การศึกษาและพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่

โดย
ชาญยุทธ อุปายโกศล
กาญบัญชา พานิชเจริญ

สนับสนุนงบประมาณโดย
มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์
ประจำปีงบประมาณ 2557

EDUCATION AND DEVELOPMENT OF NAVIGATION APPLICATION TO HELP THE BLIND ON MOBILE PHONE

By
Chanyut Aupayagoson
Kanbuncha Panichcharoen

Granted by
Rajamangala University of Technology Rattanakosin
Fiscal year 2014

กิตติกรรมประกาศ

โครงการวิจัยนี้สำเร็จลุล่วงด้วยดี เพราะได้รับการช่วยเหลือจากหลายๆ ท่าน โดยเฉพาะ ดร.รัฐศักดิ์ พรหมมาศ คณบดีคณะวิศวกรรมศาสตร์ และผู้ช่วยศาสตราจารย์ ดร.ไกรฤกษ์ เชยชื่น รองคณบดีฝ่ายวิจัยและบริการวิชาการ มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ ที่ให้ความอนุเคราะห์สนับสนุนการทำวิจัย แด่คณะทีมงานผู้วิจัยด้วยดีเสมอมา ขอขอบพระคุณคณะผู้บริหารและเจ้าหน้าที่ สถาบันวิจัยและพัฒนา ที่คอยช่วยเหลือตรวจทานข้อมูล ให้การสนับสนุนการทำ วิจัย จนงานวิจัย แล้วเสร็จ ขอกราบขอบพระคุณ ศาสตราจารย์ ดร.โกสินทร์ จำนงไทย และผู้ช่วยศาสตราจารย์ ดร.วีรพล จิรจริต ณ ห้องปฏิบัติการ Vision Lab มหาวิทยาลัยเทคโนโลยีพระจอมเกล้า ธนบุรี ที่ประสิทธิประสาทวิชาให้แก่ผู้วิจัย ขอขอบคุณ นายบรรพชา อุปายโกศล และผู้ช่วยศาสตราจารย์ ดร.ศิริชัย รุจิพัฒนพงศ์ มหาวิทยาลัย เทคโนโลยี พระจอมเกล้า พระนครเหนือ ที่ช่วยเหลือด้านข้อมูลการโปรแกรม และคำแนะนำต่างๆ ขอขอบ พระคุณครอบครัวอุปายโกศล รวมทั้งพี่ๆ เพื่อนๆ และน้องๆ ที่ให้คำปรึกษา และคอยเป็นกำลังใจที่ดีมาโดยตลอดเสมอมา

ขอขอบพระคุณมหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ ที่ได้ให้ความไว้วางใจและสนับสนุนทุนการทำวิจัย แด่คณะผู้วิจัยจนวิจัยแล้วเสร็จ

ชาญยุทธ อุปายโกศล และคณะ
มีนาคม 2558

บทคัดย่อ

รหัสโครงการ : A28/2557
ชื่อโครงการ : การศึกษาและพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่
ชื่อนักวิจัย : นายชาญยุทธ อุปายโกศล, นายกาญบัญชา พานิชเจริญ

งานวิจัยนี้เสนอการศึกษาและพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่ โดยได้เล็งเห็นถึงปัญหา การเดินทาง ของผู้พิการทางสายตาไปในสถานที่ต่างๆ ที่ไม่สามารถเดินทางได้สะดวก อาทิเช่นการเดินทางโดยรถโดยสาร การเดินทางไปยังสถานศึกษา และสถานที่อื่นๆ เป็นต้น โ ดยได้ทำการสร้างโปรแกรมประยุกต์เพื่อช่วยนำทาง การนำทางบนระบบปฏิบัติการ แอนดรอยด์ ร่วม กับ ระบบจีพีเอส ที่ประมวลผล บนโทรศัพท์เคลื่อนที่ โดยมีการออกแบบโปรแกรมประยุกต์ให้มีความเหมาะส มกับการใช้งานของผู้พิการทางสายตา ด้วยฟังก์ชันการสั่งการด้วยเสียง จากผู้ใช้งาน การแสดงผลในรูปแบบเสียงเพื่อให้ผู้พิการทางสายตาได้ยินเสียงเมื่อมีการเรียกใช้ฟังก์ชันใดๆในโปรแกรม โดยส่วนของแผนที่อิเล็กทรอนิกส์นั้นมีการทำงานร่วมกับฐานข้อมูลแผนที่บนกูเกิล ซึ่งผู้ใช้งานสามารถเลือกฟังก์ชันแสดงผลการนำทางได้ 2 รูปแบบคือ แบบแผนที่ และแบบ ดาวเทีย ม โดยสามารถทำการ ค้นหาสถานที่ปลายทางที่ต้องการเดินทาง ไปได้ โดยใช้อัลกอลิทึมของไดจ์สตราในการคำนวณเส้นทางที่ใกล้ที่สุด การเดินทางในกรณีมีตัวเลือกเส้นทางหลายตัวเลือก อีกทั้งยังสามารถคำนวณดูปริมาณความหนาแน่นของการจราจร บนท้องถนนได้ จากผลการทดลองพบว่าโปรแกรมระบบนำทาง เพื่อผู้พิการทางสายตา สามารถช่วยเหลือผู้พิการทางสายตาให้เดินทางได้สะดวก มากยิ่งขึ้น แต่อย่างไรก็ตามประสิทธิภาพการทำงานของระบบนำทาง ด้านความเร็วและความแม่นยำใ นการระบุตำแหน่งของสถานที่ จะขึ้นอยู่กับประสิทธิภาพของอุปกรณ์ ประมวลผล (โทรศัพท์เคลื่อนที่) ดังนั้นผู้ใช้งานจึงควรเลือกอุปกรณ์ประมวลผลที่มีประสิทธิภาพในการใช้งานระบบนำทาง

คำสำคัญ : ผู้พิการทางสายตา, โทรศัพท์เคลื่อนที่, โปรแกรมประยุกต์, อัลกอลิทึมของไดจ์สตรา

E-mail Address : chanyut.a@rmutr.ac.th, chanyut7@hotmail.com
ระยะเวลาโครงการ : 1 ตุลาคม 2556 - 30 กันยายน 2557

Abstract

Code of project : A28/2557
Project name : The Development of Online Multiple Choice Test System for Base Knowledge Admissions Measured in Faculty of Engineering.
Researcher name : Mr. Chanyut Aupayagoson, Mr. Kanbuncha Panichcharoen

This paper presents an education and development of navigation application to help blind on mobile phone. The blind cannot conveniently travel, such as education and other places. To created an application program to help the navigation on the operating system of android with GPS processing system on mobile phones. The application is designed to be suitable for used by the blind, a voice recognition function. For the blind, hearing on the run any function in the program. A map is working with the Google Maps database. Users can choose to display the navigation function has two forms: map and satellite. The destination can be reached using Dijkstra Shotest Paht algorithm. It can calculate the density of traffic on the road. The experimental result showed that the navigation system for the blind can help, the visually impaired to travel more convenient. However, the performance of the navigation system is based on performance processing equipment. However, the performance of the navigation speed and accuracy in the identification of the location will depend on the efficiency of the processing device. Therefore, the user should select the device processing efficiency of navigation.

Keywords: Blind, Mobile phone, Application program, Dijkstra algorithm

---

E-mail Address : chanyut.a@rmutr.ac.th, chanyut7@hotmail.com
Period of project : 1 October 2012 - 30 September 2013

สารบัญ

หน้า

กิตติกรรมประกาศ ..... ก
บทคัดย่อ ..... ข
Abstract ..... ค
สารบัญ ..... ง
สารบัญภาพ ..... ฉ
สารบัญตาราง ..... ฌ
บทที่

1 บทนำ

1. ความสำคัญและที่มาของปัญหา 1
2. วัตถุประสงค์ของโครงการวิจัย 2
3. ขอบเขตของการวิจัย 2
4. กรอบแนวคิดการวิจัย 2
5. ประโยชน์ที่คาดว่าจะได้รับ 3
6. นิยามศัพท์ 3

2 ทฤษฎีแนวคิดและงานวิจัยที่เกี่ยวข้อง

1. แอนดรอยด์ (Android) 4
2. กูเกิล แอนดรอยด์ (Google Android) 5
3. สถาปัตยกรรมของแอนดรอยด์ (Android Architecture) 7
4. สมาร์ทโฟน (Smart Phone) 15
5. จีพีเอส (GPS) 17
6. การรู้จำเสียงพูด (Speech Recognition) 20
7. ระบบฐานข้อมูลเอสคิวไลท์ (Database SQLite) 22
8. การหาเส้นทางที่สั้นที่สุด (Short Path) 23
9. บลูทูธ (Bluetooth) 26
10. การทบทวนวรรณกรรม/สารสนเทศ (information) ที่เกี่ยวข้อง 28

3 วิธีดำเนินการวิจัย

1. ศึกษาความต้องการจากผู้ใช้งาน 29
2. การทดสอบและใช้งานอิคลิปส์สำหรับแอนดรอยด์เอสดีเค (Android SDK) 29
3. การออกแบบแอพพลิเคชั่นนำทาง 37
4. การออกแบบและพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตา 63
5. **แอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่** 73

สารบัญ (ต่อ)

บทที่	หน้า
4 ผลการดำเนินการวิจัย	
1. ขั้นตอนการใช้งานแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่	83
2. การทดลองใช้งานแอพพลิเคชั่น	91
3. การทดสอบเสถียรภาพของแอพพลิเคชั่น	98
5 สรุปผล อภิปรายผลและข้อเสนอแนะ	
1. สรุปผลการวิจัย	99
2. อภิปรายผลการวิจัย	99
3. ข้อเสนอแนะแนวทางในการประยุกต์ และพัฒนาต่อยอดงานวิจัย	100
บรรณานุกรม	101
ภาคผนวก	102
ประวัติผู้วิจัย	104

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์

สารบัญภาพ

ภาพที่ **หน้า**

1 กรอบแนวคิดการพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่ 2
2 สัญลักษณ์แอนดรอยด์ (Android) 5
3 สัญลักษณ์กูเกิลแอนดรอยด์ (Google Android) 5
4 สถาปัตยกรรมของแอนดรอยด์ (Android Architecture) 8
5 ตัวอย่างแอพพลิเคชัน (Application) 8
6 ชั้นแอพพลิเคชั่นเฟรมเวิร์ค (Application Framework Layer) 9
7 ไลบรารี (Library) และแอนดรอยด์ รันไทม์ (Android Runtime) 10
8 ชั้นลีนุกซ์เคอร์เนล (Linux Kernel) 11
9 แอนดรอยด์เอสดีเคอีมูเลเตอร์ (Android SDK Emulator) 11
10 วงรอบชีวิตของแอพพลิเคชั่น (Android Activity Life Cycle) 14
11 ตัวอย่างสมาร์ทโฟน (Smart Phone) 15
12 ดาวเทียม NAVSTAR ของสหรัฐ 17
13 องค์ประกอบของระบบดาวเทียมจีเอส (GPS) 18
14 ความแม่นยำระบบรู้จำเสียงพูด (Speech recognition) จากอดีตถึงปัจจุบัน 20
15 หลักการทำงานของการรู้จำเสียงพูด (Speech Recognition) 21
16 สัญลักษณ์ฐานข้อมูลเอสคิวไลท์ (Database SQLite) 22
17 ตัวอย่างกราฟน้ำหนักทั่วไป 24
18 วิธีคิดของไดจ์สตรา (Dijkstra) ที่จะค้นหารูปแบบเส้นทางสั้นที่สุดจาก a ถึง z 25
19 ตัวอย่างการสื่อสารของอุปกรณ์บลูทูธ (Bluetooth) 27
20 ตัวอย่างชุดหูฟังที่ใช้ บลูทูธ (Bluetooth) 28
21 ดาวน์โหลดแอนดรอยด์เอสดีเค (Android SDK Download) 31
22 Add Repository 31
23 การค้นหา Developer Tools 32
24 หน้าต่างการจัดการแอนดรอยด์เอสดีเค (Android SDK Manager) 33
25 การตั้งค่าเอดีทีปลั๊กอิน (Configuring the ADT Plug-in) 33
26 หน้าต่าง Create New AVD 34
27 แพ็คเกจเอ็กซ์โพลเลอร์ (Package Explorer) 36
28 หน้าต่าง Open Perspective 36
29 อีมูเลเตอร์คอนโทรล (Emulator Control) 37
30 การทำงานของข้อมูลแผนที่ของกูเกิลบนโทรศัพท์เคลื่อนที่ 38
31 ตัวอย่าง directory ของ debug.keystore 38
32 ตัวอย่าง Certificate Fingerprint MD5 39

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
33	ตัวอย่างการสร้าง API Key	39
34	ตัวอย่าง API Key เพื่อในการพัฒนาแผนที่กูเกิ้ล (Google Map)	40
35	ตัวอย่างแผนที่กูเกิลบนแอนดรอยด์ (Google Map on Android)	42
36	แผนภาพการเรียกใช้งานข้อมูลแผนที่ของกูเกิล (Google Map)	42
37	หลักการทำงานของตำแหน่งพิกัดจีพีเอสในแผนที่	43
38	ตัวอย่างการกำหนดตำแหน่งสถานที่บนแผนที่กูเกิล	46
39	แผนภาพการทำงานระบบติดตามด้วยจีพีเอส (GPS Tracking)	47
40	ตัวอย่างสภาพการจราจรบนแผนที่กูเกิล (Google Map Traffic Marker)	48
41	แผนภาพการทำงานสภาพการจราจรบนท้องถนนบนแผนที่	49
42	การเปลี่ยนรูปแบบโหมดแผนที่จากโหมดแผนที่ (ซ้าย) เป็นโหมดดาวเทียม (ขวา)	50
43	หลักการทำงานของการดึงข้อมูลผ่าน Google Location-based Service	51
44	การหาค่าพิกัดจีพีเอสผ่านสถานที่	53
45	แผนภาพการทำงานการค้าหาสถานที่ (Location Finder)	54
46	การหาเส้นทาง (Route Path) และการกำหนดสถานที่ (Location Marker)	59
47	แผนภาพการทำงานการค้นหาเส้นทาง (Route Path)	60
48	การตั้งค่าการรักษาความปลอดภัย (Action Security)	61
49	เมนูตัวเลือก (Option Menu)	62
50	หลักการทำงานของการสั่งงานด้วยเสียง (Speech Recognition)	63
51	ชุดคำสั่งเสียงที่ใช้ในแอพพลิเคชั่น	64
52	หน้าการทำงานการสั่งงานด้วยเสียง (Speech Recognition)	65
53	แผนภาพการทำงานการเก็บสถานที่ปลายทางบนฐานข้อมูลเอสคิวไลท์	72
54	แผนภาพการทำงานส่วนติดต่อผู้ใช้งาน	74
55	ส่วนติดต่อผู้ใช้งาน 1	75
56	ส่วนติดต่อผู้ใช้งาน 2	76
57	ส่วนติดต่อผู้ใช้งาน 3	77
58	ส่วนติดต่อผู้ใช้งาน 4	78
59	ส่วนติดต่อผู้ใช้งาน 5	79
60	ส่วนติดต่อผู้ใช้งาน 6	80
61	ส่วนติดต่อผู้ใช้งาน 7	81
62	ส่วนติดต่อผู้ใช้งาน 8	82
63	การใช้งานหน้าแอพพลิเคชั่นหลัก	83
64	หน้าแอพพลิเคชั่นรับคำสั่งเสียง	84
65	เมนูตัวเลือก (Option Menu)	85

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
66	หน้าต่างหลักระบบนำทางผู้พิการทางสายตา (Navigation for Blind)	85
67	การสร้างฐานข้อมูล (Edit Database)	86
68	หน้าต่าง My Location	87
69	เปลี่ยนรูปแบบโหมดแผนที่	88
70	หน้าต่างแสดงสภาพการจราจร (Traffic)	88
71	การค้นหาสถานที่ (Navigation Location Finder)	89
72	การหาเส้นทาง (Route Path)	90
73	การตั้งค่าตัวรับสัญญาณจีพีเอสและเครือข่ายไร้สาย	91
74	การทดลองการแก้ไขข้อมูลสถานที่ปลายทางในฐานข้อมูล SQLite	92
75	การทดลองการนำทางผู้พิการทางสายตาด้วยเสียง	93
76	การทดลองการดูสภาพการจราจรบนท้องถนน ณ บริเวณลานวัฒนธรรม	94
77	การทดลองการดูสภาพการจราจรบนท้องถนน ณ อำเภอพุทธมณฑล	94
78	การทดลองการเลือกการแสดงผลของแผนที่นำทาง	95
79	การทดลองการค้นหาสถานที่ปลายทาง	95
80	การทดลองการนำทางจากพุทธมณฑลสาย 4 – คณะวิศวกรรมศาสตร์	96
81	การทดลองการนำทางจากตลาดศาลายา – อาคารเฉลิมพระเกียรติฯ	96
82	การทดสอบการหาตำแหน่ง ณ ปัจจุบัน	97
83	การทดสอบการกำหนดสถานที่การเดินทางบนแผนที่	97
84	การทดสอบเสถียรภาพของการนำทาง	98

สารบัญตาราง

ตารางที่ **หน้า**

1 ผลการสำรวจของ Nielsen ในอังกฤษ เทียบ Q2/Q3 2009 16
2 เปรียบเทียบข้อดี - ข้อเสียของระบบการรู้จำเสียงพูด (Speech Recognition) 22
3 การเปรียบเทียบโอเพนซอร์ส (Open Source) 23
4 ตัวอย่างข้อมูลในตาราง location2 71

บทที่ 1
บทนำ

1. ความสำคัญและที่มาของปัญหา

จากสำนักงานสถิติแห่งชาติในการสำรวจคนพิการ พ .ศ. 2544 พบว่า มีจำนวนคนตาบอดที่ขึ้นทะเบียนทั้งสิ้น 123,157 คน คิดเป็นร้อยละ 11.2 ของจำนวนคนพิการทั่วประเทศ และคิดเป็นร้อยละ 0.2 ของจำนวนประชากรทั้งประเทศ ทั้งนี้โดยแยกเป็น ชายจำนวน 70,556 คน เป็นหญิง 52,601 คน แต่อย่างไรก็ตามองค์กรภาคเอกชนหลายแห่งมักใช้ตัวเลขคาดประมาณจำนวนผู้พิการทางการเห็นขององค์การอนามัยโลก ซึ่งระบุว่าประเทศกำลังพัฒนาทั้งหลายจะมีจำนวนผู้พิการประเภทนี้ราวร้อยละ 1 ของจำนวนประชากรทั้งประเทศ ดังนั้นประเทศไทยจะมีมีผู้พิการทางการเห็นอยู่ไม่น้อยกว่า 600,000 คน ของประชากรในประเทศ [1]

จากรายงานทางสถิติพบว่ามีผู้พิ การทางสายตาจำนวนไม่น้อยที่รอการ ช่วยเหลือจากภาครัฐและหน่วยงานที่เกี่ยวข้อง โดยปัญหาหลักของคนพิการทางสายตาส่วนใหญ่คือการเดินทางไปในสถานที่ต่างๆนั้นจำเป็ นต้องมีคนดูแลเป็นพิเศษ อาทิเช่นการเดินทางไปทำงาน การเดินทางไปสถานศึกษา และสถานที่อื่นๆเป็นต้น ทำให้เป็นภาระต่อผู้ดูแล ไม่มีความเป็นส่วนตัว อีกทั้งยังเป็นการบั่นทอน ต่อจิตใจของผู้พิการเนื่องจากผู้พิการส่วนใหญ่ต้องการอยู่ได้ในสังคมโดยการพึ่งพาตนเองเสมือนหนึ่งเป็นคนธรรมดาที่สามารถช่วยเหลือตัวเองได้

ระบบการนำทางโดยใช้ จีพีเอส ในปัจจุบันได้มีการใช้งานอย่างแพร่หลาย แต่เมื่อพิจารณาแล้วยังเป็นระบบที่ไม่เหมาะกับการนำมาใช้กับคนพิการทางสายตาทั้งนี้เนื่องจาก ระบบนำทางปัจจุบันนั้นเน้นการนำทางบนท้องถนนเป็นหลัก ซึ่งไม่สา มารถนำทางไปในพื้นที่ที่ไม่มีแผนที่ได้ อีกทั้งผู้พิการทางสายตานั้นๆไม่สามารถมองเห็นปุ่ม หรือภาพบนจอของตัวอุปกรณ์นำทางได้ ทำให้ไม่สามารถใช้งานอุปกร์นำทางด้วยจีพีเอสได้เหมือนคนปกติ ดังนั้นจึงจำเป็นที่ต้องมีการพัฒนาระบบนำทางสำหรับผู้พิการโดยตรง โดยการผ่านตัวอุปกรณ์ด้วยคำสั่งเสียงเป็นหลัก และแผนที่นั้นต้องมีความละเอียดมากพอที่จะให้คนพิการทางสายตานั้นสามารถเดินทางได้อย่างสะดวก และปลอดภัย

ดังนั้นผู้วิจัยจึงได้มีแนวคิดการสร้างระบบนำทาง ด้วยจีพิเอสสำหรับคนพิการบนโทรศัพท์เคลื่อนที่ (Smart Phone) ในระบบปฏิบัติการแอนดรอยด์ (Android OS) เนื่องจากเป็นอุปกรณ์ที่มีขนาดเล็กพกพาได้สะดวก โดยปรับปรุงและพัฒนารูปแบบการสั่งการให้เหมาะสมกับผู้พิการด้วยการใช้คำสั่งเสียง การระบุพิกัดในแผนที่ให้มีความเหมาะสมต่อการใช้งานต่อผู้พิการทางสายตา

2. วัตถุประสงค์ของโครงการวิจัย

2.1 เพื่อออกแบบ และพัฒนาระบบแผนที่ และการจัดเก็บข้อมูลแผนที่บนมือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์

2.2 เพื่อออกแบบระบบการใช้งานฟังก์ชันการสั่งการในตัวอุปกรณ์นำทางให้เหมาะสมกับผู้พิการทางสายตา

2.3 เพื่อพัฒนาระบบการนำทางสำหรับผู้พิการทางสายตา

3. ขอบเขตของการวิจัย

3.1ออกแบบ และพัฒนาระบบแผนที่ และการจัดเก็บ ข้อมูลแผนที่บนมือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์

3.2 สร้างโปรแกรมประยุกต์ สำหรับฟังก์ชันการสั่งการตัวอุปกรณ์ด้วยคำสั่งเสียง

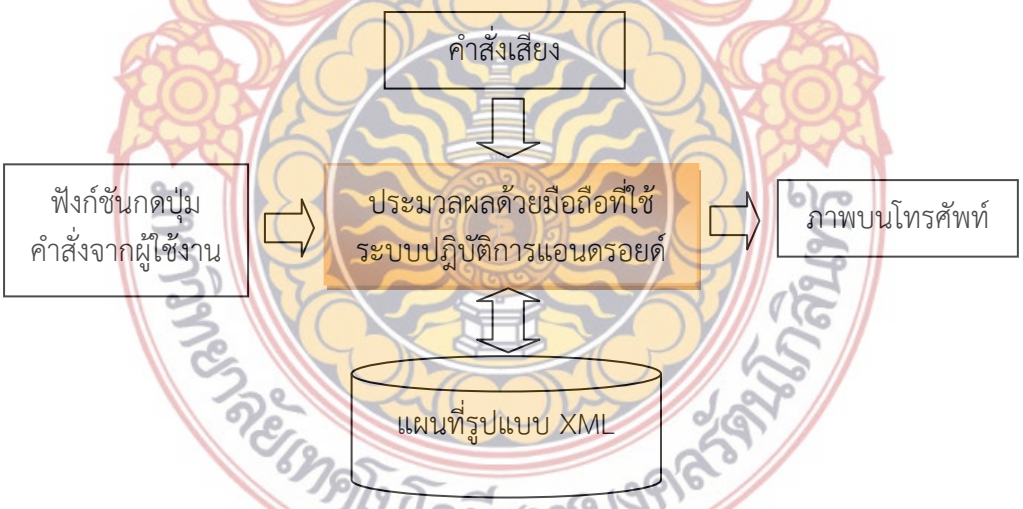
3.3 สร้างโปรแกรมประยุกต์ในการกำหนดแผนที่และการนำทางคนพิการทางสายตา

3.4 สร้างแบบจำลองระบบตามกรณีศึกษา

3.5 ทดสอบระบบตามแบบจำลอง และนำมาติดตั้งใช้งานจริง โดยใช้มหาวิทย าลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ ศาลายา เป็นกรณีศึกษา

4. กรอบแนวคิดการวิจัย

ผู้วิจัยได้มีกรอบแนวคิดในการสร้าง แอพพลิเคชั่นช่วยนำทางเพื่อผู้พิ การทางสายตาบนโทรศัพท์เคลื่อนที่โดยแสดงดังภาพที่ 1



ภาพที่ 1 กรอบแนวคิดการพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่

จากภาพที่ 1 ผู้วิจัยได้มีแนวคิดในการวิ จัยโดยจะนำระบบประมวลผลด้วยมือถือภายใต้การทำงานบนระบบปฏิบัติการ แอนดรอยด์ โดยใช้โทรศัพท์ที่สามารถรับสัญญาณจีพีเอสได้เป็นตัวรับสัญญาณ

เพื่อระบุพิกัดของผู้พิการทางสายตา จากนั้นจึงทำการสร้างแผนในรูปแบบเอ็กเอ็มแอลโดยจะทำการสร้างฟังก์ชันการทำงานให้เหมาะสมกับผู้พิการทางสายตาโดยใช้คำสั่งเสียงในการควบคุมการทำงาน

5. ประโยชน์ที่คาดว่าจะได้รับ

5.1 ช่วยในการแก้ไขปัญหาการเดินทางของผู้พิการทางสายตาได้

5.2 ช่วยให้มีทางเลือกใหม่ในการเดินทางของผู้พิการทางสายตา

5.3 ช่วยเพิ่มความสามารถในเดินทาง และมีสภาพแวดล้อมที่หลากหลายมากขึ้น

5.4 นำผลการวิจัยสู่ประชากรเป้าหมาย

5.5 นำผลการวิจัยสู่การผลิตเชิงพาณิชย์และอุตสาหกรรม ในกรณีมีผู้สนใจ

5.6 เป็นการวิจัยพื้นฐานที่จะนำไปสู่การวิจัยต่อยอดต่อไป

6. นิยามศัพท์

6.1 ผู้พิการทางสายตา หมายถึง บุคคลที่สูญเสียการเห็นตั้งแ ต่ระดับเล็กน้อยจนถึงตาบอดสนิทอาจแบ่งได้ 2 ประเภทดังนี้

6.1.1 คนตาบอด หมายถึง บุคคลที่สูญเสียการเห็นมากจนต้องสอนให้อ่านอักษรเบรลล์หรือใช้วิธีการฟังเทปหรือแถบเสียง หากตรวจวัดความชัดของสายตาข้างดีเมื่อแก้ไขแล้วอยู่ในระดับ 6 ส่วน 60 เมตร (6/60) หรือ 20 ส่วน 200 ฟุต (20/200) ลงมาจนถึงบอดสนิท (หมายถึงคนตาบอดสามารถมองเห็นวัตถุได้ในระยะห่างน้อยกว่า 6 เมตร หรือ 20 ฟุต ในขณะที่คนปกติสามารถมองเห็นวัตถุเดียวกัน ได้ในระยะห่าง 60 เมตร หรือ 200 ฟุต) หรือมีลาน สายตาแคบกว่า 20 องศา (หมายถึงมองเห็นได้กว้างน้อยกว่า 20 องศา)

6.1.2 คนเห็นเลือนร าง หมายถึง บุคคลที่สูญเสียการเห็นแต่ยังสามา รถอ่านอักษรตัวพิมพ์ ที่ขยายใหญ่ได้ หรือต้องใช้แว่นขยายอ่าน หากตรวจวัดความชัดเจนของสายตาข้างดีเมื่อแก้ไขแล้วอยู่ในระดับระหว่าง 6 ส่วน 18 เมตร (6/18) หรือ 20 ส่วน 70 ฟุต (20/70) ถึง 6 ส่วน 60 เมตร (6/60) หรือ 20 ส่วน 200 ฟุต (20/200) หรือมีลานสายตาแคบกว่า 30 องศา

6.2 XML ย่อมาจาก “Extensible Markup Language” มีต้นแบบมาจาก SGMLเป็นภาษา Markup Language เช่นเดียวกับ HTML Markup ประกอบด้วยโค้ดหรือที่เรียกว่า แท็ก (Tag) โดย XML เป็นภาษาประเภท Meta language ภาษาหนึ่งเช่นเดียวกับ SGML จึงสามารถนำXMLไปสร้างภาษาอื่นๆ หรือคำศัพท์ (Vocabulary) ต่างๆได้

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์

บทที่ 2
ทบทวนวรรณกรรมที่เกี่ยวข้อง / ทฤษฎีที่เกี่ยวข้อง

ในการทำงานวิจัยเรื่อง การศึกษาและพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่ผู้วิจัยได้ศึกษาทฤษฎี และงานวิจัยที่เกี่ยวข้องดังนี้

1. แอนดรอยด์ (Android)
2. กูเกิล แอนดรอยด์ (Google Android)
3. สถาปัตยกรรมของแอนดรอยด์ (Android Architecture)
4. สมาร์ทโฟน (Smart Phone)
5. จีพีเอส (GPS)
6. การรู้จำเสียงพูด (Speech Recognition)
7. ระบบฐานข้อมูลเอสคิวไลท์ (Database SQLite)
8. การหาเส้นทางที่สั้นที่สุด (Short Path)
9. บลูทูธ (Bluetooth)
10. การทบทวนวรรณกรรม/สารสนเทศ (information) ที่เกี่ยวข้อง

1. แอนดรอยด์ (Android) [2]

แอนดรอยด์ (Android) เป็นระบบปฏิบัติ (Operation System) การสำหรับอุปกรณ์พกพา เช่นโทรศัพท์มือถือแท็บเล็ตคอมพิวเตอร์ เน็ตบุ๊ก ทำงานบนลินุกซ์ เคอร์เนล เริ่มพัฒนาโดยบริษัทแอนดรอยด์จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดยกูเกิล และนำแอนดรอยด์ไปพัฒนาต่อ ภายหลังถูกพัฒนาในนามของ โอเพ่นแฮนด์เซตอัลไลแอนซ์ (Open Handset Alliance) ทางกูเกิล (Google) ได้เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่างๆด้วยภาษาจาวา (JAVA) และควบคุมอุปกรณ์ผ่านทางชุดจาวาไลบรารี (Java libraries) ที่กูเกิลพัฒนาขึ้น

แอนดรอยด์ได้เป็นที่รู้จักต่อสาธารณชนเมื่อวันที่ 5 พฤศจิกายน พ.ศ. 2550 โดยทางกูเกิลได้ประกาศก่อตั้ง โอเพ่นแฮนด์เซตอัลไลแอนซ์ (Open Handset Alliance) กลุ่มบริษัทฮาร์ดแวร์, ซอฟต์แวร์ ที่ร่วมมือกันเพื่อพัฒนา มาตรฐานเปิด สำหรับอุปกรณ์มือถือลิขสิทธิ์ของโค้ดแอนดรอยด์นี้จะใช้ในลักษณะของซอฟต์แวร์เสรี

โทรศัพท์เครื่องแรกที่สามารถใช้งานระบบปฏิบัติการแอนดรอยด์ (Android Operating System) ได้คือเอชทีซี ดรีม (HTC Dream) ออกจำหน่ายเมื่อ 22 ตุลาคม 2551 เวอร์ชันล่าสุด ของแอนดรอยด์คือ 4.2 เจลลี่บีน (Jelly Bean) ความสามารถใหม่ของ แอนดรอยด์ 4.2 ที่เพิ่มขึ้นมาคือ โฟโต้สเปียร์ (Photo Sphere) ที่สามารถถ่ายรูปได้ 360 องศา และ คีย์บอร์ดเกสเตอร์ (Keyboard - Gestures) ที่สามารถลากนิ้วแทนการสัมผัสตัวอักษรได้

ภาพที่ 2 สัญลักษณ์แอนดรอยด์ (Android)

2. กูเกิล แอนดรอยด์ (Google Android) [3]

ภาพที่ 3 สัญลักษณ์กูเกิลแอนดรอยด์ (Google Android)

กูเกิลแอนดรอยด์ (Google Android) คือ ระบบปฏิบัติการที่เป็นซอฟต์แวร์แพลตฟอร์มบนมือถือสร้างขึ้นมาจากระบบปฏิบัติการลีนุกซ์ (Linux Operating System) พัฒนาขึ้นมาโดยกูเกิลแอนดรอยด์ (Google Android) นั้นได้เปิดให้นักพัฒนาสามารถเข้ามาจัดการเขียนโค้ดต่างๆ ได้ด้วยภาษาจาวา (JAVA) และเขียนควบคุมอุปกรณ์ต่างๆ ผ่านทางจาวาไลบลารี่ที่ทางกูเกิลพัฒนาขึ้นโดยเฉพาะ (Google Developed Java Libraries) โปรแกรมต่างๆที่รันบนกูเกิลแอน ดรอยด์ (Google Android สามารถเขียนได้ด้วยภาษาซี (C Language) และภาษาอื่นส่วนการพัฒนาผ่านการคอมไพล์ (Compiler) ด้วยสถาปัตยกรรมแบบ อามเนทีฟโค้ด (ARM Native Code (32bit)) นั้นยังไม่ได้รับการสนับสนุนจา กทางกูเกิลลิขสิทธิ์ของกูเกิลแอนดรอยด์นั้นจะอยู่ในลักษณะของฟรีซอฟต์แวร์และ โอเพนซอร์ส (open source) โดยอยู่ภายใต้สิทธิบัตรของ ครีเอทีฟคอมมอนส์แอททริบิว 2.5 (Creative Commons Attribute) ซึ่งทำให้ผู้ใช้นั้นสามารถดาวน์โหลดซอฟต์แวร์ของกูเกิล แอนดรอยด์ไปใช้ได้ฟรี และยังสามารถนำซอฟต์แวร์ที่ได้ไปแชร์แจกต่อได้แต่ไม่อนุญาตให้แก้ไขโด ยการนำเอาชื่อผู้เขียนซอฟต์แวร์ หรือรายการสิทธิบัตรของซอฟต์แวร์นั้นออกตัวโปรแกรมแอนดรอยด์ (Android) คือ ระบบปฏิบัติการ (Operation System) หรือแพลตฟอร์ม (Platform) ที่จะใช้ควบคุมการทำงานบนอุปกรณ์ อิเล็กทรอนิกส์ต่างๆ สำหรับโทรศัพท์มือถือและอุปกรณ์พกพา โดยมี กูเกิลอิงก์, ที-โมบาย, เอชทีซี, ควอลคอมม์, โมโตโรลา และบริษัทชั้นนำอีกมากมายร่วมพัฒนาโปรเจคแอนดรอยด์ ผ่านกลุ่มพันธมิตรเครื่องมือสื่อสารระบบเปิ ด (Open Handset- Alliance) ซึ่งเป็นกลุ่มพันธมิตรชั้นนำระดับนานาชาติด้านเทคโนโลยีและเครื่องมือสื่อสารเคลื่อนที่ ซึ่ง แอนดรอยด์ (Android) ประกอบด้วยระบบปฏิบัติการ (Operation System) ไลบรารี (Library)

เฟรมเวิร์ค (Framework) และซอฟต์แวร์อื่นๆ ที่จำเป็นในการพัฒนา ซึ่งเทียบเท่ากับ Windows Mobile, Palm OS, Symbian, OpenMoko และMaemo ของโนเกียโดยใช้องค์ประกอบที่เป็นโอเพนซอร์สหลายอย่างเช่น Linux Kernel, SSL, OpenGL, Free Type, SQLite, WebKit และเขียนไลบรารี เฟรมเวิร์ค ของตัวเองเพิ่มเติม ซึ่งทั้งหมดจะโ อเพนซอร์ส ใช้ อาปาเช่ ไลเซ่นส์ (Apache License)

ความร่วมมือครั้งนี้มีเป้าหมายในการส่งเสริมนวัตกรรมบนเครื่องมือสื่อสารเพื่อให้ได้รับประ สบการณ์ที่เหนือกว่าแพลตฟอร์มโมบายทั่วไปที่มีอยู่ในปัจจุบันทั้งนี้การนำเสนอมิติใหม่ ของแพลต ฟอร์มระบบเปิดให้แก่นักพัฒนาจะทำช่วยให้กลุ่มคนเหล่านี้ทำงานร่วมกันได้มีประสิทธิภาพยิ่งขึ้นโดยแอนดรอยด์ (Android) จะช่วยเร่งและผลักดันบริการระบบสื่อสารรูปแบบใหม่ไปสู่ผู้บริโภคได้อย่างที่ไม่เคยเกิดขึ้นมาก่อน

2.1 คุณสมบัติและความสามารถของแอนดรอยด์ [1], [4]

คุณสมบัติและความสามารถหลักของระบบแอนดรอยด์มีดังต่อไปนี้

2.1.1 การเชื่อมต่อ เทคโนโลยีของแอนดรอยด์นั้นมีความสามารถในการเชื่อมต่อที่ประกอบด้วย GSM/EDGE, CDMA, Bluetooth, Wi-Fi, NFC และ WiMAX

2.1.2 การรับส่งข้อความ (Messaging) สนับสนุกการทำงานของ SMS, MMS

2.1.3 ระบบฐานข้อมูล แอนดรอยด์นั้นมีเอสคิวไลท์ (SQLite) ในการเก็บข้อมูล (Data)

2.1.4 เว็บบราวเซอร์ (Web Browser) แอนดรอยด์นั้นจะมีกูเกิลโครม (Google - Chrome) ถูกติดตั้งมาให้พร้อมใช้งานทันที

2.1.5 มีเดีย (Media) แอนดรอยด์นั้นสนับสนุนไฟล์วีดีโอ เสียง และรูปภาพในรูปแบบของ MPEG4, H.264, MP3, AAC, JPG และ PNG

2.1.6 สตรีมมิง (Streaming) แอนดรอยด์สนับสนุน RTP/RTSP และ HTML Progressive Download

2.1.7 มัลติทัช (Multi - touch) แอนดรอยด์นั้นรองรับการสั่ง ที่หน้าจอได้มากกว่าการสัมผัส 1 จุด

2.1.8 การสนับสนุนฮาร์ดแวร์อื่นๆ ในแอนดรอยด์นั้นสามารถรองรับการทำงานเช่น กล้องถ่ายรูป, จีพีเอส (GPS), เข็มทิศ, การวัดอัตราความเร่งและเทอร์โมมิเตอร์ เป็นต้น

2.2 ข้อดีของแอนดรอยด์

2.2.1 แอนดรอยน์เป็นโปรแกรมเสรี บริษัทมือถือสามารถนำไปใช้กับโทรศัพท์ของตัวเองได้ฟรี พัฒนาต่อยอดได้ ทำให้โทรศัพท์มีราคาต่อคุณภาพคุ่มค่า เกิดความหลากหลาย

2.2.2 แอนดรอยด์มีชุดพัฒนาแอพพลิเคชันให้ใช้ฟรี หมายความว่าเราสามารถเขียนแอพพลิเคชันขึ้นมาเพื่อใช้งานเอง หรือเพื่อการค้า

2.2.3 มีตลาด (Market) ให้คุณโหลดแอพพลิเคชันฟรี

2.2.4 การทำงานบนพื้นฐานของลินุกซ์ (LINUX) แอนดรอยด์จึงมีประสิทธิภาพที่ยอดเยี่ยมในการเชื่อมต่อกับ ดาวเทียม กล้อง และอินเทอร์เน็ต สิ่งเหล่านี้คือจุดประสงค์ของแอนดรอยด์

2.2.5 มีความปลอดภัย ความเสถียรภาพสูง

2.2.6 แอนดรอยด์ใช้งานสะดวกเพราะมีบริการต่างๆของกูเกิลติดมากับแอนดรอยด์

2.2.7 แอนดรอยด์อนุญาตให้ผู้ใช้งานสามารถ อัพเดต (Update) ตัวระบบปฏิบัติการ (Operation System) ได้เองไม่ต้องรอจากทางผู้ผลิตมือถือหากมีปัญหาสามารถหาคนช่วยเหลือได้มาก

2.3 ข้อเสียของแอนดรอยด์

2.3.1 การประมวลผล (Process) ไม่สามารถปิด Process เองได้ ถ้าเปิดโปรแกรมอะไรขึ้นมา กล่าวคือ จะรันอยู่อย่างนั้นตลอดซึ่งจะทำให้เครื่องช้าลงเรื่อยๆ ต้องมาลงโปรแกรมที่จัดการทำงาน (Task Manager) คอยปิดการประมวลผล (Process) ทำให้ยุ่งยากมากขึ้น

2.3.2 เมื่อเทียบกับวินโดวส์โมบายล์ (Window Mobile) ในแง่ความแพร่หลายของโปรแกรม, การใช้งานจีพีเอส (GPS) และการใช้งานร่วมกับคอมพิวเตอร์ที่เป็นวินโดว์ (Windows) แอนดรอยด์ (Android) ยังสู้ไม่ได้อย่างแน่นอน อีกทั้งการใช้งานร่วมกับภาษาไทยยังไม่รู้ว่าจะทำได้ดีขนาดไหน

2.3.3 ใช้งานยากเพราะเมนูซับซ้อน ต้องทำความเข้าใจก่อน

2.3.4 ต้องต่ออินเตอร์เน็ตตลอดเวลาจึงจะใช้ฟังก์ชันได้เต็มที่

2.4 ข้อเด่นของแอนดรอยด์

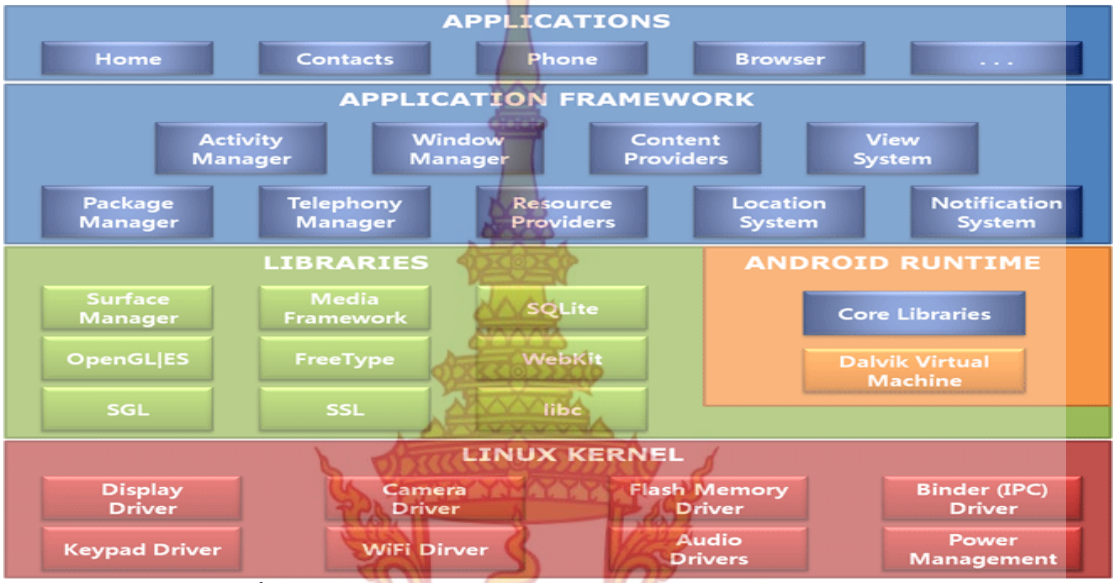
เนื่องจากระบบปฏิบัติ การแอนดรอยด์ (Android Operating System)มีการเจริญเติบโตอย่างรวดเร็ว และมีส่วนแบ่งตลาดของอุปกรณ์ด้านนี้ขึ้นทุกขณะทำให้กลุ่มผู้ใช้งาน และกลุ่มนักพัฒนาโปรแกรม (Programmer) ให้ความสำคัญกับระบบปฏิบัติการแอนดรอยด์เพิ่มมากขึ้น เมื่อมองในด้านของกลุ่มผลิตภัณฑ์บริษัทที่มีการพัฒนาผลิตภัณฑ์รุ่นใหม่มีการนำเอาระบบปฏิบัติการแอนดรอยด์ไปใช้ในสินค้าของตนเอง พร้อมทั้งยังมีการปรับแต่งให้ระบบปฏิบัติการมีความสามารถการจัดวางโปรแกรมและลู กเล่นใหม่ๆ ที่แตกต่างจากคู่แข่งในท้องตลาด โดยเฉพาะอย่างยิ่งกลุ่มสินค้าที่เป็น มือถือรุ่นใหม่ (Smart Phone) และอุปกรณ์จอสัมผัส (Touch Screen) โดยมีคุณลักษณะแตกต่างกันไป เช่นขนาดหน้าจอ ระบบโทรศัพท์ ความเร็วของหน่วยประมวลผล ปริมาณหน่วยความจำแม้กระทั่งอุปกรณ์ตรวจจับต่างๆ (Sensor)

หากมองในด้านของการพัฒนาโปรแกรมทางบริษัทกูเกิ้ลได้มีการพัฒนาแอพพลิเคชันเฟรมเวิร์ค (Application Framework) ไว้สำหรับนักพัฒนาใช้งานได้อย่างสะดวกและไม่เกิดปัญหาเมื่อนำชุดโปรแกรมที่พัฒนาขึ้นมาไปใช้กับอุป กรณ์ที่มีคุณลักษณะต่างกัน เช่นขนาดจออุปกรณ์ ไม่เท่ากันก็ยังสามารถใช้งานโปรแกรมได้เหมือนกัน เป็นต้น

3. สถาปัตยกรรมของแอนดรอยด์ (Android Architecture) [3], [4]

แอนดรอยด์เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อนหรือแบบสแต็ก (Stack) ซึ่งรวมระบบปฏิบัติการ (Operating System), มิดเดิลแวร์ (Middleware) และแอพพลิเคชั่นที่สำคัญเข้าไว้ด้วยกันเพื่อใช้สำหรับทำงานบนอุปกรณ์พกพาเคลื่อนที่ (Mobile Devices) เช่นโทรศัพท์มือถือ การทำงานของแอนดรอยด์มีพื้นฐานอยู่บนระบบลีนุกซ์ เคอร์เนล (Linux Kernel) ซึ่งใช้ Android SDK (Software Development Kit) เป็นเครื่องมือสำหรับการพัฒนาแอพพลิเคชั่น

บนระบบปฏิบัติการ แอนดรอยด์ (Android) และใช้ภาษา จาวา (Java) ในการพัฒนา โดยสถาปัตยกรรมของแอนดรอยด์ (Android Architecture) นั้นถูกแบ่งออกเป็นลำดับชั้น ออกเป็น 4 ชั้นหลักดังภาพที่ 4



ภาพที่ 4 สถาปัตยกรรมของแอนดรอยด์ (Android Architecture)

3.1 ชั้นแอพพลิเคชั่น (Application Layer)

ชั้นนี้จะเป็นชั้นที่อยู่บนสุดของโครงสร้างสถาปัตยกรรมแอนดรอยด์ (Android) ซึ่งเป็นส่วนของแอพพลิเคชั่นที่พัฒนาขึ้นมาใช้งาน เช่น แอพพลิเคชั่นรับ / ส่งอีเมล์ (E-mail) ข้อความ (SMS) ปฏิทิน แผนที่เว็บเบราเซอร์ รายชื่อผู้ติดต่อเป็นต้น ซึ่งแอพพลิเคชั่นจะอยู่ในรูปแบบของไฟล์ .apk โดยทั่วไปแล้วจะอยู่ในไดเร็คทอรี data/app ดังภาพที่ 5

ภาพที่ 5 ตัวอย่างแอพพลิเคชัน (Application)

3.2 ชั้นแอพพลิเคชั่นเฟรมเวิร์ค (Application Framework)

ชั้นนี้จะอนุญาตให้นักพัฒนาสามารถเข้าเรียกใช้งาน โดยผ่าน API (Application - Programming Interface) ซึ่งแอนดรอยด์ (Android) ได้ออกแบบไว้เพื่อลดความซ้ำซ้อนในการใช้งาน ส่วนประกอบของ แอพพลิเคชัน (Application Component) โดยในชั้นนี้ประกอบด้วย แอพพลิเคชั่นเฟรมเวิร์คดังนี้

3.2.1 วิวซิสเต็มส์ (View System) เป็นส่วนที่ใช้ในการควบคุมการทำงานสำหรับการสร้างแอพพลิเคชั่น เช่น lists, grids, text boxes, buttons และ embeddable web browser

3.2.2 โลเคชั่น เมเนเจอร์ (Location Manager) เป็นส่วนที่จัดการเกี่ยวกับค่าตำแหน่งของเครื่องอุปกรณ์พกพาเคลื่อนที่

3.2.3 คอนเทนต์โพรไวเดอร์ (Content Provider) เป็นส่วนที่ใช้ควบคุมการเข้าถึงข้อมูลที่มีการใช้งานร่วมกัน (Share data) ระหว่างแอพพลิเคชั่นที่แตกต่างกัน เช่น ข้อมูลผู้ติดต่อ

3.2.4 รีซอร์สเมเนเจอร์ (Resource Manager) เป็นส่วนที่จัดการข้อมูลต่างๆที่ไม่ใช่ส่วนของโค้ดโปรแกรม เช่น รูปภาพ, localized strings, layout ซึ่งจะอยู่ในไดเร็คทอรี res/

3.2.5 โนติฟิเคชันเมเนเจอร์ (Notification Manager) เป็นส่วนที่ควบคุมอีเวนต์ (Event) ต่างๆ ที่แสดงบนแถบสถานะ (Status Bar) เช่น ในกรณีที่ได้รับข้อความหรือสายที่ไม่ได้รับและการแจ้งเตือนอื่นๆ เป็นต้นแอคทิวิตี้เมเนเจอร์ (Activity Manager) เป็นส่วนควบคุมวงจรชีวิต (Life Cycle) ของแอพพลิเคชั่น โดยชั้นแอพพลิเคชั่นเฟรมเวิร์ค (Application Framework Layer) แสดงได้ดังภาพที่ 6

ภาพที่ 6 ชั้นแอพพลิเคชั่นเฟรมเวิร์ค (Application Framework Layer)

3.3 ชั้นไลบรารี (Library Layer)

แอนดรอยด์ (Android) ได้รวบรวมกลุ่มของไลบรารี (Library) ต่างๆที่สำคัญและมีความจำเป็นเอาไว้มากมาย เพื่ออำนวยความสะดวกให้กับนักพัฒนาและง่ายต่อก การพัฒนาโปรแกรม โดยตัวอย่างของไลบรารีที่สำคัญเช่น

- System C library เป็นกลุ่มของไลบรารี มาตรฐานที่อยู่บนพื้นฐานของภาษา ซี (C Language) ไลบรารี (Library) สำหรับระบบสมองกลฝังตัว (Embedded System) มีพื้นฐานมาจากลินุกซ์ (Linux)

- Media Libraries เป็นกลุ่มการทำงานมัลติมีเดีย (Multimedia) เช่น MPEG4, H.264, MP3, AAC, AMR, JPEG, และ PNG

- เซอร์เฟสเมเนเจอร์ (Surface Manager) เป็นกลุ่มการจัดการรูปแบบหน้าจอ การวาดหน้าจอ

- 2D/3D library เป็นกลุ่มของกราฟิกแบบ 2 มิติ หรือ SGL (Scalable - Graphics Library) และแบบ 3 มิติ หรือ OpenGL

- Free Type เป็นกลุ่มของบิตแมป (Bitmap) และเวคเตอร์ (Vector) สำหรับการเรนเดอร์ (Render) ภาพ

- เอสคิวไลท์ (SQLite) เป็นกลุ่มของฐานข้อมูล (Database) โดยนักพัฒนาสามารถใช้ฐานข้อมูลนี้เก็บข้อมูลแอพพลิเคชั่นต่างๆ ได้

- บราวเซอร์เอนจิน (Browser Engine) เป็นกลุ่มของการแสดงผลบนเว็บเบราเซอร์ (Web Browse) โดยอยู่บนพื้นฐานของ เว็บคิท (Web Kit) ซึ่งมีลักษณะคล้ายกับ กูเกิลโครม (Google Chrome)

3.3.1 แอนดรอยด์ รันไทม์ (Android Runtime) เป็นชั้นย่อยที่อยู่ในชั้นไลบรารี ซึ่งประกอบด้วย 2 ส่วนหลักคือ

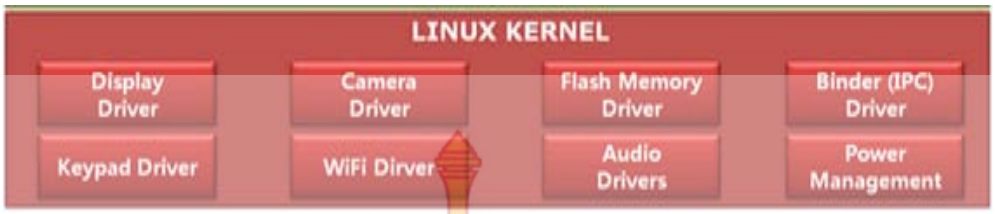
- Dalvik VM (Virtual Machine) ส่วนนี้ถูกเขียนด้วยภาษา จาวา (Java) เพื่อใช้เฉพาะการใช้งานในอุปกรณ์เคลื่อนที่ Dalvik VM จะแตกต่างจาก Java VM (Virtual Machine) คือ Dalvik VM จะรันไฟล์ .dex ที่คอมไพล์มาจากไฟล์ .class และ .jar โดยมี tool ที่ชื่อว่า dx ทำหน้าที่ในการบีบอัดคลาส (Class) จาวา (Java) ทั้งนี้ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์เคลื่อนที่มากกว่า .class เพื่อต้องการใช้พลังงานจากแบตเตอรี่ อย่างมีประสิทธิภาพสูงสุด

- Core Java Library ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็มีความแตกต่างจากไลบรารีของ Java SE (Java Standard Edition) และ Java ME (Java Mobile Edition)

ภาพที่ 7 ไลบรารี (Library) และแอนดรอยด์ รันไทม์ (Android Runtime)

3.3.2 ชั้นลีนุกซ์เคอร์เนล (Linux Kernel)

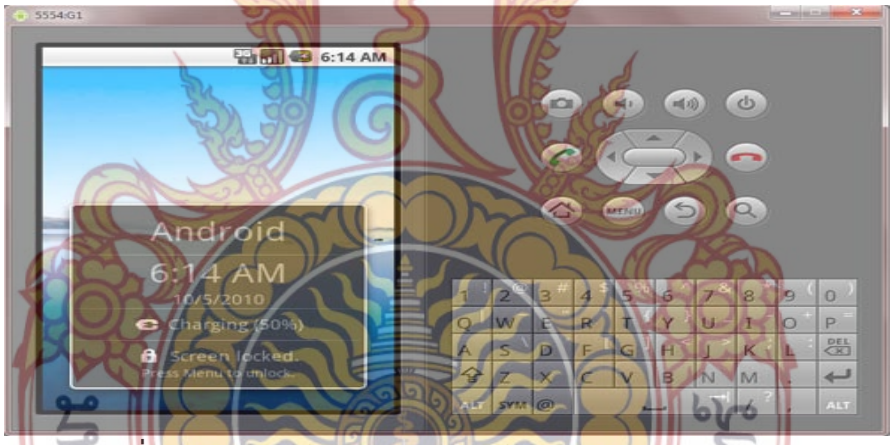
ระบบแอนดรอยด์ (Android) นั้นถูกสร้างบนพื้นฐานของระบบปฏิบัติการ ลินุกซ์ (Linux) โดยในชั้นนี้จะมีฟังก์ชันการทำงานหลาย ๆส่วน แต่โดยส่วนมากแล้วจะเกี่ยวข้องกับฮาร์ดแวร์ (Hard Ware) โดยตรงเช่นการจัดการหน่วยความจำ (Memory Management) การจัดการโพรเซส (Process Management) การเชื่อมต่อเครือข่าย (Networking) เป็นต้น



ภาพที่ 8 ชั้นลีนุกซ์เคอร์เนล (Linux Kernel)

3.4 แอนดรอยด์เอสดีเค (Android SDK)

แอนดรอยด์เอสดีเค (Android SDK) ย่อมาจาก Android Software - Development Kit ซึ่งเป็นชุดโปรแกรมที่ทางกูเกิล (Google) พัฒนาออกมาเพื่อแจกจ่ายให้นักพัฒนาแอพพลิเคชั่น (Application) ทั่วไปดาวน์โหลดไปใช้กันโดยไม่มีค่าใช้จ่าย ซึ่งก็เป็นหนึ่งในปัจจัยที่ทำให้แอพพลิเคชั่นบนแอนดรอยด์นั้นเพิ่มขึ้นอย่างรวดเร็ว ซึ่งในชุดเอสดีเค (SDK) นั้นจะมีโปรแกรมและไลบรารี่ที่ จำเป็นต่อการพัฒนาแอพพลิเคชั่นบนแอนดรอยด์ เช่น อีมูเลเตอร์ (Emulator) ซึ่งทำให้ผู้ใช้สามารถสร้างแอพพลิเคชั่น และนำมาทดลองรันบน อีมูเลเตอร์ (Emulator) ก่อน โดยมีสภาวะแวดล้อมเหมือนมือถือที่รันระบบปฏิบัติการแอนดรอยด์ (Android Operating System)



ภาพที่ 9 แอนดรอยด์เอสดีเคอีมูเลเตอร์ (Android SDK Emulator)

3.5 ส่วนประกอบของแอพพลิเคชัน (Application Component)

คุณลักษณะอย่างหนึ่งของแอนดรอยด์ (Android) คือเป็นแอพพลิเคชั่น (Application) ที่สามารถใช้เป็นส่วนประกอบของแอพพลิเคชั่นอื่นๆ ได้ ตัวอย่างเช่น ถ้าต้องการสร้างแอพพลิเคชั่น (Application) ให้แสดงการเลื่อนของรายการรูปภาพ โดยที่อาจมีแอพพลิเคชั่น(Application) ส่วนอื่นที่ได้พัฒนาไว้แล้ว สามารถเรียกใช้แอพพลิเคชั่น (Application) ในส่วนที่มีอยู่มาพัฒนาต่อได้โดยที่ไม่จำเป็นต้องพัฒนาขึ้นเอง เพื่อความสะดวกและรวดเร็วในการพัฒนา แอพพลิเคชั่น (Application) ซึ่งเรียกแอพพลิเคชั่นเหล่านี้ว่า แอพพลิเคชั่นคอมโพเนนท์ (Application Component) ซึ่งสามารถแบ่งออกเป็น 4 ส่วนคือ Activity, Service, Content Provider และ Broadcast and Intent Receiver

3.5.1 แอคทิวิตี้ (Activity) คือหน้าจอที่ติดต่อกับผู้ใช้ (Graphic User - Interface) ทั้งนี้ในแต่ละแอพพลิเคชั่นอาจมีได้มากกว่า 1 หน้าจอ หรือ แอคทิวิตี้ (Activity) เดียวซึ่งแต่ละแอคทิวิตี้ (Activity) จะทำหน้าที่เก็บสถานะการใช้งานส่วนต่างๆ ตัวอย่าง ในการแสดงรายการเมนู สามารถเลือกให้รายการเมนูที่แสดงออ กมามีภาพและคำบรรยายใต้ภาพได้ สำหรับแอพพลิเคชั่นส่งข้อความอาจมี แอคทิวิตี้ (Activity) หนึ่งจะเป็นส่วนของการเลือกการติดต่อ และแอคทิวิตี้ (Activity) อื่นๆจะทำหน้าที่ดูข้อความเก่าที่ถูกส่งมาแล้ว เป็นต้น

3.5.2 เซอร์วิสโพรไวเดอร์ (Service Provider) เป็นส่วนที่ไม่มีการแสดงผล แต่ถูกเรียกใช้ให้รันอยู่ในลักษณะของ back ground process โดยเซอร์วิส (Service) นั้นอาจจะมีการกระทำอะไรบางอย่าง เช่นติดต่อรับส่งข้อมูลผ่านเครือข่าย หรือคำนวณค่าต่ างๆแล้วทำการส่งข้อมูลไปแสดงยัง แอคทิวิตี้ (Activity) ก็ได้หรือการเปิดเพลงในขณะที่เรากำลังทำงานบนแอพพลิเคชั่นอื่น

3.5.3 บอร์ดแคส รีซีฟเวอร์ (Broadcast receiver) และอินเทนท์รีซีฟเวอร์ (Intent - Receiver) หรือ ดาต้าโพรไวเดอร์ (Data Provider) คือ ตัวที่ใช้สำหรับคอยรับและตอบสนองต่อเหตุการณ์ต่างๆที่เกิดขึ้น เช่น เมื่อแบตเตอรี่ต่ำ, การเปลี่ยนภาษา, มีการโทรออก, มีข้อความเข้าและอื่นๆ ถึงแม้ บอร์ดแคส รีซีฟเวอร์ (Broadcast receiver) จะไม่มีส่วนของการแสดงผลแต่สามารถที่จะเรียก แอคทิวิตี้ (Activity) ขึ้นมาแสดงผลให้ผู้ใช้ทราบได้ เรียกว่า โนติฟิเคชัน เมเนเจอร์ (Notification Manager) ซึ่งจะเป็นตัวที่แจ้งเตือนในรูปแบบต่างๆ เช่น การสั่น, การแสดงไฟกระพริบที่หน้าจอ หรือการส่งเสียงออกมาโดยจะมีไอคอน (Icon) แสดงอยู่บน สเตตัสบาร์ (Status Bar) เพื่อแจ้งเหตุการณ์ที่เกิดขึ้น นอกจากนี้ อินเทนท์รีซีฟเวอร์ (Intent Receiver) คือเป็นส่วนทำให้แอพพลิเคชั่นอื่นๆ เข้าถึงการทำงานของ แอคทิวิตี้ (Activity) และเซอร์วิส (Service) ซึ่งในการปฏิบัติงานแต่ละอย่างเป็นการตอบสนองการร้องขอจากข้อมูลหรือบริการของ แอค ทิวิตี้ (Activity) อื่นๆ

3.5.4 Content provider (System Event Listener) คือส่วนของการบริการข้อมูลสำหรับแต่ละแอพพลิเคชั่ น ทั้งนี้ข้อมูลสาม ารถเก็บอยู่ในรูปแบบของระบบไฟล์ หรือฐานข้อมูล เช่น Google สามารถเข้าใช้งานข้อมูลร่วมกันกับผู้ใช้งานได้ในแอพ พลิเคชั่นที่ต้องการข้อมูลของผู้ใช้งานโดยปกติแล้วในแต่ละแอพพลิเคชั่นจะทำงานแยกกันในแต่ละโพรเซส (Process) ซึ่งมีการทำงานที่แยกจากกันโดยชัดเจน ในกรณีที่นักพัฒนาต้องการเข้าถึงการทำงาน หรือขอสิทธิ์เข้าใช้นั้นนักพัฒนาจะต้องกำหนดการขอเข้าใช้ที่ไฟล์ AndroidManifest.xml เช่น

- INTERNET เป็นการอนุญาตให้เข้าใช้อินเตอร์เน็ต
- READ_CONTACTS เป็นการอนุญาตให้อ่านข้อมูลผู้ใช้
- WRITE_CONTACTS เป็นการอนุญาตให้เขียนข้อมูลผู้ใช้
- RECEIVE_SMS เป็นการอนุญาตให้ตรวจสอบการรับSMS
- ACCESS_COARSE_LOCATION เป็นการอนุญาตให้ใช้การอ้างอิงตำแหน่งแบบคร่าวๆ (Coarse)
- ACCESS_FINE_LOCATION เป็นการอนุญาตให้ใช้การอ้างอิงตำแหน่งที่มีความถูกต้องสูง (Fine) เช่น จีพีเอส (GPS)

3.6 วงรอบชีวิตของแอพพลิเคชั่น (Android Activity Life Cycle)

โดยปกติ แล้วแอพพลิเคชั่นจะทำงานแยกกันในแต่ละโพรเซส และในแต่ละโพรเซสอาจจะมีแอคทิวิตี (Activity) / เซอร์วิส (Service) ที่ทำงานอยู่มากกว่า 1 แอคทิวิตี (Activity) / เซอร์วิส (Service) ดังนั้นในแต่ละแอพพลิเคชั่นอาจจะมีมากกว่า 1 แอคทิวิตี (Activity) ซึ่งในการเริ่มทำงานของแอคทิวิตี (Activity) จะเริ่มด้วย startActivity() สำหรับแบบซิงโครนัส (Synchronous) และจะเริ่มด้วย startSubActivity () สำหรับแบบอะซิงโครนัส (Asynchronous) โดยในแต่ละแอคทิวิตี (Activity) จะมีวงรอบชีวิต (Life Cycle) ที่แยกจากกันโดยชัดเจนซึ่งมีสถานการณ์ทำงานหลักดังนี้

3.6.1 onCreate()

แอนดรอยด์ (Android) จะเรียก onCreate() เมื่อ Activity Start ในหนึ่งช่วงเวลาของแอพพลิเคชั่นนั้นอาจมีการสร้าง (Create) และเดสทรอยแอคทิวิตี (Destroy Activity) อยู่เรื่อยๆ ยกตัวอย่างเช่น เมื่อผู้ใช้งาน (User) ทำการ โร-เทท สกรีน (Rotate Screen) จะส่งผลให้แอคทิวิตี (Activity) ถูกเดสทรอย (Destroy) และInstance ใหม่ของแอคทิวิตี (Activity) เดิมจะถูกสร้าง (Create) อีกครั้ง เป็นไปได้ที่แอพพลิเคชั่นอาจถูกล้างข้อมูล (Kill) ถ้าหากแอพพลิเคชั่นทำงาน (Run) อยู่ในพื้นหลัง (Background) และระบบ (System) กำลังอยู่ในสภาวะ โลว์รีซอร์ส (Low Resources) ซึ่งถ้าแอพพลิเคชั่นถูกล้างข้อมูล (Kill) เวลาที่แอพพลิเคชั่นกลับขึ้นมาอยู่บนพื้นหน้า (Foreground) สิ่งที่จะเกิดตามมาก็คือ อินเทนท์ (Instance) ใหม่ของ แอคทิวิตี (Activity) ตัวเดิม จะถูกสร้างขึ้น และ onCreate() ก็จะถูกเรียกอีกครั้ง นอกจากนั้น onCreate() ยังเป็นเพียงโอกาสเดียว ที่จะสามารถสั่ง setContentView ให้กับแอคทิวิตี (Activity) ได้

3.6.2 onStart()

หลังจาก onCreate() ถูกเรียก onStart() จะถูกเรียกตามมาถ้าหากแอพพลิเคชั่นถูกสั่งให้ไปอยู่ในพื้นหลัง (Background) โดยอาจจะโดยการสั่ง ลันช์ (Launch) แอพพลิเคชั่น (Application) onStart() จะถูกเรียกเมื่อกลับมาที่แอพลิเคชันอีกครั้ง

3.6.3 onResume()

เมื่อแอพพลิเคชั่ นถูก Start ขึ้นมา onResume() จะถูกเรียกหลังจาก onCreate() และonStart() หรือเมื่อแอพพลิเคชั่นเปลี่ยนสถานะจาก พื้นหลัง (Background) กลับมาอยู่บนพื้นหน้า (Foreground) อีกครั้ง onResume() ก็จะถูกเรียกเช่นเดียวกัน onResume() จะถูกเรียกก่อนที่แอคทิวิตี (Activity) จะสามารถมองเห็นได้บนหน้าจอ (Screen)

3.6.4 onPause()

onPause() จะถูกเรียกเมื่อแอพพ ลิเคชั่นกำลังเปลี่ยนจากสถานะ พื้นหน้า (Foreground) ไปยังพื้นหลัง (Background) ถ้าหากเรามีการประมวลผลอะไรก็ตามที่ควรจะ ทำงาน (Run) เฉพาะตอนที่ แอคทิวิตี (Activity) อยู่บนหน้าจอ (Screen) (ยกตัวอย่างเช่นการแสดงแอนิเมชั่น (Animation)) เราควรจะสั่งหยุด การประมวลผลดังกล่าวใน onPause() onPause() จะถูกเรียกในเวลาที่สั่ง ลันช์แอคทิวิตี (Launch Activity) อื่นขึ้นมาอยู่เหนือ แอคทิวิตี (Activity) ที่กำลังแสดงผลอยู่ แอพพลิเคชัน (Application) อาจถูกล้างข้อมูล (Kill) เมื่อระบบ (System) อยู่ในสภาวะ Low Resources ซึ่งกระบวนการล้างข้อมูล (Kill) เกิดขึ้นหลังจาก onPause()

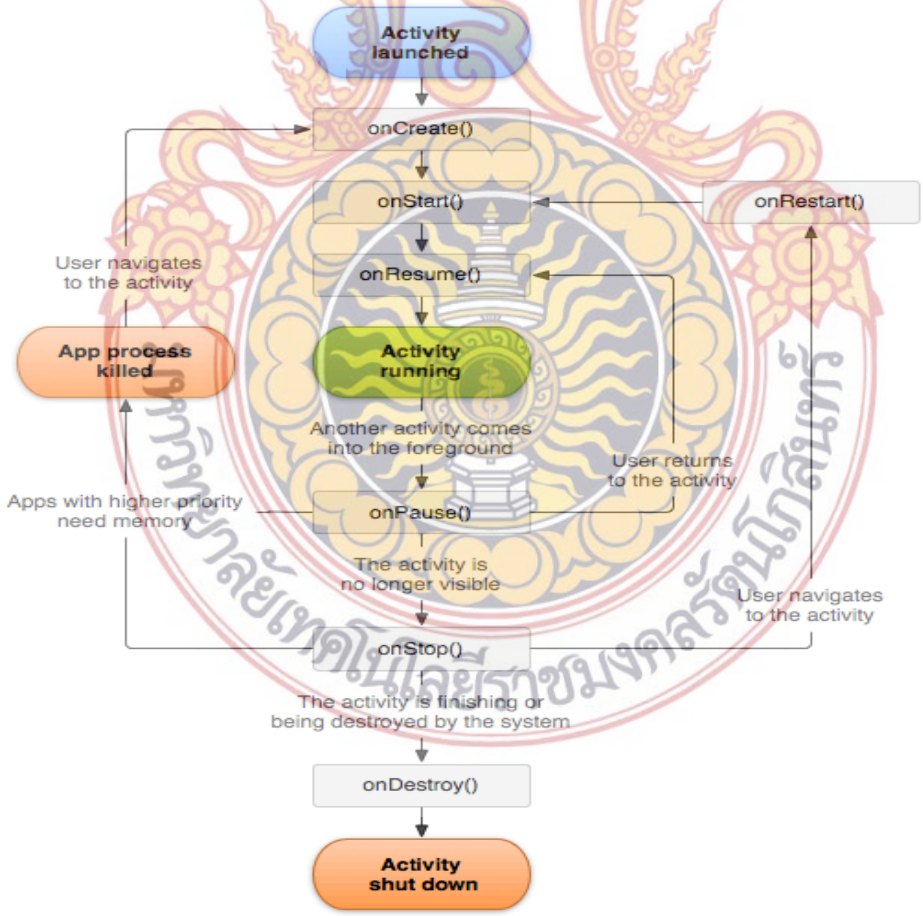
ซึ่งไม่ได้เกิดเป็นปกติ แต่อย่างน้อยก็ควรเตรียมรับมือไว้ onPause() เป็น เมธอด (Method) ที่สำคัญ เพราะเป็นเหมือนการเตือนครั้งสุดท้ายก่อนที่ แอคทิวิตี (Activity) จะหายออกจากหน้าจอ (Screen) ซึ่งเมธอด (Method) นี้เป็นที่ที่ควรจะ คัดลอกข้อมูล (Save) ข้อมูลที่สำคัญลง Disk, Database หรือ Preferences

3.6.5 onStop()

onStop() จะถูกเรียกเมื่อ แอคทิวิตี (Activity) ออกจาก หน้าจอ (Screen) เรียบร้อยแล้วหรือเมื่อเปลี่ยนไป Interact กับอีกแอคทิวิตี (Activity) หนึ่งแทน (เปลี่ยนจากสถานะ แอคทิวิตี (Activity) ไปเป็น Inactive) แต่นี่ไม่ได้หมายความแอคทิวิตี (Activity) ถูกปิดตัวลงเพียง เปลี่ยนมาอยู่ในสถานะ Inactive ถ้าหากมีการประมวลผลใดที่ต้องการ ทำงาน (Run) เฉพาะตอนที่ แอคทิวิตี (Activity) นี่คือ เมธอด (Method) ที่เหมาะสมที่เราจะสั่งหยุดการประมวลผลดังกล่าว

3.6.6 onDestroy()

onDestroy() เป็น เมธอด (Method) สุดท้ายที่จะถูกเรียกก่อนที่ แอคทิวิตี (Activity) จะปิดตัวลงอย่างถาวรเป็น เมธอด (Method) ที่ใช้ในการคืนค่าทรัพยากร (Resources) หรือล้างข้อมูล (Clean Up) ใดๆก่อนที่ แอคทิวิตี (Activity) จะถูกเก็บกวาดด้วย Garbage Collector ภายใน เมธอด (Method) นี้ควรจะสั่งปิดการทำงาน (Process) ใดๆที่สั่งทำงาน (Run) ไว้ใน พื้นหลัง (Background)



ภาพที่ 10 วงรอบชีวิตของแอพพลิเคชั่น (Android Activity Life Cycle)

4. สมาร์ทโฟน (Smart Phone) [5]

สมาร์ทโฟน (Smart Phone) คือโทรศัพท์ที่มีความสามารถมากกว่าการโทรออกและรับสาย ด้วยความที่สมาร์ทโฟน (Smart Phone) มีระบบปฏิบัติการอยู่ภายใน ทำให้สามารถทำงานได้ในลักษณะเดียวกับเครื่องคอมพิวเตอร์นอกจากนี้จะมีโปรแกรมต่างๆที่ติดตั้งมาพร้อมกับเครื่องโทรศัพท์จำนวนหนึ่ง โปรแกรมเหล่านี้มักถูกเรียกว่า แอพ (App) โดยมีทั้งแบบที่สามารถ ดาวน์โหลด (Download) มาใช้งานได้ฟรี และแบบที่ต้องเสียสตางค์ซื้อ โดยสิ่งสำคัญที่ต้องมีคู่กับ สมาร์ทโฟน (Smart Phone)ก็คืออินเทอร์เน็ต (Internet) ซึ่งปัจจุบันมีผู้ให้บริการหลายค่าย และมี อินเทอร์เน็ต (Internet) ความเร็วต่างๆ ให้ได้เลือกใช้งาน เช่น Wifi, EDGE / GPRS, 3G เป็นต้น

ภาพที่ 11 ตัวอย่างสมาร์ทโฟน (Smart Phone)

4.1 คุณสมบัติของสมาร์ทโฟน (Smart Phone)

4.1.1 การเชื่อมต่ออุปกรณ์ไร้สาย นี่เป็นคุณสมบัติอย่างหนึ่งที่จะทำให้ สมาร์ทโฟน (Smart Phone) เชื่อมต่อกับอุปกรณ์อื่นๆ ไม่ว่าจะเป็น คอมพิวเตอร์ พีดีเอ (Population and Community Development Association : PDA) โทรศัพท์เครื่องอื่นพริ้นเตอร์ หรือกล้องดิจิตอล ผ่านทาง อินฟราเรด บลูทูธ หรือวายฟาย (Wi-Fi)

4.1.2 สามารถรองรับไฟล์ มัลติมีเดีย (Multimedia) ได้หลากหลายรูปแบบ เช่นไฟล์ ภาพ , ภาพเคลื่อน ไหว เช่นภาพเคลื่อนไหวสกุล .gif เสียง ซึ่งก็จะมีหลายรูปแบบ เช่นไฟล์ Wave, MP3, Midi ต่อไปเป็นไฟล์วิดีโอ ซึ่งจะสามารถรองรับภาพเคลื่อน ไหว หรือภาพเคลื่อนไหวพร้อมเสียง เช่นสกุล .3gp .mp4 เป็นต้น

4.2 คุณสมบัติเด่นของ สมาร์ทโฟน (Smart Phone)

ระบบปฏิบัติการ หรือ OS (Operating System) เป็นระบบที่ช่วยให้การทำงาน ของโทรศัพท์มีประสิทธิภาพ และเป็นตัวกำหนดว่าโปรแกรมต่างๆ จะสามารถติดตั้งเข้ากับสมาร์ทโฟน (Smart Phone) ได้หรือไม่ด้วย สำหรับระบบปฏิบัติการที่เป็นที่นิยมใช้งานบน สมาร์ทโฟน (Smart Phone) ได้แก่ Symbian OS, Android, Windows Mobile, Palm OS หรือแม้กระทั่ง Linux OS

4.3 อุปกรณ์ต่อพ่วงที่ใช้สำหรับสมาร์ทโฟน (Smart Phone)

4.3.1 หูฟังบลูทูธ (Bluetooth)

หูฟังแบบไร้สาย ที่อาศัยเทคโนโลยี บลูทูธ (Bluetooth) ในการสื่อสาร โดยสามารถพูดคุยได้ โดยไม่จำเป็นต้องวางโทรศัพท์ไว้ใกล้ตัวเรา ปกติจะสามารถใช้งานในระยะประมาณ 10 เมตร ทั้งนี้ขึ้นกับประสิทธิภาพของ บลูทูธ (Bluetooth)

4.3.2 แป้นพิมพ์ - Keyboard

ช่วยให้เกิดความสะดวกในการพิมพ์ข้อความ โดยเฉพาะอีเมล์

4.3.3 จอยสติ๊ก (Joy Stick)

สำหรับเล่นเกมบนมือถือเพื่อความสนุกสนานมากยิ่งขึ้น

4.4 รายชื่อระบบปฏิบัติการของ สมาร์ทโฟน (Smart Phone) ที่เป็นนิยมได้แก่

4.4.1 ซิมเบียน (Symbian)

4.4.2 แบล็กเบอร์รีโอเอส (BlackBerry OS)

4.4.3 แอนดรอยด์ (Android)

4.4.4 ไอโอเอส (iOS)

4.4.5 วินโดวส์โมบาย (Windows Mobile)

4.4.6 บาดา (Bada)

4.4.7 เว็บโอเอส (webOS)

4.4.8 มีโก (MeeGo)

ตารางที่ 1 ผลการสำรวจของ Nielsen ในอังกฤษ เทียบ Q2/Q3 2009

Fatest – growing UK mobile phone media activities, Q2 2009 – Q3 2009				
Rank	Media Activity	Q3 (millions)	Q2 (millions)	Q3 % of UK Mobile Owners
1	Internet	10.4	8.8	21%
2	Downloading apps	4.1	3.1	8%
3	Email	5.8	5.1	12%
4	Text alerts	4.3	3.5	9%
5	Text messaging/SMS	37.6	36.9	78%
6	Video	1.8	1.3	4%
7	Location-based services	3.3	2.9	7%
8	Instant messaging	3.4	3.0	7%
9	Picture messaging	10.8	10.4	22%
10	Uploading content	2.6	2.3	5%

5. จีพีเอส (GPS) [6]

จีพีเอส (GPS) หรือ Global Positioning System ชื่อภาษาไทยบัญญัติโดยคณะกรรมการบัญญัติศัพท์เทคโนโลยีสารสนเทศ ราชบัณฑิตยสถาน เมื่อเดือนพฤษภาคม 2541 ไว้ว่า "ระบบกำหนดตำแหน่งบนโลก" ชื่อเต็มของระบบนี้คือ NAVSTAR Global Positioning System คำว่า NAVSTAR เป็นอักษรย่อมาจาก Navigation Satellite Timing and Ranging ภาคของคำว่าดาวเทียม สำหรับนำร่อง คือระบบที่ระบุตำแหน่งทุกแห่งบนโลกจากกลุ่มดาวเทียม 24 ดวงที่โคจรอยู่รอบโลก ในระดับสูงที่พ้นจากคลื่นวิทยุรบกวนของโลกและวิธีการที่สามารถให้ความถูกต้องเพียงพอที่จะใช้ชี้บอกตำแหน่งได้ทุกแห่งบนโลกตลอดเวลา 24 ชั่วโมง จากการนำมาใช้งานจริงจะให้ความถูกต้องสูง โดยที่ความคลาดเคลื่อนมาตรฐานของตำแหน่งทางราบต่ำกว่า 50 เมตร และถ้าเป็นแบบวิธี "อนุพันธ์" (Differential) จะให้ความถูกต้องถึงระดับเซนติเมตรจากการพัฒนาทางด้านอุปกรณ์คอมพิวเตอร์ทำให้สามารถผลิตเครื่องรับสัญญาณกำหนดตำแห น่งบนโลกที่มีขนาดลดลงและมีราคาถูกลงกว่าเครื่องรับระบบทรานซิส (TRANSIT)

ภาพที่ 12 ดาวเทียม NAVSTAR ของสหรัฐ

5.1 องค์ประกอบของระบบดาวเทียมจีเอส (GPS)

5.1.1 ส่วนศูนย์ควบคุมกลาง (Control Station Segment)

ซึ่งเป็นศูนย์ควบคุม ระบบและบัญชาการการทำงานข องระบบจีเอส (GPS) รวมไปถึงการตรวจตราดูความเรียบร้อยของระบบตั้งอยู่ที่ฐานทัพอากาศเมือง Colorado Spring สหรัฐอเมริกา และศูนย์ควบคุมกลางประกอบด้วย

- สถานีสังเกตการณ์ (Monitor Station) จำนวน 5 แห่ง กระจายอยู่ตามจุดต่างๆของโลก
- หมู่เกาะฮาวาย (Hawaii) ในมหาสมุทรแปซิฟิก
- หมู่เกาะอัสเซนชัน (Ascension) มหาสมุทรแอตแลนติก
- หมู่เกาะดิเอโกการ์เซีย (Diego Garcia) มหาสมุทรอินเดีย
- หมู่เกาะควาจาเลียน (Kwajalein) ประเทศฟิลิปปินส์
- เมืองโคโลราโดสปริงส์, สหรัฐอเมริกา

5.1.2 จานส่งสัญญาณภาคพื้นดิน (Ground Antennas) ซึ่งมีอยู่ด้วยกัน 3 จุดได้แก่

- หมู่เกาะควาจาเลียน
- หมู่เกาะดิเอโกการ์เซีย
- หมู่เกาะอัสเซนชัน

5.1.3 ศูนย์บัญชาการ (Master Control Station) ตั้งอยู่ฐานทัพอากาศสหรัฐฯ

- Schriever AFB รัฐ Colorado

5.1.4 ส่วนอวกาศ (Space Segment) จะประกอบด้วย

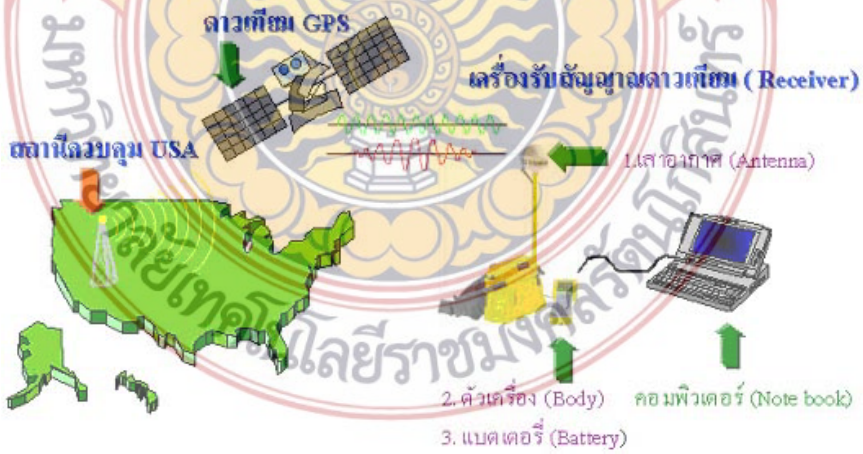
- ดาวเทียมทั้งหมด 24 ดวง แต่ละดวงโคจรรอบโลกเป็นเวลา 12 ชั่วโมง
- มีความสูงของวงโคจรอยู่ประมาณ 11,000 ไมล์จากพื้นโลก
- ดาวเทียมแต่ละดวงจะมีนาฬิกาอะตอม (Atomic Clock) ติดตั้งอยู่ถึง 4 เครื่องซึ่งจะให้เวลาที่ถูกต้องมาก
- มีระนาบของวงโคจร 6 ระนาบ แต่ละระนาบมีดาวเทียม 4 ดวง และเอียงทำมุมกับเส้นศูนย์สูตร (Equator) เป็นมุม 55 องศา

5.1.5 ส่วนผู้ใช้งาน (User Segment) ประกอบด้วย 2 ส่วนใหญ่ๆ คือ

ส่วนที่เกี่ยวข้องกับทางทหาร (Military) และทางพลเรือน (Civilian) ซึ่งทางพลเรือนจะได้รับสัญญาณฟรี แต่ผู้ใช้ต้องรับผิดชอบหาซื้อจาน รับ (Antenna) และเครื่องรับ (Receiver) ด้วยตนเอง

ส่วนประกอบของเครื่องรับสัญญาณดาวเทียม GPS โดยทั่วไปเครื่องรับสัญญาณดาวเทียม (Receiver) ประกอบด้วย 3 ส่วนคือ

- ส่วนเสาอากาศ (Antenna)
- ตัวเครื่อง (Body)
- ส่วนให้พลังงาน (Battery)



ภาพที่ 13 องค์ประกอบของระบบดาวเทียมจีเอส (GPS)

5.2 หลักการทำงานของจีเอส (GPS)

หลักการพื้นฐานของจีเอส (GPS) เป็นเรื่องง่ายๆ แต่อุปกรณ์ของเครื่องมือถูกสร้างขึ้นด้วยวิทยาการขั้นสูงการทำงานจีเอส (GPS) คือ

- จะอาศัยหลักพื้นฐานของ GPS: Satellites Triangulation หลักการ: อาศัยตำแหน่งของดาวเทียมในอวกาศเป็นจุดอ้างอิง แล้ววัดระยะจาก ดาวเทียม 4 ดวง และใช้หลักการทางเรขาคณิตในการคำนวณหาตำแหน่งบนพื้นโลก

- วัดระยะทางระหว่างเครื่องรับ GPS กับดาวเทียม GPS โดยการวัดระยะเวลาคลื่นวิทยุใช้ในการเดินทางจากดาวเทียมสู่เครื่องรับใช้เวลาเดินทางของคลื่นวิทยุ

สูตร : ระยะทาง = ความเร็ว * เวลาที่ใช้เดินทาง

คลื่นวิทยุ : ความเร็ว = 186,000 ไมล์ต่อนาที

การวัดระยะเวลาในการเดินทาง คือโดยการเทียบกันของคลื่นสัญญาณที่ดาวเทียมส่งมากับคลื่นสัญญาณที่เครื่องรับ จีพีเอส (GPS Receiver) ส่งมาส่วนคลื่นที่ใช้ในการส่ง จะเป็นพีเสโด-แรน-ดอม-นอยส์-โค้ด (Pseudo Random Noise Code: PNR)

- การวัดระยะเวลาที่คลื่นวิทยุใช้ในการเดินทางของ จีพีเอส (GPS) จะต้องใช้นาฬิกาที่ แม่นยำมาก ถ้า พี-เอ็น-อาร์-โค้ด (PRN CODE) จากดาวเทียมมีข้อมูลเวลาที่คลื่นเริ่มออกเดินทางจากดาวเทียมเมื่อคลื่นสัญญาณจากดาวเทียมและคลื่นสัญญาณจากเครื่องรับ จีพีเอส (GPS Receiver) สมมาตรกัน (Synchronize) และจะต้องใช้ นาฬิกาอะตอม (Atomic Clock) ในการวัดเวลาส่วนเวลาที่ใช้ในการเดินทางจะสั้นมากประมาณ 0.06 วินาที คือเวลาของเครื่องรับ (GPS Receiver) * เวลาของดาวเทียม ส่วนการบอกตำแหน่ง จีพีเอส (GPS) ยังเป็นเวลาที่มีความแน่นอนถึง 10 นาโนวินาทีหรือดีกว่า

5.2.1 ปัญหาทำงานของจีพีเอส (GPS)

- ต้องรู้ตำแหน่งของดาวเทียมจีพีเอส (GPS) ที่แน่นอนในอวกาศ

- วงโคจรอาจคลาดเคลื่อน (Ephemeris Errors) เนื่องจากแรงโน้มถ่วงดวงจันทร์และดวงอาทิตย์

- สถานีควบคุมจะใช้เรดาร์ตรวจสอบการโคจรของดาวเทียม จีพีเอส (GPS) ตลอดเวลาแล้วส่งข้อมูลไปปรับแก้ข้อมูลวงโคจรและเวลาของดาวเทียมเมื่อข้อมูลได้รับการปรับแก้แล้วจะถูกส่งมายังเครื่องรับจีพีเอส (GPS Receiver) ต้องแก้ไขความคลาดเคลื่อนที่เกิดจากการเดินทาง ของคลื่นวิทยุมาสู่โลกสาเหตุที่ของความคลาดเคลื่อน (GPS Errors) ของค่าพิกัดที่คำนวณได้

- เกิดจากการเดินทางสู่ชั้นบรรยากาศไอโนโนสเฟียร์ (Ionosphere) จะมีประจุไฟฟ้าและชั้นโทรโพสเฟียร์ (Troposphere) จะมีทั้งความชื้น อุณหภูมิ ความหนาแน่นที่แปรเปลี่ยนได้ตลอดเวลาใน

- การสะท้อนของคลื่นสัญญาณไปในหลายทิศทาง (Multipath Error) ซึ่งที่ผิวโลกคลื่นสัญญาณต้องกระทบกับวัตถุ ก่อนถึงเครื่องรับจีพีเอส (GPS Receiver) จะทำให้มีการหักเหและสัญญาณอ่อน

- ปัญหาที่เกิดจากดาวเทียม (Check error, Ephemeris error) อาจเกิดจากวงโคจรคลาดเคลื่อนเนื่องจากแรงโน้มถ่วงของดวงจันทร์และดวงอาทิตย์หรืออาจจะเกิดจาก

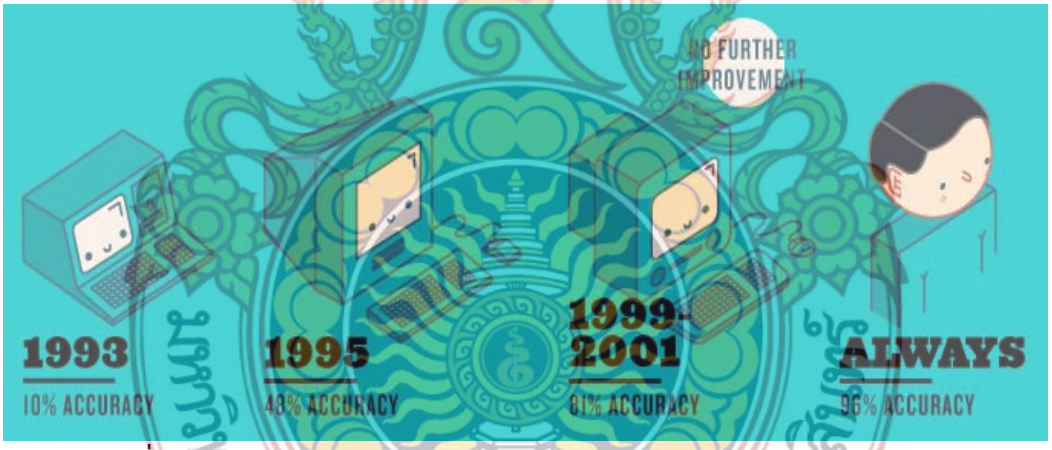
ความคลาดเคลื่อนของน าหิกาเพียงเล็กน้อยจะทำให้การคำนวณระยะทางผิดพลาดได้มากเนื่องจากดาวเทียมอยู่สูงมาก

- ความสัมพันธ์ทางเรขาคณิตระหว่างตำแหน่งของดาวเทียมและตำแหน่งของเครื่องรับเครื่องรับจีพีเอส (GPS Receiver) ซึ่งจะคำนวณเป็นค่า GDOP = Geometic Dlution of - recision เนื่องจากลักษณะการวางตัวของดาวเทียม และ GDOP

- อาจจะเกิดจากความผิดพลาดอื่นๆ เช่นความผิดพลาด (Error) ของคอมพิวเตอร์ หรือมนุษย์ที่ควบคุมสถานี 1 เมตร ถึง 100 เมตร ซึ่งผิดพลาดได้มาก หรือความผิดพลาดของเครื่องรับจีพีเอส (GPS Receiver), ซอฟต์แวร์ (Software), ฮาร์ดแวร์ (Hardware), ผู้ใช้งาน (User) ซึ่งความผิดพลาดนี้ไม่แน่นอน

6. การรู้จำเสียงพูด (Speech Recognition) [7]

เป็นเทคโนโลยีที่ใช้วิเคราะห์ และควบคุ มการทำงานคอมพิวเตอร์ด้วยเสียง โดยจะพยายามทำความเข้าใจในชุดข้อมูลเสียงที่ได้รับ ด้วย การตรวจสอบโครงสร้างและรูปแบบของเสี ยงว่าถูกต้องตามที่กำหนดไว้หรือไม่ และจะดำเนินการตามเ งื่อนไขของโปรแกรมที่ได้กำหนดไว้ โดยระบบจะสามารถแจกแจงคำพูดต่างๆ ที่มนุษย์สามารถพูดใส่ไมโ ครโฟน โทรศัพท์หรืออุปกรณ์อื่นๆ และเข้าใจคำศัพท์ทุกคำอย่างถูกต้องเกือบร้อยละ 100



ภาพที่ 14 ความแม่นยำระบบรู้จำเสียงพูด (Speech recognition) จากอดีตถึงปัจจุบัน

Recognizer (Speech recognition engine) เป็นซอฟต์แวร์ (software driver) ที่แปลงสัญญาณเสียงเป็น สัญญาณดิจิตอล (Digital) และส่งคำที่แปลงได้ออ กมาเป็นข้อความให้โปรแกรมทำงานรู้จำเสียงพูด (Recognizers) ส่วนมากจะสนับสนุนคำพูดที่มีความต่อเนื่องทำให้สามารถพูดได้อย่างเป็นธรรมชาติผ่านไมโครโฟนเข้าไปด้วยความเร็ว ที่ใช้ในการสนทนากันตามปกติ ซึ่ง การรู้จำเสียงพูด (Recognizers) ในปัจจุบันนี้จะสนับสนุนการทำงาน 2 แบบ คือ

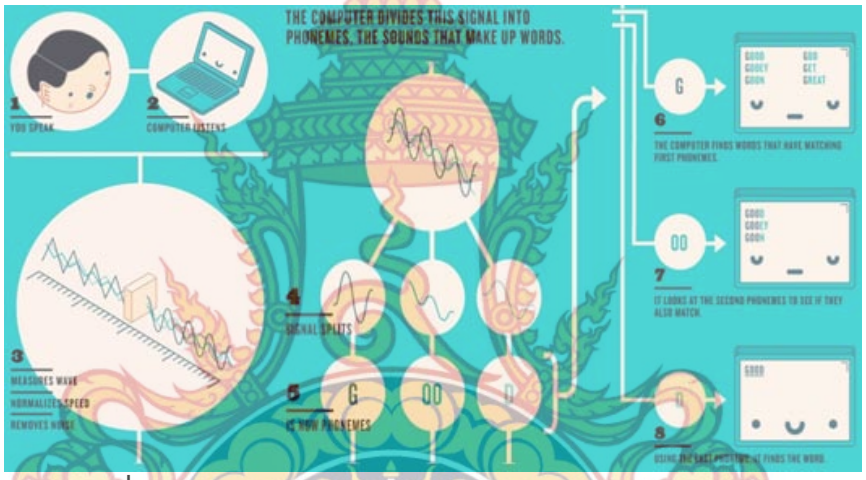
- ผู้ใช้สามารถป้อน ข้อมูลให้คอมพิวเตอร์โดยตรง เช่น การอ่านบันทึก จดหมายหรืออีเมลล์ (E-mail) ให้โปรแกรมพิมพ์ตามเป็นต้น

- ผู้ใช้สั่งให้โปรแกรมเริ่มการทำงานโดยการพูดคำสั่งหรือการถามคำถาม ด้วยการสังเคราะห์เสียงพูด (Speech Synthesis)

6.1 คุณสมบัติ

การรู้จำเสียงพูด (Speech Recognition) หรือ Speech-to-Text จะเกี่ยวข้องกับการจับคลื่นเสียงและการแปลงคลื่นเสียงเป็นข้อมูล ดิจิตอล (Digital), การแปลงให้อยู่ใน หน่วยเสียง (Phonemes), การสร้างคำจาก หน่วยเสียง (Phonemes) และการวิเคราะห์คำรอบข้างเพื่อให้แน่ใจว่าสะกดคำได้ถูกต้องสำหรับคำที่ออกเสียงคล้ายกันเช่น write กับ right

หน่วยเสียง (Phonemes) คือ หน่วยเสียงย่อยที่สุดที่นำใช้ในการทดลอง มารวมกันในการสร้างคำแต่ละคำออกมา เช่นคำว่า Good นั้นก็จะแยกออกเป็น G OO และ D มี 3 หน่วยเสียง (Phonemes)



ภาพที่ 15 หลักการทำงานของการรู้จำเสียงพูด (Speech Recognition)

หลักการทำงาน ของการรู้จำเสียงพูด (Speech Recognition) ซึ่งมีลำดับดังต่อไปนี้

1. ผู้ใช้พูดผ่านไมโครโฟน
2. จากนั้นเครื่องคอมพิวเตอร์ หรืออุปกรณ์อื่นๆจะทำการจับสัญญาณเสียงนั้น
3. ฮาร์ดแวร์และซอฟต์แวร์ของตัวเครื่องจะทำการวัดความถี่และขนาดของคลื่น จากนั้นทำการ Normalize ความเร็วของคลื่น และทำการแยกแยะเอาเสียงรบกวน (Noise) ออก
4. และ 5. คอมพิวเตอร์จะทำการแบ่งแยกสัญญาณเสียงออกเป็นองค์ประกอบย่อยๆ ที่เรียกว่า Phonemes
6. 7. และ 8. จะเป็นหน้าที่ของซอฟต์แวร์ที่จะนำ Phonemes แต่ละตัวมาค้นหาคำที่มีเสียงตรงกับหน่วยเสียง (Phonemes) นั้นๆ โดยจะค่อยๆ ทำไปทีละหน่วยเสียง (Phonemes) ซึ่งตรงนี้จุดสำคัญอยู่ที่ตัวฐานข้อมูลที่ ซอฟต์แวร์จะใช้เปรียบเทียบเสียงของแต่ละหน่วยเสียง (Phonemes)

ตารางที่ 2 เปรียบเทียบข้อดี - ข้อเสียของระบบการรู้จำเสียงพูด (Speech Recognition)

ข้อดี	ข้อเสีย
1. สามารถใช้ได้ในหลายสภาวการณ์ทั้งที่ปกติและไม่ปกติ เช่น ขณะที่มือไม่ว่าง ต้องการความคล่องตัว สายตาไม่ว่าง ไม่ต้องการใช้คีย์บอร์ดมีข้อจำกัดทางร่างกาย ฯลฯ 2. ช่วยประหยัดเวลาในการทำงาน 3. ช่วยอำนวยความสะดวกมากยิ่งขึ้น	1. ผู้ใช้ต้องออกเสียงชัดเจน และพูดตามอักขระให้ถูกต้อง 2. ขณะใช้ต้องปราศจากเสียงรบกวนจากภายนอก เพื่อไม่ให้โปรแกรมประมวลผลผิดพลาด 3. โปรแกรมสามารถแสดงคำพูดได้เป็นหลายคำ เช่น “Hello” อาจจะกลา ยเป็น “Little” “good old” หรือ“who told” 4. ถ้าใช้การ์ดเสียง (Sound Card) ที่ต่ำกว่ามาตรฐานที่โปรแกรมกำหนด จะทำให้ได้ผลที่ได้ไม่ดีนัก

7. ระบบฐานข้อมูลเอสคิวไลท์ (Database SQLite) [8]

เอสคิวไลท์ (SQLite) เป็นฐานข้อมูล (Database) ขนาดเล็กที่ได้รับความนิยมอย่างมากกับแอพพลิเคชัน (Application) ที่ทำงานบน สมาร์ทโฟน (Smart Phone) ประเภทต่างๆ รูปแบบการทำงานของ เอสคิวไลท์ (SQLite) เป็นแบบ Standalone ทำงานอยู่ในแอพพลิเคชัน (Application) นั้นๆ เอสคิวไลท์ (SQLite) มีโครงสร้างง่ายต่อการจัดเก็บและนำไปใช้ และไฟล์ที่จัด เก็บนั้นก็มีขนาดเล็กมาก เกือบเท่ากั บการเก็บข้อมูลจริง เพราะฉะนั้น ฐานข้อมูลเอสคิวไลท์ (Database SQLite) จึงเหมาะสมกับ แอพพลิเคชัน (Application) ที่ทำงานบน Smartphone อย่างยิ่ง โดยเฉพาะอันเนื่องจากข้อจำกัดทางด้าน ฮาร์ดแวร์ (Hardware) และหน่วยความจำ (Memory) รวมทั้งความสามารถในการ ประมวลผล (Process) ข้อมูลต่าง ๆ ในโทรศัพท์เคลื่อนที่ (Smartphone) ย่อมน้อยกว่า PC Desktop เป็นธรรมดา

ภาพที่ 16 สัญลักษณ์ฐานข้อมูลเอสคิวไลท์ (Database SQLite)

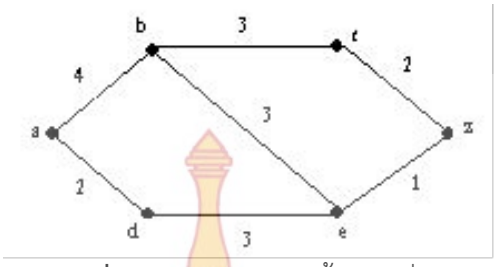
ตารางที่ 3 การเปรียบเทียบโอเพนซอร์ส (Open Source)

ข้อเปรียบเทียบ	ข้อเปรียบเทียบ	MySQL	Postgresql
ลักษณะเด่น	- เป็นแบบ high concurrency(ใช้งานพร้อมกันได้) - เป็นแบบ fine-grained access control - มี function ให้ใช้มากมาย - มี stored procedure	- เป็นโปรแกรมฐานข้อมูลที่มีหน้าที่เก็บข้อมูลอย่างมีโครงสร้าง - รองรับคำสั่ง SQL - เครื่องมือสำหรับเก็บข้อมูลอย่างมืออาชีพ	- Object-Relational DBMS โดยสามารถใช้รูปแบบของภาษา SQL ได้เกือบทั้งหมด - สามารถใช้ subselects , transactions , user-defined types และ functions ได้
ข้อดี	- ง่ายต่อการติดตั้ง - เหมาะสำหรับการเก็บข้อมูลชั่วคราว หรือทดสอบ - พัฒนาได้เร็วกว่า	- เป็นตัวเลือกที่ดีกว่าสำหรับการปรับแต่งประสิทธิภาพ - เหมาะสำหรับข้อมูลขนาดใหญ่ - สามารถจัดการสิทธิผู้ใช้งานให้ใช้งานง่าย - MySQL ทำงานได้เร็ว	- ง่ายในการติดตั้งใช้งาน - รองรับการใช้งาน transaction - ความปลอดภัยสูง - กำหนดขนาดของฐานข้อมูลได้
ข้อเสีย	- ประสิทธิภาพโดยรวมด้อยกว่า MySQL - ไม่เหมาะสำหรับข้อมูลขนาดใหญ่ - ไม่มีการจัดการผู้ใช้งาน	- การติดตั้งซับซ้อนกว่า - ความปลอดภัยต่ำ	- ไม่สามารถเลือก replicate databases หรือ tables เฉพาะที่ต้องการ ในการใช้งานค่อนข้างยุ่งยาก

8. การหาเส้นทางที่สั้นที่สุด (Short Path) [9]

การหาเส้นทางที่สั้นที่สุด (Short Path) มีหลายวิธีคิดที่จะค้นหาเส้นทางสั้นที่สุดระหว่างสองจุดในกราฟในที่นี้จะแสดงวิธีคิดที่ค้นพบโดยนักคณิตศาสตร์ชาวดัชท์ ชื่อ เอด ส์เกอร์ ไดจ์สตรา (Edsger Dijkstra) ในปี ค.ศ.1959

ตัวอย่างที่1 ความยาวของเส้นทางที่สั้นที่สุดระหว่าง a และ z ในกราฟน้ำหนักที่แสดงดังภาพที่ 17



ภาพที่ 17 ตัวอย่างกราฟน้ำหนักทั่วไป

การแก้ปัญหา ตัวอย่างที่ 1 วิธีคิดเส้นทางสั้นที่สุดที่ถูกค้นพบได้โดยง่ายโดยการ ตรวจสอบอย่างละเอียด จะพัฒนาจากแนวความคิดของ ไดจ์สตรา (Dijkstra) โดยการค้นหาความยาวของเส้นทางที่สั้นที่สุดจากจุด a ถึงจุด z เส้นทางเริ่มต้นจากจุด a เท่านั้น ซึ่งเป็นจุดที่ต้องไม่มีจุดอื่นอยู่นอกจากจุด a (จนกระทั่งจุดปลาย) คือ a, b และ a, d เพราะว่าความยาวของ a, b และ a, d คือ 4 และ 2 ตามลำดับ d คือจุดปลายที่ใกล้จุด a ที่สุด

สามารถค้นหาจุดปลายถัดไปที่ใกล้ที่สุดโดยการดูทุกเส้นทางที่เริ่มจาก a ไปถึง d เท่านั้น(จนกระทั่งถึงจุดปลายทาง) เส้นทางที่สั้นที่สุดถึง b คือ a, b มีความยาว 4 และเส้นทางที่สั้นที่สุดถึง e คือa, d, e มีความยาว 5 ผลที่สุดก็คือ จุดถัดไปที่ใกล้จุด a ที่สุดคือจุด b

เพื่อค้นหาจุดปลายที่สามที่ใกล้ a ที่สุดเราต้องการตรวจดูเส้นทางที่ผ่าน a, d, และ b (จนกระทั่งถึงจุดปลาย) มี เส้นทางหนึ่งเส้นทางของความยาว 7 ถึง c กล่าวคือ a, b, c และเส้นทางของความยาว 6 ถึง z กล่าวคือ a, d, e, z ผลที่สุดก็คือ z คือจุดปลายถัดไปที่ใกล้ a ที่สุดและความยาวของเส้นทางสั้นที่สุดถึง z คือ 6

ตัวอย่างที่ 1 โดยใช้วิธีคิดของไดจ์สตรา (Dijkstra) บันทึกว่าเส้นทางสั้นที่สุดจาก a ถึง z สามารถถูกค้นพบโดยการตรวจสอบอย่างละเอียด อย่างไรก็ตาม การตรวจสอบอย่างละเอียดทำไม่ได้กับกราฟที่มี edges จำนวนมากมาย

เราจะคิดแก้ปัญหาโดยการค้นหาความยาวของเส้นทางสั้นที่สุดระหว่าง a และ z ในไม่มีทิศทาง (Undirected) โดยการค้นหาความยาวของเส้นทางสั้นที่สุดจากจุด a ถึงจุดปลายที่หนึ่งความยาวของเส้นทางสั้นที่สุดจาก a ถึงจุดปลายที่สองและอื่นๆจนกระทั่งความยาวของเส้นทางสั้นที่สุดจาก a ถึง z ถูกค้นพบ

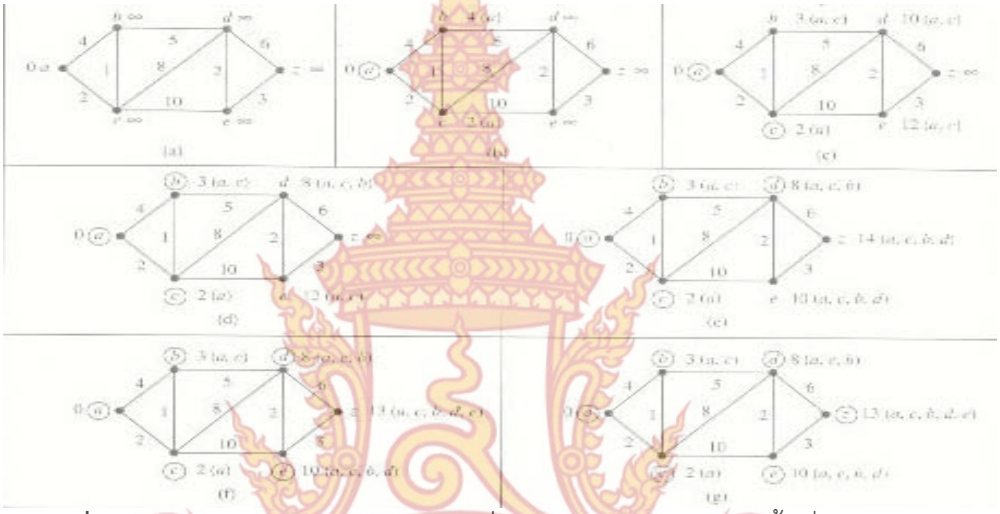
วิธีคิดด้วยอนุกรมของการทำซ้ำส่วนหนึ่งของจุดถูกสร้างโดยเพิ่มจุ ดที่ทำซ้ำแต่ละอันโพรซิเดอร์กับจุดที่ดำเนินการทำซ้ำแต่ละอันในโพรซิเดอร์ โดย จุด w ถูกกำหนดด้วยความยาวของเส้นทางสั้นที่สุดจากa ถึง w จุดที่เพิ่มคือค่าที่มี cost น้อยมากท่ามกลางจุดอื่นๆ

เราให้รายละเอียดวิธีคิดของไดจ์สตรา (Dijkstra) เริ่มต้นโดยการกำหนดค่า a เป็น 0 และจุดอื่นๆเป็น ∞ เราใช้หมายเหตุ L0 (a) = 0 และ L0 (V) = ∞ สำหรับการกำหนดก่อนทำซ้ำ จากนั้นกำหนด

วิธีคิดที่ 1วิธีคิดของไดจ์สตรา (Dijkstra) โพรซิเดอร์ ไดจ์สตรา (Dijkstra) (G: weighted - connected simple graph, กับน้ำหนักทั้งหมดที่เป็นบวก) {G มีจุด a = v0,v1,...,vn =z และ น้ำหนัก w (vi, vj) ซึ่ง w(vi, vj)= ∞ ถ้า{vi, vj} ไม่เป็น edge ใน G} for I := 1 to n L(vi) := 0 S := Ø {ค่าที่ได้ประกาศเริ่มต้นให้ ค่าของ a จะมีค่าเป็น 0 และค่าอื่นๆทั้งหมดมีค่าเป็น ∞

และ S คือค่าที่ว่างเปล่า} while z U S begin u := a จุดที่ไม่อยู่ใน S มี L(v) เป็นค่าต่ำสุด S := S U {u} for จุด v ทั้งหมดไม่อยู่ใน S\ if L(u) + w(u, v) < L(v) then L(v) := L(u) + w(u, v) {เพิ่มค่าที่น้อยที่สุดให้ S และ อัพเดทค่าของจุดที่ไม่อยู่ใน S} end {L(z) = ความยาวของเส้นทางที่น้อยที่สุดจาก a ถึง z}

ตัวอย่างที่ 2 เขียนภาพประกอบวิธีคิดของไดจ์สตรา (Dijkstra) แสดงว่าวิธีคิดนี้จะทำให้ได้ความยาวของเส้นทางสั้นที่สุดระหว่างสองจุด ซึ่งตัวอย่างที่ 2 ใช้วิธีคิดของ ไดจ์สตรา (Dijkstra) เพื่อค้นหาความยาวของเส้นทางสั้นที่สุดระหว่างสองจุด a และ z



ภาพที่ 18 วิธีคิดของไดจ์สตรา (Dijkstra) ที่จะค้นหารูปแบบเส้นทางสั้นที่สุดจาก a ถึง z

การแก้ปัญหา ตัวอย่างที่ 2 ขั้นตอนที่ใช้วิธีคิดของไดจ์สตรา (Dijkstra) เพื่อค้นหาเส้นทางสั้นที่สุดระหว่าง a และ z ที่แสดงในภาพที่ 18 การทำซ้ำแต่ละอันของวิธีคิด จุดของการตั้ง Sk คือวงกลมเส้นทางสั้นที่สุดจาก a ถึงจุดแต่ละจุดที่รวมกันเท่านั้นใน Sk ถูกชี้บอกสำหรับแต่ละการทำซ้ำวิธีคิดสิ้นสุดลงเมื่อ z ถูกวงกลม เราพบว่ารูปแบบเส้นทางสั้นที่สุดจาก a ถึง z คือa, c, b, d, e, z มีความยาวคือ 13

ถัดไป ใช้ข้อความอุปนัยที่จะแสดงวิธีคิดของ ที่ได้ความยาวของเส้นทางสั้นที่สุดระหว่างสองจุด a และ z ใน undirected connected weighted graph จากการสมมติฐานโดยอ้างอิงจากหลักการดังต่อไปนี้: การทำซ้ำที่ kth (i) ค่าของทุกจุด v ใน S คือ ความยาวของเส้นทางที่สั้นที่สุดจากจุด a ไปยังจุดนี้ (ii) ค่าของทุกจุดใน s คือ ความยาวของเส้นทางที่สั้นที่สุดจากจุด a ไปจุดนี้เท่านั้น (นอกจากจุดตัวเอง) จุด S

เมื่อ k=0 ก่อนการทำซ้ำการดำเนินการใดๆ S = Ø, ดังนั้นความยาวของเส้นทางที่สั้นที่สุดจาก a ถึงจุดที่มากกว่าจุดอื่นคือ ∞ ดังนั้น กรณีพื้นฐานถูกต้อง

สันนิษฐานว่าอุปนัยสำหรับการทำซ้ำที่ kth ให้ v จุดที่เพิ่มใน S ที่ (k+1) st เพื่อว่า v คือจุดใน S ที่จบของการทำซ้ำที่ kth กับค่าที่เล็กที่สุด

จากสมมติฐานเราจะเห็นว่าจุดใน S ก่อนการทำซ้ำที่ (k+1)st ถูกกำหนดค่าด้วยความยาวของเส้นทางที่สั้นที่สุดจาก a อีกด้วย v ถูกกำหนดค่าด้วยความยาวของเส้นทางที่สั้นที่สุดให้มันจาก a ถ้าสิ่งนี้ไม่มีกรณี, ที่จบของการทำซ้ำที่ kth เป็นเส้นทางหนึ่งเส้นทางของความยาวที่น้อยกว่า Lk (v) ที่บรรจุจุดไม่อยู่ใน S (เพราะว่า Lk(v)คือความยาวของเส้นทางที่สั้นที่สุดจาก a ถึง v ที่อยู่ใน S เท่านั้น หลังจากการทำซ้ำที่ kth)ให้ u เป็นจุดแรกที่ไม่อยู่ใน S ในเส้นทาง มีเส้นทางกับความยาวน้อยกว่า Lk(V)จาก a ถึง u ที่อยู่ใน S เท่านั้น สิ่งนี้ปฏิเสธตัวเลือกของ v ดังนั้น (i) ถือว่าจบการทำซ้ำที่ (k+1) st

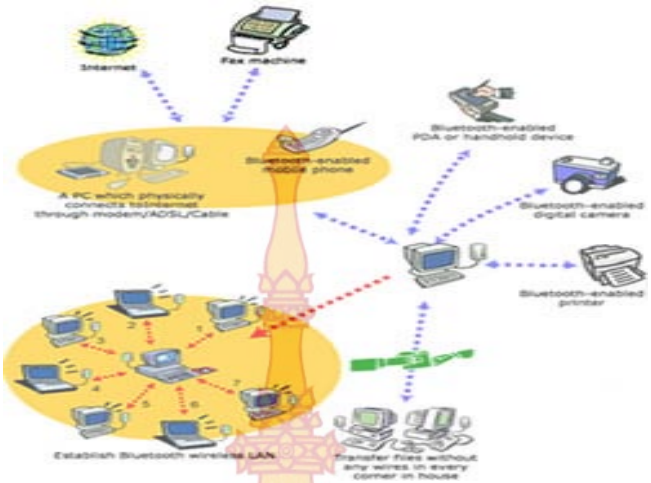
ให้ u ไม่อยู่ใน S หลังจากการทำซ้ำที่ k+1 เส้นทางสั้นที่สุดจาก a ถึง u ที่บรรจุค่าของ S อันใดอันหนึ่งบรรจุ v หรือไม่บรรจุ ถ้ามันไม่บรรจุ v แล้วให้สันนิษฐานว่าความยาวของมันคือ Lk(u) ถ้ามันบรรจุ v แล้วให้มันสร้างเส้นทางจาก a ถึง v ของความยาวที่สั้นที่สุดที่เป็นไปได้ของ S ที่มากกว่า v เพราะถูกปฏิบัติตามด้วย edge จาก v ถึง u ในกรณีนี้ความยาวของมันจะเป็น Lk (v)+w(v, u) สิ่งนี้แสดงว่า ii) นั่นถูกต้อง เพราะว่า L k +1(u)= min{ Lk (u), Lk (v)+w(v,u)}
ทฤษฎีบทที่1ถูกพิสูจน์ว่า

ทฤษฎีบทที่ 1 วิธีคิดของ ไดจ์สตรา (Dijkstra) หาความยาวของเส้นทางที่สั้นที่สุดระหว่างสองจุดใน connected simple undirected weighted graph สามารถคะเนความซับซ้อนของกระบวนการคำนวณวิธีคิดของ ไดจ์สตรา (Dijkstra) (ในแง่ของการเพิ่มและการเปรียบเทียบ) วิธีคิดใช้ไม่ได้กับทำซ้ำที่มากกว่า n - 1 เพราะว่าหนึ่งจุดที่ถูกเพิ่มขึ้นเพื่อการทำซ้ำแต่ละอัน สามารถคะเนการคำนวณจำนวนมากมายที่ใช้สำหรับทำซ้ำแต่ละอัน และ สามารถระบุจุดใน Sk ต่อมาใช้การเพิ่มและการเปรียบเทียบที่จะปรับปรุงค่าของจุดแต่ละอันใน Sk โดยค่าต้องไม่มากกว่า 2(n-1) มีทฤษฎีบทดังต่อไปนี้

ทฤษฎีบทที่ 2 วิธีคิดของไดจ์สตรา (Dijkstra) ใช้การคำนวณ O (n^2) (การเพิ่มและการเปรียบเทียบ) ที่จะค้นหาความยาวของเส้นทางที่ สั้นที่สุดระหว่างสองจุดใน connected simple undirected weighted graph ที่จุด n

9. บลูทูธ (Bluetooth) [10]

บลูทูธ (Bluetooth) คือระบบสื่อสารของอุปกรณ์อิเล็คโทรนิ คแบบสองทางด้วยคลื่นวิทยุระยะสั้น (Short - Range Radio Links) โดยปราศจากการใช้สาย เคเบิ้ล หรือสายสัญญาณเชื่อมต่อ และไม่จำเป็นจะต้องใช้การเดินทางแบบเส้นตรงเหมื อนกับอินฟราเรด ซึ่งถือว่าเพิ่มความสะดวกมากกว่าการเชื่อมต่อแบบอินฟราเรดที่ใช้ในกา รเชื่อมต่อระหว่างโทรศัพท์มือถือกับอุปกรณ์ในโทรศัพท์เคลื่อน ที่รุ่นก่อนๆ และในการวิจัย ไม่ได้มุ่งเฉพาะการส่งข้อมูลเพียงอย่างเดียว แต่ยังศึกษาถึงการส่งข้อมูลที่เป็นเสียง เพื่อใช้สำหรับชุดหูฟัง (Headset) บนโทรศัพท์มือถือด้วย



ภาพที่ 19 ตัวอย่างการสื่อสารของอุปกรณ์บลูทูธ (Bluetooth)

9.1 การทำงานของ บลูทูธ (Bluetooth)

จะใช้สัญญาณวิทยุความถี่สูง 2.4 GHz. แต่จะแยกย่อยออกไปตามแต่ละประเทศ อย่างในแถบยุโรปและอเมริกา จะใช้ช่วง 2.400 ถึง 2.4835 GHz. แบ่งออกเป็น 79 ช่องสัญญาณ และจะใช้ช่องสัญญาณที่แบ่งนี้ เพื่อส่งข้อมูลสลับช่องไปมา 1,600 ครั้งต่อ 1 วินาที ส่วนที่ญี่ปุ่น จะใช้ความถี่ 2.402 ถึง 2.480 GHz. แบ่งออกเป็น 23 ช่อง ระยะทำการของ บลูทูธ (Bluetooth) จะอยู่ที่ 5-10 เมตร โดยมีระบบป้องกันโดยใช้การป้อนรหัสก่อนการเชื่อมต่อ และป้องกันการดักสัญญาณระหว่างสื่อสารความสามารถการส่งถ่ายข้อมูลของ บลูทูธ (Bluetooth) จะอยู่ที่ 1 Mbps โดยระบบจะสลับช่องสัญญาณไปมา จะมีความสามารถในการเลือกเปลี่ยนความถี่ที่ใช้ในการติดต่อเองอัตโนมัติโดยที่ไม่จำเป็นต้องเรียงตามหมายเลขช่อง ทำให้การดักฟังหรือลักลอบขโมยข้อมูลทำได้ยากขึ้น

โดยหลักของบลูทูธจะถูกออกแบบมา เพื่อใช้กับอุปกรณ์ที่มีขนาดเล็ก เนื่องจากใช้การขนส่งข้อมูลในจำนวนที่ไม่มาก อย่างเช่น ไฟล์ภาพ, เสียง, แอพพลิเคชั่นต่างๆ และสามารถเคลื่อนย้ายได้ง่าย ขอให้อยู่ในระยะที่กำหนดไว้เท่านั้น (ประมาณ 5-10 เมตร) นอกจากนี้ยังใช้พลังงานต่ำกินไฟน้อย และสามารถใช้งานได้นาน

9.2 ประโยชน์ของ บลูทูธ (Bluetooth) ในโทรศัพท์มือถือ กับ ชุดหูฟัง (Smalltalk)

ชุดหูฟัง หรือสมอลล์ทอล์ค (Smalltalk) อุปกรณ์โทรศัพท์มือถือที่ผู้ใช้เกือบทุกคนต้องมีใช้กัน ในด้านการใช้งานบนเครื่องโทรศัพท์มือถือ หากเป็นชุดหูฟังแบบมีสาย ข้อจำกัดจะอยู่ที่เราไม่สามารถเคลื่อนตัวไปไหนได้ไกลกว่าที่สายจะยาวถึง แล้วก็ต้องคอยระวังสายไม่ให้ไปเกี่ยวกับสิ่งของต่างๆ บางทีอาจจะทำให้สายหลุดออกจากเครื่องด้วย แต่เมื่อนำ บลูทูธ (Bluetooth) มาแทนที่การใช้งาน จะเพิ่มความสะดวกและความปลอดภัยในการใช้มือทั้งสองข้างทำงานอย่างอื่นไปพร้อม ๆ กันด้วย ทั้งในเวลาขับรถ ขณะออกกำลังกาย หรือ ขณะปฏิบัติกิจต่าง ๆ ก็สามารถขยับตัวไปไหนได้อย่างสะดวก

ภาพที่ 20 ตัวอย่างชุดหูฟังที่ใช้ บลูทูธ (Bluetooth)

10. การทบทวนวรรณกรรม/สารสนเทศ (information) ที่เกี่ยวข้อง

Djunknic and Richtion [11] ได้เสนอเทคโนโลยีการระบุพอกัดตำแหน่งทาทางของภูมิศาสตร์ และระบบจีพีเอส โดยอธิบายถึงคุณสมบัติต่างๆของวิธีการบอกพิกัดตำแหน่งของแต่ละวิธี และหลักการทำงานของระบบจีพีเอสโดยสรุปงานได้กล่าวถึงระบบจีพีเอานั้นเป็นทางเลือกที่เหมาะสมที่สุดในการที่จะประยุกต์ใช้กับงานบริการข่าวสารโดยใช้พิกัดตำแหน่ง (Lacation Based Service: LBS)

Adusei et al [12] งานวิจัยนี้ได้เสนอการเปรียบเทียบประสิทธิภาพของเทคโนโลยีต่างๆในการค้นหาพิกัดของโทรศัพท์เคลื่อนที่ ซึ่งมีปัจจัยหลักในการวัดประสิทธิภ าพ คือความละเอียดถูกต้อง ความน่าเชื่อถือได้ ความสามรถในการหาตำแหน่ง เวลาที่ใช้ และการประยุกต์เอาไปใช้งานได้ จากการเปรียบเทียบเทคโนโลยีจีพีเอส และเอจีพีเอส พบว่าการใช้งานระบบนั้นให้ความน่าเชื่อถือ และการนำไปประยุกต์ใช้งานได้ดีกว่า ในขณะที่ระบบเอจีเอสจะมีควา มสามารถในการหาตำแหน่งได้ และเวลาได้ดีกว่า โดยความถูกต้องนั้นอยู่ในระดับเดียวกัน

GT&T [13] ได้เสนอการระบุตำแหนงของยานพาหนะโดยอัตโนมัติที่มีการใช้งานจริงในเชิงพานิชย์ในประเทศสิงค์โปร ที่เรียกทางการค้าว่า GFleet โดยโครงสร้างของระบบนั้นประกอบไปด้วย ศูนย์ควบคุม ศูนย์ปฏิบัติการและอุปกรณ์ที่ติดตั้งบนรถยนต์โดยพนักงานประจำสถานีกลาง จะปฏิบัติงานโดยใช้โปรแกรมสพเร็จรูป และมีอุปกรณ์ที่ติดตั้งบนรถยนต์ คือ เครื่องรับสัญญาณจีพีเอส และโมเด็มจีพีเอส

บทที่ 3
วิธีดำเนินการวิจัย

ในการออกแบบและการทดลองการพัฒนาระบบนำทางเพื่อผู้พิการทางสายตา ผู้วิจัยได้ทำการวิเคราะห์และออกแบบของการพัฒนาระบบ โดยได้แบ่งขั้นตอนการดำเนินงานเป็น 5 ขั้นตอน ดังนี้

1. ศึกษาความต้องการจากผู้ใช้งาน
2. การทดสอบและใช้งานอิคลิป (Eclipse) สำหรับแอนดรอยด์เอสดีเค (Android SDK)
3. การออกแบบแอพพลิเคชั่นนำทาง
4. การออกแบบและพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตา
5. แอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่

1. ศึกษาความต้องการจากผู้ใช้งาน

ในส่วนนี้จะเป็นการศึกษาความต้องการจากผู้ใช้งาน คือผู้พิการทางสายตาจำนวน 150 กลุ่มตัวอย่าง ทั้งนี้เพื่อใช้ในการออกแบบระบบให้ตรงความต้องการของผู้ใช้งาน และมีประสิทธิภาพ โดยได้สรุปผลความต้องการในระบบนำทางจากแบบสอบถามที่ได้ทำการแจกโดยมีประเด็นคำถามหลักๆดังนี้

- ระบบนำทางเพื่อผู้พิการทางสายตาต้องมีฟังก์ชันไม่ซับซ้อนใช้งานได้ง่าย
- ระบบนำทางเพื่อผู้พิการทางสายตาต้องมีฟังก์ชันการสั่งการด้วยเสียงได้
- ระบบนำทางเพื่อผู้พิการทางสายตาต้องมีฟังก์ชันบอกสถานที่ด้วยเสียงได้
- ระบบนำทางเพื่อผู้พิการทางสายตาต้องสามารถใช้ บนโทรศัพท์ที่มีระบบปฏิบัติการแอน - ดรอยด์ได้ทุกรุ่น
- ระบบนำทางเพื่อผู้พิการทางสายตา ต้องสามารถใช้งานได้บนโทรศัพท์เคลื่อนที่โดยจะต้องไม่มีอุปกรณ์ต่อพ่วง (Hard Ware) เพิ่มเติมเนื่องจากจะเป็นภาระค่าใช้จ่าย และการใช้งานที่ยุ่งยากมากขึ้น

จากการศึกษาความต้องการของผู้พิการทางสายตาได้จึงได้ข้อสรุปข้างต้นเพื่อใช้ในการออกแบบและพัฒนาระบบต่อไปเพื่อให้เกิดประโยชน์ และมีประสิทธิภาพสูงสุด

2. การทดสอบและใช้งาน อิคลิป (Eclipse) สำหรับแอนดรอยด์เอสดีเค (Android SDK)

โปรแกรมที่เลือกใช้ในการพัฒนาแอพพลิเคชั่นคือ อิคลิป (Eclipse) ซึ่งเป็นโปรแกรมที่ใช้สำหรับพัฒนาภาษาจาวา (Java) ซึ่งโปรแกรมอิคลิป (Eclipse) เป็นโปรแกรมหนึ่งที่ใช้ในการพัฒนาแม่ข่ายประยุกต์ (Application Server) ได้อย่างมีประสิทธิภาพ และเนื่องจาก อิคลิป (Eclipse) เป็นซอฟต์แวร์ที่เปิดใช้งานฟรี (Open Source) ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของอิคลิป (Eclipse) เป็นไปอย่างต่อเนื่องและรวดเร็ว

อิคลิป (Eclipse) มีองค์ประกอบหลักที่เรียกว่า อิคลิปแพลตฟอร์ม (Eclipse Platform) ซึ่งให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่ างๆ จากภายนอกให้สามารถเข้ามาทำงานร่วมกัน ในสภาพแวดล้อมเดียวกัน และมีองค์ประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่าอิคลิปปลั๊ก-อิน (Eclipse Plug-Ins) ดังนั้นหากต้องการให้ อิคลิป(Eclipse) ทำงานใดเพิ่มเติม ก็เพียงแต่พัฒนาปลั๊ก -อิน (Plug-in) สำหรับงานนั้นขึ้นมา และนำปลั๊ก – อิน (Plug-in) นั้นมาติดตั้งเพิ่มเติมให้กับ อิคลิป(Eclipse) ที่มีอยู่เท่านั้นอิคลิปปลั๊ก-อิน (Eclipse Plug -Ins) ที่มีมาพร้อมกับอิคลิป (Eclipse) เมื่อเราดาวน์โหลด (Download) มาครั้งแรกก็คือองค์ประกอบที่เรียกว่าชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรม ภาษาจาวา (Java Development – Toolkit - :JDT) ซึ่งเป็นเครื่องมือในการเขียนและ แก้ไขข้อผิดพลาด (Debug) โปรแกรมภาษาจาวา (Java) และเพื่อให้สามารถใช้งานได้ตรงตามเป้าหมายจึงใช้ แอนดรอยด์เอสดีเค (Android SDK) ย่อมาจาก Android - Software Development Kit เป็นปลั๊ก-อิน (Plug-in) ของอิคลิป (Eclipse) ที่ทำให้อิคลิป (Eclipse) สามารถเขียนภาษาจาวา (java) ที่ใช้พัฒนาแอพพลิเคชั่นบนแอนดรอยด์ (Android) ได้

2.1 ขั้นตอนการติดตั้ง

ก่อนที่จะเริ่มใช้งาน กับอิคลิป (Eclipse) จะต้องมีตัวโปรแกรม กับอิคลิป (Eclipse) โดยสามารถ ดาวน์โหลด (Download) ตัวโปรแกรม อิคลิป (Eclipse) ได้ที่ http://www.eclipse.org/downloads/โดยอิคลิป (Eclipse) มีหลายรุ่น (Version) สามารถเลือกดาวน์โหลด (Download) ได้ตามความเหมาะสมขอ งโปรแกรมที่จะพัฒนา ขนาดของแต่ละ รุ่น (Version) จะมีขนาดของไฟล์ (File) แตกต่างกันไป

2.1.1 การดาวน์โหลดและติดตั้งแอนดรอยด์เอสดีเค (Android SDK)

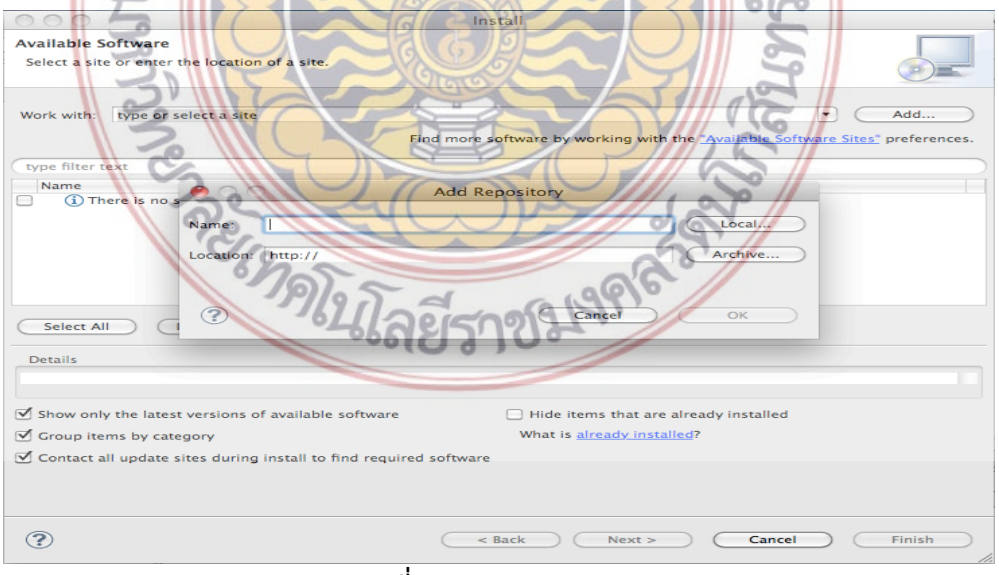
ในส่วนของแอนดรอยด์เอสดีเค (Android SDK) สามารถ ดาวน์โหลด (Download) ได้ที่ http://developer.android.com/sdk/index.html จากนั้นทำการเลือกตามแพลตฟอร์ม (Platform) ที่ผู้พัฒนานั้นใช้อยู่ถ้าวินโดว์ (Windows) ก็ทำการดาวน์โหลด (Download) ตัวที่แนะนำ (Recommend) ดังภาพที่ 21

ภาพที่ 21 ดาวน์โหลดแอนดรอยด์เอสดีเค (Android SDK Download)

ถ้าเป็นแมคแพลตฟอร์ม (Mac Platform) จะได้ไฟล์บีบอัด (Zip File) มาแล้วให้ทำการแตกไฟล์ไว้ออกมาแล้วไว้ในตำแหน่งที่อยู่ของไฟล์ (Directory) เป็นที่เข้าถึงได้ง่าย เช่น เอกสาร (Document) ถ้าเป็นวินโดว์แพลตฟอร์ม (Windows Platform) ก็จะได้ไฟล์ที่เป็น .exe ก็ทำการติดตั้งไว้ในที่เข้าถึง ได้ง่ายเช่นเดียวกัน

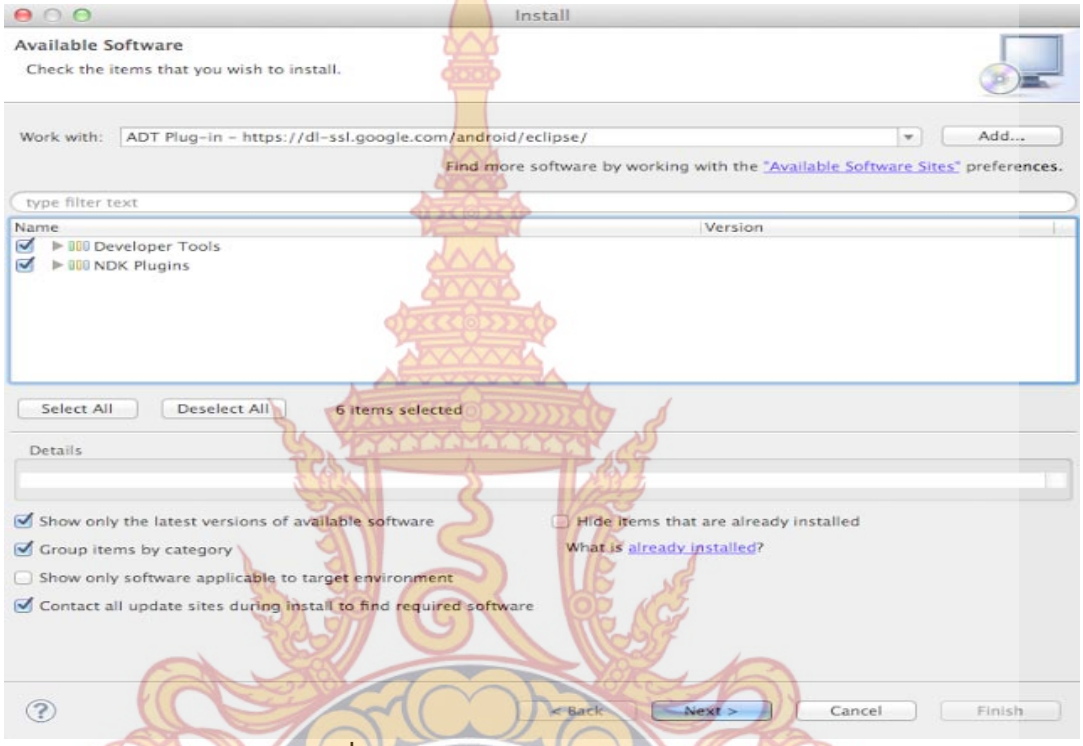
2.1.2 ขั้นตอนติดตั้งเอดีทีปลั๊กอิน (ADT Plug-In)

ทำโดยการเปิดโปรแกรมอิคลิป (Eclipse) ขึ้นมาแล้วให้ทำการเลือกเมนู (Menu) Help --> Install New Software จะทำให้ได้หน้าต่างใหม่แล้วให้กดที่ Add ทางด้านขวา แล้วจะมีขึ้น Dialog ของการเพิ่ม Add Repository ขึ้นมาดังภาพที่ 22



ภาพที่ 22 Add Repository

ทำการใส่การตั้งค่าดังนี้คือ Name: ADT Plug - in, Location: https://dl-ssl.google.com/android/eclipse/ แล้วกดปุ่ม OK แล้วรอจนกว่าการค้นหานั้นจะขึ้น Developer Tools ขึ้นมาก่อนแล้วจึงทำการเลือกถูกข้างหน้า Developer Tools แล้วกด Next ดังภาพที่ 23

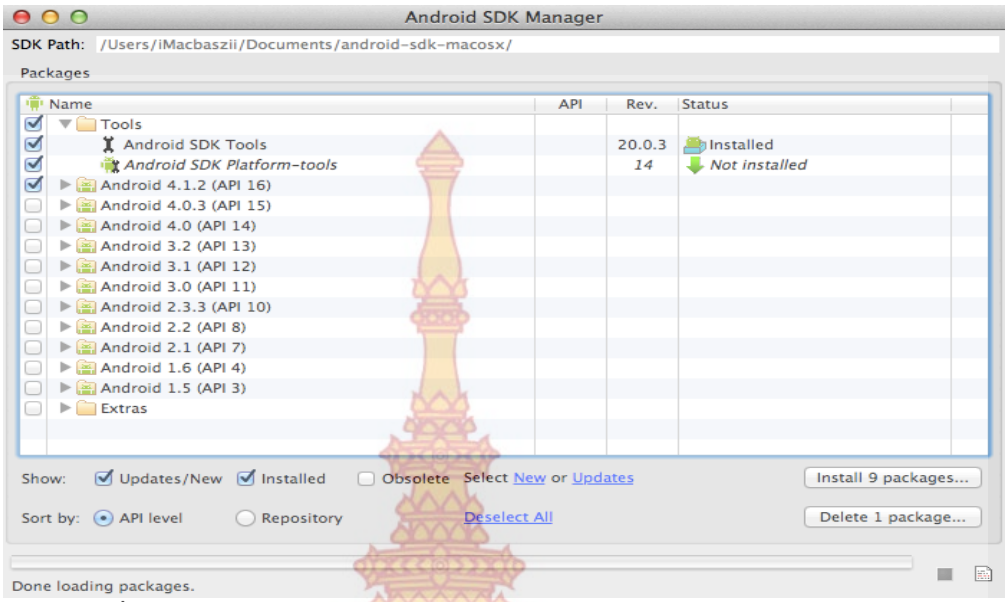


ภาพที่ 23 การค้นหา Developer Tools

จากนั้นจะมีการแจ้งว่าจะลงอะไรให้เราบ้างให้กด Next อีกครั้งแล้วการติดตั้งจะให้ยืนยันเกี่ยวกับข้อตกลง ให้เลือกที่ตกลง (I accept) แล้วนั้นกดเสร็จสิ้น (Finish) จากนั้นจะมีการโหลดตัว เอดีทีปลั๊กอิน (ADT Plug-In) เข้ามาให้รอไปเรื่อยๆ จนเสร็จ แล้วจึงทำการ Restart Eclipse

2.1.3 การเพิ่มแพลตฟอร์ม และคอมโพเนนท์ (Adding Platforms and Other - Components)

ทำการเลือก เมนู (Menu) ที่มีหน้าต่าง การจัดการแอนดรอยด์ เอสดีเค (Android SDK Manager) ซึ่งในอิคลิปสำหรับแมค (Eclipse for Mac) จะอยู่ในวินโดว์ (Window) จะได้หน้าต่างการจัดการแอนดรอยด์เอสดีเค (Android SDK Manager) ดังภาพที่ 24

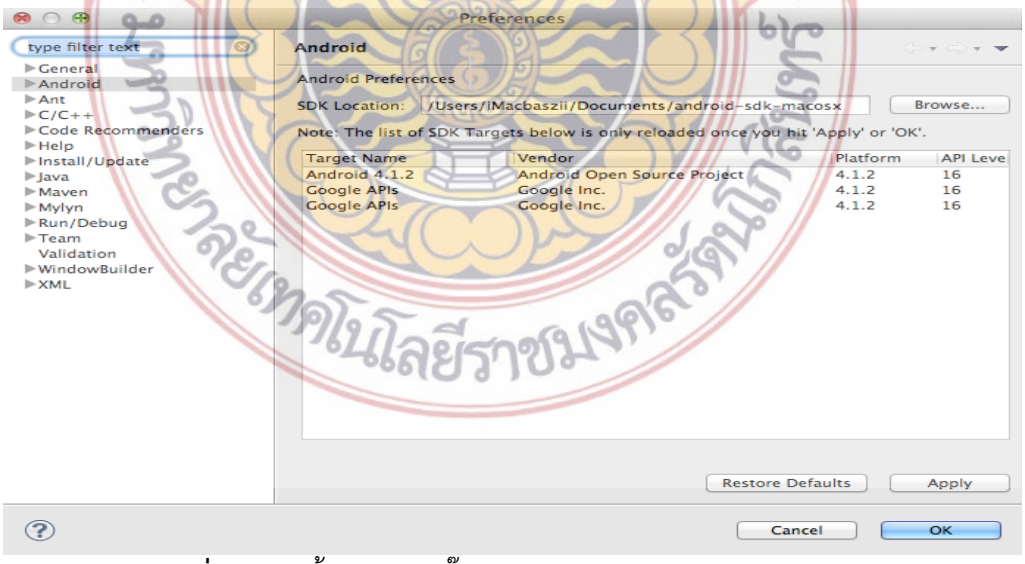


ภาพที่ 24 หน้าต่างการจัดการแอนดรอยด์เอสดีเค (Android SDK Manager)

ทำการเลือกว่าเราจะลงเวอร์ชันอะไรบ้าง ทั้งนี้ควรที่จะเลือกเว อร์ชันเดียวกันกับเครื่องโทรศัพท์ แล้วกดปุ่ม Install packge รอจนกว่าโปรแกรมก็ทำการดาวน์โหลด (Download) และติดตั้ง (Install) เสร็จสิ้น

2.1.4 การตั้งค่าเอดีทีปลั๊กอิน (Configuring the ADT Plug-in)

เมื่อเข้าโปรแกรม อิคลิป (Eclipse) อีกครั้งให้เลือกเมนู Window --> Preferences สำหรับวินโดว์แพลตฟอร์ม (Windows Platform) หรือ Eclipse --> Preferences สำหรับระบบปฏิบัติการแมค แพลตฟอร์ม (Mac OS Platform) แล้วให้เลือกที่ Android จะได้ดังภาพที่ 25

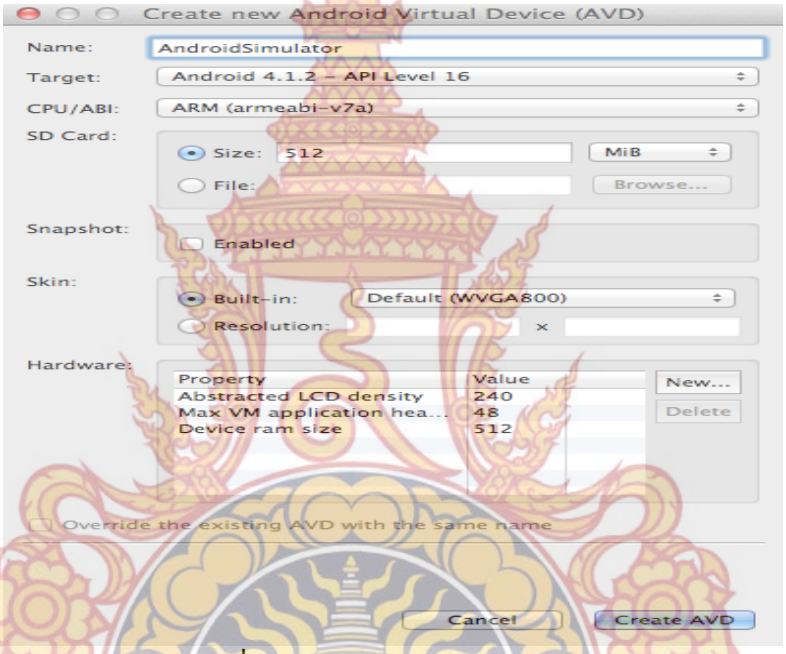


ภาพที่ 25 การตั้งค่าเอดีทีปลั๊กอิน (Configuring the ADT Plug-in)

ในช่อง SDK Location นั้นให้กดที่ Browse แล้วให้เลือก Folder ที่ทำการติดตั้งแอนดรอยด์เอสดีเค (Android SDK) ไว้ (หัวข้อที่ 2.1.1 การดาวน์โหลดและติดตั้งแอนดรอยด์เอสดี - เค) จากนั้นกดปุ่ม Apply แล้วปุ่ม OK

2.1.5 การตั้งค่าโทรศัพท์จำลองบนแอนดรอยด์ (Android Emulator)

เนื่องจากผู้พัฒนาบาง ท่านอาจจะไม่มีเครื่องโทรศัพท์ แอนดรอยด์ (Android) จริงทดสอบดังนั้นจึงต้องทำการติดตั้งตัวจำลอง โทรศัพท์จำลองบนแอนดรอยด์ (Android Phone) เพื่อจะทดสอบแอพพลิเคชั่นที่เขียนขึ้นมาในโปรแกรมอิคลิป (Eclipse) เลือก Window --> AVD Manager แล้วให้กดที่ปุ่ม New จะได้หน้าต่าง Create New AVD ดังภาพที่ 26



ภาพที่ 26 หน้าต่าง Create New AVD

ในเฉพาะวิมนโดว์แพลตฟอร์ม (Windows Platform) ให้เลือกช่อง Built-in เป็น HVGA ด้วย สำหรับผู้พัฒนาบางคนที่จะทดสอบอะไรพิเศษเพิ่มเติมสามารถที่จะ Custom Emulator ตามที่ต้องการได้ ในส่วนของ Hardware ผู้พัฒนาสามารถกด New เพื่อเพิ่มเนื้อที่ของแรม (Device Ram Size) หรืออื่นๆ ได้ จากนั้นคลิกที่ Create AVD ถือว่าเป็นการเสร็จสิ้น

ถ้าผู้พัฒนาต้องการทดสอบหลายเวอร์ชันก็ต้อง สร้าง AVD ไว้หลายๆ ตัว อย่างที่บอกว่า าเขียนแอพพลิเคชั่นเวอร์ชันไหน ก็ต้องใช้โทรศัพท์เวอร์ชันนั้นในเชิงของ AVD (Android Virtual Device) ก็เช่นกัน

2.1.6 ส่วนประกอบของอิคลิป (Eclipse) ที่จำเป็นต่อแอพพลิเคชั่น

ส่วนประกอบหลักของ อิคลิป (Eclipse) ที่จำเป็นต่อการพัฒนาแอพพลิเคชั่นนำทางผู้พิการทางสายตานั้นจะมีส่วนต่างๆ ดังนี้

2.1.6.1 แพ็คเกจเอ็กซ์โพลเลอร์ (Package Explorer)

แพ็คเกจเอ็กซ์โพลเลอร์ (Package Explorer) จะแสดงรายการ พื้นที่การทำงาน (Workspace) ต่างๆของโปรเจคที่สร้างขึ้นที่ทำในตัวโปรแกรม อิคลิป (Eclipse) ซึ่งจะมีส่วนประกอบหลักตามหมายเลขดังรูปภาพที่ 27 คือ

- หมายเลข 1 บอกถึง รุ่น (version) ของ ระบบปฏิบัติการแอนดรอยด์ (Android - Operation System)

- หมายเลข 2 SCR (Source code) เป็นส่วนของซอสโค้ดที่เราสร้างขึ้นและเขียนขึ้นเป็นไฟล์ .java ซึ่งจะเห็นชื่อคลาส (Class) แอตทริบิวส์ (Attribute) และเมธอด (Methods)

- หมายเลข 3 GEN (Generated Java Files) เป็นส่วนที่ถูกสร้างขึ้นโดยอัตโนมัติ เช่น ไฟล์ R.java ซึ่งภายในประกอบด้วย text และ UI Element โดยถูกนำเข้ามาในโปรเจคผ่าน แอนดรอยด์ปลั๊กอิน (Android Plug-in) ซึ่งไฟล์นี้จะเป็นเหมือนพอยด์เตอร์ (Pointer) ไปยัง drawable, layout, values directories

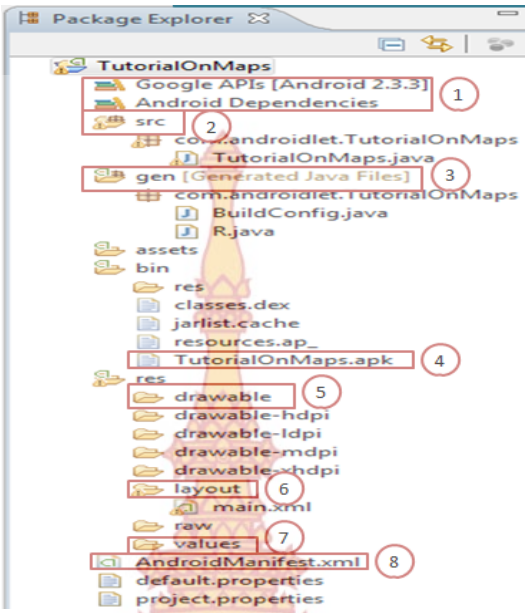
- หมายเลข 4 ไฟล์.apk ซึ่งใช้ในการติดตั้งลงบนโทรศัพท์เคลื่อนที่ ใช้ในการติดตั้งแอพพลิเคชั่นลงบนโทรศัพท์เคลื่อนที่

- หมายเลข 5 ใช้ในเก็บไฟล์ (File) รูปภาพที่ใช้ในแอพพลิเคชั่น

- หมายเลข 6 ใช้ในการออกแบบส่วนติดต่อผู้ใช้งาน (Graphic User Interface) ของแอพพลิเคชั่น

- หมายเลข 7 จัดการค่าต่างๆ ที่นำไปใช้ในแอพพลิเคชั่น ทั้งนี้ ก็ยังสามารถสร้าง ไดเรกทอรี (Directory) ต่าง ๆ ขึ้นเองเพื่อเก็บข้อมูลอื่นๆ ที่จะใช้ในแอพพลิเคชั่นได้

- หมายเลข 8 เป็นโครงสร้างของเอกซ์เอ็มแอล (XML) ไฟล์ ซึ่ง ในเอกซ์เอ็มแอล (XML) นั้นจะเป็นการกำหนดคุณสมบัติและการตั้งค่าต่างๆ ของ แอพพลิเคชั่นเช่น ชื่อของแอพพลิเคชั่นเวอร์ชันของโค้ดการกำหนดสิทธิ์ ต่าง ๆ ในการเข้าถึงของแอพพลิเคชั่น และอื่นๆดังภาพที่ 27

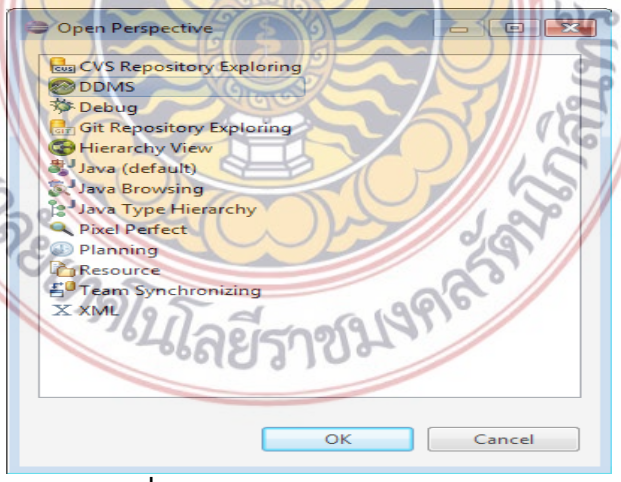


ภาพที่ 27 แพ็คเกจเอ็กซ์โพลเลอร์ (Package Explorer)

2.1.6.2 DDMS (Dalvik Debug Monitor Server)

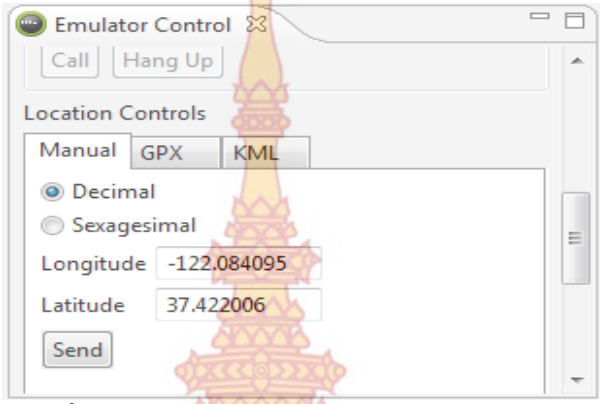
ในการรันแอพพลิเคชั่นบนอิมูเลเตอร์ (Emulator) ที่สร้างขึ้นมาแล้วนั้น นักพัฒนา จะต้องส่งค่าพิกัดจีพีเ อสที่แตกต่างกันให้แอพพลิเคชั่น เพื่อตรวจสอบว่าแอพพลิเคชั่นสามารถแสดงค่าพิกัดจีพีเอสได้ถูกต้องหรือไม่

ในส่วนนี้นั้นใช้ไ นการจำลองการ รับค่าพิกัดจีพีเอส เพื่อใช้ในการตรวจสอบความถูกต้องของตำแหน่งพิกัดจีพีเอส โดยเลือกที่เมนู Window => Open Perspective แล้วเลือกที่ DDMS ดังภาพที่ 28



ภาพที่ 28 หน้าต่าง Open Perspective

จากนั้นเลือกที่อีมูเลเตอร์คอนโทรล (Emulator Control) แล้วทำการใส่ค่าพิกัดจีพีเอสที่จำลองลงไปก็จะทำให้สามารถตรวจสอบความถูกต้องของตำแหน่งพิกัดจีพีเอสได้ดังภาพที่ 29



ภาพที่ 29 อีมูเลเตอร์คอนโทรล (Emulator Control)

3. การออกแบบแอพพลิเคชั่นนำทาง

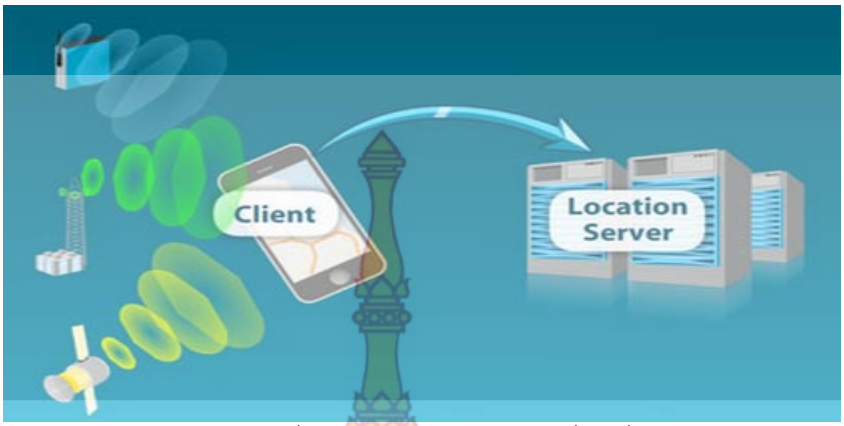
ในส่วนนี้นั้นเหมาะสำหรับ ใช้ในผู้ที่มีสายตาปกติ ซึ่งจะมีองค์ ประกอบต่างๆ ที่ช่วยเพิ่มความสะดวกให้กับผู้ใช้ การดูสถานที่ที่อ ยู่ปัจจุบัน การดูสภาพการจราจร การเปลี่ยนรูปแบบแผนที่การนำทาง และเมนูต่างๆ ซึ่งจะมีการอธิบายโดยละเอียดถึงหลักการทำงานดังนี้

3.1 การเรียกใช้งานข้อมูลแผนที่ของกูเกิล (Google Map)

ในการออกแบบแอพพลิเคชั่นจะแบ่งกลุ่มของชุดคำสั่งออกเป็น 2 ส่วนเพื่อให้ง่ายต่อการแก้ไขและใช้พัฒนาต่อ โดยแต่ละส่วนแบ่งออกเป็น

ส่วนที่ 1 Device เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อและควบคุ มการทำงานของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการกูเกิลแอนดรอยด์ (Google Android)

- com.google.android.maps.MapActivity ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในก ารเรียกใช้งานแผนที่ของกูเกิล (Google Map)
- com.google.android.maps.MapController ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในการควบคุมการทำงานของแผนที่ เช่น การซูม (Zoom) และเลื่อนแผนที่
- android.provider.Settings ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในการตั้งค่า เครือข่าย (Network) ต่างๆส่วนที่ 2 เซ็นเซอร์ (Sensors) เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อ และควบคุมเซนเซอร์ (Sensors) ชนิดต่างๆ ที่ใช้งานในโทรศัพท์เคลื่อนที่ระบบปฏิบัติการ กูเกิลแอนดรอยด์ (Google Android)
- android.location.LocationListener ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในติดต่อและควบคุมการรับสัญญาณจีพีเอส (GPS) ดังภาพที่ 30



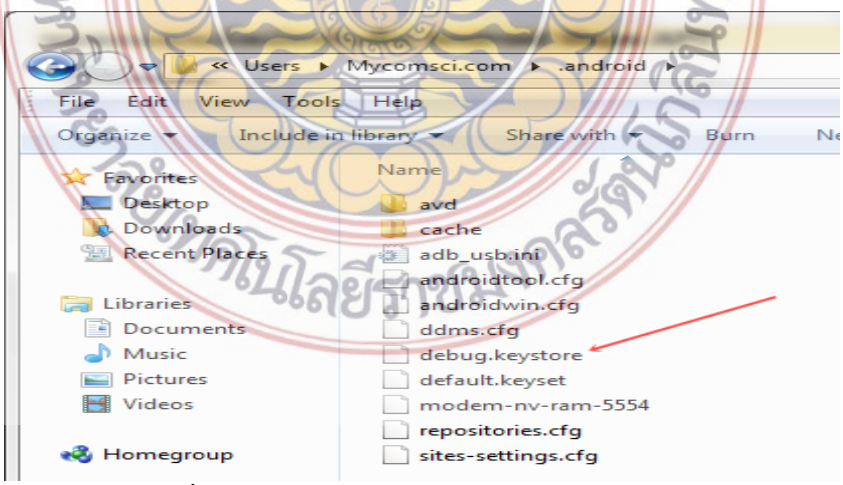
ภาพที่ 30 การทำงานของข้อมูลแผนที่ของกูเกิลบนโทรศัพท์เคลื่อนที่ (Google Maps for Mobile)

3.1.1 ขั้นตอนการขอ Google Map API Key

เอพีไอ (API) ย่อมาจาก Application Programing Inteface คือการที่ผู้ใช้ (Client) สามารถเรียกใช้ข้อมูล (Data) ความสามารถ (Method) ของอีกโปรแกรมหนึ่ง หรือเว็บหนึ่ง ผ่านทางช่องทางใดช่องทางหนึ่ง ที่ผู้ให้บริการ (Service Provider) ได้กำหนดไว้ แค่รู้ว่าจะเรียกดูข้อมูลหรือเรียก ใช้บริการนั้นอย่างไรสร้างการเชื่อมต่ออย่างไร ส่งข้อมูลอะไรให้บ้าง และจะได้ผลลัพธ์ออกมาอย่างไรเท่านั้น

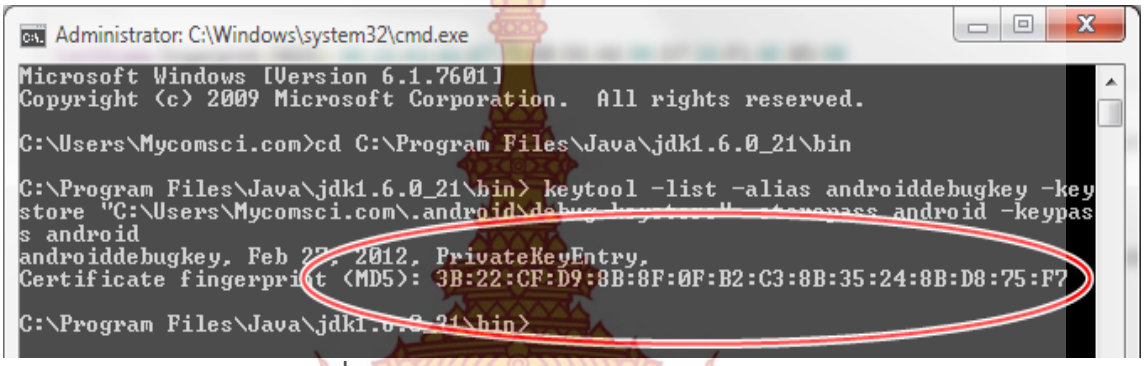
สรุปคำว่า Google Maps API นั้นเป็นบริการของกูเกิล (Google) อีกรูปแบบหนึ่งที่เราสามารถนำข้อมูลของแผนที่ของกูเกิล (Google Map) ที่ทางกูเกิล (Google) เข้าให้บริการโดยส่วนมากจะนำมาใช้กับเว็บไซต์ของ บริษัทฯหรือเว็บไซต์ห้างร้านต่างๆ เพื่อเป็นอีกช่องทางที่ให้ลูกค้ารู้ว่าบริษัทฯหรือห้างร้านนั้นอยู่ที่ใดของประเทศ

การขอ Google Maps API Key เพื่อนำ key มาใช้บนแอพพลิเคชั่นของเราเพื่อสามารถเรียกใช้แผนที่บนแอพพลิเคชั่นของเรา ก่อนอื่นเราต้องหาไฟล์ debug.keystore เพื่อจะนำ Certificate Fingerprint MD5 ออกมาดังภาพที่ 31



ภาพที่ 31 ตัวอย่าง directory ของ debug.keystore

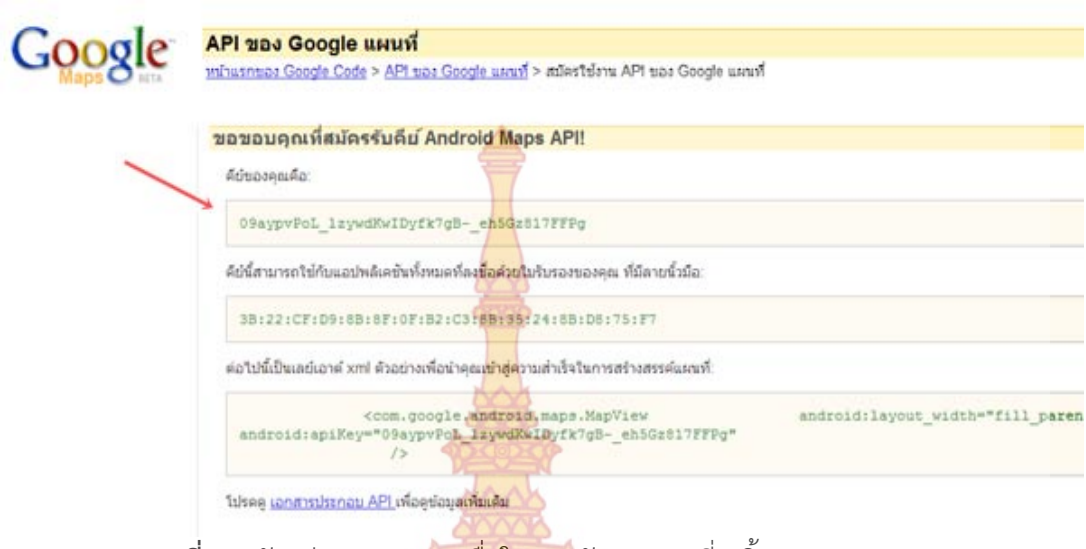
สำหรับวินโดว์ (Window) มักจะอยู่ที่ c:/Users/user ที่คุณใช้อยู่ /.android เปิด cmd พิมพ์: cd C:\Program Files\Java\jdk1.6.0_21\bin จากนั้นพิมพ์ : keytool - list - alias androiddebugkey - keystore "C:\Users\Mycomsci.com\.android\debug.keystore"- storepass android - keypass android ก็จะทำให้ได้ Certificate Fingerprint MD5 ออกมา ดังภาพที่ 32



ภาพที่ 32 ตัวอย่าง Certificate Fingerprint MD5

เมื่อได้ MD5แล้วไปยังเว็บhttps://developers.google.com/android/maps-api-signup?hl=th ดังภาพที่ 33 และภาพที่ 34

ภาพที่ 33 ตัวอย่างการสร้าง API Key



ภาพที่ 34 ตัวอย่าง API Key เพื่อในการพัฒนาแผนที่กูเกิ้ล (Google Map)

3.1.2 ขั้นตอนการขอ Google Map API Key

API ย่อมาจาก Application Programming Interface คือการที่ผู้ใช้ (Client) สามารถเรียกใช้ข้อมูล (Data) ความสามารถ (Method) ของอีกโปรแกรมหนึ่ง (หรือเว็บหนึ่ง) ผ่านทางช่องทางใดช่องทางหนึ่ง ที่ผู้ให้บริการ (Service provider) ได้กำหนดไว้ แค่รู้ว่าจะเรียกดูข้อมูลหรือเรียกใช้บริการนั้นอย่างไรสร้างการเชื่อมต่ออ อย่างไร ส่งข้อมูลอะไรให้บ้าง และจะได้ผลลัพธ์ออกมาอย่างไรเท่านั้น

3.1.3 การสร้างชุดคำสั่งการทำงาน

เพิ่มซอสโค้ด (Source Code) เข้าไปในไฟล์ AndroidManifest.xml เพื่อระบุว่าใช้ที่จัดเก็บแผนที่ (Maps Library) จากกูเกิ้ล (Google)

```
<uses-library android: name = “com.google.android.maps” />
```

และซอสโค้ด (Source Code) เข้าไปที่เพื่ อระบุว่าแอพพลิเคชั่นนี้ต้องใช้อินเตอร์เน็ต (Internet) ในการใช้งานด้วยเนื่องจากต้องดึงข้อมูลแผนที่จากกูเกิล (Google Map)

```
<uses-permission android: name = “android.permission.INTERNET” />
```

เพิ่มซอสโค้ด (Source Code) เข้าไปในไฟล์ main.xml เพื่อออกแบบส่วนติดต่อผู้ใช้งาน (User Interface File) ของแอพพลิเคชั่นแล้วทำการใส่ Google API Key เข้าไปตรง android:apiKey

```
<?xml version = “1.0” encoding = “utf-8”? />
<RelativeLayout
        xmlns : android = http://schemas.android.com/apk/res/android”
        android : Layout_width = “fill_parent”
        android : Layout_heigh = “fill_parent”
        android : orientation = “vertical”>
<com.google.android.maps.mapView
        android : id = “@+id/mapView”
        android : layout_width = “fill_parent”
        android : layout_height = “fill_parent”
        android : clickable = “true”
        android : apiKey = “”>
</com.google.android.maps.MapView>
</RelativeLayout>
```

เพิ่มซอสโค้ด (Source Code) เข้าไปในไฟล์ .java เพื่อออกแบบการทำงานของแอพพลิเคชั่ ที่ใช้ในการแสดงภาพแผนที่ (mapView) ซึ่งจะทำให้ได้หน้าตา แอพพลิเคชั่นดังในภาพที่ 35

```
/** Called when the  activity is first created. */
LinearLayout linearLayout;
MapView mapView;
@ Override
Public void onCreate (Bundle savedInstancestate) {
        super.onCreate (savedInstanceState);
        setContentView(R.layout.main);
        mapView = (MapView)findViewById(R.id.mapView);
        mapView.setBuiltInZoomControls(true);
        }
```

ภาพที่ 35 ตัวอย่างแผนที่กูเกิลบนแอนดรอยด์ (Google Map on Android)

3.1.4 แผนภาพการเรียกใช้งานข้อมูลแผนที่ของกูเกิล (Google Map)

ภาพที่ 36 แผนภาพการเรียกใช้งานข้อมูลแผนที่ของกูเกิล (Google Map)

3.2 การติดตามด้วยจีพีเอส (GPS Tracking)

ในการออกแบบแอพพลิเคชั่นจะแบ่งกลุ่มของชุดคำสั่งออกเป็น 2 ส่วน เพื่อให้ง่ายต่อการแก้ไขและใช้พัฒนาต่อ โดยแต่ละส่วนแบ่งออกเป็น

ส่วนที่ 1 Device เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อและควบคุมการทำงานของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการ Google Android

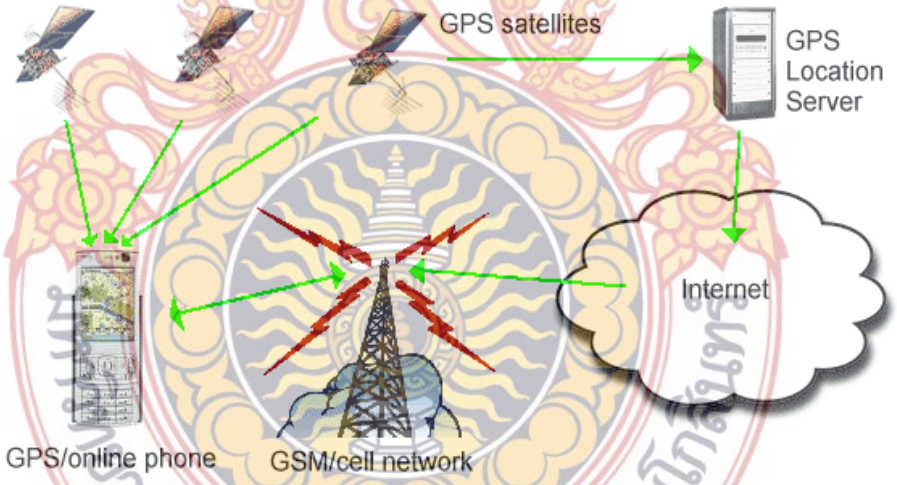
- com.google.android.maps.MapActivity ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในการเรียกใช้งานแผนที่ของ Google

- com.google.android.maps.Overlay ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในการใส่รูปภาพโดยใช้ตำแหน่งพิกัดจีพีเอสลงบนแผนที่

ส่วนที่ 2 เซ็นเซอร์ (Sensors) เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อ และควบคุมเซนเซอร์ชนิดต่างๆ ที่ใช้งานในโทรศัพท์เคลื่อนที่ระบบปฏิบัติการ กูเกิลแอนดรอยด์ (Google - Android)

- android.location.LocationListener ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในติดต่อการรับสัญญาณจีพีเอส

- android.location.LocationManager ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในควบคุมการรับสัญญาณจีพีเอสดังภาพที่ 37



ภาพที่ 37 หลักการทำงานของตำแหน่งพิกัดจีพีเอสในแผนที่

3.2.1 การสร้างชุดคำสั่งการทำงาน

ในส่วนแรกคือการกำหนดตัวแปรสำคัญให้กับแอพพลิเคชั่นเพื่อใช้สำหรับติดต่อและควบคุมการทำงานของการรับสัญญาณพิกัดจีพีเอส (GPS)

```
locMgr = (LocationManager)getSystemService(Context.LOCATION_SERVICE) ;
locLstnr = new MyLocationListener();
locMgr.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, locLstnr);
```

แล้วทำการกำหนด แอตทริบิวต์ (Attribute) เพื่อใช้ในการเก็บค่าของพิกัดของตำแหน่งที่ถูกเก็บในตัวแปรที่เรียกว่า ละติจูด (Latitude) และลองติจูด (Longitude)

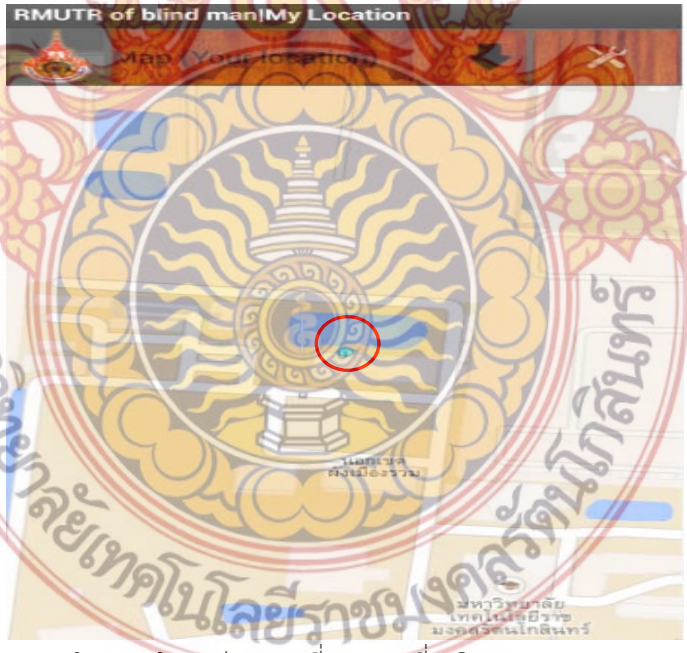
```
public double lat; // Latitude
public double lng; // Longitude
```

จากนั้นสร้างคลาส MyLocationListener () ที่ใช้ในการดึงค่าข้อมูลสัญญาณจีพีเอส (GPS) โดยมีฟังก์ชั่น on LocationChanged()ที่ใช้ดึงค่าข้อมูลสัญญาณจีพีเอสโดยใช้คำสั่ง loc.getLatitude() และ loc.getLongitude แล้วเก็บค่าไว้ใน แอททริบิวต์ (Attribute) แล้วมีฟังก์ชั่น onProviderDisabled() และ onProviderEnabled() ไว้ตรวจสอบการทำงานของการเปิด – ปิด อุปกรณ์รับสัญญาณจีพีเอส (GPS)

```
public class MyLocationlistener implements LocationListener
{
        @ Override
        Public void onLocationChanged (Location loc)
        {
        loc.getLatitude(); // get parameter Latitude
        loc.getLongitude(); // get parameter longitude
        String coordinates [ ] = { “”+loc.getLatitude(), “”+loc.getLongitude()} ;
        Lat = Double.parseDouble(coordinates[0]);
        Lng =  Double.parseDouble(coordinates[1]);
GeoPoint p = new GeoPoint((int) (lat * 1E6), (int) (lng * 1E6)); // draw marker
location
}
@Override
Public void onProviderDisabled (String provider)
}
        // Check GPS Provider Disabled
{
@Override
Public void onProviderEnabled (String provider)
{
        // Check GPS Provider Enabled
```

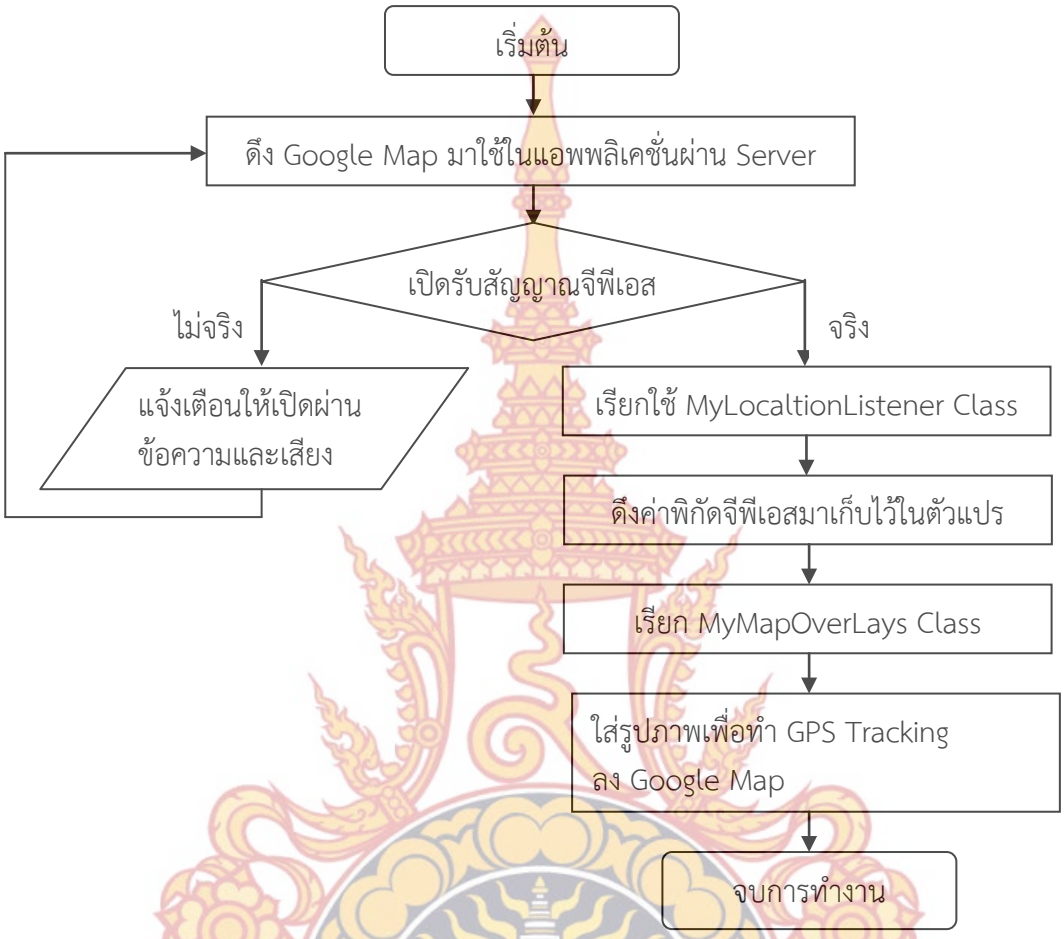
จากนั้นสร้างคลาส (Class) MyMapOverlays() ใช้การใส่รูปภาพโดยใช้ตำแหน่งพิกัดจีพีเอส (GPS) ลงบนแผนที่ เพื่อจะได้ทราบว่าอยู่ส่วนไหนข องแผนที่ซึ่งจะ ทำให้ได้หน้าตาแอพพลิเคชั่นดังภาพที่ 38

```
Class MyMapOverlays extends com.google.android.maps.Overlay
{
        GeoPoint location = null;
        Public MyMapOverlays (GeoPoint location)
        {
        Super ();
        This.location = location;
        }
@Override
Public void draw (Canvas canvas, MapView mapView, Boolean shadow)
        {
        Super.draw(canvas, mapView, shadow);
        // translate the screen pixels
        //add  the image on the map
        }
}
```



ภาพที่ 38 ตัวอย่างการกำหนดตำแหน่งสถานที่บนแผนที่กูเกิล (Google Map Location Marker)

3.2.2 แผนภาพการทำงานระบบติดตามด้วยจีพีเอส (GPS Tracking)



ภาพที่ 39 แผนภาพการทำงานระบบติดตามด้วยจีพีเอส (GPS Tracking)

3.3 สภาพการจราจรบนท้องถนนในแผนที่

ในแอพพลิเคชั่นนี้สามารถที่จะดูสภาพการจราจรบนท้องถนนได้ โดยไปดึงข้อมูลสภาพการจราจรมาจาก บริการ (Service) ของทาง กูเกิล (Google) ที่มีการเก็บข้อมูลไว้เป็นแบบเรียลไทม์ (Real Time) ซึ่งจะแบ่งสภาพการจราจรบนท้องถนนออกเป็นสามส่วน โดยจะแสดงสภาพการจราจรออกเป็นเส้นสีตามท้องถนน คือ การจราจรติดขัดมาก การจราจรติดขัดปานกลาง และการจราจรคล่องตัวซึ่งจะมีสีแสดงบนท้องถนน คือ สีแดง สีเหลือง และสีเขียว ตามลำดับ

ในการออกแบบแอ พพลิเคชั่นจะมีกลุ่มของชุดคำสั่ง เพื่อให้ง่ายต่อการแก้ไขและใช้พัฒนาต่อ โดยมี Device เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อ และควบคุมการทำงานของโทรศัพท์ เคลื่อนที่ระบบปฏิบัติการกูเกิลแอนดรอยด์ (Google Android Operation System)

- com.google.android.maps.MapActivity ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในการเรียกใช้งานแผนที่ของกูเกิล (Google Map)

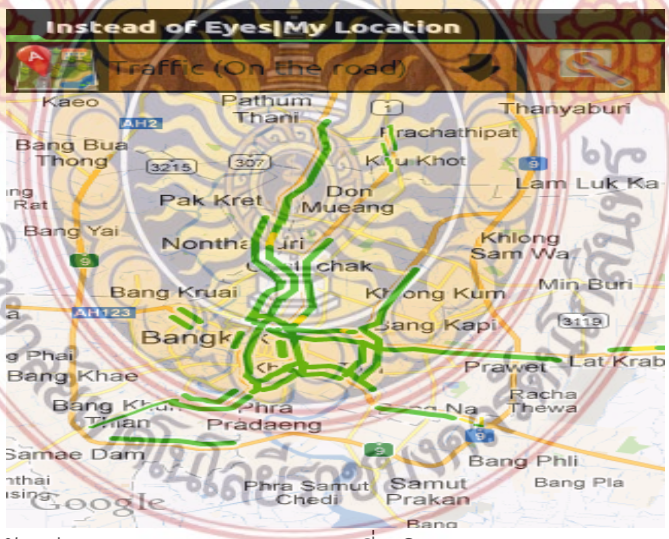
- com.google.android.maps.Overlay ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชันในกา รใส่รูปภาพโดยใช้ตำแหน่งพิกัดจีพีเอสลงบนแผนที่

3.3.1 การสร้างชุดคำสั่งการทำงาน

ในส่วนนี้คือฟังก์ชันการทำงานของการแสดงสภาพการจราจรบนท้องถนนลงในแผนที่กูเกิล (Google Map) โดยจะมีการทำงานเมื่อทำการสัมผัสหน้าจอที่เป็น ปุ่มดูการจราจร (Button Traffic) ว่าจะให้แสดงหรือไม่

```
Public void  toggleTraffic() {
        If (mapView.isTraffic())
                mapView.setTraffic(flase); // Disable traffic on map
else
                mapView.setTraffic(true); // Enabled traffic on map
mapView.invalidate();
}
```

จากนั้นก็ทำการแสดงสภาพการจราจรด้วยสีเส้น โดยจะแสดงสภาพการจราจรออกเป็นเส้นสีตามท้ องถนน คือ การจราจรติดขัดมาก การจราจรติดขัด ปานกลาง และการจราจรคล่องตัว ซึ่งจะมีสีแสดงบนท้องถนนคือ สีแดง สีเหลือง และสีเขียวตามลำดั บ ลงไปในแผนที่กูเกิล (Google Map) จะทำให้ได้การทำงานดังภาพที่ 40



ภาพที่ 40 ตัวอย่างสภาพการจราจรบนแผนที่กูเกิล (Google Map Traffic Marker)

3.3.2 แผนภาพการทำงานสภาพการจราจรบนท้องถนนบนแผนที่กูเกิล (Google Map)

ภาพที่ 41 แผนภาพการทำงานสภาพการจราจรบนท้องถนนบนแผนที่

3.2.3 การเปลี่ยนรูปแบบแผนที่

การเปลี่ยนรูปแบบแผนที่นั้นทำเพื่อการเปลี่ยนโหมดรูปแบบแผนที่จากโหมดแผนที่ (Map) เป็นโหมดดาวเทียม (Satellite) ซึ่งจะทำให้ได้มุมมองที่ต่างกันในด้านสภาพแวดล้อม

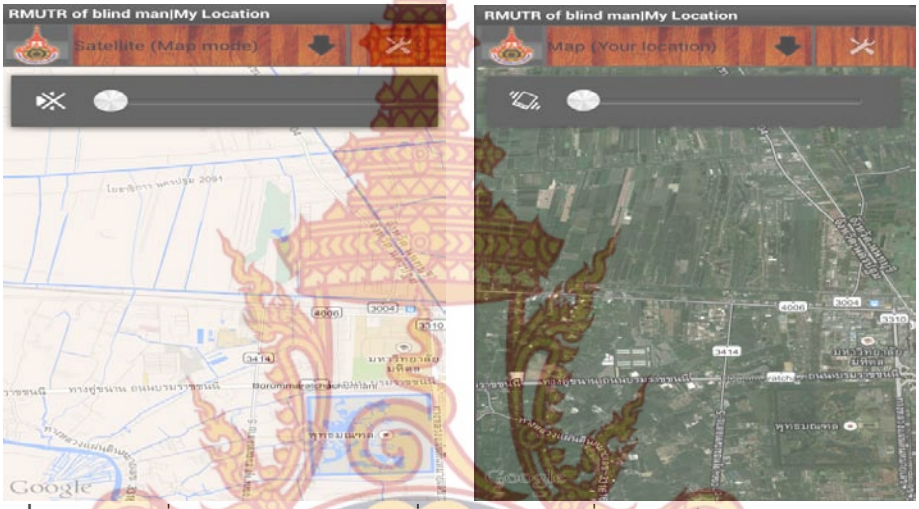
ในการออกแบบแอพพลิเคชั่นจะมีกลุ่มของชุดคำสั่ง เพื่อให้ง่ายต่อการแก้ไขและใช้พัฒนาต่อ โดยมี Device เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อและควบคุมการทำงานของโทรศัพท์ เคลื่อนที่ระบบปฏิบัติการกูเกิลแอนดรอยด์ (Google Android Operation System)

- com.google.android.maps.MapActivity ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการเรียกใช้งานแผนที่กูเกิล (Google Map)

3.2.3.1 การสร้างชุดคำสั่งการทำงาน

ในส่วนนี้คือฟังก์ชั่นการทำงานของการเปลี่ยนโหมดรูปแบบแผนที่ กูเกิล (Google Map) โดยจะมีการทำงานเมื่อทำการสัมผัสหน้าจอที่เป็น การรับสัญญาณด าวเทียม (Spinner Satellite) ว่าจะให้แสดงโหมดอะไรตามความต้องการของผู้ใช้ ซึ่งจะทำให้ได้หน้าตาแอพพลิเคชั่นดังภาพที่ 42

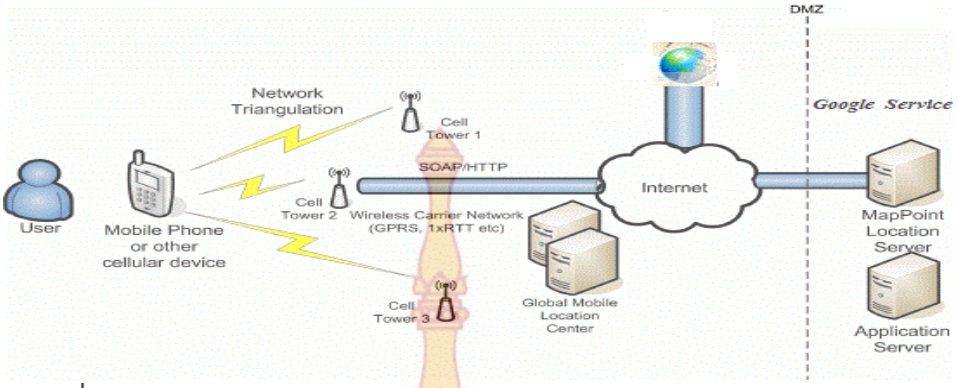
```
public void toggleSatellite() {
if (mapView.isSatellite))
mapView.setSatellite (false); // Change to Map Mode
else
mapView.setSatellite(true); // Change to Satellite Mode
mapView.invalidate();
}
```



ภาพที่ 42 การเปลี่ยนรูปแบบโหมดแผนที่จากโหมดแผนที่ (ซ้าย) เป็นโหมดดาวเทียม (ขวา)

3.2.4 การนำทาง (Navigation)

ในส่วนของการนำทางนี้มีคุณส มบัติที่จะสามารถนำทางไปได้ในทุก ๆ สถานที่ โดยการ ค้นหาสถานที่ปลายทางลงไปไม่ว่าจะเป็นภาษาไทยหรืออังกฤษก็ตาม แล้วจะมีค่าการตอบกลับเป็นค่าพิกัดจีพีเอส โดยอ้างอิงข้อมูลจาก Google Location-based Service ที่ให้บริการการค้นหาตำแหน่งพิกัดจีพีเอสผ่านสถานที่ ซึ่งได้อาศัยไลบรารีในการเข้าถึงข้อมูล นั้นๆ เมื่อได้รับค่าพิกัดจีพีเอสแล้วจึงทำการค้นหาเส้นทาง (Route Path) โดยใช้หลักการในการหาเส้นทางที่ใกล้ที่สุดในการที่จะเดินทางจากต้นทางไปปลายทาง (Shorted Path) เพื่อลดระยะทางและเวลาที่ใช้ในการเดินทางให้มากที่สุดดังภาพที่ 43



ภาพที่ 43 หลักการทำงานของการดึงข้อมูลผ่าน Google Location-based Service

3.2.4.1 การค้นหาสถานที่ (Location Finder)

ในการออกแบบแอพ พลิเคชั่นจะมีกลุ่มของชุดคำสั่ง เพื่อให้ง่ายต่อการแก้ไขและใช้พัฒนาต่อ โดยมีเซ็นเซอร์ (Sensors) เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อข้อมูลและควบคุมเซนเซอร์ชนิดต่างๆ ที่ใช้งานในโทรศัพท์เคลื่อนที่ระบบปฏิบัติการ กูเกิลแอนดรอยด์ (Google Android Operation System)

- android.location.Geocoder ชุดคำสั่งนี้ใช้สำหรับส ร้างฟังก์ชั่นในการหาตำแหน่งพิกัดจีพีเอสผ่านการค้นหาชื่อสถานที่
- android.location.Address ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการหาตำแหน่งพิกัดจีพีเอสผ่านการค้นหาชื่อบ้านเลขที่ของสถานที่สำคัญ
- android.location.LocationListener ชุดคำสั่งนี้ใช้สำ หรับสร้างฟังก์ชั่นในติดต่อการรับสัญญาณจีพีเอส
- android.content.Intent ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการเรียกหน้าการแสดงผลอื่นที่อยู่ในแอพพลิเคชั่น

3.2.4.1.1 การสร้างชุดคำสั่งการทำงาน

ในขั้นตอนแรกคือการเรียกใช้งานหน้าแอพพลิเคชั่นในกา รที่จะหาพิกัดจีพีเอสจากสถานที่ซึ่งจะใช้ชุดคำสั่งดังนี้เพื่อให้แยกการทำงานจากหน้าแอพพลิเคชั่นเดิม

```
startActivity (new Intent(LocationAndTraffic.this,LocationFinder.class));
```

ซึ่งเมื่อทำการเรียกหน้าแอพพลิเคชั่นในการหาพิ กัดจีพีเอสจากสถาน ที่มาแล้วนั้น ก็จะใช้ไลบรารี่ android.location.Geocoder และ android.location.Address ในการที่จะ ตรวจสอบและดึงค่าพิกัดจีพีเอสมา เพื่อใช้เป็นพิกัดปล ายทางที่ผู้ใช้ต้องการให้นำทางไป โดยเก็บค่าพิกัดจีพี เอสที่ได้จากสถานที่ไว้ในตัวแปล Dest lat และ Dest_lng ซึ่งมีเงื่อนไขการตรวจสอบเพื่อป้องกันแอพพลิเคชั่นเกิดข้อผิดพลาดเอาไว้แล้วเมื่อมีการหาค่าพิกัดจีพีเอสจากสถานที่นั้นๆ ได้จึงจะสามารถทำงาน

```
// Set up the button for “Location Finder”
Button geoBtn = (Button) findViewById(R.id.geocodeBtn);
Geocoder = new Geocoder (this);
geoBtn.setOnClickListener(new OnClickListener() {
        @Override
        Public void onClick(View v) {
        try {
        EditText locale = (EditText) findViewById(R.id.Location);
         //pull data form EditText
        String locationName = locale.getText().toString();
        List<Address> addressList = geocoder.getFromLocationName(
        locationName, 5);
        // check Latitude and Longitude from Address and location
        If (adressList ! = null && addressList.size() > 0) {
        Dest_lat = addressList.get(0).getLatitude();
        //get Latitude from Address and location
        }
        } catch (Exception e) {
        }
        }
});
```

ส่วนพิกัดจีพีเอสต้นทางนั้นจะรับค่าจากพิกัดจีพีเอสตำแหน่งปัจจุบันที่ได้จากอุปกรณ์รับสัญญาณจีพีเอสโดยตรงในโทรศัพท์เคลื่อนที่ โดยเก็บค่าพิกัดจีพีเอสที่ได้จากอุปกรณ์รับสัญญาณจีพีเอสไว้ในตัวแปล Src_lat และ Src_lng ที่อยู่ในคลาส MyLocationListener ซึ่งอธิบายไว้แล้วในหัวข้อที่ 3.2 ซึ่งจะทำให้ได้หน้าตาแอพพลิเคชั่นดังในภาพที่ 44

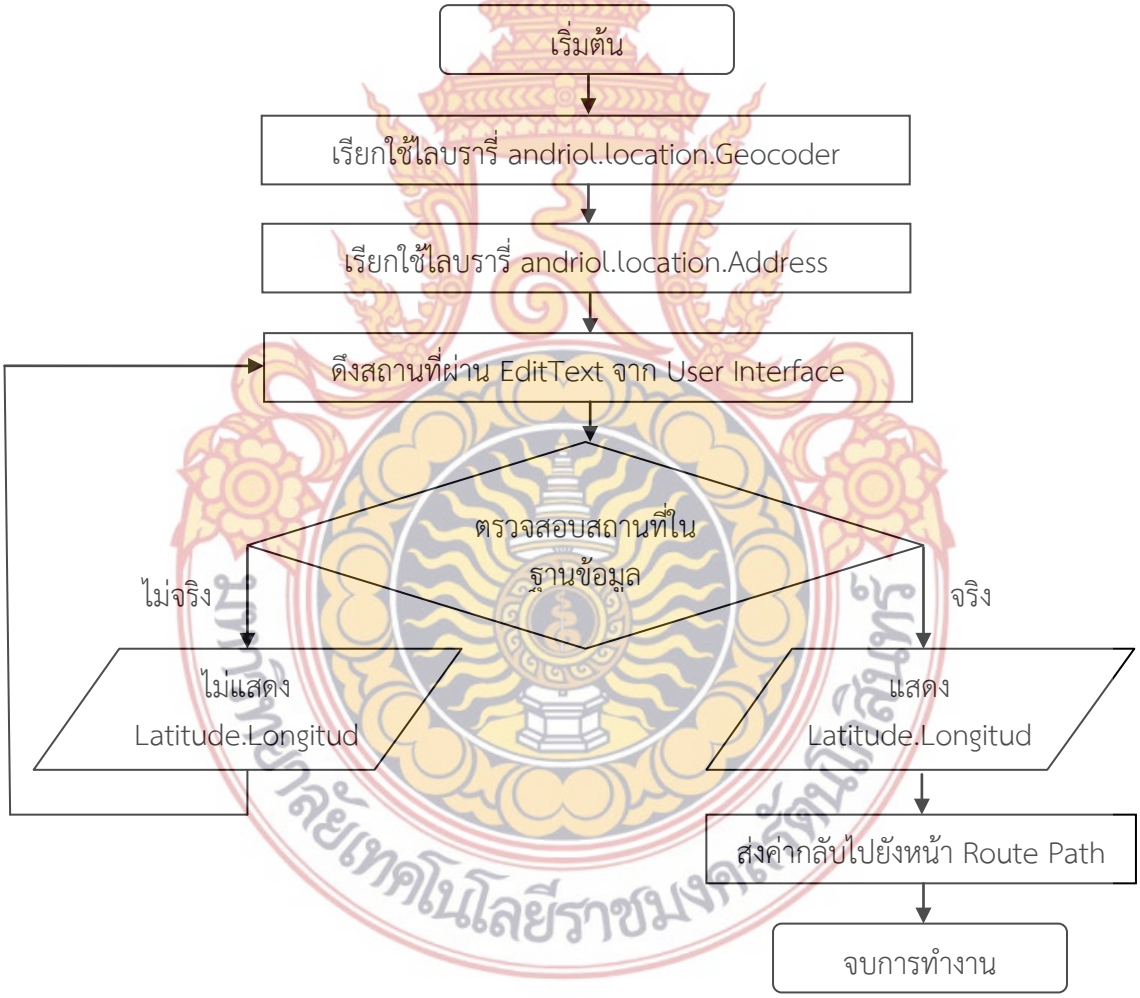
```
public class MyLocationListener implements LocationListenner
{
        @ Override
        public void onLocationChanged (Location loc)
{
        loc.getLatitude();
        loc.getLongitude();
        String coordinates[] = {“”+loc.getLatitude(), “”+loc.getLongitude()};
        Src_lat = Double.parseDouble(corrdinates[0]); // get Latitude
        Src_lng = Double.parseDouble(corrdinates[1]);//get Longitude
}
```

ภาพที่ 44 การหาค่าพิกัดจีพีเอสผ่านสถานที่

จากนั้นก็จะทำการส่งค่าในตัวแปลที่เก็บพิกัดจีพีเอสต้นทางและปลายทาง และ API Key ซึ่งอธิบายไว้ในหัวข้อที่ 3.1 ไปในอีกหน้าแอพพลิเคชั่นที่ใช้ในการ การค้นหาเส้นทาง (Route Path) เพื่อนำทางไปในสถานที่ต้องการไปโดยมีลักษณะการส่งค่าดังนี้

```
SharedData data = SharedData.getInstance();
data.setAPIKEY (“  ”); //to geneate API key
data.setSrc_lat(Src_lat);  // sent data source latitude
data.setSrc_lng(Src_lng); // sent  data source longitude
data.setDest_lat(Dest_lat); // sent data destination latitude
data.setDest_lng(Dest_lng); //sent data destination longitude
startActivity(new Intent(LocationFinder.this,RoutePathNormal.class));
// go to new GUI application
```

3.2.4.1.2 แผนภาพการทำงานการค้าหาสถานที่ (Location Finder)



ภาพที่ 45 แผนภาพการทำงานการค้าหาสถานที่ (Location Finder)

3.2.4.2 การค้นหาเส้นทาง (Route Path)

ในการออกแบบแอพพลิเคชั่นจะแบ่งกลุ่มของชุดคำสั่งออกเป็น 2 ส่วน เพื่อให้ง่ายต่อการแก้ไขและใช้พัฒนาต่อ โดยแต่ละส่วนแบ่งออกเป็น

ส่วนที่ 1 Device เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อข้อมูล และควบคุมการทำงานของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการ กูเกิลแอนดรอยด์ (Google Android - Operation System)

- com.google.android.maps.MapActivity ชุดคำสั่งนี้ใช้สำหรับสร้าง ฟังก์ชั่นในการเรียกใช้งานแผนที่ของ Google

- com.google.android.maps.Overlay ชุดคำสั่งนี้ใช้สำหรับสร้าง ฟังก์ชั่นในการ ใส่รูปภาพโดยใช้ตำแหน่งพิกัดจีพีเอสลงบนแผนที่

- android.content.Intent ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการ เรียกหน้าการแสดงผลอื่นที่อยู่ในแอพพลิเคชั่น

- org.w3c.dom.Node ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการข้าถึง หลักข้อมูลของเอกสาร HTML หรือ XML

- javax.xml.parsers ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการแสดง ข้อ ผิดพลาดการกำหนดค่า

ส่วนที่ 2 เซ็นเซอร์ (Sensors) เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการ ติดต่อ และควบคุมเซนเซอร์ชนิดต่าง ๆ ที่ใช้งานใน โทรศัพท์เคลื่อนที่ระบบปฏิบัติการ กูเกิลแอนดรอยด์ (Google Android Operation System)

- android.location.LocationListener ชุดคำสั่งนี้ใช้สำหรับสร้าง ฟังก์ชั่นในติดต่อและควบคุมการรับสัญญาณจีพีเอส

3.2.4.2.1 การสร้างชุดคำสั่งการทำงาน

เมื่อมีการรับค่ามาจากหน้าแอพพลิ เคชั่นการหาพิกัดจีพีเอสผ่าน สถานที่มาแล้วนั้นก็จะทำการคำนวณหาเส้นท างที่สั้นที่สุดหรือที่เรียกว่า Shorted Path ในการที่จะนำทางไป ถึงจุดหมายปลายทางที่ได้ค้นหาไว้แล้ว โดยใช้หลักการการเรียกข้อมูลเส้นทางมาจาก บริการของกูเกิล (Google Service) ที่เปิดให้บริการในการของตำแหน่งเส้นทางพิกัดจีพีเอสที่อยู่ใน บริเวนต้นทางและปลายทางที่อยู่ ในฟังก์ชั่น fetchdata() ซึ่งต้องทำการส่งค่าพิกัดจีพี เอสที่ต้นทาง และปลายทางไปให้ทางบริการของกูเกิล (Google Service) พร้อมกับโหมดเส้นทาง (Route Type - Mode) ไปพร้อมกันด้วย ซึ่งในแอพพลิเคชั่นนี้เลือกใช้โหมดการเดิน (Walking Mode) เพื่อให้ได้พิกัด จีพีเอสที่ละเอียดมากยิ่งขึ้น

```
StringBuilder urString = new StringBuilder();
urlString.append("http://mapsgoogle.com/maps/api/directions/aml?origin=");
        //fetch data from google service
urlString.append (Doible.tostring(double)srcGeoPoint.getLatitudeE6()1.0E6));
        //sent source latitude
urlString.append(" , ");
urlString.append(Double.toString((double) srcGeoPoint.getLongitudeE6()1.0E6));
        //sent source longitude
urlString.append(" &destination="); // to
urlString.append(Double.toString((double) srcGeoPoint.getLongitudeE6()1.0E6));
        //sent destination latiitude
urlString.append(" , ");
urlString.append(Double.toString((double) srcGeoPoint.getLongitudeE6()1.0E6));
        //sent destination longitude
urlString.append("&sensor=true&mode=walking"); //route type walking mode
Log,d ("url" , ": :" +urlString.toString());
```

แล้วจึงทำการเรียกข้อมูลมาเพื่อเตรียมการคำนวณหาเส้นทางที่สั้นที่สุด (Shorted Path) ในขั้นต่อไป โดยมีฟังก์ชั่น onPreExecute() ในการเช็คการรอเรียกข้อมูลและฟังก์ชั่น doInBackground() ที่บ่งบอกว่าให้ทำการรันข้อมูล (Run Data) ในส่วนของพื้นหลัง ที่เกี่ยวกับหน้าของแอพพลิเคชั่น

```
 @Override
Protected void onPreExecute() { // TODO Auto – generated method stub
        super.onPreExcute();  //Fetching route, Please wait...
}
@Override
protect  doInBackground(Void...programs) {
        // TODO Auto – generated method stub
        // call fetchData() ; function
        return null;
}
```

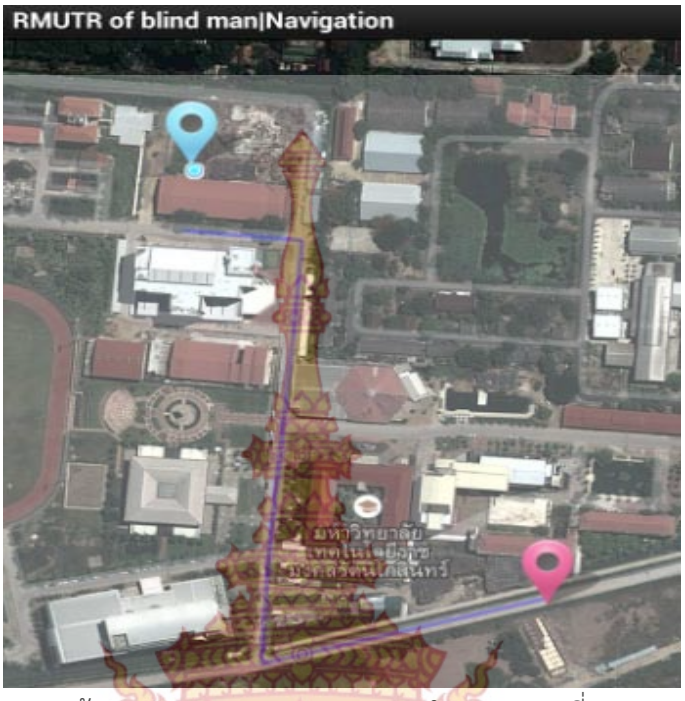
จากนั้นก็นำข้อมูลที่เรียกมาได้นั้นมาทำการ หาเส้นทาง (Route Path) ออกมาจากค่าพิกัดจีพีเอส (GPS) ที่เรียกมาได้

```
Overlay ol = MyOverLay (_activity,srcGeoPoint,srcGeoPoint,1);
        mOverLays.add(ol);
NodeList_nodelist = doc.getElementsByTagName(“status”);
Node node1 = _node1.getChildnode().item(0).getNodeValue();
        If (_status1.equalsIgnoreCase(“OK”)){
NodeList_nodelist_path = doc.getElementsByTagName(“overview_polyline”
Node node_path = _nodelist_path.item(0);
Element _status_path = (Element)node_path;
Nodelist _nodelist_destination_path = _status_path.getElementsByTagName_
(“points”);
Node _nodelist_dest = _nodelist_destination_path.item(0);
String_path = _nodelist_dest.getChildNodes().item(0).getNodeValue();
List<Geopoint> _geopoints = decodePoly(_path) ;
        GeoPoint gp2;
        Gp2 = _geopoints.get(0);
        Log.d(“_geopointsget(0);
        Log.d(“geopoints “,” : : “ +_geopoints.size());
        for (int i=1;i<geopoints.size(); i++) // the last one would be crash
        {
        gp1 = gp2;
        gp2 = _geopoints.get(i);
Overlay ol1 = new MyOverLay(gp1, gp2, 2, Color. BLUE);
mOverLays.add(old2);
progressDialog.dismiss();
        }else{
        showAlert(“Unable to find the route”);
}
```

จากนั้นทำการวาดเส้นการหาเส้นทาง (Route Path) และใส่ตำแหน่งจุดเริ่มต้นและปลายทางการเดินทางเป็นรูปแสดงลักษณะ

```
Private List<Geopint> decodePoly(String encoded) {
List<GeoPoint> poly = new ArrayList<Geopoint> ();
I       nt index = 0, len = encoded.length();
        int lat 0, lng = 0;
        while (index < len) {
        int b, shift = 0,result = 0;
        do {
        b = encode.charAt(index++) -63;
        result | = (b & 01f) << shift;
        shift +=5;
        { while (b>=0×20);
        int dlat = ((result &1) !=0? ~(result >>1 : (result >>1));
        lat += dlat;
        shift = 0
        result = 0
        do {
        b  = encoded.charAt(index++) – 63;
        result | = (b&o×1f) << shift;
        shift += 5;
        { while (b>=0×20);
        int dlng = ((result & 1) != 0 ? ~(result >>1) : (result >>1));
        lng += dlng;
        GeoPoint p =new Geopoint ((int) (((double) lat / 1E5) * 1E6),
        (int) (((double) lng / 1E5) * 1E6));
        poly.add(p);
}
        return poly;
}
```

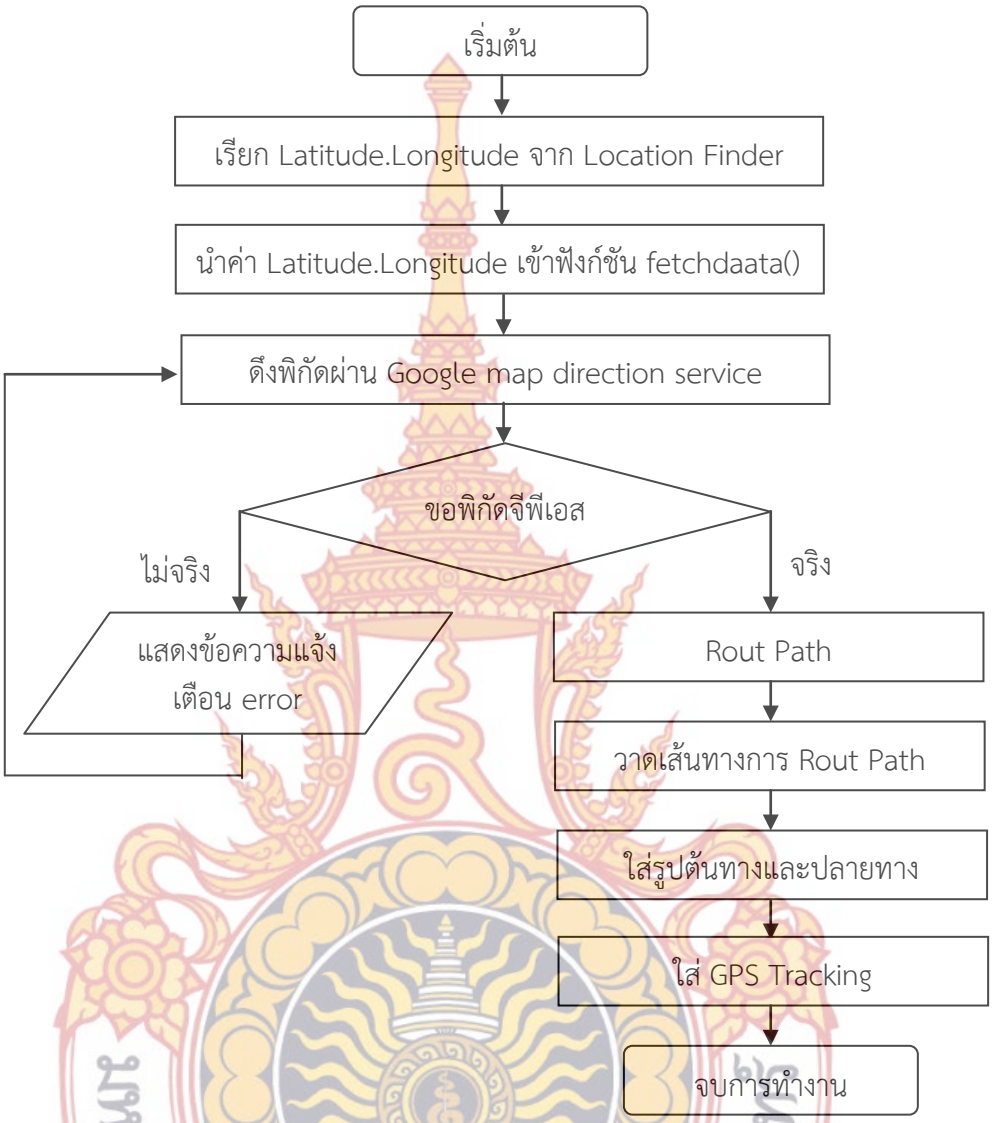
ในขั้นตอนสุดท้ายคือการ กำหนด (Marker) ตำแหน่งปัจจุบันที่ผู้ในนั้นอยู่ เพื่อใช้ในการตรวจสอบว่า ผู้ใช้นั้นได้เดินไปในการหาเส้นทาง (Route Path) ไว้หรือไม่ โดยใช้คลาส (Class) ซึ่งได้กล่าวไว้แล้ว ในหัวข้อที่ 3.2 ซึ่งจะทำให้ได้หน้าตาแอพพลิเคชั่นดังในภาพที่ 46



ภาพที่ 46 การหาเส้นทาง (Route Path) และการกำหนดสถานที่ (Location Marker)

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์

3.2.4.2.2 แผนภาพการทำงานการค้นหาเส้นทาง (Route Path)



ภาพที่ 47 แผนภาพการทำงานการค้นหาเส้นทาง (Route Path)

3.2.5 การตั้งค่า (Setting)

การตั้งค่านั้นจะเป็นส่วนในการที่จะเปิด -ปิด ตัวรับสัญญาณจีพีเอส และตัวรับสัญญาณเครือข่ายไร้สาย (WIFI) หรือ 3G ที่จำเป็นที่จะต้องเปิดไว้ในขณะใช้แอพพลิเคชั่นเพื่อดึงข้อมูลแผนที่ และบริการต่างๆของทางกูเกิล (Google)

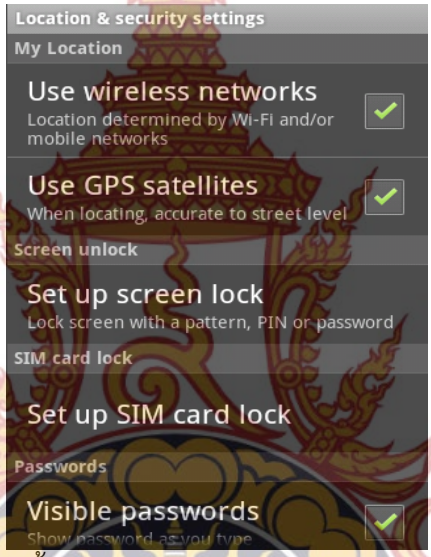
ในการออกแบบแอพ พลิเคชั่นจะมีกลุ่มของชุดคำสั่ง เพื่อให้ง่ายต่อการแก้ไขและใช้พัฒนาต่อ โดยมี Device เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อและควบคุมการทำงานของโทรศัพท์ เคลื่อนที่ระบบปฏิบัติการกูเกิลแอนดรอยด์ (Google Android Operation System)

- android.content.Intent ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการเรียกหน้าการแสดงผลอื่นที่อยู่ในแอพพลิเคชั่น

3.2.5.1 การสร้างชุดคำสั่งการทำงาน

ซึ่งการทำงานนี้จะเรียก ว่าการตั้งค่าการรักษาความปลอดภัย (Action Security) ที่ใช้ในการตั้งค่าของผู้ให้บริการเครือข่าย (Network Provider) ต่างๆขึ้นมา ดังภาพที่ 48

```
Intent myIntent =new Intent (Setting.ACTION_SECURITY_SETTING;
// call intent ACTION SECURITY SETTINGS
strartActivity(myIntent); //Activity start
```



ภาพที่ 48 การตั้งค่าการรักษาความปลอดภัย (Action Security)

3.2.6 ตัวเลือกเมนู (Option Menu)

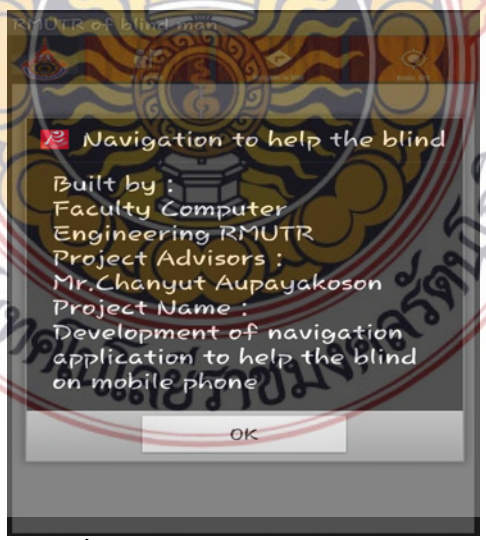
ในส่วนเมนูนี้นั้นภายในจะมีรายละเอี ยดเกี่ยวกับผู้จัดทำแอพพลิเคชั่น เช่น ชื่อ-นามสกุล ชื่อมหาวิทยาลัย ชื่อโครงการ เป็นต้น

ในการออกแบบแอพ พลิเคชั่นจะมีกลุ่มของชุดคำสั่ง เพื่อให้ง่ายต่ อการแก้ไข และใช้พัฒนาต่อ โดยมี Device เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อ และควบคุมการทำงานของโทรศัพท์ เคลื่อนที่ระบบปฏิบัติการกูเกิลแอนดรอยด์ (Google Android Operation System)

- android.view.Menu ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการสร้าง ตัวเลือกเมนู (Option Menu) โดยการกดที่ปุ่มเมนู (Menu) ของโทรศัพท์ดังภาพที่ 49

3.2.6.1 การสร้างชุดคำสั่งการทำงาน

```
public boolean onOptionsItemSelected (MenuItem item) {
        switch (item.getItemId()) {
        case ID_MENU_ABOUT:
        (new AlertDialog.Builder(SharedDataActivity.this)
        .setTitle("Navigation to help the blid") //title
        .setIcon(R.drawable.eye2)
        .setPositiveButton ("OK", new DialogInterface.OnClickListener() {
// create OK button
        public void onClick(DialogInterface dialog, int whichButton) {
/* Do something for when button is checked*/
        }}).create()).show();
        break;
        default:
        break;
        default:
        break;
}
        return true;
}
```



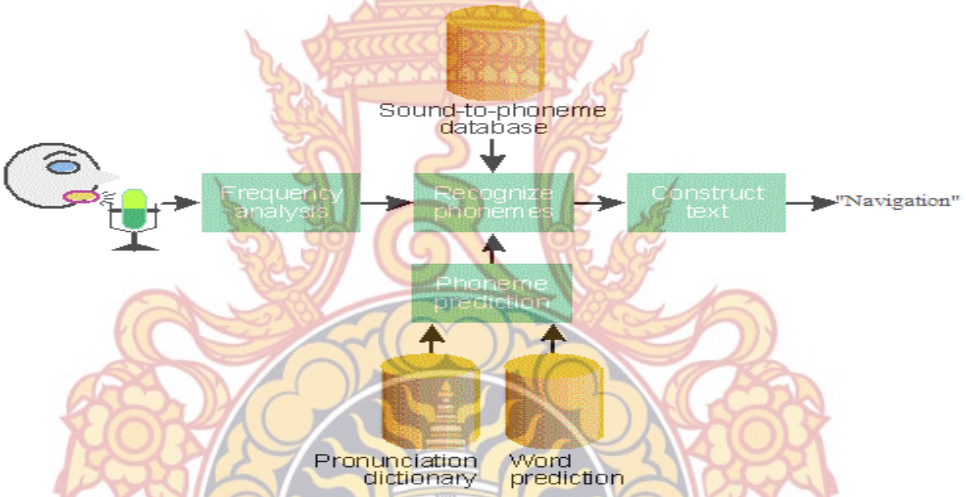
ภาพที่ 49 เมนูตัวเลือก (Option Menu)

4. การออกแบบและพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตา

ในส่วนนี้นั้นจะเพิ่มการทำงานเพิ่มเข้ามาจาก หัวข้อที่ 2. เพื่อให้เหมาะสำหรับการนำทางผู้พิการทางสายตาแต่มีการทำงานพื้นฐานที่เหมือนกันการทำงานที่เพิ่มเข้ามานั้นได้แก่ การสั่งงานด้วยเสียงการเก็บพิกัดสถานที่ปลายทางไว้ใน ฐานข้อมูลเอสคิวไลท์ (Database SQLite) มีเสียงตอบสนองและการเตือนและการนำทางด้วยสียงที่บอกรูปแบบเส้นทางตลอดการนำทาง ซึ่งมีรายละเอียดดังต่อไปนี้

4.1 การสั่งงานด้วยเสียง (Speech Recognition)

การสั่งงานด้วยเสียงหรือที่เรียกกันว่า Speech Recognition โดยมีหลักการการทำงาน คือ รับเสียงมาจากไมโครโฟนบนโทรศัพท์เคลื่อนที่ (Mobile Microphone) แล้วทำการวิเคราะห์เสียงผ่านฐานข้อมูล (Database) ที่เก็บไฟล์ตรวจสอบเสียงไว้ จนได้เป็นข้อความ (Text) ออกมาแล้วจึง นำข้อความ (Text) เหล่านั้นมาเปรียบเทียบข้ อความที่ต้องการให้ทำตามคำสั่ง เพื่อใช้ในการควบคุมการทำงานต่อไปดังภาพที่ 50



ภาพที่ 50 หลักการทำงานของการสั่งงานด้วยเสียง (Speech Recognition)

ในการออกแบบแอพ พลิเคชั่นจะมีกลุ่มของชุดคำสั่ง เพื่อให้ง่ายต่อการแก้ไข และใช้พัฒนาต่อ โดยมีเซ็นเซอร์ (Sensors) เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อข้อมูลและควบคุมเซนเซอร์ชนิดต่างๆ ที่ใช้งานในโทรศัพท์เคลื่อนที่ ระบบปฏิบัติการ กูเกิลแอนดรอยด์ (Google - Android Operation System)

- android.speech.RecognizerIntent ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการใช้งานการสั่งงานด้วยเสียง (Speech Recognition)

4.1.1 การสร้างชุดคำสั่งการทำงาน

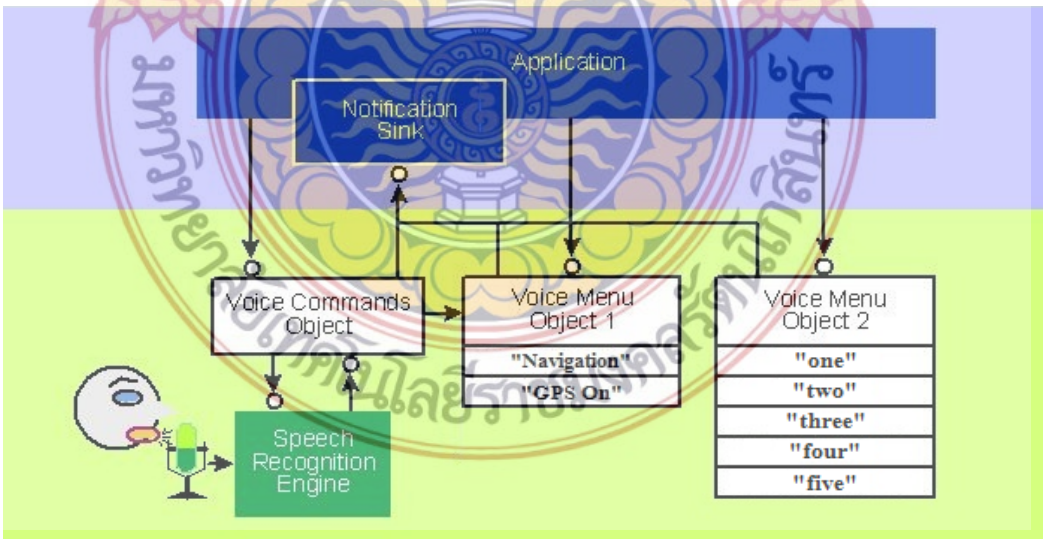
ในการสั่งงานด้วยเสียงนั้ นภาษาที่ใช้สั่งการคือภาษาอังกฤษ โดยการเรียกหน้า RecognizerIntent ขึ้นมาเพื่อเป็นหน้าการทำงานในการที่จะรับคำสั่งด้วยเสียง

```
Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
//call Recognizer Intent
Intent.putExtra (RecognizerIntent.EXTRA_LANGUAGE_MODEL, “en-US”);
// Language speech input is English
```

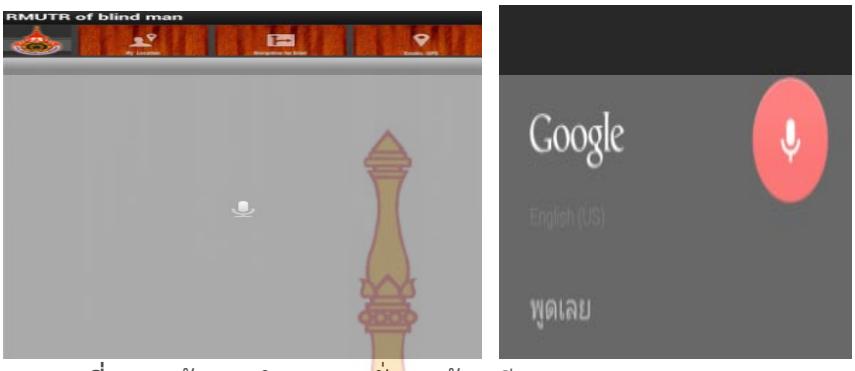
เมื่อรับคำสั่งด้วยเสียงก็จะทำการวิ เคราะห์เสียงให้ออกมาเป็น ข้อความ (Text) โดยมีวิธีการดั่งเช่นใน ภาพที่ 50 แล้วจึงนำข้อความ (Text) นั้นมาเปรียบเทียบกับชุด ตัวแปรสตริง (String) ที่เตรียมไว้ ที่ต้องการให้ใช้คำสั่งด้วยเสียง ถ้าหากข้อความ (Text) นั้นมีข้อความตรงกับชุดตัวแปรสตริง (String) ที่เตรียมไว้ก็จะทำการเรียกฟังก์ชั่นการทำงานที่อยากจะให้ทำต่อไป

```
ArrayList<String> textGet_SPEECH = data
        .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS); // text output
String textCheck_SPEECH1 = “ ”; // string check
If (textCheck_SPEECH1.equals(textGet_SPEECH.get(0)))
{
        // call function to run
}
```

ซึ่งคำสั่งด้วยเสียงที่ใช้ในแอพพลิเคชั่นนี้นั้นมีอยู่ หลักๆ ด้วยกันอยู่สองชุดคำสั่งซึ่งมีชุดคำสั่งการดังภาพที่ 51 และมีหน้าการทำงานเช่นในดังภาพที่ 52



ภาพที่ 51 ชุดคำสั่งเสียงที่ใช้ในแอพพลิเคชั่น



ภาพที่ 52 หน้าการทำงานการสั่งงานด้วยเสียง (Speech Recognition)

4.2 การเก็บสถานที่ปลายทางบนฐานข้อมูลเอสคิวไลท์ (Destination Location on - SQLite)

การเก็บสถานที่ปลายทางไว้ในฐานข้อมูลเอสคิวไลท์ (Database SQLite) นั้นมีจุดประสงค์เพื่อเก็บสถานที่ปลายทางที่ต้องการที่จะไปสำหรับผู้พิการทางสายตา ซึ่งเป็นสถานที่ที่ผู้พิการทางสายตานั้นใช้เดินทางไปเป็นประจำ จำนวนทั้งสิ้นห้าเส้นทางหลัก ซึ่งในห้าเส้นทางนั้นจำสามารถแก้ไขสถานที่ปลายทางได้เพื่อให้ยืดหยุ่นในการใช้งานให้ได้มากที่สุด

ในการออกแบบแอพพลิเคชั่นจะมีกลุ่มของชุดคำสั่งเพื่อให้ง่ายต่อการแก้ไขและใช้พัฒนาต่อ โดยมี Device เป็นกลุ่มของชุดคำสั่งเกี่ยวกับการติดต่อและควบคุมการทำงานของโทรศัพท์ เคลื่อนที่ระบบปฏิบัติการกูเกิลแอนดรอยด์ (Google Android Operation System)

- android.database.sqlite.SQLiteDatabase ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการใช้ในการเปิด/ปิด Connection ของ Database และการ Query , Insert , Update , Delete
- android.database.sqlite.SQLiteOpenHelper ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการใช้สร้าง (Create) และเพิ่มฐานข้อมูล (Update Database) ของเรา
- android.database.Cursor ชุดคำสั่งนี้ใช้สำหรับสร้างฟังก์ชั่นในการใช้จัดการกับคิวรี่ (Query) ที่ได้จากฐานข้อมูลเอสคิวไลท์ (SQLite Database) ซึ่งโดยปกติแล้วจะใช้ในคู่กับการเลือกข้อมูล

4.2.1 การสร้างชุดคำสั่งการทำงานของการสร้าง วัตถุ (Object) และไฮเปอร์คลาส (Helper Class)

ทำการสร้างคลาส (Class) ชื่อ Contact.java ในคลาสนี้สร้างขึ้นเพื่อเก็บสร้างวัตถุ (Object) ต่างๆในตาราง (Table) ในคลาสนี้ได้ทำการสร้างคลาสที่มีการเก็บฟิลล์ (Field) ต่างๆของตาราง (Table) ตามตัวอย่างนี้มีสี่คอลัมน์ (Column) คือ id, name, latitude และ longitude

```
Public class Contact { // private variables
Int _id;
String_name;
double_latitude;
double_longitude; // Empty constructor
public Contact() {
}
Public Contact(int id) { // constructor
this Contact (int id);
{
Public Contact(int id, string name, double latitude, double longitude){
this._id = id; // id location (1-5)
this._name=name; //latitude;
this._longitude = longitude;
}
```

ทำการสร้าง Class DatabaseHandler ขึ้นมาคลาส (Class) จำเป็นจะต้องมีเมธอด (Method) โดยมี method DatabaseHandler ที่เป็น constructor และ onCreate() และ onUpdate() ที่สร้างมาให้อัตโนมัติ และจำเป็นจะต้องมี

```
plblic class DatabaseHandler extends SQLiteOpenHelper {
// All Static variables
// Database Version
private static final int DATABASE_VERSION = 1;
//Database Name
Private static final String DATABASE_NAME = “LocationManager2”;
// Contacts table name
Private static final String TABLE_LOCATION = “location2”;
// Contact Table Columns names
Private static final String KEY_ID = “id”;
Private static final String KEY_NAME_LOC = “name_location”;
Private static final String KEY_LAT = “latitude”;
Private static final String KEY_LNG = longtitude;
Public DatabaseHandler (Context context) {
super (context, DATABASE_NAME, null, DATABASE_VERSION);
}
//Creating Table
@Override
Public void onUpgrade (SQLiteDatabase db2) {
String CREATE_CONTACTS_TABLE = “CREATE TABLE” + TABLE_LOCATION + “(“
+ KEY_ID + “INTEGER PRIMARY KEY,” + KEY_NAME_LOC +” TEXT,”
+KEY_LAT + “ DOUBLE,” + KEY_LNG + “ DOUBLE “+”);
db2.exeSQL(CREATE_CONTACTS_TABALE);
@Override
Public void onUpgrade(SQLiteDatabase db2,int oldVersion, int newVersion) {
// Drop older table if existed
db2.execSQL (“DROP TABLE IF EXISTS” + TABLE_LOCATION);
//create table again
onCreate(db2);
}
```

ในฐานข้อมูล (Database) นี้นั้นมีชื่อว่า LocationManager2 และมีชื่อตารางคือ location2 ซึ่งจะประกอบไปด้วยสี่คอลัมน์ (column) คือ id, name, latitude และlongitude

ในคลาส (Class) ของ SQLiteOpenHelper ที่จะใช้จัดการกับ ฐานข้อมูล (Database) ของเอสคิวไลท์ (SQLite)มีคลาส (Class) ที่เป็นโอเวอร์รายด์เมธอด(Override Method)

อยู่ที่ 2 ตัวที่จำเป็นจะต้องใช้คือ onCreate()และ onUpgrade() โดย เมธอด(Method) จะทำงานภายใต้เหตุการณ์วัตถุที่เกิดขึ้นระหว่างการเรียกใช้งานฐานข้อมูลเอสคิวไลท์ (SQLite Database) เช่น เมื่อมีการเรียกใช้ Propertyว่า getReadableDatabase() และ getWriteableDatabase() ซึ่งเป็น Property ที่เกี่ยวข้องกับการ Read-Write ของฐานข้อมูล(Database) การทำงานของ method onCreate() and onUpgrade() ก็จะเกิดขึ้น

ดังนั้น onCreate() จึงนิยมแทรกคำสั่งสำหรับการสร้างฐานข้อมูล และการสร้างตารางซึ่งจะทำงานแรกสุดเมื่อโปรแกรม ฐานข้อมูลเอสคิวไลท์ (SQLite Database) เริ่มทำงาน onUpgrade() นิยมใช้สำหรับการเปลี่ยนแปลง รุ่น (Version) หรือโครงสร้างของ ฐานข้อมูล (Database) และตาราง (Table)

ใน method onCreate() มีการเรียกใช้งานคำสั่ง CREATE TABLE ซึ่งทุกครั้งที่เรียกใช้ฐานข้อมูลเอสคิวไลท์ (SQLite Database) โปรแกรมจะตรวจสอบก่อนว่ามีตารางอยู่หรือไม่ ถ้าไม่มีก็ให้สร้างขึ้นมาใหม่ทุกครั้ง

จากนั้นทำการเพิ่มเมธอด (Method) ลงไปใน Class DatabaseHandler ให้สร้างฐานข้อมูลเอสคิวไลท์ (SQLite Database) ขึ้นมาเพื่อเรียกใช้คำสั่ง insert และ ข้อมูลที่ต้องการ insert ก็ใช้คลาส ContentValues เพื่อทำการใส่ข้อมูลลงไปผ่าน method put

```
//Adding new contact
Void addContact (Contact contact) {
        SQLiteDatabase db2 = this.getWritableDatabase();
        ContentValues values = new ContentValues();
// inport android.content.ContentValues
        values.put(KEY_NAME_LOC, contact.getNameLocation()); // Name location
        values.put(KEY_LAT, contact.getLatitude()); //Latitude
        values.put(KEY_LNG, contact.getLongitude()); //Longitude
// Inserting Row
        db2.insert (TABLE_LOCATION, null, values);
        db2.close (); //Closing database connection
}
```

ในการดึงข้อมูลก็ให้สร้างรายการ (List) ของคลาส (Class) Contact ขึ้นมาก่อน จากนั้นก็เขียนคำสั่ง SELECT ลงไปแล้วในการดูผลลัพธ์เราจะใช้ เคอร์เซอร์ (Cursor) เพื่อวนลูปดึงข้อมูลแต่ละแถวเพื่อใส่ข้อมูลเข้าไปใน List ของ return ก็ส่งเป็น List ไปแสดงในหน้าที่เรียกใช้ โดยในการวนลูปจะเริ่มจาก แถวแรก (moveToFirst) แล้ววนไปเรื่อยถึงแถวสุดท้าย (ตรวจสอบจาก moveToNext ถ้ามีข้อมูลในแถวถัดไปก็ให้ทำคำสั่ง do ถ้าไม่มีก็ออกจากคำสั่ง do - while)

```
// Getting All  Contacts
public List<Contact> getAllContacts() {
List<Contact> contactList = new ArrayList<Contact>();
// Select All Query
String selectQuery = “SELECT * FROM “ + TABLE_LOCATION;
SQLiteDatabase db2 = this.getWritableDatabase();
Cursor cursor = db2.rawQuery(selectQuery, null);
// import android.database.Cursor;
// looping through all rows and adding to list
If (cursor.moveToFirst()) {
do {
        Contact contact =new Contact();
        contact.setID(Integer.parseInt(cursor.getString(0)));//first column
        contact.setNameLocation(cursor.getString(1)); //second column
        contact.setLatitude(cursor.getDouble(2)); //third column
        contact.setLongitude(cursor.getDouble(3)); //fourth column
        // Adding contact to list
        contctList.add(contact);
        } while (cursor.moveToNext());
}
        // return contact list
        return contactList;
}
```

สำหรับการเพิ่ม (Update) ไม่มีอะไรที่ซับซ้อนโดยใส่ ContentValue ของข้อมูลที่ต้องการเพิ่ม (Update) เข้ามา และต้องกำหนดด้วยว่าต้องการให้เพิ่ม (Update) ที่ column ID แถวไหน โดยจะมีการรับค่ามาจาก contact.getID

```
//Update single contact
Public int updateContact(Contact contact) {
        SQLiteDatabase db2 = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(KEY_NAME_LOC, contact.getNameLocation());
        values.put(KEY_LAT, contact.getLatitude());
        values.put(KEY_LNA, contact.getLongitude());
        // updating row
        return db2.update(TABLE_LOCATION, values, KEY_ID + " = ?",
        new String[] { String.valueOf(contact.getID()) });
}
```

สำหรับการลบ (Delete) เหมือนกับการอัพเดท (Update) โดยต้องกำหนด contact.getID ที่ต้องการลบก่อน

```
// Deleting single contact
Public void deleteContact(Contact contact) {
        SQLiteDatabase db2 = this.getWritableDatabase();
        new String[] [String.valueof(contact.getID()) });
        db2.close();
}
```

4.2.2 การสร้างชุดคำสั่งการทำงานของการเรียกใช้ไฮเปอร์คลาส (Helper Class)

ในการเรียกใช้งานในขั้นตอนแรกคือการแทรก (Insert) ข้อมูลลงในตารางโดยใส่ชื่อสถานที่ละติจูด (Latitude) และลองติจูด (Longitude) ที่ตั้งให้เป็นข้อมูลพื้นฐานให้กับ ฐานข้อมูล (Database) เพื่อจะได้มีข้อมูลในแอพพลิเคชั่นในตอนเริ่มแรกการใช้งานโดยเมื่อมีการ แทรก (Insert) แล้วจะได้ตารางฐานข้อมูล (Database) ดังตารางที่ 4

```
// Inserting Data
    db2.addContact(new Contact("A", 13.1111, 100.1111)); // add first
    db2.addContact(new Contact("B", 13.2222, 100.2222)); // add second
    db2.addContact(new Contact("C", 13.3333, 100.3333)); // add third
    db2.addContact(new Contact("D", 13.4444, 100.4444)); // add forth
    db2.addContact(new Contact("E", 13.5555, 100.5555)); // add fifth
```

ตารางที่ 4 ตัวอย่างข้อมูลในตาราง location2

id	name_location	latitude	Longitude
1	A	13.1111	100.1111
2	B	13.2222	100.2222
3	C	13.3333	100.3333
4	D	13.4444	100.4444
5	E	13.5555	100.5555

ในขั้นตอนต่อไปคือการป้องกันข้อผิดพลาด (Bug) โดยใช้การลบ (Delete) ข้อมูลในตารางไม่ให้มีข้อมูลเกินห้าข้อมูล เพื่อให้อยู่ในจุดประสงค์ที่ต้องการจะให้มีสถานที่ปลายทางที่จะนำทางไปนั้นมีห้าสถานที่ โดยต้องมีการกำหนดค่า i เพื่อจะกำหนดแถวที่ต้องการจะลบข้อมูลนั้นๆ

```
// Delete Data
For (int I = 6; I <=10; I ++) {
Db2.deleteContact(new Contact(i));
}
```

ในขั้นตอนต่อมาคือการดึงข้องมูลหรือการ คิวรี่ (Query) จากตารางฐานข้อมูล (Database) เพื่อแสดงข้อมูลทั้งหมดในตารางว่ามีข้อมูลอะไรบ้าง ะทำให้ทราบว่ามีสถานที่ปลายทางอะไร มีค่าพิกัดจีพีเอส (GPS) ละติจูด (Latitude) และลองติจูด (Longitude) อะไรจะใช้ในการแก้ไขข้อมูลต่อไปได้ โดยจะทำการนับจำนวนแถวทั้งหมดในตารางแล้วเก็บไว้ในตัวแปล cn แล้วจึงทำการวนลูปแล้วดึงข้อมูลตามจำนวนแถวทั้งหมดที่มี

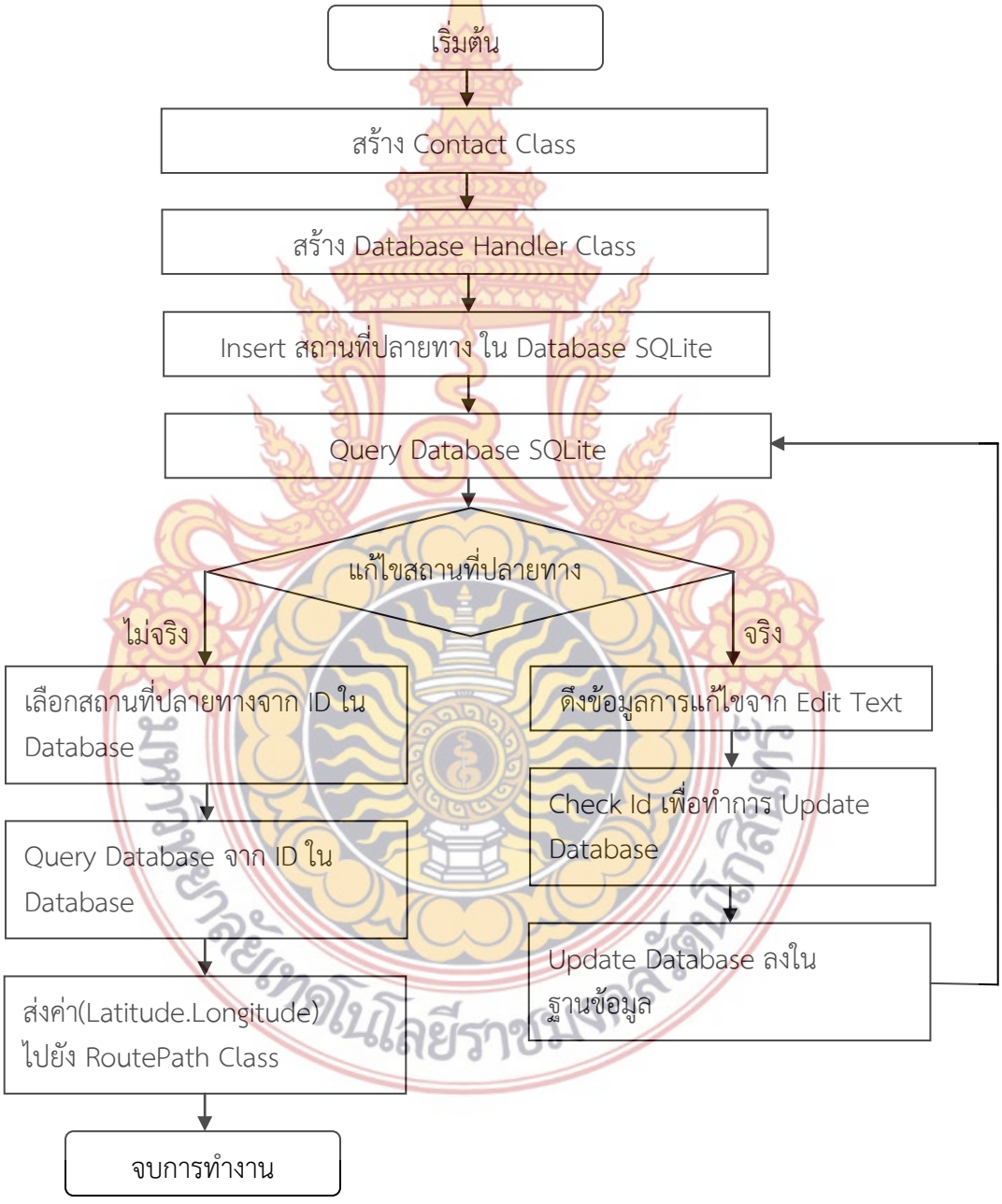
```
//Reading all Data
List<Contact>contacts = db2.getAllContacts());
for (Contact cn : contact) {
String log = ":Id: "+cngetID()+ " , " cn.getNameLocation() + " ,Lat: " _
+cn.getLatitude ()+ " ,Lng: " + cn.getLongitude();
 Toast.makeText (getApplicationContext(). log, Toast.LENGTH_SHORT).show();
}
```

ในขั้นตอนสุดท้ายคือการ เพิ่ม (Update) ข้อมูลลงในตาราง ซึ่งใช้ในการแก้ไขข้อมูลในตารางเพื่อให้เกิดการยืดหยุ่นในการนำทาง โดยการนำข้อมูลที่จะแก้ไขมาจาก EditText ซึ่งอยู่ในหน้าแอพพลิเคชั่นค้นหาสถานที่ (Location Finder) ซึ่งอธิบายไว้ในหัวข้อที่ 3.2.5.1 แต่มีการ

เพิ่มช้องไว้สำหรับใส่ id ของแถวตารางข้อมูลที่เก็บอยู่ในตัวแปล getButtonID2 เพื่อจะทำให้แก้ไขข้อมูลในแถวที่ต้องการแก้ไขได้ตามที่ต้องการเปลี่ยนแปลงข้อมูล

```
Db2.updateContact(new Contact(getButtonID2, locationName, lat, lng)) ;
```

4.2.3 แผนภาพการทำงาน (Flow Chart) การเก็บสถานที่ปลายทางบนฐานข้อมูลเอสคิวไลท์



ภาพที่ 53 แผนภาพการทำงานการเก็บสถานที่ปลายทางบนฐานข้อมูลเอสคิวไลท์

5. แอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่

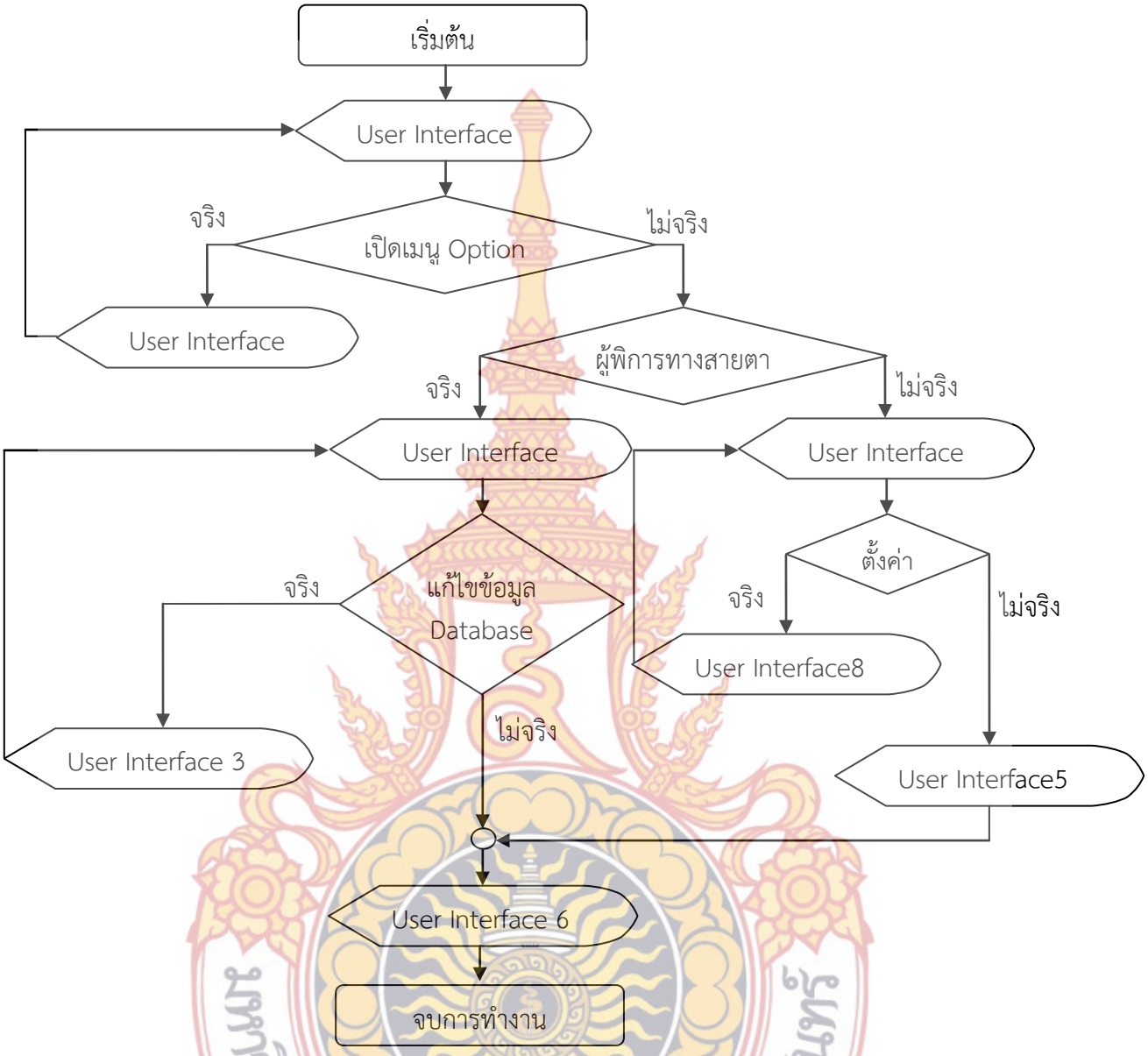
แอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่นั้นมีหน้าที่พื้นฐานในการทำงานต่างๆดังนี้

1. รับคำสั่งการทำงานผ่านฟังก์ชันการสั่งการตัวอุปกรณ์ด้วยคำสั่งเสียง หรือปุ่มกด
2. มีชุดเสียงในการนำทางคนพิการทางสายตา
3. มีการจัดเก็บสถานที่ที่ผู้ พิการทางสายตาต้องเดินทางไปลงใน ฐานข้อมูลเอสคิวไลท์ (Database SQLite)
4. สามารถดูสภาพการจราจรบทท้องถนนได้สำหรับผู้ที่ปกติทางสายตา
5. สามารถเปลี่ยนรูปแบบแผนที่ได้สำหรับผู้ที่ปกติทางสายตา
6. สามารถค้นหาสถานที่ปลายทางที่ผู้ใช้ต้องการเดินทางไปได้สำหรับผู้ที่พิการทางสายตาและปกติทางสายตา

ในขั้นตอนก ารพัฒนาแอพพลิเคชั่น จะมีอยู่หลายหน้าของแอพพลิเ คชั่น ซึ่งแต่และหน้าแอพพลิเคชั่นนั้นจะมีหน้าที่การทำงานต่างๆในแต่ละหน้าแอพพลิเคชั่นเหล่านี้ได้ถูกแบ่งแยกออกเป็น 3 ส่วนการทำงานใหญ่ๆ ได้แก่ ส่วนควบคุมการทำงาน ส่วนแสดงผล และระบบฐานข้อมูล ซึ่งจะนำมาอธิบายในแต่ละหน้าแอพพลิเคชั่นในหัวข้อต่อไป

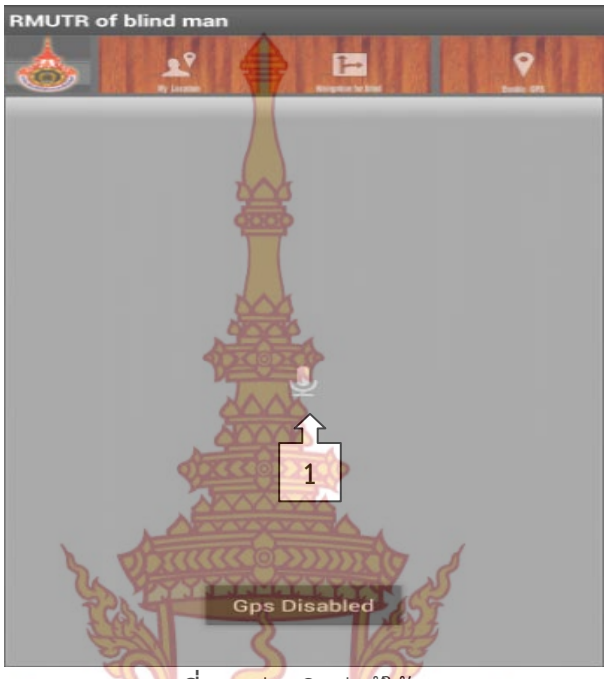
ในแต่ละหน้าแอพพลิเคชั่นนั่นจะมี ลำดับการทำงานหรือลักษณะการทำงาน ที่จะอธิบายด้วยแผนภาพแสดงการทำงานโปรแกรมดังภาพที่ 54

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์



ภาพที่ 54 แผนภาพการทำงานส่วนติดต่อผู้ใช้งาน

5.1 ส่วนติดต่อผู้ใช้งาน 1 (User Interface 1)



ภาพที่ 55 ส่วนติดต่อผู้ใช้งาน 1

5.1.1 ส่วนควบคุมการทำงาน

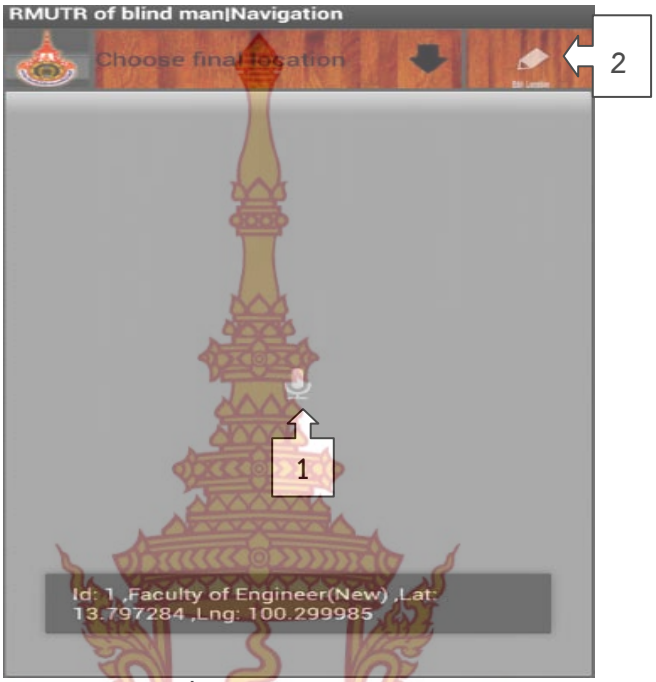
ส่วนควบคุมการทำงานของหน้า ส่วนติดต่อผู้ใช้งาน 1 (User Interface 1) นั้นจะเป็นหน้าแรกของ แอพพลิเคชั่น ที่เป็นหน้าของการสั่งการผ่านปุ่มบนหน้าแอพพลิ เคชั่นหรือด้วยเสียงการสั่งการ ซึ่งส่วนควบคุมนี้จะมีฟังก์ชั่นการใช้งานทั้งหมด 4 ฟังก์ชั่นได้แก่

1. My Location: ใช้ในการเรียกหน้าแอพพลิเคชั่นต่อไปสำหรับผู้ปกติทางสายตาซึ่งอธิบายไว้ในหัวข้อที่ 3
2. Navigation: ใช้ในการเรียกหน้าแอพพลิเคชั่นต่อไปสำหรับนำทางผู้พิการทางสายตาซึ่งอธิบายไว้ในหัวข้อที่ 4
3. GPS: ใช้ในการตั้งค่าเปิด – ปิด ตัวรับสัญญาณจีพีเอสซึ่งอธิบายไว้ในหัวข้อที่ 3.2.6
4. ภาพที่ 55 เลขที่ 1: ใช้ในการที่จะสั่งงานด้วยเสียงของผู้ใช้ซึ่งอธิบายไว้ในหัวข้อที่ 4.1

5.1.2 ส่วนการแสดงผล

ส่วนการแสดงผลนั้นจะแสดงเป็นข้อความและเสียงที่คอยแจ้งเตือนสถานะต่างๆ เช่น เตือนให้เปิดตัวรับสัญญาณจีพีเอสแอพพลิเคชั่นพร้อมทำงาน และเตือนให้รอจนกว่าแอพพลิเคชั่นจะพร้อมทำงาน

5.2 ส่วนติดต่อผู้ใช้งาน 2 (User Interface 2)



ภาพที่ 56 ส่วนติดต่อผู้ใช้งาน 2

5.2.1 ส่วนควบคุมการทำงาน

ส่วนควบคุมการทำงานของหน้า ส่วนติดต่อผู้ใช้งาน 1 (User Interface 2) นั้นจะเป็นของผู้พิการทางสายตาเป็นหลักที่เป็นหน้าของการสั่งการผ่านปุ่มบนหน้าแอพพลิเ คชั่น หรือด้วยเสียงการสั่งการซึ่งส่วนควบคุมนี้จะมีฟังก์ชั่นการใช้งานทั้งหมด 3 ฟังก์ชันได้แก่

5.2.1.1ใช้ในการเลือกสถานที่ปลายทางที่ทำการดึงข้อมูลที่ถูกเก็บไว้ในฐานข้อมูลเอสคิวไลท์ (Database SQLite) ซึ่งอธิบายไว้ในหัวข้อที่ 4.2.2

5.2.1.2 ภาพที่ 56 เลขที่ 1: ใช้ในการเรียกหน้าแอพพลิเคชั่นที่ใช้ในการแก้ไขข้อมูลลงในฐานข้อมูลเอสคิวไลท์ (Database SQLite)

5.2.1.3 ภาพที่ 56 เลขที่ 2: ใช้ในการที่จะสั่งงานด้วยเสียงของผู้ใช้ซึ่งอธิบายไว้ในหัวข้อที่ 4.1

5.2.2 ส่วนการแสดงผล

ส่วนการแสดงผลนั้นจะแสดงเป็นข้อความ และเสียงที่คอยแจ้งถึงสถานที่ปลายทางทั้งห้าเส้นทาง ที่ได้ทำการดึงข้อมูลมาจากฐานข้อมูลเอสคิวไลท์ (Database SQLite)

5.3 ส่วนติดต่อผู้ใช้งาน 3 (User Interface 3)

ภาพที่ 57 ส่วนติดต่อผู้ใช้งาน 3

5.3.1 ส่วนควบคุมการทำงาน

ส่วนควบคุมการทำงานของหน้า ส่วนติดต่อผู้ใช้งาน 3 (User Interface 3) นั้นจะเป็นหน้าของการค้นหาสถานที่ปลายทาง ซึ่งส่วนควบคุมนี้จะมีฟังก์ชั่นการใช้งานได้แก่

5.3.1.1 Search Location: ใช้ในการค้นหาสถานที่ปลายทางซึ่งอธิบายไว้ในหัวข้อที่ 3.2.5.1

5.3.1.2 Save Change: ใช้ในการบันทึกการแก้ไขข้ อมูลลงในฐานข้อมูลซึ่งอธิบายไว้ในหัวข้อที่ 4.2.2

5.3.2 ส่วนการแสดงผล

ส่วนการแสดงผลนั้นจะแสดงเป็นข้อความ ที่จะบ่งบอกถึงค่าของพิกัด จีพีเอส (GPS) ละติจูด (Latitude) และลองติจูด (Longitude) ที่ได้จากการค้นหาสถานที่ปลายทาง

5.3.3 ส่วนระบบฐานข้อมูล

ส่วนระบบฐานข้อ มูลนั้นจะเป็นการแก้ไขข้อมูลสถานที่ปลายทาง ซึ่งอธิบายไว้ในหัวข้อที่ 4.2 ที่มีส่วนประกอบของข้อมูลที่จะนำไปใส่ไว้ในฐานข้อมูลดังนี้

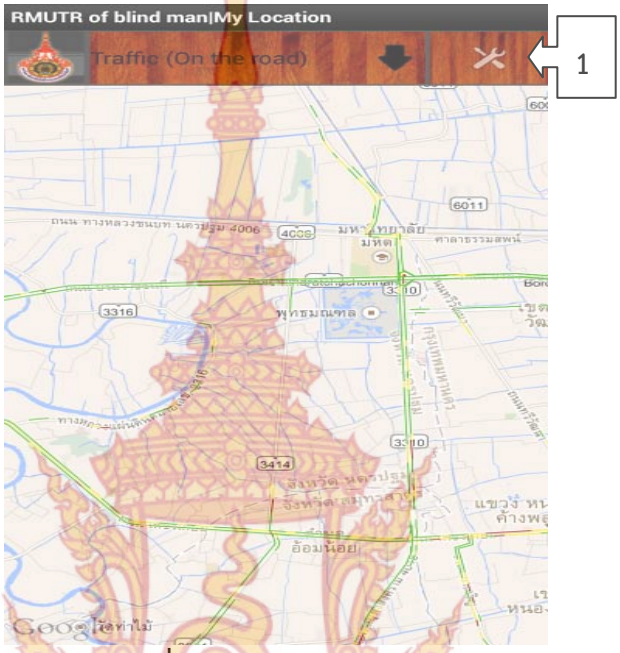
5.3.3.1 Button ID: ใช้ในการใส่ id ของข้อมูลที่ต้องการแก้ไขลงในฐานข้อมูล

5.3.3.2 Final Place: ใช้ในการใส่สถานที่ปลายทางที่ ต้องการเดินทางไปที่ต้องการแก้ไขลงในฐานข้อมูล

5.3.3.3 Latitude: แสดงถึงค่าพิกัดจีพีเอสละติจูด (Latitude) ที่ต้องการแก้ไขลงในฐานข้อมูล

5.3.3.4 Longitude: แสดงถึงค่าพิกัดจีพีเอสลองติจูด (Longitude) ที่ต้องการแก้ไขลงในฐานข้อมูล

5.4 ส่วนติดต่อผู้ใช้งาน 4 (User Interface 4)



ภาพที่ 58 ส่วนติดต่อผู้ใช้งาน 4

5.4.1 ส่วนควบคุมการทำงาน

ส่วนควบคุมการทำงานของหน้า ส่วนติดต่อผู้ใช้งาน 4 (User Interface 4) นั้นจะเป็นหน้าของผู้ปกติทางสายตา ที่เป็นหน้าของการสั่งการผ่านปุ่มบน หน้าแอพพลิเคชั่น ซึ่งส่วนควบคุมนี้จะมีฟังก์ชั่นการใช้งานทั้งหมด 5 ฟังก์ชั่นได้แก่

5.4.1.1 Satellite: ใช้ในการเปลี่ยนรูปแบบของแผนที่ซึ่งอธิบายไว้ในหัวข้อที่ 4.2

5.4.1.2 Traffic: ใช้ในการดูสภาพการจราจรบนท้องถนนซึ่งอธิบายไว้ในหัวข้อที่ 3.3

5.4.1.3 Navigation: ใช้ในการเรียกหน้าแอพพลิเคชั่นต่อไปสำหรับนำ ทางผู้ปกติทางสายตาซึ่งอธิบาย ไว้ใน หัวข้อที่ 3.

5.4.1.4 ภาพที่ 58 เลขที่ 1: ใช้ในการตั้งค่าเปิด – ปิด ตัวรับสัญญาณจีพีเอสซึ่งอธิบายไว้ในหัวข้อที่ 3.2.6

5.4.1.5 Zoom In - Out: ใช้ในการขยาย – ย่อแผนที่

5.4.2 ส่วนควบคุมการทำงาน

ส่วนควบคุมการทำงานของหน้า ส่วนติดต่อผู้ใช้งาน 4 (User Interface 4) นั้นจะเป็นหน้าของผู้ปกติทางสายตา ที่เป็นหน้าของการสั่งการผ่านปุ่มบน หน้าแอพพลิเคชั่น ซึ่งส่วนควบคุมนี้จะมีฟังก์ชั่นการใช้งานทั้งหมด 4 ฟังก์ชั่นได้แก่

5.4.3 ส่วนการแสดงผล

ส่วนการแสดงผลนั้นจะแสดงออ กเป็นฟังก์ชั่นการแสดงผลทั้งหมด 2 ฟังก์ชั่น ได้แก่

5.4.3.1 แสดงผลในรูปแบบแผนที่: ซึ่งอธิบายไว้ในหัวข้อที่ 3.1

5.4.3.2 แสดงผลในการกำหนด (Mark) ตำแหน่งจีพีเอส: ซึ่งอธิบายไว้ในหัวข้อที่ 3.2

5.5 ส่วนติดต่อผู้ใช้งาน 5 (User Interface 5)

ภาพที่ 59 ส่วนติดต่อผู้ใช้งาน 5

5.5.1 ส่วนควบคุมการทำงาน

ส่วนควบคุมการทำงานของหน้า ส่วนติดต่อผู้ใช้งาน 5 (User Interface 5) นั้นจะเป็นหน้าของการค้นหาสถานที่ปลายทาง ซึ่งส่วนควบคุมนี้จะมีฟังก์ชันการใช้งานได้แก่

5.5.1.1 Final Place: ใช้ในการใส่สถานที่ปลายทางที่ต้องการเดินทางไปซึ่งอธิบายไว้ในหัวข้อที่ 3.2.5.1

5.5.1.2 Search Location: ใช้ในการค้นหาสถานที่ปลายทางซึ่งอธิบายไว้ในหัวข้อที่ 3.2.5.1

5.5.1.3 Go!! Navigation: ใช้ในการเรียกหน้าแอพพลิเคชั่นในการ หาเส้นทาง (Route Path) ซึ่งอธิบายไว้ในหัวข้อที่ 3.2.5.2

5.5.1.4 Point on map: ใช้ในการเรียกหน้าแอพพลิเคชั่นในการสัมผัสหน้าจอเพื่อรับค่าพิกัดจีพีเอส

5.5.2 ส่วนการแสดงผล

ส่วนการแสดงผลนั้นจะแสดงเป็นข้อความ ที่จะบ่งบอกถึงค่าของพิกัดจีพีเอสละติจูด (Latitude) และลองติจูด (Longitude) ที่ได้จากการค้นหาสถานที่ปลายทาง ซึ่งอธิบายไว้ในหัวข้อที่ 3.2.5.1

5.6 ส่วนติดต่อผู้ใช้งาน 6 (User Interface 6)

ภาพที่ 60 ส่วนติดต่อผู้ใช้งาน 6

5.6.1 ส่วนการแสดงผล

ส่วนการแสดงผลของหน้า ส่วนติดต่อผู้ใช้งาน 6 (User Interface 6) นั้นจะแสดงออกเป็นฟังก์ชั่นการแสดงผลทั้งหมด 5 ฟังก์ชั่นได้แก่

5.6.1.1 แสดงผลในรูปแบบแผนที่: ซึ่งอธิบายไว้ในหัวข้อที่ 3.1

5.6.1.2 แสดงผลในการกำหนด (Mark) ตำแหน่งจีพีเอส: ซึ่งอธิบายไว้ในหัวข้อที่ 3.2

5.6.1.3 การแสดงผลในการกำหนด (Mark) ตำแหน่งของต้นทางและปลายทางของการนำทางซึ่งอธิบายไว้ในหัวข้อที่ 3.2.5.2

5.6.1.4 การแสดงผลในรูปแบบข้อความแจ้งเตือน: ใช้ในการแจ้งเตือนที่จำเป็นต่อการนำทาง เช่น เลี้ยวซ้าย เลี้ยวขวา ฯลฯ

5.6.1.5 การแสดงผลในรูปแบบเสียงแจ้งเตือน: ใช้ในการแจ้งเตือนที่จำเป็นต่อการนำทาง เช่น เลี้ยวซ้าย เลี้ยวขวาฯลฯ

5.7 ส่วนติดต่อผู้ใช้งาน 7 (User Interface 7)

ภาพที่ 61 ส่วนติดต่อผู้ใช้งาน 7

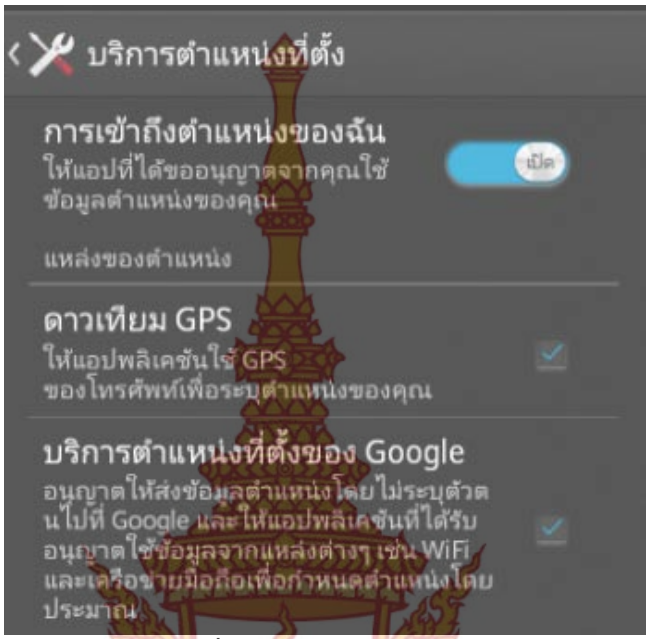
5.7.1 ส่วนควบคุมการทำงาน

ส่วนควบคุมการทำงานของหน้า ส่วนติดต่อผู้ใช้งาน 7 (User Interface 7) นั้นจะเป็นหน้าของการเปิดใช้ Option Menu ซึ่งส่วนควบคุมนี้จะมีฟังก์ชั่นการใช้งานได้แก่ ฟังก์ชัน OK: ใช้ในการกลับไปยังหน้าการทำงานหลักของแอพพลิเคชั่น

5.7.2 ส่วนการแสดงผล

ส่วนการแสดงผลนั้นจะแสดงเป็นข้อความ ที่จะบ่งบอกถึงรายละเอีย ดเกี่ยวกับผู้จัดทำแอพพลิเคชั่น เช่น ชื่อ – นามสกุล ชื่อมหาวิทยาลัย ชื่อโครงการ เป็นต้น ซึ่งอธิบายไว้ในหัวข้อที่ 3.2.7

5.8 ส่วนติดต่อผู้ใช้งาน 8 (User Interface 8)



ภาพที่ 62 ส่วนติดต่อผู้ใช้งาน 8

5.8.1 ส่วนควบคุมการทำงาน

ส่วนควบคุมการทำงานของหน้า ส่วนติดต่อผู้ใช้งาน 8 (User Interface 8) นั้นจะเป็นหน้าของการเปิดใช้หน้า Location & security setting ซึ่งส่วนควบคุมนี้จะมีฟังก์ชั่นการใช้งานได้แก่

5.8.1.1 Use wireless networks: ใช้ในการเปิดการเชื่อมต่อเครือข่ายไร้สาย (Wireless Networks)

5.8.1.2 Use GPS satellite: ใช้ในการเปิดการเชื่อมต่อสัญญาณจีพีเอสจากดาวเทียม (GPS Satellite)

บทที่ 4
ขั้นตอนการทดลอง

ในขั้นตอนการทดลองการพัฒนาระบบนำทางเพื่อผู้พิการทางสายตา ผู้วิจัยได้ทำการทดลอง โดยได้แบ่งขั้นตอนการดำเนินงานเป็น 4 ขั้นตอน ดังนี้

1. การใช้งานแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่
2. การทดลองใช้งานแอพพลิเคชั่น
3. การทดสอบเสถียรภาพของแอพพลิเคชั่น

1. ขั้นตอนการใช้งานแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่

การใช้งานช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เค ลื่อนที่จะประกอบด้วยขั้นตอนหลักๆ ดังนี้

1.1 การใช้งานหน้าแอพพลิเคชั่นหลัก

เป็นหน้าแรกของการทำงานจะเป็นหน้าตัวเลือกในการใช้งานว่าจะใช้งาน แอพพลิเคชั่นในส่วนใดซึ่งแบ่งออกเป็น 2 คือ การใช้งานส่วนผู้พิการทางสายตาและการใช้งานส่วนผู้ปกติทาสายตา เมื่อทำการเลือกที่นำทาง (Navigation) หรือตำแหน่งสถานที่ของฉัน (My Location) ตามลำดับซึ่งจะแสดงดังภาพที่ 63

ภาพที่ 63 การใช้งานหน้าแอพพลิเคชั่นหลัก

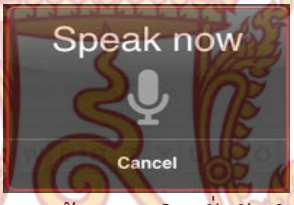
ทำการเลือกการใช้งานผ่านปุ่มบนหน้าแอพพลิ เคชั่นหรือด้วยเสียงการสั่งการ ซึ่งส่วนเลือกการทำงานนี้จะมีการใช้งานทั้งหมด 4 การใช้งานได้แก่

1.1.1 My Location: ใช้เรียกหน้าแอพพลิเคชั่นต่อไปสำหรับผู้ปกติทางสายตาโดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นดังภาพที่ 63 เลขที่ 1 หรือด้วยเสียงการสั่งการว่า“My Location”

1.1.2 Navigation for blind: ใช้ในการเรียกหน้าแอพพลิเคชั่นต่อไปสำหรับนำทางผู้พิการทางสายตาโดยการเลือกปุ่มบนหน้าแอพพลิเคชั่น ดังภาพที่ 63 เลขที่ 2 หรือด้วยเสียงการสั่งการว่า “Navigation”

1.1.3 GPS: ใช้ในการตั้งค่าเปิดตัวรับสัญญาณจีพีเอส (GPS Receiver) สำหรับผู้พิการทางสายตา โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่น ดังภาพที่ 63 เลขที่ 3 หรือด้วยเสียงการสั่งการว่า “GPS” ซึ่งสามารถทราบได้ว่าทำการเปิดตัวรับสัญญาณจีพีเอส (GPS Receiver) เช่นจากข้อความแจ้งว่า “Application Ready to Use!” จากเสียงแอพพลิเคชั่นแจ้งเตือนว่า “แอพพลิเคชั่นพร้อมนำทาง” และจากสัญลักษณ์ แสดงว่ามีการรับสัญญาณพิกัดจีพีเอส (GPS) ซึ่งจะอยู่ในส่วนของสถานะของโทรศัพท์เคลื่อนที่

1.1.4 Speech Recognition: ดังภาพที่ 63 เลขที่ 4 ใช้ในการที่จะสั่งงานด้วยเสียงสำหรับผู้พิการทางสายตา โดยการเลือก ปุ่มบนหน้าแอพพลิเคชั่น ซึ่งจะใช้สั่งงานด้วยเสียงในข้อที่ 1.1.1 – 1.1.3 ที่กล่าวมาก่อนหน้านี้ ซึ่งจะเรียกหน้าแอพพลิ เคชั่นของการรับคำสั่งด้วยสียงขึ้นมาดังในภาพที่ 64 แล้วจึงทำการพูดคำสั่ง ที่ต้องการให้แอพพลิเคชั่นทำงานลงไป



ภาพที่ 64 หน้าแอพพลิเคชั่นรับคำสั่งเสียง

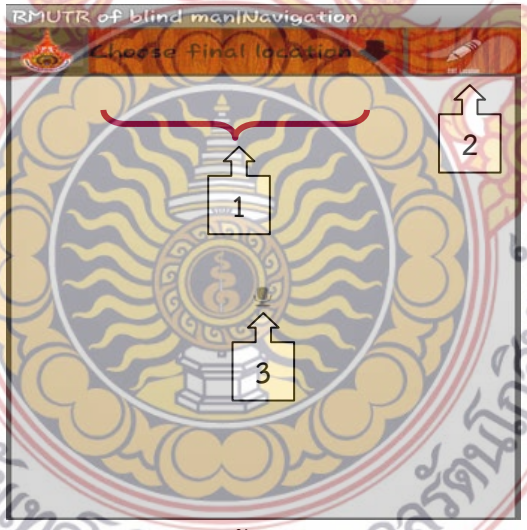
1.2 เมนูตัวเลือก (Option Menu)

การเปิดใช้เมนูตัวเลือก (Option Menu) สำหรับผู้ปกติทางสายตาซึ่งในส่วนนี้จะไม่มีปุ่มบนหน้าแอพพลิเคชั่นแต่จะกดปุ่ม เมนู (Menu) ที่ตัวโทรศัพท์เคลื่อน ที่แทนส่วนการแสดง ผลนั้นจะแสดงเป็นข้อความที่จะบ่งบอกถึงรายละเอียดเกี่ยวกับผู้จัดทำแอพพลิเคชั่นเมื่อดูข้อมูลเสร็จแล้วสามารถกลับไปยังหน้าแอพพลิเคชั่นหลักโดยการเลือกที่ปุ่มดังภาพที่ 65 เลขที่ 1

ภาพที่ 65 เมนูตัวเลือก (Option Menu)

1.3 การใช้งานส่วนผู้พิการทางสายตา

ในส่วนของหน้าแอพพลิเคชั่นนั้นจะเป็นหน้าแอพพลิเคชั่นหลักในการนำทางของผู้พิการทางสายตาซึ่งจะถูก เรียกใช้โดยหน้าแอพพลิเคชั่นหลัก โดยการเลือกปุ่มบนหน้าแอพพลิ เคชั่น ดังภาพที่ 63 เลขที่ 2 หรือด้วยเสียงการสั่งการว่า “Navigation” ดังภาพที่ 63 ซึ่งได้กล่าวมาก่อนในหน้านี้หน้าแอพพลิเคชั่นหลักในการนำทางของผู้พิการทางสายตานั้นจะแสดงดังในภาพที่ 66



ภาพที่ 66 หน้าต่างหลักระบบนำทางผู้พิการทางสายตา (Navigation for Blind)

ทำการเลือกการใช้งานผ่านปุ่มบนหน้าแอพพลิ เคชั่นหรือด้วยเสียงการสั่งการ ซึ่งส่วนเลือกการทำงานนี้จะมีการใช้งานทั้งหมด 3 การใช้งานได้แก่

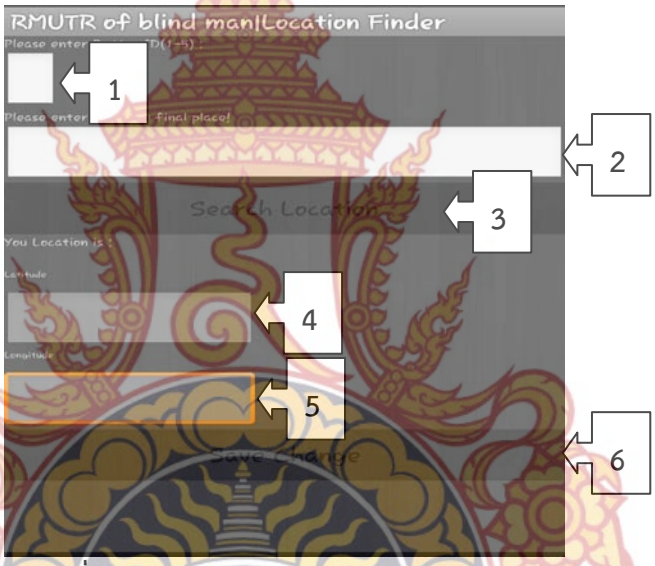
1.3.1 ในการเลือกสถานที่ปลายทางที่ต้องการให้นำทางได้ทั้งหมด 5 สถานที่ที่ได้จากการบันทึกข้อมูลไว้แล้วจากนั้นจึงจะทำการเรียกหน้าการ ค้นหาเส้นทาง (Route Path) เพื่อนำทางด้วยเสียงต่อไปโดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นดังภาพที่ 66 เลขที่ 1 เพื่อให้คนรู้จักของผู้พิการ

ทางสายตาเลือกสถานที่ปลายทางใ ห้ หรือด้วยเสียงการสั่งการว่า “one”, “two” ถึง “five” สำหรับผู้พิการทางสายตา ซึ่งจะแสดงสถานที่ปลายทางไว้ในด้านล่างของแอพพลิเคชั่น

1.3.2 Edit Location: ใช้ในการเรียกหน้าแอพพลิเคชั่นที่ใช้ในการแก้ไขข้อมูลลงใน ฐานข้อมูลเอสคิวไลท์ (Database SQLite) โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่น ดังภาพที่ 66 เลขที่ 2 แล้วจะทำการเรียกหน้าแอพพลิเคชั่นในหัวข้อที่ 1.4 ดังภาพที่ 67

1.3.3 Speech Recognition: ดังภาพที่ 63 เลขที่ 4 ใช้ในการที่จะสั่งงานด้วยเสียง สำหรับผู้พิการทางสายตา โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่น ซึ่งจะใช้สั่งงานด้วยเสียงในข้อที่ 1.3.1 - 1.3.2 ที่กล่าวมาก่อนหน้านี้

1.4 การสร้างฐานข้อมูล (Edit Database)



ภาพที่ 67 การสร้างฐานข้อมูล (Edit Database)

ในหน้าแอพพลิเคชั่นนี้นั้นใช้สำหรับผู้ที่ปกติทางสายตานั้นช่วยผู้พิการทางสายตาแก้ไขข้อมูลสถานที่ปลายทางลงใน ฐานข้อมูล (Database) ทำโดยการเลือกการใช้งานผ่านปุ่มบนหน้าแอพพลิเคชั่นและด้วยการใส่สถานที่ ดังภาพที่ 67 ซึ่งส่วนเลือกการทำงานนี้จะมีการใช้งานทั้งหมด 6 ฟังก์ชันการใช้งานได้แก่

1.4.1 Button ID: ใช้ในการใส่ id ของข้อมูลที่ต้องการแก้ไขลงในฐานข้อมูล ซึ่งมีเลขตั้งแต่ 1-5 ตามเลขหมายของปุ่มบนหน้าแอพพลิเคชั่นหลักสำหรับผู้พิการทางสายตา

1.4.2 Final Place: ใช้ในการใส่สถานที่ปลายทางที่ต้องการเดินทางไปที่ต้องการแก้ไขลงในฐานข้อมูลเช่น “มหาวิทยาลัยเทคโนโลยี ราชมงคลรัตนโกสินทร์ ”, “13.7609021, 100.6356597” เป็นต้น

1.4.3 Search Location: ใช้ในการค้นหาสถานที่ปลายทางที่ได้จากการใส่ Final Place ดังภาพที่ 67 เลขที่ 2 ลงไปเพื่อเช็คว่าสถานที่นั้นมีข้อมูลอยู่ในระบบของฐานข้อมูลหรือไม่โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นดังภาพที่ 67 เลขที่ 3

1.4.4 Latitude: แสดงถึงค่าพิกัดจีพีเอส latitudeที่ต้องการแก้ไขลงในฐานข้อมูล เพื่อเช็คว่ าสถานที่นั้นมีข้อมูลอยู่ในระบบของฐานข้อมูลหรือไม่ซึ่งจะแสดงหลังเลือกปุ่มบนหน้า แอพพลิเคชั่น Search Location ดังภาพที่ 67 เลขที่ 4

1.4.5 Longitude: แสดงถึงค่าพิกัดจีพีเอส ลองติจูด (longitude) ที่ต้องการแก้ไขลง ในฐานข้อมูลเพื่อเช็คว่าสถานที่นั้นมีข้อมูลอยู่ในระบบของฐานข้อมูลหรือไม่ ซึ่งจะแสดงหลังเลือกปุ่ม บนหน้าแอพพลิเคชั่น Search Location ดังภาพที่ 67 เลขที่ 5

1.4.6 Save Change: ใช้ในการบันทึกการแก้ไขข้อมูลลงในฐานข้อมูล ซึ่งจะไม่สามารถ บันทึกการแก้ไขข้อมูลลงในฐานข้อมูลได้ ถ้าหากว่าสถานที่ที่ใส่ใน Final Place นั้นไม่มีในระบบ ฐานข้อมูลใช้โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นดังภาพที่ 67 เลขที่ 6

1.5 การใช้งานส่วนผู้ปกติทางสายตา

ในส่วนของหน้าแอพพลิเคชั่นนี้นั้นจะเป็นหน้าแอพพลิเคชั่นหลักของผู้ปกติทางสายตา ซึ่งจะถูกเรียกใช้โดยหน้าแอพพลิเคชั่นหลัก โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่น My Location หรือด้วยเสียงการสั่งการว่า “My Location” ดังภาพที่ 63 ซึ่งได้กล่าวมาก่อนในหน้านี้ หน้าแอพพลิเคชั่นหลักของผู้ปกติทางสายตานั้นจะแสดงดังใน ภาพที่ 68

ภาพที่ 68 หน้าต่าง My Location

ทำการเลือกการใช้งานผ่านปุ่มบนหน้าแอพพลิเคชั่น ซึ่งส่วนเลือกการทำงานนี้จะมีการ ใช้งานทั้งหมด 5 การใช้งานได้แก่

1.5.1 Satellite: ใช้ในการเปลี่ยนรูปแบบของแผนที่โดย การเลือกปุ่มบนแอพพลิเคชั่น เป็น Satellite (Map mode) ซึ่งจะเปลี่ยนไปดังเลขที่ 1 ในภาพที่ 6

ภาพที่ 69 เปลี่ยนรูปแบบโหมดแผนที่

1.5.2 Traffic: ใช้ในการดูสภาพการจราจรบนท้องถนน โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นเป็น Traffic (On the Road) ซึ่งจะเปลี่ยนไปดังเลขที่ 1 ในภาพที่ 70

ภาพที่ 70 หน้าต่างแสดงสภาพการจราจร (Traffic)

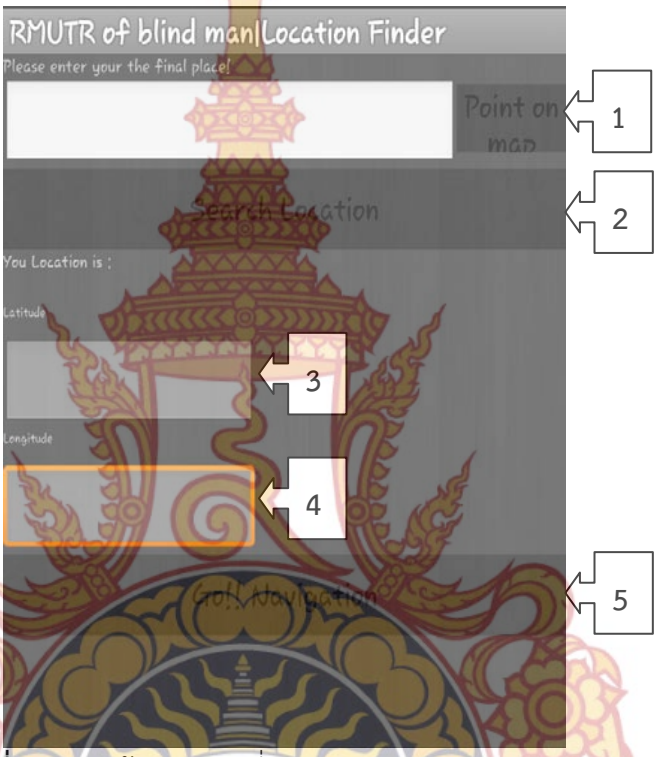
1.5.3 Navigation: ใช้ในการเรียกหน้าแอพพลิเคชั่นต่ อไปสำหรับนำทางผู้พิการทางสายตาโดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นเป็น นำทาง (Navigation) แล้วจะทำการเรียกหน้าแอพพลิเคชั่นในหัวข้อที่ 1.6 ภาพที่ 71

1.5.4 GPS: ใช้ในการตั้งค่าเปิด – ปิด ตัวรับสัญญาณจีพีเอส (GPS Receiver) โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นดังภาพที่ 70 เลขที่ 2 แล้วจะทำการเรียกหน้าแอพพลิเคชั่น

การตั้งค่าตัวรับสัญญาณจีพีเอส (GPS) และเครือข่ายไร้สาย (Wireless Network) ในหัวข้อที่ 1.6 ภาพที่ 73

1.5.5 Zoom In - Out: ใช้ในการขยาย – ย่อแผนที่โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นดังภาพที่ 67 เลขที่ 3

1.6 การนำทาง (Navigation)



ภาพที่ 71 การค้นหาสถานที่ (Navigation Location Finder)

ในหน้าแอพพลิเคชั่นนี้นั้นใช้สำหรับผู้ที่ปกติทางสายตาทำโดยการเลือกการใช้งานผ่านปุ่มบนหน้าแอพพลิเคชั่น และด้วยการใส่สถานที่ ซึ่งส่วนเลือกการทำงานนี้จะมีการใช้งานทั้งหมด 5 ฟังก์ชันการใช้งานได้แก่

1.6.1 Final Place: ใช้ในการใส่สถานที่ปลายทางที่ต้องการเดินทางไปที่ต้องการแก้ไขลงในฐานข้อมูล เช่น “มหาวิทยาลัยเทคโนโลยี ราชมงคลรัตนโกสินทร์ ”, “13.7609021, 100.6356597” เป็นต้น

1.6.2 Search Location: ใช้ในการค้นหาสถานที่ปลายทางที่ได้โดยใส่ Final Place ลงไปเพื่อเช็คว่าสถานที่นั้นมีข้อมู ลอยู่ในระบบของฐานข้อมูลหรือไม่ โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่น Search Location ดังภาพที่71 เลขที่ 2

1.6.3 Latitude: แสดงถึงค่าพิกัดจีพีเอส (GPS) ลาติจูด (Latitude) ที่ต้องการแก้ไขลงในฐานข้อมูลเพื่อเช็คว่าสถานที่นั้นมีข้อมูลอยู่ในระบบของฐานข้อมูลหรือไม่ซึ่งจะแสดงหลังเลือกปุ่มบนหน้าแอพพลิเคชั่น Search Location ดังภาพที่ 71 เลขที่ 3

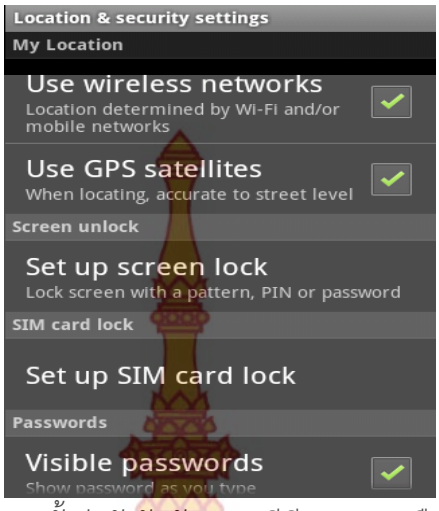
1.6.4 Longitude: แสดงถึงค่าพิกัดจีพีเอส (GPS) ลองติจูด (Longitude) ที่ต้องการแก้ไขลงในฐานข้อมูลเพื่อเช็คว่าสถานที่นั้นมีข้อมูลอยู่ในระบบของฐานข้อมูลหรือไม่ ซึ่งจะแสดงหลังเลือกปุ่มบนหน้าแอพพลิเคชั่น Search Location ดังภาพที่ 71 เลขที่ 4

1.6.5 Go!! Navigation: ใช้ในการเรียกหน้าการ หาเส้นทาง (Route Path) ดังภาพที่ 72 เพื่อนำทางด้วยเส้นสีต่อไป ซึ่งจะไม่สามารถเลือกเพื่อนำทางได้ ถ้าหากว่าสถานที่ที่ใส่ใน Final Place นั้นไม่มีในระบบฐานข้อมูลใช้โดยการเลือกปุ่มบนหน้าแอพพลิเคชั่นดังภาพที่ 71 เลขที่ 5

ภาพที่ 72 การหาเส้นทาง (Route Path)

1.7 การตั้งค่าตัวรับสัญญาณจีพีเอส (GPS Receiver) และเครือข่ายไร้สาย (Wireless - Network)

การตั้งค่าตัวรับสัญญาณจีพีเอส (GPS Receiver) และและเครือข่ายไร้สาย (Wireless - Network) นั้นจะทำหน้าที่ในการที่จะเปิด – ปิดตัวรับสัญญาณจีพีเอส และและเครือข่ายไร้สาย (Wireless - Network) ซึ่งแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตานั้นจำเป็นที่จะต้องทำการตั้งค่าเพื่อเปิดตัวรับสัญญาณจีพีเอสและเครือข่ายไร้สาย (Wireless Network) เสียก่อนจึงจะใช้งานได้อย่างสมบูรณ์ ทำโดยการกดเลือกที่ Check Box ของช่อง Use wireless networks และ Use GPS satellite ให้เปลี่ยนรูปสถานะจาก กลายเป็น ซึ่งจะแสดงดังในภาพที่ 73



ภาพที่ 73 การตั้งค่าตัวรับสัญญาณจีพีเอสและเครือข่ายไร้สาย

2. การทดลองใช้งานแอพพลิเคชั่น

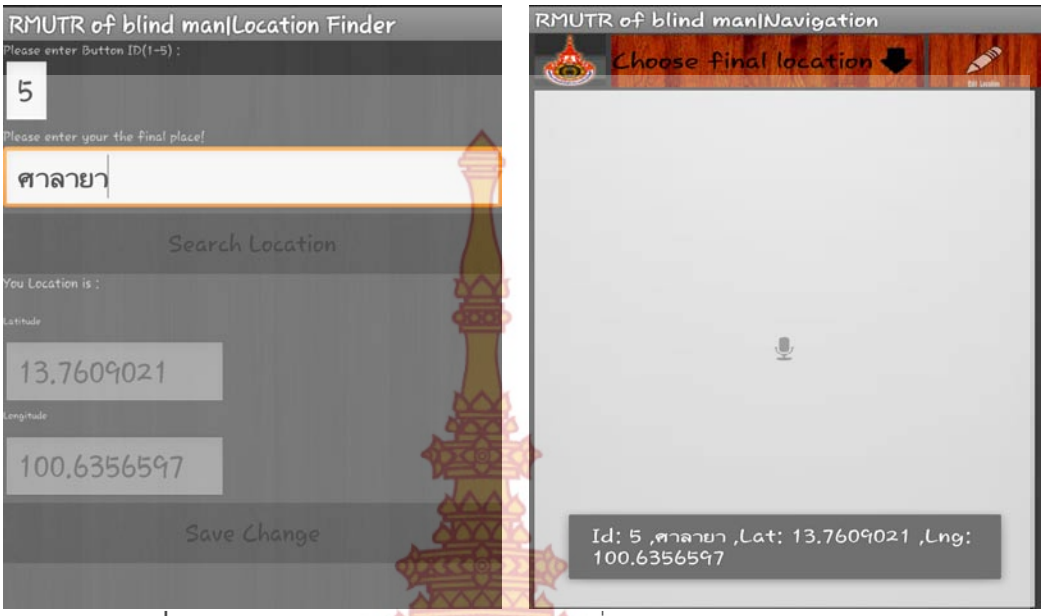
การทดลองใช้งานแอพพลิเคชั่นทางคณะ ผู้จัดทำได้เลือกเป้าหมายของการทดลองในบริเวณมหาวิทยาลัยเทคโนโลยี ราชมงคลรัตนโกสินทร์ พื้นที่ศาลายา ได้มีการทดลองที่แบ่งออกเป็น 2 ส่วน คือการทดลองใช้งานส่วนของผู้พิการทางสายตาและการทดลองใช้งานส่วนผู้ปกติทางสายตา ซึ่งมีการทดลองการใช้งานดังนี้

2.1 การทดลองใช้งานส่วนของผู้พิการทางสายตา

ในส่วนการทดลองนี้นั้นจะมีการทดลองทั้งหมด 3 การทำงานหลัก ได้แก่ การทำงานในส่วนของการแก้ไขข้อมูลสถานที่ปลายทางใน ฐานข้อมูลเอสคิวไลท์ (Database SQLite) การสั่งการทำงานด้วยเสียงและการนำทางผู้พิการทางสายตาด้วยเสียงซึ่งมีการทดลองดังนี้

2.1.1 การทดลองการแก้ไขข้อมูลสถานที่ปลายทางในฐานข้อมูลเอสคิวไลท์ (Database-SQLite)

ทำการทดลองโดยการทดสอบใส่สถานที่ปลายทางลงในหน้าแก้ไขข้อมูล โดยการแก้ไขในทุกๆ แถวตารางของ ฐานข้อมูล (Database) แล้วดึงข้อมูลที่แก้ไขนั้นขึ้นมาตรวจสอบว่าตรงตามที่ได้แก้ไขไปหรือไม่จากการทดสอบนั้นทำให้ทราบว่าทำ การแก้ไขข้อมูลสถานที่ปลายทางในฐานข้อมูลเอสคิวไลท์ (Database SQLite) ได้ดังภาพที่ 74



ภาพที่ 74 การทดลองการแก้ไขข้อมูลสถานที่ปลายทางในฐานข้อมูล SQLite

2.1.2 การทดลองการสั่งการทำงานด้วยเสียง

ทำการทดส อบโดยการทดลองพูดเสียงเข้าไปในแอพพลิเคชั่นแล้วทดสอบว่าการทำงานนั้นตรงไปตามที่ได้สั่งการทำงานด้วยเสียงหรือไม่พบว่าการทำงานนั้นตรงไปตามคำสั่งเสียงแต่ยังมีข้อผิดพลาดของด้านการรับเสียงอยู่เกิดจากเสียงรบกวนรอบข้างและสำเนียงของผู้สั่งงานนั้นไปตรงไปตามสำเนียงที่แอพพลิเคชั่นต้องการทำให้ไม่เกิดการทำงานขึ้นได้ในกรณีนี้

2.1.3 การทดลองการนำทางผู้พิการทางสายตาด้วยเสียง

ทำการทดสอบการใช้งานในสถานที่และสภาพแวดล้อมจริง โดยการเลือกสถานที่ปลายทางแล้วให้แอพพลิเคชั่นนำทางด้วยเสียงแล้วทำการเดินตามเสียงนำทาง พบว่าแอพพลิเคชั่นนี้สามารถนำทางด้วยเสียงได้ แต่ยังไม่มีความละเอียดมากเท่าที่ควรจึงทำการแก้ไขในเบื้องต้นแล้วและความแม่นยำในพิกัดตำแหน่ง จีพีเอส (GPS) นั้นยังไม่มีประสิทธิภาพมากเท่าที่ควร เนื่องจากตัวรับสัญญาณจีพีเอส (GPS Receiver) ที่มีประสิทธิภาพการรับสัญญาณที่ไม่ละเอียดมากนักทำให้เกิดการผิดพลาด (Error) ในการนำทางด้วยเสียงเกิดขึ้น โดยการทดสอบดังภาพที่ 75 จะพบว่ามีการระบุพิกัดนอกการเดินทางเกิดขึ้นอีกเช่นกัน

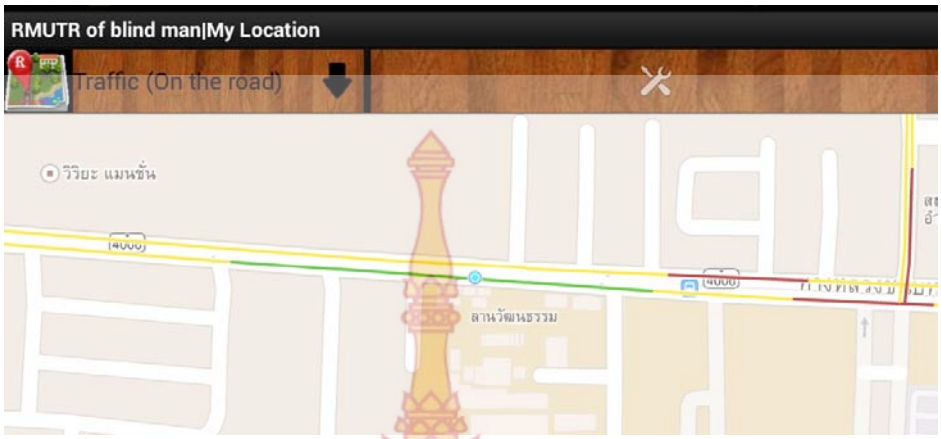
ภาพที่ 75 การทดลองการนำทางผู้พิการทางสายตาด้วยเสียง

2.2 การทดลองใช้งานส่วนของผู้ปกติทางสายตา

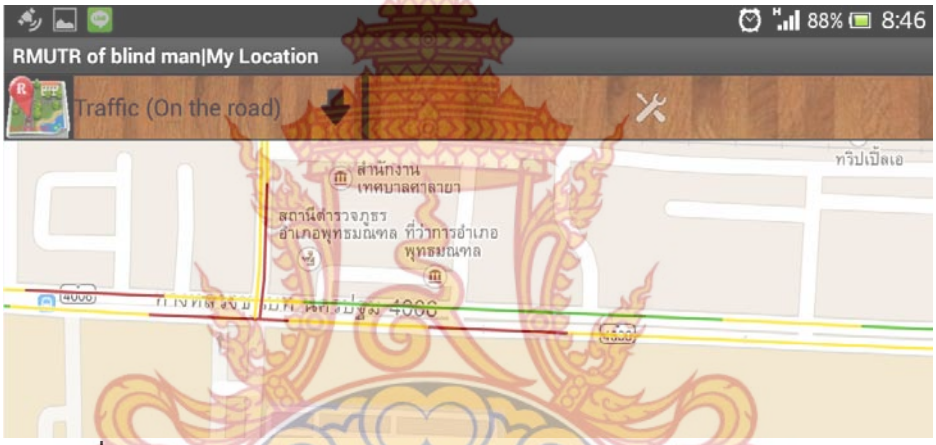
ในส่วนการทดลองนี้นั้นจะมีการทดลองทั้งหมด 5 การทำงานหลักได้แก่ การดูสภาพการจราจรบนท้องถนน การค้นหาสถานที่ปลายทาง การนำทางผู้ปกติทางสายตา การทดลองการหาตำแหน่ง ณ ปัจจุบัน และการทดลองการกำหนดสถานที่การเดินทางบนแผนที่ซึ่งมีการทดลองดังนี้

2.2.1 การทดลองการดูสภาพการจราจรบนท้องถนน

ทำการทดสอบการใช้งานในสถานที่และสภาพแวดล้อมจริง โดยการเปิดใช้งานแอพพลิเคชั่นในส่ว นของการดูสภาพการจราจรบนท้องถนน แล้วทำการไปดูในสภาพท้องถนนนั้นมีการติดขัดทางการจราจร หรือมีสภาพคล่องของการจราจรเป็นจริงอย่างที่แอพพลิเค ชั่นนั้นแจ้งขึ้นมาหรือไม่ โดยเส้นสีเขียวหมายถึงการจราจรไม่ติดขัด สีเหลืองหมายถึงการจราจรพอเคลื่อนตัวได้ สีแดงหมายถึงการจรติดขัด โดยพบว่าการแจ้งสภาพการจราจรบนท้องถนนในแอพพลิเคชั่นนั้นเป็ นไปตามสภาพการจราจรบนท้องถนนจริง เนื่องจากมีการเก็บข้อมูลของทาง บริการของ กูเกิล (Google Service) แบบเรียลไทม์ (Real Time) โดยการทดสอบดังภาพที่ 76 และ ภาพที่ 77



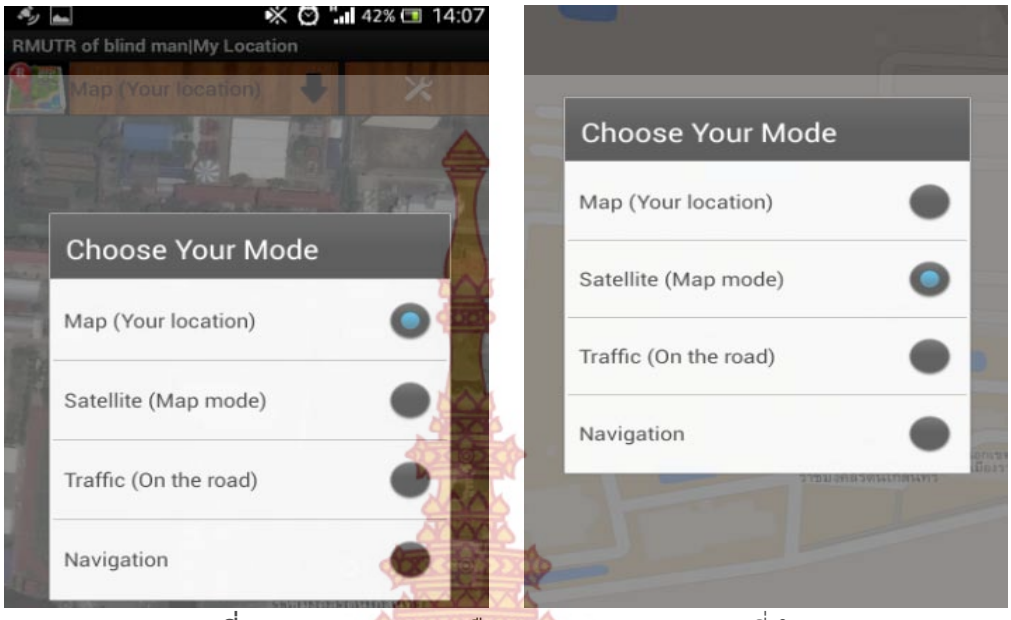
ภาพที่ 76 การทดลองการดูสภาพการจราจรบนท้องถนน ณ บริเวณลานวัฒนธรรม



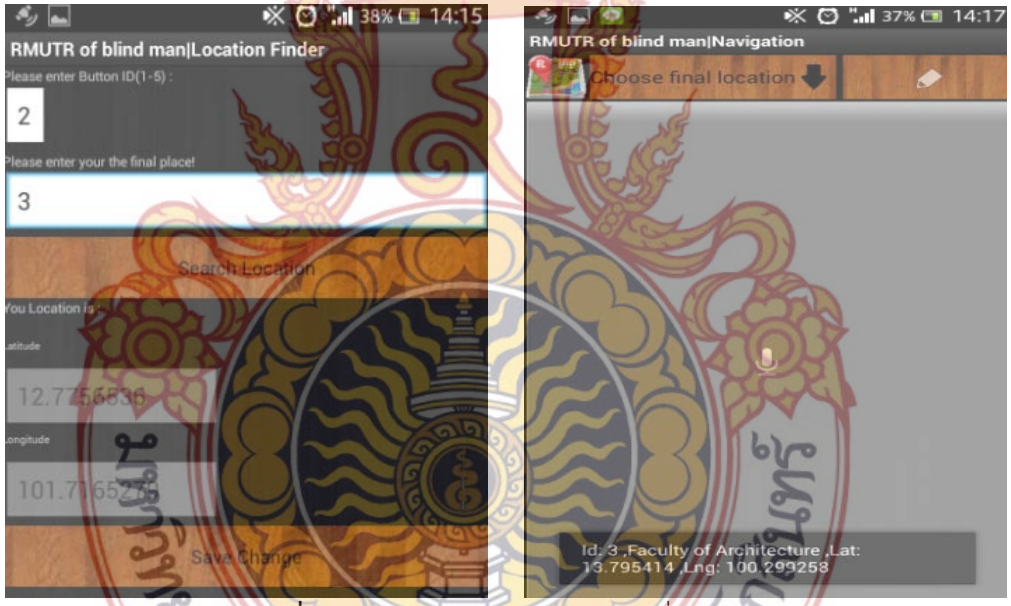
ภาพที่ 77 การทดลองการดูสภาพการจราจรบนท้องถนน ณ อำเภอพุทธมณฑล

2.2.2 การทดลองการค้นหาสถานที่ปลายทาง

ทำการทดลอง โดยการใส่สถานที่ปลายทางในรูปแบบต่าง ๆ เช่น “มหาวิทยาลัยเทคโนโลยี ราชมงคลรัตนโกสินทร์ ”, “13.7609021, 100.6356597” เป็นต้นพบว่าสามารถจะทำการค้นหาสถานที่ปลายทางในรูปแบบดังกล่าวได้ในทุกรูปแบบทำให้มีความยืดหยุ่นในการค้นหาสถานที่ปลายทางเป็นอย่างมาก ดังรูปที่ 79 และสามารถเลือกฟังก์ชันการแสดงผล 2 รูปแบบคือแบบดาวเทียมและแบบแผนที่ ดังรูปที่ 78



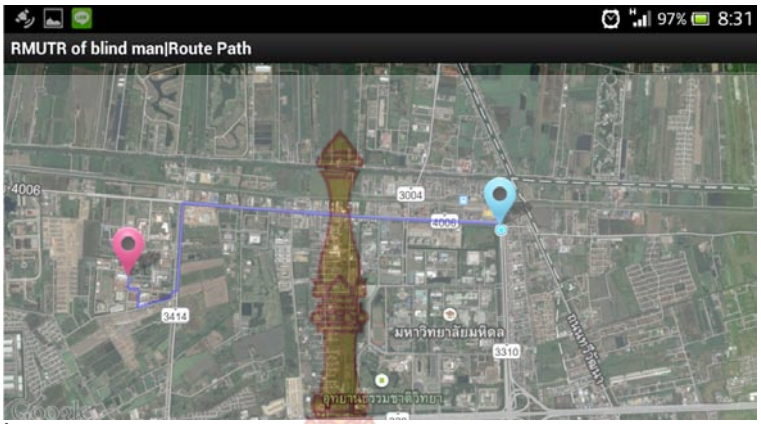
ภาพที่ 78 การทดลองการเลือกการแสดงผลของแผนที่นำทาง



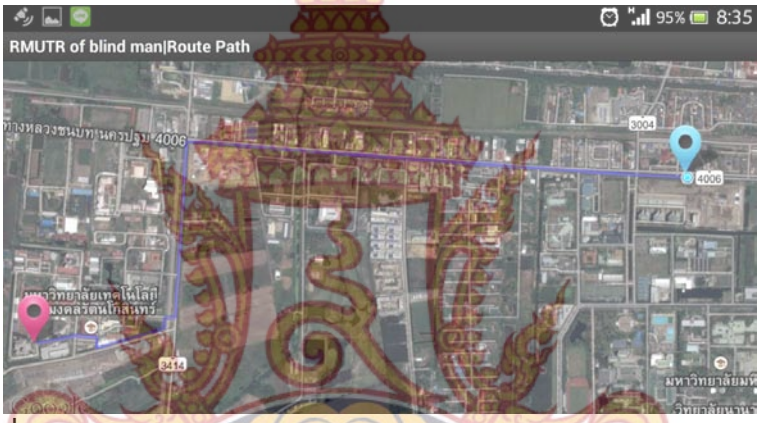
ภาพที่ 79 การทดลองการค้นหาสถานที่ปลายทาง

2.2.3 การทดลองการนำทางผู้ปกติทางสายตา

ทำการทดสอบการใช้งานในสถานที่และสภาพแวดล้อมจริง โดยการเลือกสถานที่ปลายทางแล้วให้แอพพลิเคชั่นนำทางด้วยเส้นสีนำทางแล้วทำการ ขับรถยนต์ตาม พบว่าแอพพลิเคชั่นนี้สามารถนำทางไปถึงสถานที่ปลายทางได้อย่างถูกต้องอย่างแม่นยำ



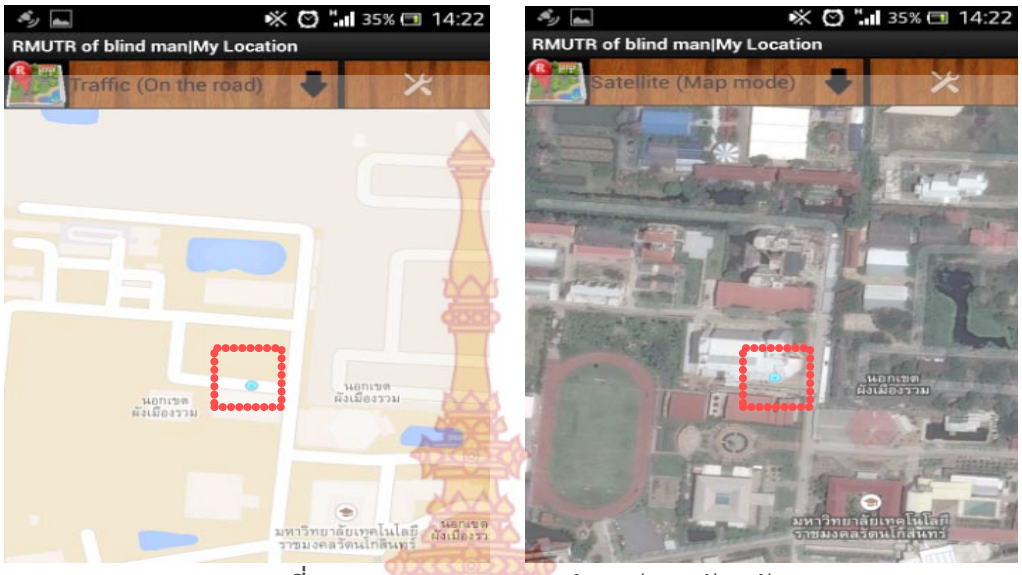
ภาพที่ 80 การทดลองการนำทางจากพุทธมณฑลสาย 4 – คณะวิศวกรรมศาสตร์



ภาพที่ 81 การทดลองการนำทางจากตลาดศาลายา – อาคารเฉลิมพระเกียรติฯ

2.2.4 การทดลองการหาตำแหน่ง ณ ปัจจุบัน

ทำการทดสอบการใช้งาน ตรวจสอบตำแหน่งที่อยู่ ณ ปัจจุบันโดยสามารถแสดงผลแบบแผนที่ และแบบดาวเทียมดังภาพที่ 82



ภาพที่ 82 การทดสอบการหาตำแหน่ง ณ ปัจจุบัน

2.2.5 การทดลองการกำหนดสถานที่การเดินทางบนแผนที่

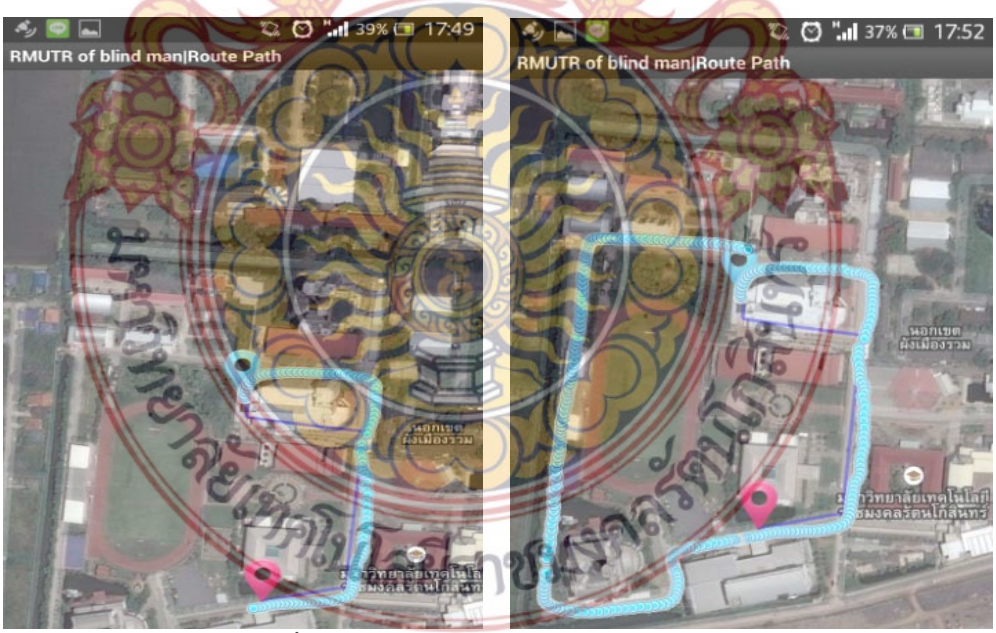
ทำการทดสอบ การเพิ่มพิกัดสถานที่โดยใช้การปักหมุดสีแดง หมายถึสถานที่ ณ ตำแหน่งใดๆบนแผนที่ พบว่าสามารถทำงานเพิ่ม สถานจุดหมายได้อย่างละเอียดทำให้ การบอกเส้นทางมีความละเอียดและแม่นยำมากขึ้นดังรูปที่ 83

ภาพที่ 83 การทดสอบการกำหนดสถานที่การเดินทางบนแผนที่

3. การทดสอบเสถียรภาพของแอพพลิเคชั่น

หลังจากที่ได้มีการนำช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่ไปทำการทดลองใช้งานจากสถานที่และ สภาพแวดล้อมจริง ทำให้ได้พบปัญหาด้านเสถียรภาพการทำงานของแอพพลิเคชั่นจึงได้มีการนำปัญหาดังกล่าวมาทำการปรับปรุงแก้ไขในส่วนของ โค้ด (Code) การทำงานของแอพพลิเคชั่น เนื่องจากการทำงานของแอพพลิเคชั่นนำทางจะต้องทำงานที่มีความแม่นยำในพิกัดตำแหน่งจีพีเอส (GPS) เพื่อใช้นำทางผู้พิการทางสายตา ซึ่งถ้าหากแอพพลิเคชั่นนำทางไม่สามารถทำงานได้แม่นยำเท่าที่ควร จะทำให้เกิดข้อผิดพลาดของการเดินทางส่งผลให้การเดินทางไปยังสถานที่เป้าหมายนั้นมีความคลาดเคลื่อนหรือเกิดการเข้าใจผิดในเส้นทางที่จะนำทางได้ โดยได้มีการทดสอบแอพพลิเคชั่นนำทางภายใต้สภาพแวดล้อมภายในบริเวณ มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ ซึ่งจะมุ่งเน้นไปในส่วนของเสถียรภาพของแอพพลิเคชั่นนำทางและความแม่นยำในพิกัดตำแหน่งจีพีเอส

ผลจากการทดลองในช่วงระยะเวลาแรกปรากฏว่าแอพพลิเคชั่นนำทางยังมีความคลาดเคลื่อนของความแม่นยำในพิกัดตำแหน่งจีพีเอส (GPS) เนื่องจากเกิดข้อผิดพลาดต่าง ๆ ในส่วนของตัวรับสัญญาณจีพีเอส (GPS Receiver) ที่มีประสิทธิภาพการรับสัญญาณที่ไม่ละเอียดมากนัก จึงได้มีการแก้ไขและพัฒนาแอพพลิเคชั่นในส่วนของ โค้ด (Code) ให้มีความยืดหยุ่นในเงื่อนไขที่เกี่ยวข้องกับตำแหน่งพิกัดจีพีเอส (GPS) จนปัจจุบันแอพพลิเคชั่น นำทางสามารถทำงานได้ดียิ่งขึ้น มีความคลาดเคลื่อนน้อยลงอย่างมาก สามารถใช้นำทางผู้พิการทางสายตาได้มีประสิทธิภาพที่ดีขึ้น โดยการทดสอบดังภาพที่ 84



ภาพที่ 84 การทดสอบเสถียรภาพของการนำทาง

บทที่ 5
สรุปผล อภิปรายผลและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงบทสรุป อภิปรายผล และข้อเสนอแนะในการพัฒนาต่อยอดโดยแบ่งได้ 3 หัวข้อดังนี้

1. สรุปผลการวิจัย
2. อภิปรายผลการวิจัย
3. ข้อเสนอแนะแนวทางในการประยุกต์ และพัฒนาต่อยอดงานวิจัย

1. สรุปผลการวิจัย

แอพพ ลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่ เป็นผล จากการที่นำเทคโนโลยีจีพีเอส (GPS) และระบบปฏิบัติการแอดดรอยด์ (Android Operation System) ที่ถูกบรรจุอยู่ภาย ในโทรศัพท์เคลื่อน (Smart Phone) มาประยุกต์ใช้ให้เกิดประโยชน์ โดยมีการพัฒนา ต่อยอดให้เหมาะสมกับการใช้งานของผู้พิการทางสายตา โดยพัฒนาระบบให้ มีการสั่งงานด้วยเสียง (Speech Recognition) เพื่อง่ายต่ อการใช้งาน สำหรับผู้พิการทางสายตา ซึ่งได้มีการทดสอบ ซอฟต์แวร์ที่ พัฒนาขึ้นมานี้ ประกอบไปด้วยระบบนำทางผู้ ที่สายตา ปกติ และผู้พิการทางสายตา โดยระบบของผู้พิการทางสายตาสามารถบันทึกเส้นทา งที่ต้องการไปเก็บไว้ได้ ทั้งหมด 5 เส้นทางลงในฐานข้อมูล (Database) และสามารถเปลี่ยนแปลงเส้นทางได้ ในการทดสอบระบบครั้งนี้มีการโปรแกรมเส้นทางไว้ 5 เส้นทางภายในมหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ ประกอบด้วย อาคารเฉลิมพระเกียรติ อาคารปฏิบัติการ คณะวิศวกรรมศาสตร์ อาคารสำนักงานอธิการบดี อาคารคณะบริหาร และอาคารคณะสถาปัตยกรรมศาสตร์ ทั้งนี้ผู้ใช้งานสามารถเปลี่ยนฐานข้อมูลสถานที่ใช้งานดังกล่าวได้ในภายหลัง ทั้ งนี้ระบบนำทางนี้ยังสามารถนำทางภายนอกมหาวิทยาลัยได้ อาทิเช่น เส้นการเดินทางไปยังจังหวัดต่างๆ เส้นทางการจราจรรวมถึงการดู ปริมาณความหนาแน่นของการจราจร เส้นทางรถไฟได้

2. อภิปรายผลการวิจัย

งานวิจัยนี้ได้มีปัญหาที่เกิดขึ้นระหว่างการทำการวิจัยโดยแบ่งประเด็นปัญห าเป็น 3 หัวข้อดังนี้

2.1 ปัญหาความคลาดเคลื่อนของการรับข้อมูล ละติจูด (Latitude) และลองติจูด (Longitude) เนื่องจากประสิทธิภาพของตัวรับ สัญญาณจีพีเอส (GPS Receiver) ของโทรศัพท์เคลื่อนที่ (Smart Phone) แต่ละตัวจะประสิทธิภาพต่างกันขึ้นฮาร์ดแวร์ (Hard Ware) ที่เลือกใช้งาน ซึ่งจะทำให้ส่งผลต่อการทำงานของแอพพลิเคชั่น

2.2 ปัญหาการสั่งการอุปกรณ์ด้วยเสียง (Speech Recognition) เนื่องจากเป็นเทคโนโลยีที่เกิดขึ้นเพื่อที่จะทำให้สามา รถแยกแยะคำพูดต่างๆที่มนุษย์พูด จึงทำให้ระบบต้อ งเข้าใจ

คำศัพท์ทุกคำอย่างถูกต้อง มากที่สุด เพราะการสั่งการอุปกรณ์ด้วยเสียง จะเป็นการนำ ไฟล์เสียง (Audio File) ที่บรรจุเสียงพูดนำมาแปลงเป็น อักขระ (Text) ก่อนที่จะนำไปค้นหาข้อมูลที่ต้องการ ดังนั้นระบบ บอาจเกิดความผิดพลาดขึ้นได้ง่ายเนื่องจาก สัญญาณรบกวน จากภายนอก อาทิเช่น การออกเสียงที่ไม่ชัดเจนของผู้พูด เสียงจากรถยนต์ หรือเสียงอื่นๆที่เข้ามาในชุดประมวลผลขณะผู้พูดกำลังสั่งการด้วยด้วยเสียง หรือ สัญญาณรบกวนจากภายในเครื่องประมวลผลสัญญาณเสียง (Smart Phone) เช่นคุณภาพของไมโครโฟน (Microphone) หากคุณภาพไม่ดีจะส่งผลให้มีสัญญาณรบกวนสูง (Noise) เช่นกัน

2.3 ปัญหาของการนำทาง เนื่องจากไม่สามารถนำทางไปในเส้นทางที่ไม่ได้อยู่บนถนนสายหลักได้เพราะฐานข้อมูล (Database) ข้อมูลละติจูด (Latitude) และลองติจูด (Longitude) ในการนำทางยังไม่มากพอ จึงยังไม่มีข้อมูลในส่วนของเส้นทางย่อย เช่น ซอยต่างๆ เส้นทางในตึก เป็นต้น

3. ข้อเสนอแนะแนวทางในการประยุกต์ และพัฒนาต่อยอดงานวิจัย

3.1 การนำระบบจีพีเอส (GPS) มาใช้ในการพัฒนาแอพพลิเคชั่นช่วยนำทางเพื่อผู้พิการทางสายตาบนโทรศัพท์เคลื่อนที่ ปัจจุบันมีการพัฒนามากขึ้น และมีการเชื่อมต่อกับอุปกรณ์ได้ง่าย และสะดวกสบายเพราะระบบ จีพีเอส (GPS) ถูกบรรจุอยู่ในสมาร์ทโฟน (Smart Phone) แต่ทั้งนี้ควรเลือกสมาร์ทโฟน (Smart Phone) ที่มีประสิทธิภาพในการประมวลผล และการรับสัญญาณจีพีเอสที่ดี เพราะจะส่งผลต่อความแม่นยำในการหาตำแหน่งสถานที่อย่างมาก

3.2 การใช้โปรแกรมจำลอง (Simulator) เพื่อพัฒนาแอพพลิเคชั่นแอนดรอยด์ (Android) ยังมีข้อจำกัดในการใช้งานอยู่ ดังนั้นในการพัฒนาจึงควรต้องมีการทดสอบกับอุปกรณ์จริง เพื่อทดสอบการใช้งาน และข้อผิดพลาดต่างๆที่จะเกิดขึ้น

3.3 แอพพลิเคชั่นนี้ที่ผู้วิจัยได้พัฒนานี้ ใช้เพียงแนะนำและค้นหาเส้นทางเฉพาะที่เท่านั้น เนื่องจากตัวอุปกรณ์นั้นมีความละเอียดในการระบุพิกัดยังไม่สูงมาก ถ้าหากมีอุปกรณ์ที่แม่นยำมากกว่านี้ก็จะสามารถพัฒนาเพิ่มได้ให้สามารถนำทางผู้พิการทางสายตาได้ โดยการเพิ่ม ส่วนการประมวลผลภาพ (Image Processing) และเนียร์ฟิลด์คอมมูนิเคชัน (Near Field Communication: NFC) ในการช่วยเพิ่มประสิทธิภาพในการบอกสิ่งกีดขวาง และสถานที่ภายในอาคารที่เป็นจุดบอดของสัญญาณจีพีเอส (GPS)

บรรณานุกรม

[1] สำนักงานสถิติแห่งชาติในการสำรวจคนพิการ [ออนไลน์]. แหล่งที่มา: http://service.nso.go.th (2 มกราคม 2555).

[2] พร้อมเลิศ หล่อวิจิตร, **คู่มือเขียนแอพ Android**. พิมพ์ครั้งที่1. กรุงเทพฯ : โรงพิมพ์โปรวิชั่น, 2555.

[3] Android Community [ออนไลน์]. แหล่งที่มา: http://source.android.com/ Thailand Android Community (4 มกราคม 2557).

[4] Ed Burnette, "**Hello, Android: Introducing Google's Mobile Development Platform (Pragmatic Programmers)**". 250 pp., Pragmatic Bookshelf, October 2009.

[5] สมาร์ทโฟน. [ออนไลน์]. แหล่งที่มา: http://th.wikipedia.org/wiki/%E0%B8%AA%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B9%8C%E0%B8%95%E0%B9%82%E0%B8%9F%E0%B8%99 (27 พฤศจิกายน 2556).

[6] ระบบการทำงานGPS. [ออนไลน์]. แหล่งที่มา: http://www.nectec.or.th (5 มกราคม 2556)

[7] รศ.ดร.โกสินทร์ จำนงไทย. "**Speech Recognition**". [ออนไลน์]. แหล่งที่มา : http://www.kmutt.ac.th/rippc/best47.htm (27 พฤศจิกายน 2556).

[8] Database SQLite. [ออนไลน์]. แหล่งที่มา : http://www.thaicreate.com/mobile/android-sqlite-database.html (1 มกราคม 2557).

[9] Dijkstra's algorithm [ออนไลน์]. แหล่งที่มา : http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm (1 ธันวาคม 2556).

[10] บลูทูธ. [ออนไลน์]. สืบค้นวันที่ 1 มกราคม 2557. แหล่งที่มา : http://itc.excise.go.th/PEOPLE/IT_Utilities/WEBPORTAL16200037212

[11] Djunknic, G.-M. and Richton, R.-E. "**Geolocation and assisted GPS**", Computer, vol. 34, pp. 123 -125, Mar. 2001.

[12] Adusei, I.K., et.al. "**Moblie Positioning Technoligies in cellular networks:An Evakution of their performance metrics**". MILCOM, California,USA, 2002.

[13] GT&T. "**Intelligent Transportation System (ITMS)**". GT&T Engineering Pte, 2005.

Mcgovern A. Geographic Distance and Azimuth Calculations. Available from: http://www.codegrn.com/Cpp/cpp/algorithms/article.php/ January 2557

ภาคผนวก

แบบสำรวจความต้องการคุณสมบัติของแอพพลิเคชั่นเพื่อนำทางผู้พิการทางสายตา

แบบสำรวจความต้องการคุณสมบัติของแอพพลิเคชั่นเพื่อนำทางผู้พิการทางสายตา

.....

คำชี้แจง: 1. แบบสำรวจฉบับนี้จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อนำข้อมูลจากการสำรวจมาพัฒนา/ปรับปรุงระบบเพื่อให้ได้ระบบตรงตามความต้องการของผู้ใช้งานแอพพลิเคชั่นนำทางผู้พิการทางสายตา

2. โปรดทำเครื่องหมาย ✓ ลงในช่องที่ตรงกับความคิดเห็นของท่าน ตามเกณฑ์การประเมินดังนี้

5 หมายถึง พึงพอใจมากที่สุด 4 หมายถึง พึงพอใจมาก 3 หมายถึง พึงพอใจปานกลาง
2 หมายถึง พึงพอใจน้อย 1 หมายถึง พึงพอใจน้อยที่สุด

ข้อมูลพื้นฐาน

1) เพศ ☐ชาย ☐หญิง
2) อายุ ☐ 20-25 ปี ☐ 26-30 ปี ☐ 31-35 ปี ☐ 36-40 ☐ 40 ปีขึ้นไป
3) การศึกษา .....

รายการสำรวจ	ระดับความต้องการ				
	มากที่สุด (5)	มาก (4)	ปานกลาง (3)	น้อย (2)	น้อยที่สุด (1)
การสำรวจด้านความต้องการนำเทคโนยีช่วยพัฒนาคุณภาพชีวิตของผู้พิการทางสายตา					
1. เทคโนโลยีสามารถช่วยพัฒนาคุณภาพชีวิตได้					
2. ท่านประสบปัญหาการเดินทางไปในสถานที่ต่างๆ					
3. ท่านต้องการเทคโนโลยีช่วยนำทาง					
การสำรวจคุณสมบัติของซอฟต์แวร์ที่ต้องการ					
1. สามารถใช้งานได้ง่าย					
2. สามารถสั่งการด้วยเสียง					
3. สามารถกำหนดเส้นทางการเดินทาง					
4. สามารถรายงานผลการจราจร					
5. อุปกรณ์เหมาะแก่การพกพา และมีขนาดเล็ก					
6. สามารถแสดงผลการทำงานด้วยเสียง					

- ข้อเสนอแนะ

.....
.....
.....
.....

ประวัติผู้วิจัย

ประวัติผู้วิจัย

ประวัติผู้วิจัย

1. ชื่อ - นามสกุล (ภาษาไทย) นายชาญยุทธ อุปายโกศล
   ชื่อ - นามสกุล (ภาษาอังกฤษ) Mr. CHANYUT AUPAYAGOSON
2. เลขหมายบัตรประจำตัวประชาชน
3. ตำแหน่ง อาจารย์
4. หน่วยงานและสถานที่อยู่ที่ติดต่อได้สะดวก
   สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
   มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ กระทรวงศึกษาธิการ
   E-mail: chanyut.a@rmutr.ac.th มือถือ 086-663-6998
5. ประวัติการศึกษา
   - ปริญญาตรี อุตสาหกรรมศาสตรบัณฑิต (เทคโนโลยีโทรคมนาคม)
     มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์
   - ปริญญาโท วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)
     มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
6. สาขาวิชาการที่มีความชำนาญพิเศษ(แตกต่างจากวุฒิการศึกษา) ระบุสาขาวิชาการ
   - การประมวลผลทางภาพ

ประวัติผู้วิจัย (ต่อ)

ประวัติผู้ร่วมวิจัย

1. ชื่อ - นามสกุล (ภาษาไทย) นายกาญบัญชา พานิชเจริญ
   ชื่อ - นามสกุล (ภาษาอังกฤษ) Mr. KANBUNCHA PANICHCHAROEN
2. เลขหมายบัตรประจำตัวประชาชน
2. ตำแหน่ง อาจารย์
3. หน่วยงานและสถานที่อยู่ที่ติดต่อได้สะดวก พร้อมหมายเลขโทรศัพท์ที่ ทำงาน โทรศัพท์มือถือ โทรสาร และไปรษณีย์อิเล็กทรอนิกส์(e-mail)
   สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ กระทรวงศึกษาธิการ
   E-mail: kanbuncha.pan@rmutr.ac.th มือถือ 081-668-8678
4. ประวัติการศึกษา
   - ปริญญาตรี วิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า)
     คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสยาม
   - ปริญญาโท ครุศาสตร์อุตสาหกรรมมหาบัณฑิต
     (คอมพิวเตอร์และเทคโนโลยีสารสนเทศ)
     คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
5. สาขาวิชาการที่มีความชำนาญพิเศษ(แตกต่างจากวุฒิการศึกษา)
   - สาขาวิชา GSM / WCDMA

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์