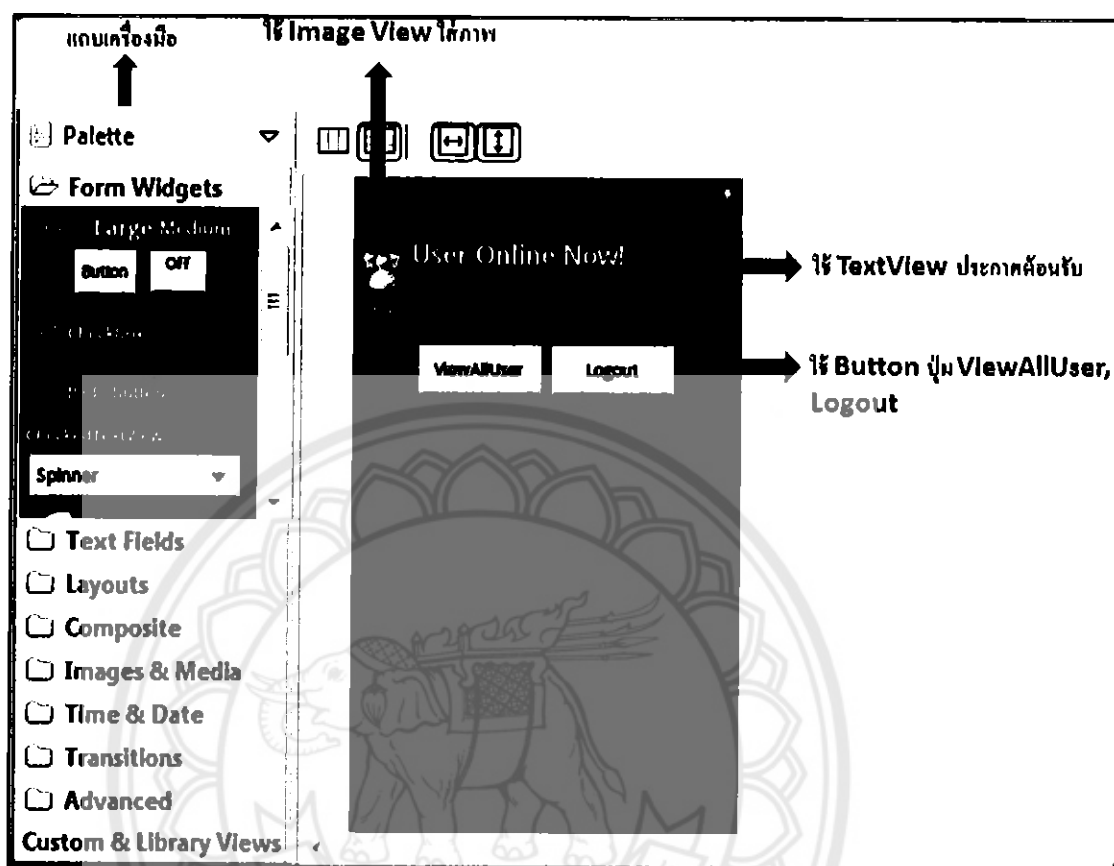
Register Page



รูป ก-4 แสดงหน้าการทำ Register Page

จากรูป ก-4 แสดงการทำ GUI ของหน้า Register โดยในส่วนแรกจะใช้ “TextView” เพื่อแสดงข้อความว่า “Please input data for register” เพื่อให้ผู้ใช้รู้ว่าอยู่ในขั้นตอนไหนแล้ว ในส่วนถัดมาจะเป็นการใช้ “TextView” เช่นเดียวกัน เพื่อนำมาระบุให้ทราบว่าให้ผู้ใช้กรอกข้อมูลได้ถูกต้อง ซึ่งจะใช้เครื่องมือ “EditText” ให้ผู้ใช้กรอกข้อมูลของตนเองเพื่อนำไปใช้งาน โดยจะมีรายละเอียดดังนี้ ช่องสำหรับใส่ Username โดย Username นี้จะเป็นชื่อที่ใช้ในการลงชื่อเข้าใช้งานโปรแกรมในหน้า Login Page ถัดมาเป็นช่องสำหรับใส่ Password และช่อง Confirm Password ที่ผู้ใช้งานต้องกรอกข้อมูลทั้งสองช่องนี้ให้เหมือนกัน ช่องถัดไปเป็นช่องสำหรับใส่ Name ซึ่งจะเป็นชื่อที่จะนำไปแสดงในหน้า Online Page ชื่อนี้เสมือนเป็นชื่อเล่นสำหรับแสดงบนหน้า Map Page ด้วย และช่องสุดท้ายจะเป็นช่องสำหรับกรอก E-mail เพื่อเป็นข้อมูลบันทึกต่อไป หลังจากกรอกข้อมูลเสร็จเรียบร้อยแล้ว จะใช้เครื่องมือ “Button” สร้างปุ่ม “Sign Up” เพื่อเป็นการยืนยันการสมัคร แต่มีข้อแม้ว่า ต้องใส่ข้อมูลให้ครบทุกช่อง มิฉะนั้นจะไม่สามารถสมัครได้ เมื่อทำตามทุกเงื่อนไขแล้ว ก็เป็นอันเสร็จสิ้นในขั้นตอนการลงทะเบียน

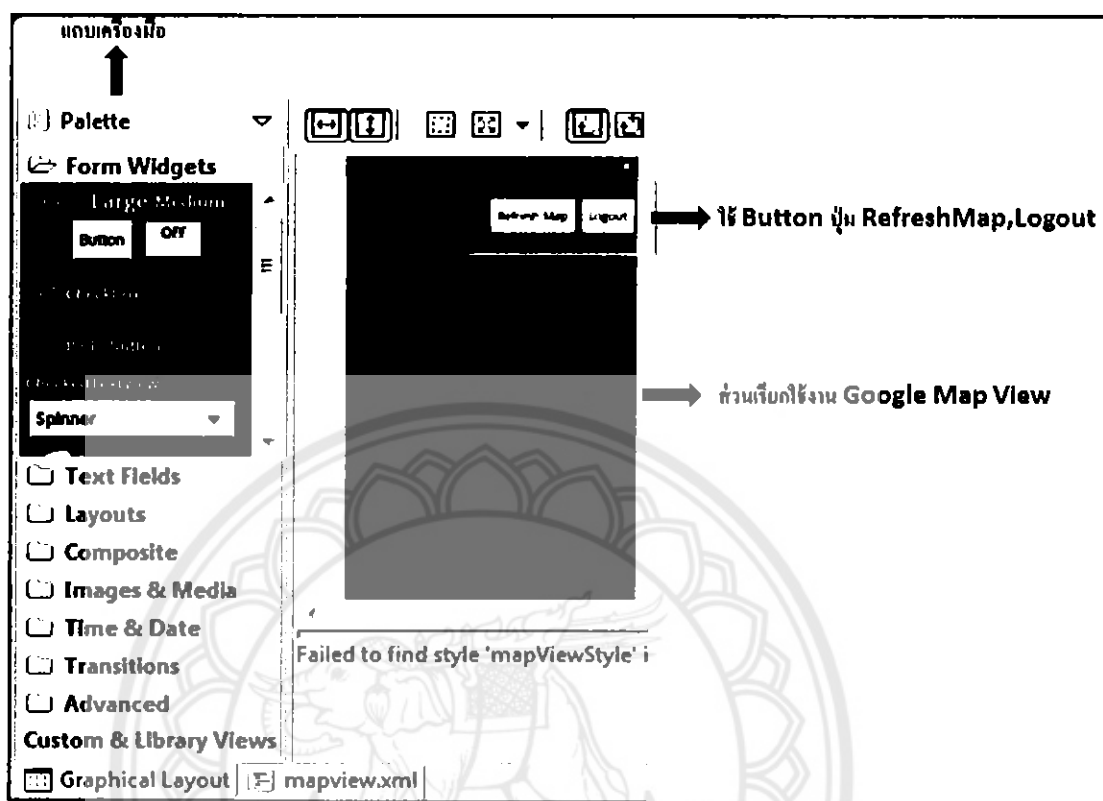
Online Page



รูป ก-5 แสดงหน้าการทำ Online Page

จากรูป ก-5 แสดงการสร้าง GUI ของหน้า Online Page ส่วนนี้จะไม่มีเครื่องมือนำมาใช้งานมากนัก เพราะจะแสดงรายชื่อผู้ใช้งานที่กำลังใช้งานอยู่ในขณะนี้ โดยในส่วนหัวข้อ จะใช้เครื่องมือ "TextView" แสดงข้อความว่า "User Online Now!" และใช้เครื่องมือ "Button" สร้างปุ่มสองปุ่ม คือ "ViewAllUser" ปุ่มสำหรับผ่านไปหน้า MapView โดยจะแสดงชื่อผู้ใช้งานบนแผนที่ และปุ่ม Logout สำหรับออกจากระบบการใช้งาน และจะกลับไปสู่หน้า Login ใหม่อีกครั้ง

Map Page



รูป ก-6 แสดงหน้าการทำ Map Page

จากรูปที่ ก-6 แสดงการสร้าง GUI ของ Map Page โดยพื้นที่ส่วนใหญ่จะเป็นพื้นที่สำหรับแสดงแผนที่ จะใช้เครื่องมือ “Button” นำมาสร้างปุ่มสองปุ่ม คือ Refresh Map เป็นปุ่มสำหรับเปลี่ยนแผนที่ เพื่ออัปเดตข้อมูลใหม่ แต่โดยปกติโปรแกรมจะทำการ Refresh ทุกๆ 1 นาที และอีกปุ่มคือ Logout สำหรับออกจากระบบ กลับไปสู่หน้า Login Page ใหม่

การติดตั้งและจัดการระบบฐานข้อมูล

หลักการทำงานในส่วนนี้ มีความสำคัญต่อการใช้งานของโปรแกรมเป็นอย่างมาก เพราะในระบบฐานข้อมูลนี้ จะต้องทำงานอยู่ตลอดการใช้งานโปรแกรม หลังจากที่ได้ออกแบบระบบฐานข้อมูล การรับส่งข้อมูลในบทที่ 3 แล้ว สิ่งที่จะต้องใช้งานคือ Web service โดยนำมาเป็นสื่อกลางในการทำงานของระบบ การรับส่งข้อมูลที่ผู้ใชกรอกมา คือ Uername, Password และอื่นๆ ที่ต้องการพื้นที่สำหรับบันทึกข้อมูลไว้ เพื่อให้ผู้ใช้งานสามารถใช้ข้อมูลเดิมในการใช้งานครั้งต่อไปได้ โดยหลักในการทำงานของระบบนี้ ทางคณะผู้จัดทำได้จำลอง Web service อย่างง่าย โดยใช้ภาษาพีเอชพี ทำงานร่วมกับ มายเอสคิวแอล และจะใช้เครื่องคอมพิวเตอร์ของผู้จัดทำโครงการนจำลองเป็นเครื่องเซิร์ฟเวอร์(Local host Server) โดยใช้โปรแกรม AppServ สร้างขึ้นมา

1. การสร้าง Local Host โดยใช้โปรแกรม AppServ

โปรแกรม AppServ เป็นโปรแกรมที่ถูกออกแบบมาเพื่อใช้สร้าง LocalHost ในเครื่องคอมพิวเตอร์แบบสำเร็จรูป และมีความสะดวกและง่ายในการติดตั้ง ทางคณะผู้จัดทำจึงได้นำโปรแกรมนี้มาใช้เป็นระบบฐานข้อมูลของเรา เพื่อจำลองรูปแบบการทำงานของโปรแกรมได้อย่างมีประสิทธิภาพมากขึ้น ในหัวข้อนี้ก็จะแสดงการสร้าง LocalHost ด้วยโปรแกรม AppServ โดยมีขั้นตอนการติดตั้งระบบอยู่ 8 ขั้นตอนด้วยกัน

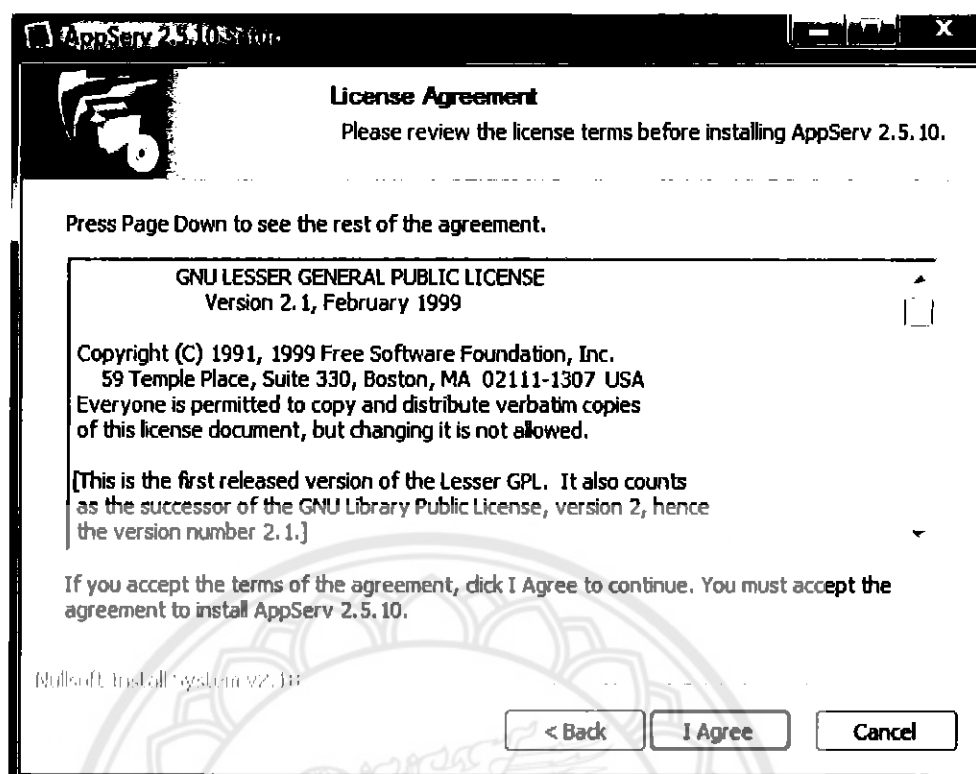
ขั้นตอนที่ 1 ดาวน์โหลดไฟล์ของ AppServ เวอร์ชันที่ต้องการ
ขั้นตอนที่ 2 เปิดไฟล์ AppServ ที่ได้ดาวน์โหลดมาจะได้ดังนี้



รูป ก-7 แสดงหน้าต่างต้อนรับของโปรแกรม AppServ

จากรูป ก-7 แสดงหน้าต่างต้อนรับของโปรแกรม AppServ ที่แสดงข้อความเกี่ยวกับโปรแกรม ในเบื้องต้น ทำการคลิกถัดไป(Next) เพื่อไปสู่ขั้นตอนต่อไป

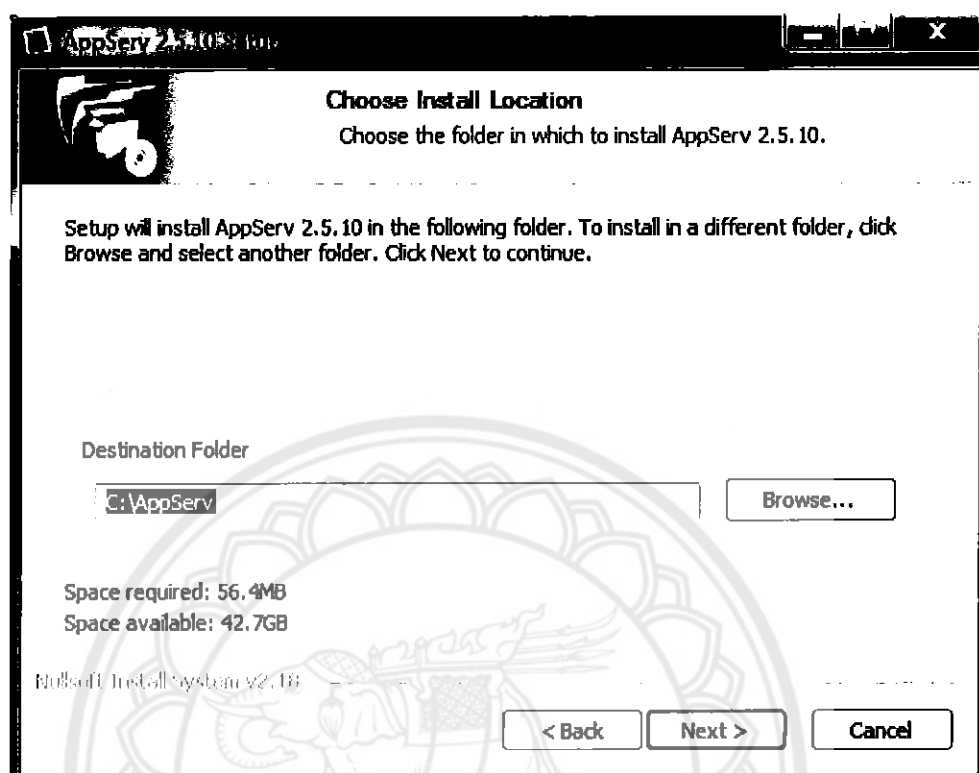
ขั้นตอนที่ 3 หลังจากคลิกถัดไปจะถึงหน้าแสดงรายละเอียดข้อตกลง ดังนี้



รูป ก-8 แสดงหน้าข้อตกลงของโปรแกรม AppServ

จากรูป ก-8 แสดงข้อตกลงของโปรแกรม AppServ ที่ควรรู้ หลังจากทำความเข้าใจแล้ว ให้คลิกที่ I Agree ในกรณีที่เราเข้าใจข้อตกลงแล้ว เพื่อไปสู่ขั้นตอนถัดไป

ขั้นตอนที่ 4



รูป ก-9 แสดงหน้าเลือกพื้นที่สำหรับเก็บข้อมูลของโปรแกรม AppServ

จากรูป ก-9 แสดงหน้าเลือกพื้นที่สำหรับเก็บข้อมูลของโปรแกรม Appserv ในเครื่องคอมพิวเตอร์ของเรา โดยสามารถกดปุ่ม “Browse” เพื่อค้นหาพื้นที่ที่ต้องการ หลังจากได้พื้นที่เก็บที่ต้องการแล้ว คลิกที่ Next เพื่อไปสู่ขั้นตอนต่อไป

ขั้นตอนที่ 5



รูป ก-10 แสดงหน้าเลือกโปรแกรมที่เกี่ยวข้องของโปรแกรม AppServ

จากรูป ก-10 แสดงหน้าสำหรับเลือกโปรแกรมที่เกี่ยวข้องที่จะติดตั้ง โดยปกติจะเลือกทั้งหมด หลังจากเลือกเรียบร้อยแล้วให้คลิก Next เพื่อไปสู่ขั้นตอนถัดไป

ขั้นตอนที่ 6 โดยขั้นตอนนี้จะเป็นการกรอกข้อมูลสำหรับการติดต่อกับอะแพชี เว็บเซิร์ฟเวอร์ (Apache server) โดยอะแพชี เว็บเซิร์ฟเวอร์เป็นซอฟต์แวร์สำหรับเปิดให้บริการเซิร์ฟเวอร์บนโปรโตคอลเอชทีทีพีโดยสามารถทำงานได้บนหลายระบบปฏิบัติการ ดังนี้

AppServ 2.5.10 Setup

Apache HTTP Server Information
Please enter your server's information.

Server Name (e.g. www.appservnetwork.com)
localhost

Administrator's Email Address (e.g. webmaster@gmail.com)
nics_losso@hotmail.com

Apache HTTP Port (Default : 80)
80

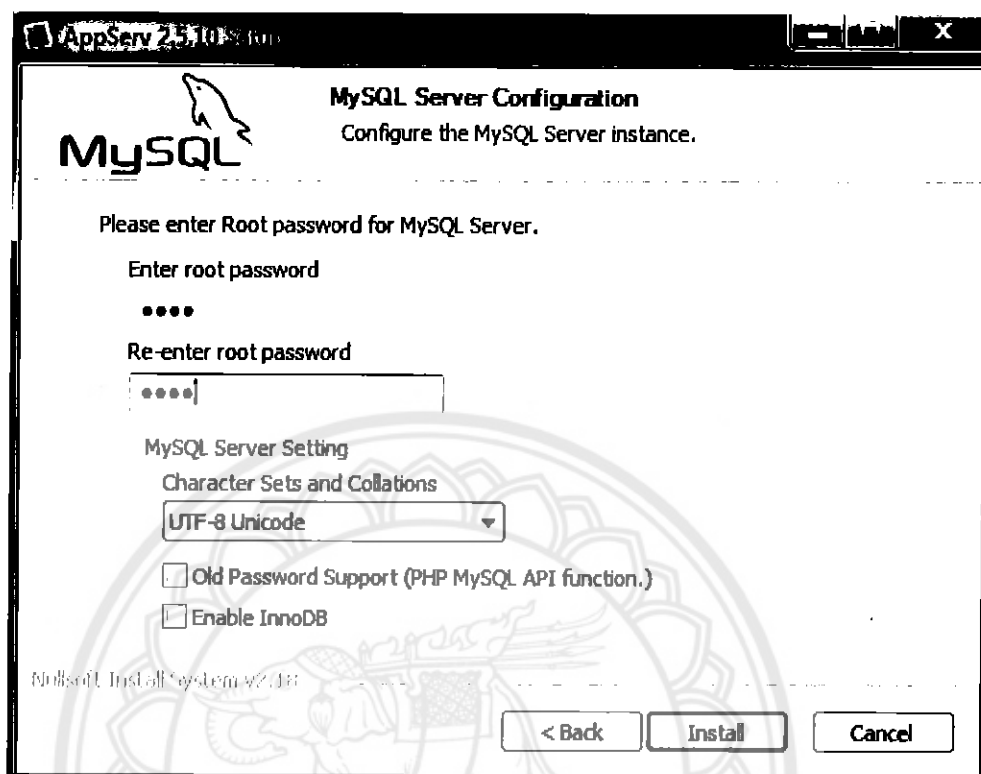
Microsoft Windows [Version 6.0.6002.18005] Copyright (c) 2006 Microsoft Corporation. All rights reserved. C:\Windows\system32\cmd.exe

< Back Next > Cancel

รูป ก-11 แสดงหน้าต่างกรอกรายละเอียดสำหรับ Apache server ของโปรแกรม AppServ

จากรูป ก-11 เป็นการกรอกข้อมูลรายละเอียดสำหรับ Apache server โดยในช่องแรกจะให้ใส่ชื่อเซิร์ฟเวอร์ของเราไป ในที่นี้ใส่ชื่อว่า Localhost ช่องถัดมาจะให้เรากรอกอีเมลที่ใช้งานลงไป ช่องสุดท้ายเป็นการใส่ช่อง Apache HTTP Port ในที่นี้ใส่ไว้ที่ Port 80 หลังจากกรอกข้อมูลเรียบร้อยแล้วให้คลิก Next เพื่อไปขั้นตอนถัดไป

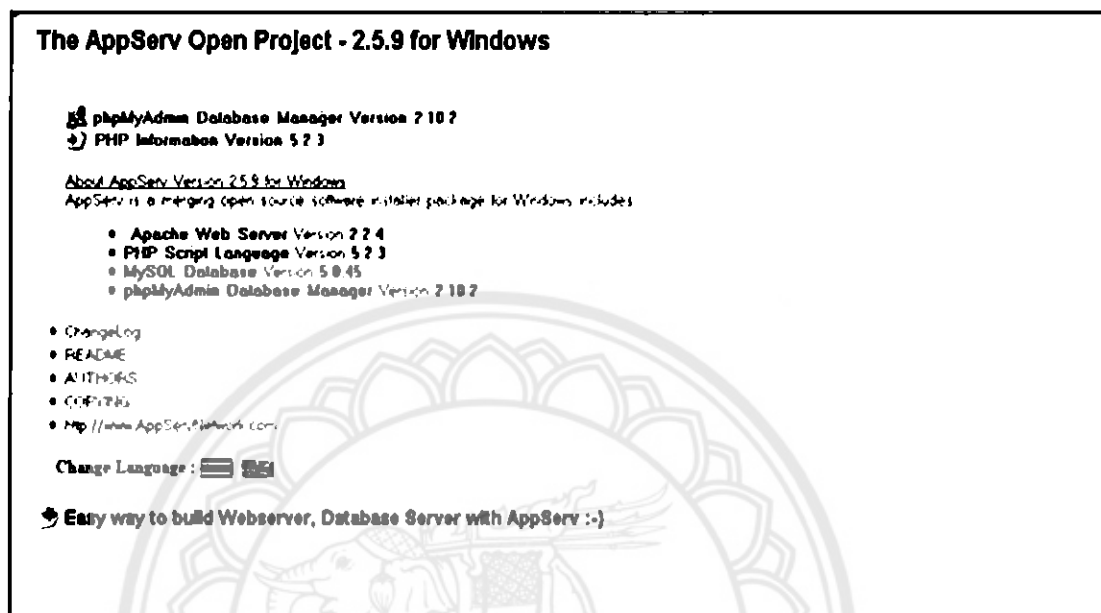
ขั้นตอนที่ 7 ขั้นตอนสำหรับมายเอสคิวแอล ดังนี้



รูป ก-12 แสดงหน้าสำหรับกรอกรายละเอียดสำหรับมายเอสคิวแอลของโปรแกรม AppServ

จากรูป ก-12 แสดงหน้าสำหรับกรอกรายละเอียดข้อมูลของ MySQL Server โดยจะมีช่องสำหรับใส่รหัสผ่านและยืนยันรหัสผ่านของเราที่สามารถจำได้ง่าย และ Character Sets and Collations ในที่นี้เลือกเป็น UTF-8 Unicode หลังจากนั้นคลิกที่ Install เพื่อทำการติดตั้ง

ขั้นตอนที่ 8 หลังจากทำการรอในกระบวนการติดตั้งเรียบร้อยแล้ว ทำการทดสอบว่าติดตั้งสมบูรณ์หรือไม่ โดยเปิดที่ Web Browser ของเรา ทำการพิมพ์ <http://localhost> ลงไปในช่อง Address ถ้าติดตั้งเสร็จสมบูรณ์จะได้ดังรูปที่ ก-13

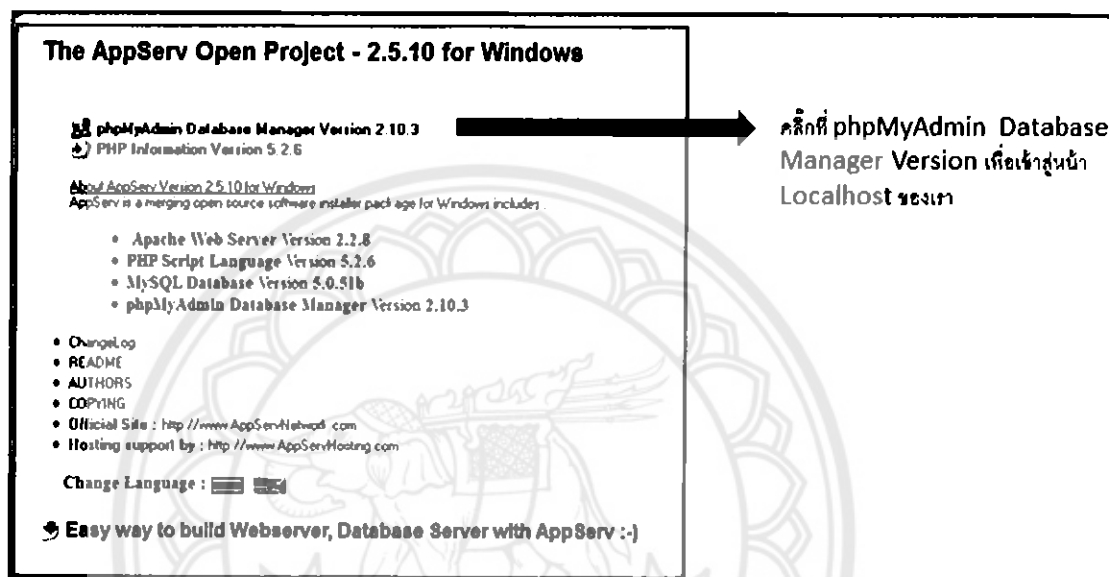


รูป ก-13 แสดงหน้าจอหลังจากติดตั้งเสร็จสมบูรณ์ของโปรแกรม AppServ

การสร้าง MySQL โดยใช้โปรแกรม AppServ

หลังจากที่ได้ลงโปรแกรม Appserv แล้ว ในขั้นตอนนี้จะเป็นขั้นตอนการพื้นฐานข้อมูลภายใน Localhost ที่เราสร้างขึ้นมา ซึ่งในโปรแกรมจะมีเครื่องมือสำหรับการทำฐานข้อมูลมายเอสคิวแอ็ด โดยมีขั้นตอนการสร้างดังต่อไปนี้

1. ส่วนเริ่มแรกนี้ จะเป็นหน้าต่างของโปรแกรม Appserv ที่ได้ติดตั้งไว้ ดังนี้



รูป ก-14 แสดงหน้าต่างหลักหลังจากติดตั้งโปรแกรม AppServ

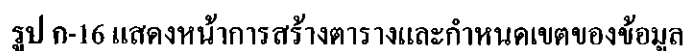
จากรูป ก-14 แสดงหน้าต่างหลักของ Web service ของเราที่ได้สร้างไว้ ในขั้นตอนนี้จะทำการสร้างฐานข้อมูลขึ้นมา โดยเลือกที่ “phpMyAdmin Database Manager Version 2.10.3” เพื่อทำการกำหนดหรือสร้างองค์ประกอบต่างๆของเซิร์ฟเวอร์จำลองของเรา ในกรณีที่ให้ใส่ชื่อผู้ใช้และรหัสผ่านเพื่อยืนยันตัวตนของเรา ก็ให้ใส่ชื่อผู้ใช้งานว่า root ส่วนรหัสผ่านก็ใช้รหัสผ่านที่ได้ใส่ไว้ในส่วนการติดตั้งโปรแกรม Appserv ในตอนแรก

2. ส่วนของการตั้งชื่อฐานข้อมูลใหม่ของเราและมาตรฐานการตรวจเทียบภาษาในฐานข้อมูล โดยมีขั้นตอนและรูปแบบดังนี้



จากรูป ก-15 แสดงการสร้างฐานข้อมูลใหม่ โดยการตั้งชื่อฐานข้อมูลที่ต้องการลงไปในช่องทางด้านซ้ายมือ หลังจากนั้นในช่องทางด้านขวามือเป็นช่องสำหรับเลือกมาตรฐานการตรวจเทียบภาษาในฐานข้อมูลของเรา ในที่นี้จะเลือกเป็นการเรียงลำดับการตรวจเทียบ หลังจากใส่รายละเอียดเรียบร้อยแล้ว ให้คลิก “สร้าง” โปรแกรมจะทำการสร้างฐานข้อมูลของเรา และไปสู่ขั้นตอนต่อไป

3. ขั้นตอนของการสร้างตารางในฐานข้อมูล มีรายละเอียดการสร้างตาราง ดังนี้



จากรูป ก-16 เป็นขั้นตอนการสร้างตารางภายในฐานข้อมูลของเรา เพื่อใช้เก็บข้อมูลอย่างเป็นระบบ ในช่องทางซ้ายมือจะเป็นช่องสำหรับใส่ชื่อตาราง ข้อแนะนำ ก็ควรตั้งชื่อให้สอดคล้อง

กับข้อมูลที่จะเก็บ เพื่อไม่ให้สับสนในภายหลัง กรณีที่มีหลายตาราง หลังจากใส่ชื่อตารางแล้ว ในช่องทางขวามือ เป็นช่อง Number of Fields คือ ช่องสำหรับใส่จำนวนของเขตข้อมูล (Fields) ในตารางว่าต้องการเก็บข้อมูลทั้งหมดกี่เขต ในที่นี้เลือกเป็น 10 เขตข้อมูล หลังจากกำหนดรายละเอียดเรียบร้อยแล้ว ให้คลิก “ลงมือ” โปรแกรมจะทำการสร้างตารางให้ และผ่านไปยังขั้นตอนถัดไป

4. ในขั้นตอนนี้ จะเป็นการกำหนดรายละเอียดของเขตข้อมูลแต่ละเขต โดยมีรายละเอียดที่ต้องกำหนด ดังนี้

ชื่อ Fields	ฟิลด์	ชนิดข้อมูลใน Fields
		VARCHAR
		VARCHAR
		VARCHAR
		VARCHAR
		VARCHAR
		VARCHAR
		VARCHAR
		VARCHAR
		VARCHAR
		VARCHAR
		VARCHAR

รูป ก-17 แสดงหน้าการใส่รายละเอียดของเขตข้อมูล(ส่วนที่ 1)

จากรูป ก-17 เป็นรูปส่วนที่ 1 ของการกำหนดรายละเอียดของเขตข้อมูล ในส่วนนี้จะมีตารางขึ้นมาให้ผู้ใช้ใส่รายละเอียด โดยมีจำนวนแถวเท่ากับจำนวน Fields ที่กำหนดไว้ในขั้นตอนก่อนหน้านี้ นั่นคือ 10 แถว สำหรับ 10 fields โดยใน คอลัมน์แรกจากซ้ายมือจะเป็นช่องสำหรับใส่ชื่อ Fields ที่ได้ออกแบบไว้ โดยทั่วไปจะเลือก Fields หลักอยู่ในแถวแรก ในคอลัมน์ถัดมาจะเป็นช่องสำหรับเลือกชนิดของข้อมูลใน Fields ของว่าจะให้เป็นชนิดอะไร เช่น ชนิดตัวเลขจำนวนเต็ม (int), ชนิดเก็บเป็นตัวอักษร (Varchar) เป็นต้น ต่อมาเป็นคอลัมน์สำหรับการกำหนดขนาดของข้อมูลใน Fields คอลัมน์ต่อไปจะเป็นการกำหนดการเรียงลำดับข้อมูล และคอลัมน์ของคุณสมบัติเฉพาะของข้อมูลที่เก็บ

กำหนดความ ว่างเปล่า	คำปรีชา	ส่วนเพิ่มเติม	ไพรมารี	ดัชนี	เอกลักษณ์	ว่าง	ข้อความเต็ม	หมายเหตุ
↓	↓	↓	↓	↓	↓	↓	↓	↓
ว่างเปล่า (null)	คำปรีชา ²	เพิ่มเติม	☐	☒	☐	☐	☐	หมายเหตุ
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								
not null ▼								

รูป ก-18 แสดงหน้าการใส่รายละเอียดของเขตข้อมูล(ส่วนที่ 2)

จากรูป ก-18 เป็นรูปส่วนที่ 2 ของการกำหนดรายละเอียดของเขตข้อมูล จะมีคอลัมน์ของการกำหนดความว่างเปล่า นั่นก็คือการกำหนด Null ให้กับ Fields ว่าว่างเปล่าได้หรือไม่ได้ ถัดมาคือส่วนของการกำหนดค่าเริ่มต้นของ Fields และคอลัมน์ของส่วนเพิ่มเติมที่ต้องการสำหรับ Fields หลังจากนั้น จะเป็นการเลือกความเฉพาะสำหรับ Fields แต่ละ Fields ดังนี้ ไพรมารี, ดัชนี, เอกลักษณ์, ว่าง, ข้อความเต็ม โดยสามารถเลือกตามที่ได้ออกแบบไว้ในแต่ละลักษณะของ Fields นั้นๆ ส่วนคอลัมน์สุดท้ายเป็นส่วนของหมายเหตุเฉพาะของ Fields ก็เป็นอันเสร็จสิ้น โดยการกำหนดรายละเอียดของ Fields ไม่จำเป็นต้องใส่รายละเอียดครบทุกหัวข้อ ขึ้นอยู่กับลักษณะของ Fields นั้นๆ ว่าต้องการรูปแบบอย่างไร เมื่อเสร็จสิ้นการกำหนดรายละเอียด Fields แล้ว ก็ทำการคลิก “บันทึก” ก็เป็นอันเสร็จสิ้น หรือต้องการสร้าง Fields เพิ่มก็สามารถเพิ่มได้ ดังรูป ก-19

บันทึก	หรือ Add 1	field(s)	ลงมือ
--------	------------	----------	-------

รูป ก-19 การบันทึกและการเพิ่ม Field

ตัวอย่างแสดงหน้าการใส่รายละเอียดของเขตข้อมูล ดังรูป ก-20

ฟิลด์	ชนิด	การโยงข้าม	แมตเทรียล	วางแปลน (คณ)	สลับไวยา	เพิ่มเติม	การตั้งค่า									
UserID	int(3)			ไม่		auto_increment	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Username	varchar(20)	utf8_general_ci		ไม่			☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Password	varchar(20)	utf8_general_ci		ไม่			☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Name	varchar(100)	utf8_general_ci		ไม่			☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Email	varchar(150)	utf8_general_ci		ไม่			☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Status	enum('ADMIN', 'USER')	utf8_general_ci		ไม่	USER		☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
SID	varchar(32)	utf8_general_ci		ไม่			☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Active	enum('Yes', 'No')	utf8_general_ci		ไม่	No		☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Lat	double			ไม่			☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Lng	double			ไม่			☒	☒	☒	☒	☒	☒	☒	☒	☒	☒

รูป ก-20 แสดงหน้าการใส่รายละเอียดของเขตข้อมูล

การสร้างฐานข้อมูลบน phpMyadmin นี้ ขั้นตอนหลักๆจะไม่ซับซ้อนมากนัก เพราะมีเครื่องมือช่วยในการสร้างค่อนข้างพร้อม และใช้งานง่าย โดยในส่วนนี้จะรอการเรียกใช้งานร่วมกับตัวโปรแกรมใน Eclipse เพื่อให้การทำงานเป็นระบบสมบูรณ์มากขึ้น

การรับพิกัดจากการเชื่อมต่อ Internet

จาก <http://developer.android.com/guide/topics/location/obtaining-user-location.html>

Obtaining User Location

Knowing where the user is allows your application to be smarter and deliver better information to the user. When developing a location-aware application for Android, you can utilize GPS and Android's Network Location Provider to acquire the user location. Although GPS is most accurate, it only works outdoors, it quickly consumes battery power, and doesn't return the location as quickly as users want. Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoors and outdoors, responds faster, and uses less battery power. To obtain the user location in your application, you can use both GPS and the Network Location Provider, or just one.

Challenges in Determining User Location

Obtaining user location from a mobile device can be complicated. There are several reasons why a location reading (regardless of the source) can contain errors and be inaccurate. Some sources of error in the user location include:

- **Multitude of location sources**

GPS, Cell-ID, and Wi-Fi can each provide a clue to users location. Determining which to use and trust is a matter of trade-offs in accuracy, speed, and battery-efficiency.

- **User movement**

Because the user location changes, you must account for movement by re-estimating user location every so often.

- **Varying accuracy**

Location estimates coming from each location source are not consistent in their accuracy. A location obtained 10 seconds ago from one source might be more accurate than the newest location from another or same source.

These problems can make it difficult to obtain a reliable user location reading. This document provides information to help you meet these challenges to obtain a reliable location reading. It also provides ideas that you can use in your application to provide the user with an accurate and responsive geo-location experience.

Requesting Location Updates

Before addressing some of the location errors described above, here is an introduction to how you can obtain user location on Android.

Getting user location in Android works by means of callback. You indicate that you'd like to receive location updates from the LocationManager ("Location Manager") by calling requestLocationUpdates(), passing it a LocationListener. Your LocationListener must implement several callback methods that the Location Manager calls when the user location changes or when the status of the service changes.

For example, the following code shows how to define a LocationListener and request location updates:

```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        // Called when a new location is found by the network location provider.
        makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};

// Register the listener with the Location Manager to receive location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
```

The first parameter in `requestLocationUpdates()` is the type of location provider to use (in this case, the Network Location Provider for cell tower and Wi-Fi based location). You can control the frequency at which your listener receives updates with the second and third parameter—the second is the minimum time interval between notifications and the third is the minimum change in distance between notifications—setting both to zero requests location notifications as frequently as possible. The last parameter is your `LocationListener`, which receives callbacks for location updates.

To request location updates from the GPS provider, substitute `GPS_PROVIDER` for `NETWORK_PROVIDER`. You can also request location updates from both the GPS and the Network Location Provider by calling `requestLocationUpdates()` twice—once for `NETWORK_PROVIDER` and once for `GPS_PROVIDER`.

Requesting User Permissions

In order to receive location updates from `NETWORK_PROVIDER` or `GPS_PROVIDER`, you must request user permission by declaring either the `ACCESS_COARSE_LOCATION` or `ACCESS_FINE_LOCATION` permission, respectively, in your Android manifest file. For example:

```
<manifest ... >
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    ...
</manifest>
```

Without these permissions, your application will fail at runtime when requesting location updates.

Defining a Model for the Best Performance

Location-based applications are now commonplace, but due to the less than optimal accuracy, user movement, the multitude of methods to obtain the location, and the desire to conserve battery, getting user location is complicated. To overcome the obstacles of obtaining a good user location while preserving battery power, you must define a consistent model that specifies how your application obtains the user location. This model includes when you start and stop listening for updates and when to use cached location data.

Flow for obtaining user location

Here's the typical flow of procedures for obtaining the user location:

1. Start application.
2. Sometime later, start listening for updates from desired location providers.
3. Maintain a "current best estimate" of location by filtering out new, but less accurate fixes.
4. Stop listening for location updates.
5. Take advantage of the last best location estimate.

Figure 1 demonstrates this model in a timeline that visualizes the period in which an application is listening for location updates and the events that occur during that time.

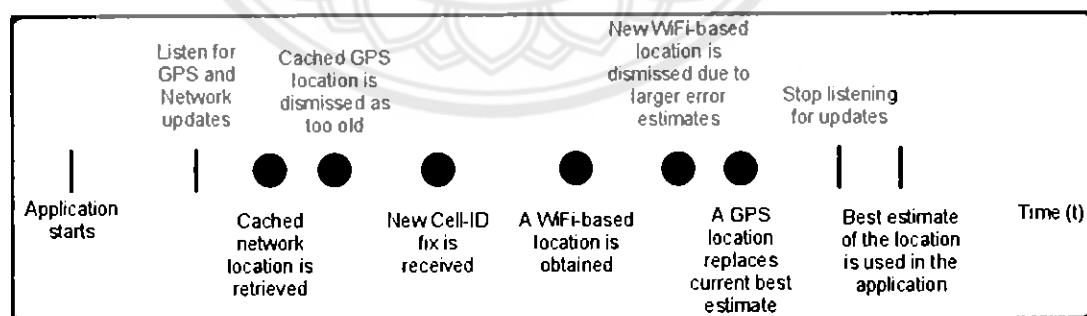


Figure 1. A timeline representing the window in which an application listens for location updates.

This model of a window—during which location updates are received—frames many of the decisions you need to make when adding location-based services to your application.

Deciding when to start listening for updates

You might want to start listening for location updates as soon as your application starts, or only after users activate a certain feature. Be aware that long windows of listening for location fixes can consume a lot of battery power, but short periods might not allow for sufficient accuracy.

As demonstrated above, you can begin listening for updates by calling `requestLocationUpdates()`:

```
LocationProvider locationProvider = LocationManager.NETWORK_PROVIDER;
// Or, use GPS location data:
// LocationProvider locationProvider = LocationManager.GPS_PROVIDER;

locationManager.requestLocationUpdates(locationProvider, 0, 0, locationListener);
```

Getting a fast fix with the last known location

The time it takes for your location listener to receive the first location fix is often too long for users wait. Until a more accurate location is provided to your location listener, you should utilize a cached location by calling `getLastKnownLocation(String)`:

```
LocationProvider locationProvider = LocationManager.NETWORK_PROVIDER;
// Or use LocationManager.GPS_PROVIDER

Location lastKnownLocation = locationManager.getLastKnownLocation(locationProvider);
```

Deciding when to stop listening for updates

The logic of deciding when new fixes are no longer necessary might range from very simple to very complex depending on your application. A short gap between when the location is acquired and when the location is used, improves the accuracy of the estimate. Always beware that listening for a long time consumes a lot of battery power, so as soon as you have the information you need, you should stop listening for updates by calling `removeUpdates(PendingIntent)`:

```
// Remove the listener you previously added
locationManager.removeUpdates(locationListener);
```

Maintaining a current best estimate

You might expect that the most recent location fix is the most accurate. However, because the accuracy of a location fix varies, the most recent fix is not always the best. You should include logic for choosing location fixes based on several criteria. The criteria also varies depending on the use-cases of the application and field testing.

Here are a few steps you can take to validate the accuracy of a location fix:

- Check if the location retrieved is significantly newer than the previous estimate.
- Check if the accuracy claimed by the location is better or worse than the previous estimate.
- Check which provider the new location is from and determine if you trust it more.

An elaborate example of this logic can look something like this:

```
private static final int TWO_MINUTES = 1000 * 60 * 2;

/** Determines whether one Location reading is better than the current Location fix
 * @param location The new Location that you want to evaluate
 * @param currentBestLocation The current Location fix, to which you want to compare the new one
 */
protected boolean isBetterLocation(Location location, Location currentBestLocation) {
    if (currentBestLocation == null) {
        // A new location is always better than no location
        return true;
    }

    // Check whether the new location fix is newer or older
    long timeDelta = location.getTime() - currentBestLocation.getTime();
    boolean isSignificantlyNewer = timeDelta > TWO_MINUTES;
    boolean isSignificantlyOlder = timeDelta < -TWO_MINUTES;
    boolean isNewer = timeDelta > 0;

    // If it's been more than two minutes since the current location, use the new location
    // because the user has likely moved
    if (isSignificantlyNewer) {
        return true;
    }
    // If the new location is more than two minutes older, it must be worse
    } else if (isSignificantlyOlder) {
        return false;
    }

    // Check whether the new location fix is more or less accurate
    int accuracyDelta = (int) (location.getAccuracy() - currentBestLocation.getAccuracy());
    boolean isLessAccurate = accuracyDelta > 0;
    boolean isMoreAccurate = accuracyDelta < 0;
    boolean isSignificantlyLessAccurate = accuracyDelta > 200;

    // Check if the old and new location are from the same provider
    boolean isFromSameProvider = isSameProvider(location.getProvider(),
        currentBestLocation.getProvider());

    // Determine location quality using a combination of timeliness and accuracy
    if (isMoreAccurate) {
        return true;
    }
    } else if (isNewer && !isLessAccurate) {
        return true;
    }
    } else if (isNewer && !isSignificantlyLessAccurate && isFromSameProvider) {
        return true;
    }
    }
    return false;
}
```

```

/** Checks whether two providers are the same */
private boolean isSameProvider(String provider1, String provider2) {
    if (provider1 == null) {
        return provider2 == null;
    }
    return provider1.equals(provider2);
}

```

Adjusting the model to save battery and data exchange

As you test your application, you might find that your model for providing good location and good performance needs some adjustment. Here are some things you might change to find a good balance between the two.

Reduce the size of the window

A smaller window in which you listen for location updates means less interaction with GPS and network location services, thus, preserving battery life. But it also allows for fewer locations from which to choose a best estimate.

Set the location providers to return updates less frequently

Reducing the rate at which new updates appear during the window can also improve battery efficiency, but at the cost of accuracy. The value of the trade-off depends on how your application is used. You can reduce the rate of updates by increasing the parameters in [requestLocationUpdates\(\)](#) that specify the interval time and minimum distance change.

Restrict a set of providers

Depending on the environment where your application is used or the desired level of accuracy, you might choose to use only the Network Location Provider or only GPS, instead of both. Interacting with only one of the services reduces battery usage at a potential cost of accuracy.

Common application cases

There are many reasons you might want to obtain the user location in your application. Below are a couple scenarios in which you can use the user location to enrich your application. Each scenario also describes good practices for when you should start and stop listening for the location, in order to get a good reading and help preserve battery life.

Tagging user-created content with a location

You might be creating an application where user-created content is tagged with a location. Think of users sharing their local experiences, posting a review for a restaurant, or recording some content that can be augmented with their current location. A model of how this interaction might happen, with respect to the location services, is visualized in figure 2.

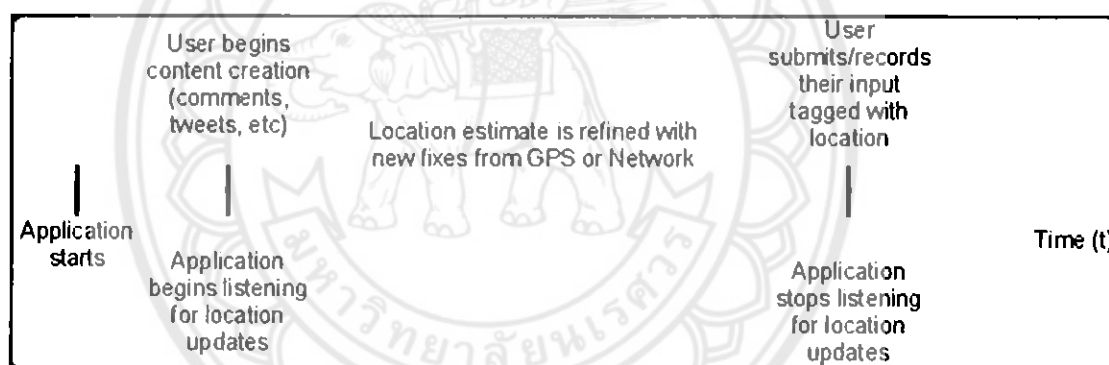


Figure 2. A timeline representing the window in which the user location is obtained and listening stops when the user consumes the current location.

This lines up with the previous model of how user location is obtained in code (figure 1). For best location accuracy, you might choose to start listening for location updates when users begin creating the content or even when the application starts, then stop listening for updates when content is ready to be posted or recorded. You might need to consider how long a typical task of creating the content takes and judge if this duration allows for efficient collection of a location estimate.

Helping the user decide on where to go

You might be creating an application that attempts to provide users with a set of options about where to go. For example, you're looking to provide a list of nearby restaurants, stores, and entertainment and the order of recommendations changes depending on the user location.

To accommodate such a flow, you might choose to:

- Rearrange recommendations when a new best estimate is obtained
- Stop listening for updates if the order of recommendations has stabilized

This kind of model is visualized in figure 3.

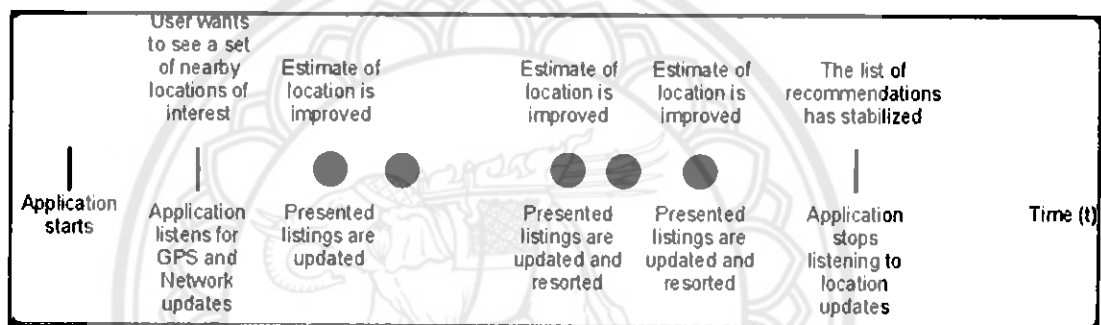


Figure 3. A timeline representing the window in which a dynamic set of data is updated each time the user location updates.

Providing Mock Location Data

As you develop your application, you'll certainly need to test how well your model for obtaining user location works. This is most easily done using a real Android-powered device. If, however, you don't have a device, you can still test your location-based features by mocking location data in the Android emulator. There are three different ways to send your application mock location data: using Eclipse, DDMS, or the "geo" command in the emulator console.