ระบบติดตามและประมาณเวลาการเดินรถไฟฟ้า

ด้วยสมาร์ทโฟนผ่านเครือข่าย 3G

TRACKING AND TIME APPROXIMATION SYSTEM FOR EV

VIA SMART PHONE ON NETWORK 3G

นายกฤษณะ วทานิยานนท์ รหัสนิสิต 53363317

นายจักรกฤษ เติมฤทธิกุล รหัสนิสิต 53363379

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2556

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 2 มิ.ย. 2558
เลขทะเบียน 1675792
เลขเรียกหนังสือ ว.
มหาวิทยาลัยนเรศวร ก281 ร
2556

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน ระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟน ผ่านเครือข่าย 3G

ผู้ดำเนินโครงงาน นายกฤษณะ วทานิยานนท์ รหัส 53363317

นายจักรกฤษ เติมฤทธิกุล รหัส 53363379

ที่ปรึกษาโครงงาน นายเศรษฐา ตั้งค้าวานิช

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2556

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

เศรษฐา ตั้งค้าวานิช .....ที่ปรึกษาโครงงาน

(นายเศรษฐา ตั้งค้าวานิช)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา)

.....กรรมการ

(ดร. สุรเดช จิตประไพกุลศาล)

ชื่อหัวข้อโครงงาน	ระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G		
ผู้ดำเนินโครงงาน	นายกฤษณะ วทานิยานนท์	รหัส	53363317
	นายจักรกฤษ เติมฤทธิกุล	รหัส	53363379
ที่ปรึกษาโครงงาน	นายเศรษฐา ตั้งค้าวานิช		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2556		

บทคัดย่อ

โครงงานมีนี้จุดมุ่งหมายเพื่ออำนวยความสะดวกให้แก่ผู้ใช้บริการรถไฟฟ้ามหาวิทยาลัยนเรศวร ผู้ใช้จะสามารถติดตามตำแหน่งของรถไฟฟ้าและประมาณระยะเวลาที่รถไฟฟ้าจะเคลื่อนที่มาจุดจอดรับผู้โดยสารในแต่ละป้ายบนสมาร์ทโฟนที่ใช้ระบบปฏิบัติการแอนดรอยด์เวอร์ชั่น 4 ขึ้นไป ระบบได้รับการออกแบบให้ครอบคลุมการใช้บริการของรถไฟฟ้าสายสีแดง ข้อมูลตำแหน่งและอัตราเร็วของรถไฟฟ้าจะถูกส่งผ่านเครือข่ายการสื่อสารแบบ 3G ไปประมวลผลที่เครื่องบริการเว็บ จากนั้นป้ายจอดรถร้องขอเวลาจากเครื่องบริการเว็บมาแสดงบนป้ายจอดรถไฟฟ้า ผู้ใช้สามารถติดตามสถานะผ่านเว็บไซต์หรือสมาร์ทโฟน จากการทดสอบผู้ใช้พึงพอใจกับระบบมาก ค่าใช้จ่ายในการติดตั้งระบบประมาณ 76,000 บาท สำหรับสมาร์ทโฟนระบุตำแหน่งปัจจุบันจำนวน 3 เครื่อง และป้ายจอดรถไฟฟ้าจำนวน 6 ป้าย

Project title	Tracking and Time Approximation System for EV via Smart Phone on Network 3G
Name	Mr. Kritsana Wathaniyanon ID. 53363317
	Mr. Chakkrit Termritthikun ID. 53363379
Project advisor	Mr. Settha Tangkawanit
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2013

Abstract

This project aims to facilitate the Naresuan University Electric Vehicle users. This system can track the EVs's position and estimated the arrival time at each bus stop and informed to users by graphic and text information on the Android 4.x smart phone. The system was designed the information to cover the red line service of EVs. The EVs's position and speed will be transmitted via 3G communication network to the web server for processing. Then the bus stop will request the estimated arrival time information from the server to display on its board. The users can track the status of the EV via website or smart phone application. From the experiments, the users were very satisfied with the system. The system implementation cost is approximately 76,000 Bath for 3 located smart phones and 6 bus stops.

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยความกรุณาเป็นอย่างยิ่งจากอาจารย์ที่ปรึกษา อาจารย์เศรษฐา ตั้งค้าวานิช ที่ให้คำปรึกษา แนะนำวิธีการแก้ไขปัญหาต่างๆ รวมถึงให้การสนับสนุนในทุกด้านจนกระทั่งโครงงานเสร็จสมบูรณ์ คณะผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวรผู้สนับสนุนงบประมาณเพื่อการดำเนินการ จากโครงการวิจัย “ระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G” ประจำปีงบประมาณ พ.ศ.2557

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา และ ดร.สุรเดช จิตประไพกุลศาล ที่สละเวลาเป็นอาจารย์กรรมการสอบโครงงานพร้อมทั้งให้คำแนะนำที่เป็นประโยชน์

ขอขอบพระคุณมหาวิทยาลัยนเรศวร ที่ปลูกฝังวิชาการความรู้และมอบประสบการณ์อันมีค่าอย่างยิ่งตลอดระยะเวลาที่ศึกษาอยู่ในมหาวิทยาลัยแห่งนี้

ขอขอบพระคุณครอบครัวเป็นอย่างสูง ซึ่งเป็นเบื้องหลังแห่งความสำเร็จทุกประการ เป็นผู้ให้การสนับสนุนในทุกๆ ด้านและคอยเป็นกำลังใจอย่างดีตลอดมา ซึ่งเป็นส่วนสำคัญให้คณะผู้จัดทำดำเนินโครงการจนเป็นผลสำเร็จ

สุดท้ายขอขอบคุณเพื่อนทุกคนที่ให้คำปรึกษา ให้คำแนะนำ แลกเปลี่ยนองค์ความรู้ที่เป็นประโยชน์ รวมทั้งคอยเป็นกำลังใจตลอดมา จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

คณะผู้จัดทำ

นายกฤษณะ วทานิยานนท์

นายจักรกฤษ เติมฤทธิกุล

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์.....ก

บทคัดย่อภาษาไทย.....ข

บทคัดย่อภาษาอังกฤษ.....ค

กิตติกรรมประกาศ.....ง

สารบัญ.....จ

สารบัญตาราง.....ช

สารบัญรูป.....ญ

บทที่ 1 บทนำ.....1

1.1. ที่มาและความสำคัญของโครงงาน.....1

1.2. วัตถุประสงค์.....2

1.3. ผลที่คาดว่าจะได้รับ.....2

1.4. ขอบเขตการดำเนินโครงงาน.....2

1.5. ขั้นตอนการดำเนินงาน.....3

1.6. แผนการดำเนินโครงงาน.....3

1.7. งบประมาณของโครงงาน.....3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....4

2.1 สมาร์ทโฟน (Smart phone).....4

สารบัญ (ต่อ)

หน้า

2.2 ระบบปฏิบัติการแอนดรอยด์ (Android).....5

2.3 เอพีไอ (Application programming interface) 12

2.4 เครื่องบริการเว็บ (Web Server).....16

2.6 IOIO Board.....21

2.7 การคำนวณหาระยะทางระหว่างจุด 2 จุดบนพิกัดภูมิศาสตร์.....24

บทที่ 3 วิธีดำเนินโครงงาน.....29

3.1 โปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า.....30

3.2 ส่วนคำนวณและจัดเก็บฐานข้อมูล.....34

3.3 ส่วนติดต่อกับผู้ใช้.....57

บทที่ 4 ผลการทดลอง.....68

4.1 การทดลองเก็บตำแหน่งของถนนเส้นทางมหาวิทยาลัยนเรศวร.....68

4.2 การทดลองความแม่นยำในการระบุตำแหน่งในกรณีรถไฟฟ้าเคลื่อนที่.....70

4.3 การทดลองความแม่นยำของโปรแกรมประยุกต์ตัวระบุตำแหน่งปัจจุบันของรถไฟฟ้า...71

4.4 การวัดประสิทธิภาพการส่งข้อมูลไปยังเครื่องบริการเว็บ.....75

สารบัญ (ต่อ)

หน้า

4.5 การทดสอบการคำนวณเวลาของเครื่องบริการ....76

4.6 การทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ....77

4.7 การทดลองแสดงผลส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน....79

4.8 การทดสอบระบบโดยรวม....85

4.9 อัตราค่าใช้จ่ายทั้งหมดของระบบ....86

บทที่ 5 สรุปผลและข้อเสนอแนะ....91

5.1 สรุปผลการทดลอง....91

5.2 เปรียบเทียบระบบติดตามรถไฟฟ้ามหาวิทยาลัยนเรศวรแต่ละรุ่น....92

5.3 ปัญหาในการดำเนินงานและแนวทางการแก้ไข....94

5.4 ข้อเสนอแนะ....94

5.5 ความรู้ที่จำเป็นต่อการพัฒนาต่อ....95

บรรณานุกรม....96

ภาคผนวก....98

ประวัติผู้ดำเนินโครงงาน....106

สารบัญตาราง

ตารางที่ หน้า

2.1 วิธีของเอชทีทีพี ในการติดต่อระหว่างผู้รับบริการกับเครื่องบริการ....18

2.2 IOIO Useful Libraries Chart....23

2.3 ข้อมูลตำแหน่งของจังหวัดพิษณุโลกและสุโขทัย....26

3.1 ตัวอย่างข้อมูลตำแหน่งจุดอ้างอิง....36

3.2 รูปแบบโพรโทคอลในการส่งข้อมูลไปยังเครื่องบริการเว็บ....45

3.3 คำอธิบายข้อมูลตอบกลับในการส่งข้อมูลไปยังเครื่องบริการเว็บ....46

3.4 รูปแบบโพรโทคอลในการขอรายชื่อป้ายจอดรถจากเครื่องบริการเว็บ....48

3.5 คำอธิบายข้อมูลตอบกลับในการขอรายชื่อป้ายจอดรถจากเครื่องบริการเว็บ....49

3.6 รูปแบบโพรโทคอลในการขอรายชื่อตำแหน่งจุดอ้างอิงจากเครื่องบริการเว็บ....50

3.7 คำอธิบายข้อมูลตอบกลับในการขอรายชื่อตำแหน่งจุดอ้างอิงจากเครื่องบริการเว็บ....51

3.8 รูปแบบโพรโทคอลในการขอรายชื่อข้อมูลจากเครื่องบริการเว็บ....52

3.9 คำอธิบายข้อมูลตอบกลับในการขอรายชื่อข้อมูลจากเครื่องบริการเว็บ....53

3.10 รูปแบบโพรโทคอลในการขอรายชื่อตำแหน่งจากเครื่องบริการเว็บ....54

3.11 คำอธิบายโพรโทคอลในการขอรายชื่อตำแหน่งจากเครื่องบริการเว็บ....55

4.1 แสดงความสัมพันธ์ระหว่างเวลาและความเร็วในการรับส่งข้อมูล....70

4.2 ค่าพิกัดตำแหน่งได้จากการกำหนดจุดบนบริการแผนที่ออนไลน์....72

4.3 ค่าพิกัดตำแหน่งได้จากการสมาร์ทโฟน....72

4.4 ผลการทดสอบหาค่าประสิทธิภาพความแม่นยำของการระบุตำแหน่งของสมาร์ทโฟน....74

สารบัญตาราง(ต่อ)

ตารางที่ หน้า

4.5 สรุปผลการทดสอบอัปโหลดข้อมูลไปยังเครื่องบริการเว็บ....75

4.6 การคำนวณเวลาของเครื่องบริการเว็บ....77

4.7 ข้อมูลการเปรียบเทียบการใช้งานโปรแกรมประยุกต์แสดงผลส่วนติดต่อกับผู้ใช้....83

4.8 กรณีที่ 1 เปิด/ปิดโปรแกรมประยุกต์ระบุตำแหน่งตามเวลาที่ใช้จริง....86

4.9 กรณีที่ 2 เปิดโปรแกรมประยุกต์ระบุตำแหน่งตลอดเวลา....86

4.10 อัตราค่าบริการสำหรับโปรแกรมประยุกต์ส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ 1 ป้าย....88

4.11 ค่าใช้จ่ายสำหรับติดตั้งอุปกรณ์ในครั้งแรก....89

4.12 ค่าใช้จ่ายสำหรับอินเทอร์เน็ต 3G และเครื่องบริการเว็บรายปี....89

5.1 เปรียบเทียบระบบติดตามรถไฟฟ้ามหาวิทยาลัยนเรศวรแต่ละรุ่น....93

5.2 ปัญหาในการดำเนินงานและแนวทางการแก้ไข....94

5.3 ข้อเสนอแนะและวิธีการ....95

สารบัญรูป

รูปที่ หน้า

2.1 สมาร์ทโฟนแอนดรอยด์รุ่น LG Nexus 5 5

2.2 สถาปัตยกรรมของการออกแบบแอนดรอยด์ 7

2.3 สถาปัตยกรรมของการออกแบบแอนดรอยด์ชั้นโปรแกรมประยุกต์ 7

2.4 สถาปัตยกรรมของการออกแบบแอนดรอยด์ชั้นโปรแกรมประยุกต์เฟรมเวิร์ค 7

2.5 สถาปัตยกรรมของการออกแบบแอนดรอยด์ชั้นไลบรารี 9

2.6 สถาปัตยกรรมของการออกแบบแอนดรอยด์ชั้นชั้นลีนุกซ์เคอร์เนล 10

2.7 การทำงานของกระบวนการที่มี 2 เทรดบน 1 โพรเซสเซอร์ 11

2.8 แผนที่ Google Maps บนเว็บไซต์ 13

2.9 แผนที่ OpenStreetMap บนเว็บไซต์ โดย Leaflet API 14

2.10 การเพิ่ม Marker บน Google Maps และบน OpenStreet 15

2.11 การเพิ่ม Marker พร้อมคำอธิบาย บน Google Maps และบน OpenStreetMap 15

2.12 การทำงานของ WWW 16

2.13 การทำงานแบบ ผู้ใช้บริการ - เครื่องบริการ 17

2.14 การทำงานของ ผู้รับบริการ – เครื่องบริการ ในลักษณะของการให้บริการฐานข้อมูล 17

2.14 องค์ประกอบของระบบฐานข้อมูล และความสัมพันธ์ 20

2.15 โครงสร้างเครือข่าย 3G มาตรฐาน UMTS 21

2.16 IOIO บอร์ด 21

2.17 ละติจูดและลองจิจูดบนโลก 24

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.18 รัศมีของโลก	25
2.19 แสดงระยะห่างระหว่างจังหวัดพิษณุโลกและสุโขทัยด้วย Haversine formula	27
3.1 โครงสร้างโดยรวมของระบบติดตามและประมาณเวลาการเดินรถไฟฟ้า	29
3.2 หลักการทำงานของโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า	30
3.3 โปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า (Sender)	31
3.4 แสดงขั้นตอนการทำงานของโปรแกรมประยุกต์ตัวระบุตำแหน่งรถไฟฟ้า	32
3.5 ตัวอย่างการอ่านค่า GPS ของสมาร์ทโฟน	32
3.6 รหัสเทียม (pseudo code) การอ่านค่า GPS ของสมาร์ทโฟน	33
3.7 โครงสร้างของส่วนคำนวณและจัดเก็บฐานข้อมูล	34
3.8 การติดต่อสื่อสารระหว่างผู้ใช้กับเครื่องบริการเว็บผ่าน API	35
3.9 รหัสเทียมสำหรับคำนวณระยะทางระหว่างจุด 2 จุด	36
3.10 ตำแหน่งจุดอ้างอิงโดยมีการกำหนดรัศมี 50 เมตร	37
3.11 แสดงเส้นทางจากการวาดเส้นตรงของตำแหน่งแต่ละจุดอ้างอิง	38
3.12 แสดงการเดินรถไฟฟ้าในกรณีที่รถไฟฟ้ามีมากกว่า 1 คัน	39
3.13 แสดงการเคลื่อนที่ของรถไฟฟ้า โดยป้ายจอดรถเลือกคำนวณเวลาโดยประมาณ	40
3.14 โครงสร้างของฐานข้อมูล	43
3.15 แสดงขั้นตอนการทำงานของเครื่องบริการบน API ส่งข้อมูล	44
3.16 ตัวอย่างการส่งข้อมูลพร้อมได้รับข้อมูลตอบกลับ	44

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.17 ข้อมูลตอบกลับในกรณีที่ส่งข้อมูลถูกต้อง โดยแสดงในรูปแบบ JSON....45

3.18 ข้อมูลตอบกลับในกรณีที่ส่งข้อมูลไม่ครบ โดยแสดงในรูปแบบ JSON....46

3.19 แสดงขั้นตอนการทำงานของเครื่องบริการบน API ขอข้อมูล....47

3.20 ข้อมูลตอบกลับในการขอรายชื่อป้ายจอดรถจากเครื่องบริการเว็บ รูปแบบ JSON....49

3.21 ข้อมูลตอบกลับในการขอรายชื่อตำแหน่งจุดอ้างอิงจากเครื่องบริการเว็บรูปแบบ JSON....51

3.22 ข้อมูลตอบกลับในการขอรายชื่อข้อมูลจากเครื่องบริการเว็บรูปแบบ JSON....53

3.23 ข้อมูลตอบกลับในการขอรายชื่อตำแหน่งจากเครื่องบริการเว็บรูปแบบ JSON....55

3.24 ตั้งค่าตำแหน่งจุดอ้างอิงของแผนที่เพื่อกำหนดเส้นทางของการเดินรถไฟฟ้า....56

3.25 ตั้งค่าป้ายจอดรถของแผนที่เพื่อกำหนดป้ายจอดรถในการแสดงเวลา....57

3.26 โครงสร้างของส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ....57

3.27 แสดงการทำงานของโปรแกรมประยุกต์ในส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ....58

3.28 การตั้งค่า USB Debugging....59

3.29 แสดงการทำงานโปรแกรมประยุกต์ระบุตำแหน่งในการร้องขอเวลาจากเครื่องบริการเว็บ....60

3.30 IOIO บอร์ด....61

3.31 แสดง Switching Power Supply Output 12VDC 2A....61

3.32 แสดงวงจรแปลง BCD เป็น Seven Segment....62

3.33 แสดงแผงวงจรแปลง BCD เป็น Seven Segment....63

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.34 แสดง Seven Segment 4 หลัก....63

3.35 แสดงวงจรสำหรับใช้งานร่วมกับ Seven Segment....64

3.36 โครงสร้างของส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน....64

3.37 แสดงขั้นตอนการทำงานของส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน....65

3.38 ส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟนแสดงผลเฉพาะแผนที่....66

3.39 ส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟนแสดงผลเฉพาะการประมาณเวลา....67

3.40 ส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟนในส่วนการตั้งค่าและระบบแจ้งเตือนเวลาอัตโนมัติ....67

4.1 ตัวอย่างค่า GPS บนสมาร์ทโฟน....68

4.2 การสื่อสารระหว่างสมาร์ทโฟนบนรถไฟฟ้าและเครื่องบริการเว็บ....69

4.3 การส่งข้อมูลถูกต้อง (ซ้าย) การส่งข้อมูลผิดพลาด (ขวา) 69

4.4 ตำแหน่งที่เก็บจากโปรแกรมประยุกต์ตัวระบุตำแหน่งปัจจุบันของรถไฟฟ้า....72

4.5 ความคลาดเคลื่อนของตำแหน่งป้ายจอดรถเมล์ประมาณ 2.486 เมตร....73

4.6 ความคลาดเคลื่อนของตำแหน่งป้ายจอดรถไฟฟ้าสามแยกตึก EE 73

4.8 แสดงจำนวนครั้งในแต่ละช่วงเวลาที่มีการอัปโหลดข้อมูล....75

4.9 แสดงอัตราเฉลี่ยของเวลาในการส่งข้อมูลไปยังเครื่องบริการเว็บ....76

4.10 ขั้นตอนการเลือกป้ายจอดรถของโปรแกรมประยุกต์ที่ใช้ติดต่อกับผู้ใช้บนป้ายจอดรถ....77

4.11 ขั้นตอนการแสดงเวลาของโปรแกรมประยุกต์ที่ใช้ติดต่อกับผู้ใช้บนป้ายจอดรถ....78

4.12 การแสดงเวลาบนสมาร์ทโฟนกับป้ายจอดรถไฟฟ้า....78

4.13 การแสดงผลในส่วนของแผนที่บนสมาร์ทโฟนและแสดงแผนที่บนเว็บไซต์....80

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.14 การแสดงผลในส่วนของประมาณเวลาบนสมาร์ทโฟนและแสดงเวลาบนเว็บไซต์.....81

4.15 การแสดงผลในส่วนของประมาณเวลากรณีที่รถไฟฟ้าวิ่งป้ายจอดรถ Bus stop.....82

4.16 การตั้งค่าระบบแจ้งเตือนอัตโนมัติโดยเลือกป้ายจอดรถ.....83

4.17 กราฟความพึงพอใจสาหรับผู้ใช้ในกลุ่มตัวอย่างที่มีสมาร์ทโฟน.....84

4.18 กราฟการแสดงเวลาที่ป้ายจอดรถในขณะที่รถไฟฟ้าเคลื่อนที่.....85

4.19 กราฟเปรียบเทียบปริมาณการใช้ 3G กรณีที่ 1 และ 2.....87

4.20 กราฟเปรียบเทียบค่าใช้จ่ายกรณีที่ 1 และ 2.....87

4.21 กราฟค่าใช้จ่ายสำหรับการติดตั้งอุปกรณ์และค่าบริการรายปี.....90

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ปัจจุบันรถไฟฟ้ามีความสำคัญในการเดินทางภายในมหาวิทยาลัยนเรศวรเป็นอย่างมาก เนื่องจากอาคารแต่ละแห่งมีระยะทางที่ไกลกันมาก จากการสำรวจ ปัญหาการใช้งานรถไฟฟ้าภายในมหาวิทยาลัยนเรศวร พบว่าการรอรถไฟฟ้าเป็นปัญหาที่สำคัญที่ผู้ใช้งานรถไฟฟ้าประสบ ปัญหานี้เกิดมาจากสาเหตุหลัก 2 ประการ คือ ประการแรกผู้ใช้บริการไม่ทราบเวลาการเดินรถไฟฟ้า ประการที่สองรถไฟฟ้ามีปัญหาด้านเทคนิค ซึ่งส่งผลให้ผู้ใช้เสียเวลาในการเดินทางเพิ่มมากขึ้นและไม่สามารถวางแผนการเดินทางได้

แนวทางหนึ่งที่จะช่วยให้ผู้ใช้สามารถทราบเวลาการเดินรถไฟฟ้า คือ ระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G ที่ผู้ใช้สามารถใช้งานผ่านสมาร์ทโฟนโดยสามารถติดตามและทราบเวลาในการเดินรถไฟฟ้า เพื่อให้ผู้ใช้สามารถวางแผนการเดินทางได้

รถพลังงานไฟฟ้าถือว่าเป็นพลังงานทางเลือกในการลดมลพิษทางอากาศ โดยปัจจุบันนิยมใช้รถพลังงานไฟฟ้ามากขึ้น อันเนื่องมาจากพลังงานเชื้อเพลิงมีจำนวนน้อยลงและราคาสูงขึ้น พลังงานไฟฟ้าซึ่งถือว่าเป็นพลังงานสะอาด (Green energy) เพราะไม่มีการปล่อยแก๊สเรือนกระจก (Greenhouse gases) [1]

โครงการระบบติดตามและแสดงตำแหน่งรถไฟฟ้ามหาวิทยาลัยนเรศวร ภายใต้การสนับสนุนจากงานวิจัยรถไฟฟ้า ได้มีการดำเนินการวิจัยมา 3 โครงการ ดังนี้ โครงการแรก การติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย [2] โครงการที่สอง ระบบเครือข่ายและแสดงผลสำหรับติดตามรถไฟฟ้า [3] และโครงการที่สาม ระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยเครือข่าย Zigbee [4]

ปัญหาของเครือข่าย Zigbee คือการติดตั้งเราเตอร์จำนวนมาก เพื่อให้เครือข่าย Zigbee สามารถส่งข้อมูลระหว่างเราเตอร์ได้ โดยจำเป็นต้องมีการบำรุงรักษาและค่าใช้จ่ายจำนวนมาก หากมีการใช้เราเตอร์จำนวนมาก

ด้วยเหตุนี้ จึงมีการพัฒนาระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G มาใช้แทนเครือข่าย Zigbee ซึ่งจะเพิ่มความสะดวกของผู้ให้บริการและผู้ใช้บริการทั้งด้านการติดตั้งที่ลดลงและการเลือกใช้ผ่านระบบเครือข่าย 3G ที่มีความครอบคลุมทั่วประเทศ โดยหวังว่าการพัฒนาในครั้งนี้จะทำให้ผู้ให้บริการและผู้ใช้บริการจะสามารถวางแผนการเดินทางได้ พร้อมทั้งนำไปพัฒนาในระบบขนส่งมวลชนให้มีประสิทธิภาพมากยิ่งขึ้น

1.2 วัตถุประสงค์

1) เพื่อพัฒนาระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G

2) เพื่อบันทึกข้อมูลการเคลื่อนที่ของรถไฟฟ้า

1.3 ผลที่คาดว่าจะได้รับ

1) เพื่อใช้งานสมาร์ทโฟนให้เกิดประโยชน์มากขึ้น (ใช้งานนอกจาก โทรศัพท์, SMS, อินเทอร์เน็ต) และประยุกต์ใช้งานอุปกรณ์ที่เกี่ยวกับสมาร์ทโฟนผ่านเครือข่าย 3G

2) วิเคราะห์พฤติกรรมการใช้งานรถไฟฟ้าภายในมหาวิทยาลัยนเรศวร

3) อาจจะมีผู้ใช้บริการรถไฟฟ้าเพิ่มมากขึ้นและลดปริมาณการใช้รถยนต์ส่วนบุคล และรถจักรยานยนต์ ของนิสิต คณาจารย์ และบุคลากรภายในมหาวิทยาลัย

1.4 ขอบเขตการดำเนินโครงงาน

1) โปรแกรมประยุกต์ (Application) สำหรับระบบปฏิบัติการ Android 4.x ขึ้นไป ที่ระบุตำแหน่งปัจจุบันของรถไฟฟ้าผ่านเครือข่าย 3G

2) เครื่องบริการเว็บที่สามารถคำนวณด้วยภาษา PHP 5.2 ขึ้นไป และจัดเก็บฐานข้อมูลด้วย MySQL 5.0 ขึ้นไป เพื่อติดตามและประมาณเวลาของการเดินรถไฟฟ้ามหาวิทยาลัยนเรศวร

3) แสดงเวลาการเดินรถไฟฟ้าของมหาวิทยาลัยนเรศวรได้ตั้งแต่ 1 - 3 คัน

1.5 ขั้นตอนการดำเนินงาน

1) ศึกษาข้อมูลเกี่ยวกับการใช้งานและทำงานของเว็บเซิร์ฟเวอร์

2) ศึกษาข้อมูลเกี่ยวกับการเขียนโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์

3) ศึกษาข้อมูลเกี่ยวกับการส่งข้อมูลด้วยสมาร์ทโฟนผ่านเครือข่าย 3G

4) พัฒนาระบบติดตามรถไฟฟ้า

5) พัฒนาระบบประมาณเวลารถไฟฟ้า

6) ทดสอบและปรับปรุงระบบติดตามและประมาณเวลารถไฟฟ้า

1.6 แผนการดำเนินโครงงาน

กิจกรรม	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1) ศึกษาข้อมูลเกี่ยวกับการใช้งานและทำงานของเว็บเซิร์ฟเวอร์									
2) ศึกษาข้อมูลเกี่ยวกับการเขียนโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์									
3) ศึกษาข้อมูลเกี่ยวกับการส่งข้อมูลด้วยสมาร์ทโฟนผ่านเครือข่าย 3G									
4) พัฒนาระบบติดตามรถไฟฟ้า									
5) พัฒนาระบบประมาณเวลารถไฟฟ้า									
6) ทดสอบและปรับปรุงระบบติดตามและประมาณเวลารถไฟฟ้า									

1.7 งบประมาณของโครงงาน

1.	หนังสือประกอบการทำโครงการ	500 บาท
2.	ค่าเอกสาร	1,000 บาท
3.	อื่นๆ	500 บาท
	รวม	2,000 บาท

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้กล่าวถึงหลักการและทฤษฎีต่างๆ ที่ใช้ในการจัดทำโครงงาน โดยเป็นการศึกษาข้อมูลเกี่ยวกับ อุปกรณ์ที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ ประกอบไปด้วย สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ บอร์ด IOIO รวมทั้งระบบเอพีไอบนการทำงานของเครื่องบริการเว็บ และยังต้องศึกษาเกี่ยวกับการคำนวณหาระยะทางระหว่างจุด 2 จุดบนพิกัดภูมิศาสตร์ ทั้งหมดนี้เป็นการศึกษาเพื่อให้เกิดความรู้ความเข้าใจและนำมาประยุกต์ใช้งาน

2.1 สมาร์ทโฟน (Smart phone)

โทรศัพท์เคลื่อนที่แบบพกพาที่มีความสามารถมากกว่าการโทรออกและรับสาย ทำให้สมาร์ทโฟนมี [5] ความสามารถในลักษณะเดียวกับคอมพิวเตอร์ สมาร์ทโฟนมีระบบปฏิบัติการรุ่นใหม่ๆ เช่น Android, iOS, Symbian, Ubuntu Phone, BlackBerry, Windows Phone ฯลฯ โปรแกรมที่ติดตั้งมาพร้อมกับเครื่องสมาร์ทโฟนเรียกว่า โปรแกรมประยุกต์ (Application) ด้านการติดต่อสื่อสารนั้น สมาร์ทโฟนสามารถเชื่อมต่อกับอินเทอร์เน็ต (Internet) ผ่าน Wi-Fi, EDGE/GPRS, 3G เป็นต้น ความสามารถในด้านอื่นๆ สมาร์ทโฟนมีโปรแกรมประยุกต์พื้นฐาน เช่น ปฏิทิน, เครื่องคิดเลข, สมุดบันทึก, นาฬิกา, แผนที่, กล้องถ่ายรูป, อีเมล์, วิทยุ ฯลฯ

สมาร์ทโฟนที่นำมาใช้พัฒนาจำเป็นต้องมีระบบปฏิบัติการที่มีความสามารถในการระบุตำแหน่งผ่าน GPS การเชื่อมต่ออินเทอร์เน็ตและสามารถเชื่อมต่อกับ IOIO บอร์ด เนื่องจากส่วนระบุตำแหน่งจะต้องนำข้อมูลตำแหน่งปัจจุบันและความเร็วมาใช้งาน ในส่วนอื่นๆ รวมทั้งส่วนระบุตำแหน่ง จำเป็นต้องมีการเชื่อมต่ออินเทอร์เน็ตเพื่อติดต่อกับเครื่องบริการเว็บเพื่อจัดการกับฐานข้อมูลและในส่วนแสดงผลต้องมีการเชื่อมต่อกับ IOIO บอร์ดเพื่อส่งค่าไปยังวงจร Seven Segment ดังนั้นจึงเลือกสมาร์ทโฟนที่มีความสามารถดังกล่าวคือ ระบบปฏิบัติการแอนดรอยด์

2.2 ระบบปฏิบัติการแอนดรอยด์ (Android)

2.2.1 ความหมายของระบบปฏิบัติการแอนดรอยด์

ระบบปฏิบัติการแอนดรอยด์ [6] คือซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อน (Stack) โดยรวมระบบปฏิบัติการ (Operating system) มิดเดิลแวร์ (Middleware) และโปรแกรมประยุกต์ (Application) เข้าไว้ด้วยกันเพื่อใช้สำหรับการทำงานบนอุปกรณ์พกพา เช่น สมาร์ทโฟน ดังรูปที่ 2.1

รูปที่ 2.1 สมาร์ทโฟนแอนดรอยด์รุ่น LG Nexus 5 [7]

แอนดรอยด์ทำงานบนพื้นฐานของระบบลีนุกซ์ เคอร์เนล (Linux Kernel) ซึ่งมีการผนวก Android SDK (Software Development Kit) เป็นเครื่องมือในการพัฒนาโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์ โดยพัฒนาด้วยโครงสร้างของภาษาจาวา (Java)

ระบบปฏิบัติการแอนดรอยด์ เริ่มพัฒนาโดยบริษัทแอนดรอยด์ ต่อมาได้ผนวกเข้ากับบริษัท Google ในเดือนพฤศจิกายน ปีพุทธศักราช 2550 โดยมีการร่วมมือระหว่างบริษัทชั้นนำ 33 บริษัท เพื่อพัฒนาระบบปฏิบัติการแอนดรอยด์ ในการพัฒนามีทั้งบริษัทผู้ให้บริการโทรศัพท์เคลื่อนที่ บริษัทผู้ผลิตอุปกรณ์โทรศัพท์เคลื่อนที่ บริษัทซอฟต์แวร์ และบริษัทเอกชนต่างๆ ซึ่งเรียกว่า OHA (Open Handset Alliance)

ทั้งนี้กลุ่ม OHA ได้ร่วมมือพัฒนามาตรฐานสำหรับการพัฒนาซอฟต์แวร์ระบบเปิด (Open System) หรือ โอเพ่นซอร์ส (Open Source) โดยมีลิขสิทธิ์ตาม Apache Version 2 license ซึ่งหลักลิขสิทธิ์ของ Apache อนุญาตให้ผู้พัฒนาสามารถนำรหัสต้นฉบับ (Source code) ไปพัฒนาต่อได้ ทั้งในส่วนแบบการค้า (Commercial) หรือซอฟต์แวร์กรรมสิทธิ์ (Proprietary) และแบบใช้ฟรีหรือฟรีแวร์ (Freeware)

2.2.2 ประเภทของระบบปฏิบัติการแอนดรอยด์

เนื่องจากระบบปฏิบัติการแอนดรอยด์ เป็นซอฟต์แวร์ระบบเปิดจึงอนุญาตให้นักพัฒนาหรือผู้ที่สนใจสามารถดาวน์โหลด (Download) รหัสต้นฉบับได้ ทำให้ผู้พัฒนาจากหลายๆ ฝ่ายนำรหัสต้นฉบับไปปรับแต่งและพัฒนาโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์ ในแบบฉบับของตนเอง โดยสามารถแบ่งประเภทของระบบปฏิบัติการแอนดรอยด์ ออกเป็นกลุ่มได้ 3 ประเภท ดังนี้

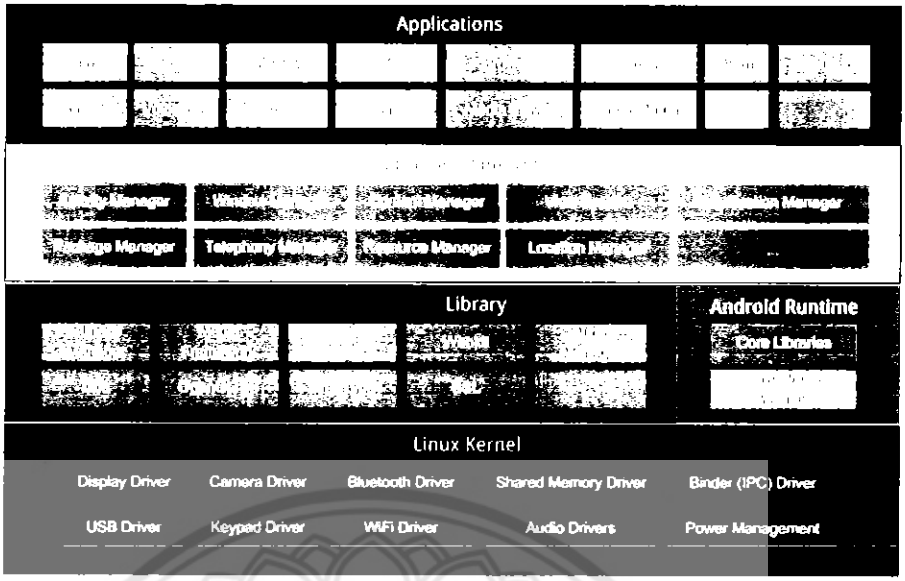
- AOSP - Android Open Source Project ระบบปฏิบัติการแอนดรอยด์ประเภทแรก ที่บริษัท Google เปิดให้สามารถนำรหัสต้นฉบับไปติดตั้งและใช้งานในอุปกรณ์ต่างๆ ได้โดยไม่เสียค่าใช้จ่าย

- OHM - Open Handset Mobile ระบบปฏิบัติการแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่ม OHA ซึ่งจะพัฒนาระบบในแบบฉบับของตนเอง โดยมีรูปร่างหน้าตาการแสดงผลและฟังก์ชันการใช้งานที่แตกต่างกัน รวมไปถึงอาจจะมีความเป็นเอกลักษณ์และรูปแบบการใช้งานเป็นของแต่ละบริษัท และโปรแกรมประยุกต์บนแอนดรอยด์ ประเภทนี้จะได้รับสิทธิบริการเสริมต่างๆ จาก Google ที่เรียกว่า GMS - Google Mobile Service ซึ่งเป็นบริการเสริมที่ทำให้ระบบปฏิบัติการแอนดรอยด์มีประสิทธิภาพมากขึ้น

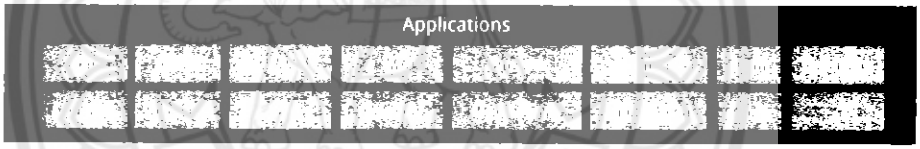
- Cooking หรือ Customize ระบบปฏิบัติการแอนดรอยด์ที่นักพัฒนานำเอารหัสต้นฉบับจากแหล่งต่างๆ มาปรับแต่งให้อยู่ในแบบฉบับของตนเอง ซึ่งการพัฒนาจะต้องปล่อย (Unlock) สิทธิในการใช้งานอุปกรณ์ก่อนจึงจะสามารถติดตั้งได้ ทั้งนี้ระบบประเภทนี้ถือได้ว่าเป็นประเภทที่มีความสามารถสูงที่สุด เนื่องจากจะได้รับการปรับแต่งขีดความสามารถต่างๆ ให้มีความเข้ากันได้กับอุปกรณ์นั้นๆ จากผู้ใช้งานจริง

2.2.3 สถาปัตยกรรมของแอนดรอยด์ (Android Architecture)

สถาปัตยกรรมของการออกแบบแอนดรอยด์นั้นถูกแบ่งออกเป็นลำดับชั้น (Layer) โดยแต่ละชั้นจะเรียกใช้บริการจากระดับชั้นที่อยู่ด้านล่าง ซึ่งสามารถแบ่งออกเป็น 4 ชั้นหลักดังรูปที่ 2.2 คือ ชั้นลินุกซ์เคอร์เนล (Linux Kernal), ชั้นไลบรารี (Library), ชั้นโปรแกรมประยุกต์เฟรมเวิร์ค (Application Framework) และชั้นโปรแกรมประยุกต์ (Application)

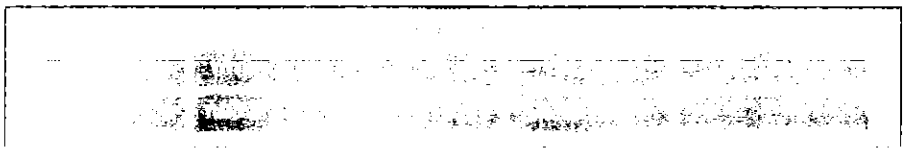


รูปที่ 2.2 สถาปัตยกรรมของการออกแบบแอนดรอยด์ [6]



รูปที่ 2.3 สถาปัตยกรรมของการออกแบบแอนดรอยด์ชั้นโปรแกรมประยุกต์ [6]

จากรูปที่ 2.3 ชั้นโปรแกรมประยุกต์ (Application) ชั้นนี้เป็นชั้นบนสุดของโครงสร้างแอนดรอยด์ ซึ่งเป็นส่วนของโปรแกรมประยุกต์ที่พัฒนาขึ้นมาใช้งาน เช่น โปรแกรมประยุกต์รับส่งอีเมล (E-mail), โปรแกรมประยุกต์โทรศัพท์ (Phone Dial), โปรแกรมประยุกต์เว็บเบราว์เซอร์ (Web Browser) เป็นต้น ทั้งนี้โปรแกรมในชั้นโปรแกรมประยุกต์นั้นจะอยู่ในรูปแบบของไฟล์ APK - Android application package file ซึ่งโดยทั่วไปแล้วจะอยู่ในสารบบ (Directory) data/app ของสมาร์ทโฟน



รูปที่ 2.4 สถาปัตยกรรมของการออกแบบแอนดรอยด์ชั้นโปรแกรมประยุกต์เฟรมเวิร์ค [6]

จากรูปที่ 2.4 ชั้นโปรแกรมประยุกต์เฟรมเวิร์ค (Application Framework) นักพัฒนาสามารถเรียกใช้งานแอนดรอยด์ผ่านเอพีไอ (Application programming interface) ได้ ซึ่งแอนดรอยด์ได้ออกแบบไว้เพื่อลดความซ้ำซ้อนในการใช้งานซ้ำของ Application Component ซึ่งมีตัวอย่างโปรแกรมประยุกต์ เฟรมเวิร์ค ดังรูปที่ 2.4 โดยมีส่วนประกอบดังนี้

- View System เป็นส่วนควบคุมการทำงานสำหรับการสร้างโปรแกรมประยุกต์ เช่น lists, grids, textboxs, buttons และ embeddable web browser

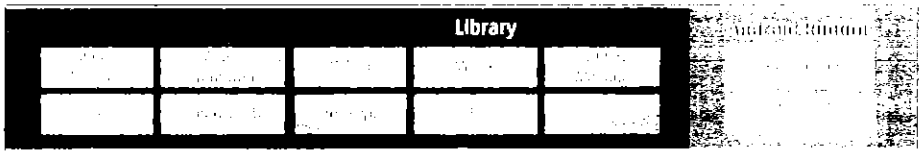
- Location Manager เป็นส่วนดูแลค่าตำแหน่งของเครื่องอุปกรณ์พกพาเคลื่อนที่

- Content Provider เป็นส่วนควบคุมการเข้าถึงของข้อมูลที่มีการใช้งานร่วมกัน (Share Data) ระหว่างโปรแกรมประยุกต์ที่แตกต่างกัน เช่น ข้อมูลผู้ใช้ (Contact)

- Resource Manager เป็นส่วนดูแลการเข้าใช้ข้อมูลต่างๆ ที่ไม่ใช่โค้ด เช่น localized string, graphics และ layout ซึ่งจะอยู่ในสารบบ res/ ทั้งนี้ข้อมูลต่างๆ ในส่วนนี้จะถูกแปลโปรแกรม (Compile) และผนวกเข้ากับโปรแกรมประยุกต์ที่เขียนนั้น ณ เวลาการ Build โดยแอนดรอยด์จะใช้เครื่องมือ The Android Asset Packaging Tool (aapt) สำหรับการแปลโปรแกรม และหลังจากการแปลโปรแกรมแล้วจะสร้างคลาส (Class) ที่ชื่อว่า R ซึ่งเป็นส่วนที่ระบุถึงข้อมูลต่างๆ ที่ใช้อ้างอิงสำหรับโปรแกรมประยุกต์ที่ถูกพัฒนาขึ้น

- Notification Manager เป็นส่วนควบคุมเหตุการณ์ (Event) ต่างๆ ที่แสดงบนสถานะ (Status bar) เช่น ในกรณีที่ได้รับข้อความและการแจ้งเตือนต่างๆ

- Activity Manager เป็นส่วนควบคุม Life Cycle ของโปรแกรมประยุกต์



รูปที่ 2.5 สถาปัตยกรรมของการออกแบบแอนดรอยด์ชั้นไลบรารี [6]

จากรูปที่ 2.5 ชั้นไลบรารี (Library) แอนดรอยด์ได้รวบรวมกลุ่มของไลบรารีต่างๆ ที่สำคัญและมีความจำเป็นต่อการพัฒนาโปรแกรมประยุกต์ไว้มากมาย ซึ่งถูกเขียนด้วยภาษา C และ C++ โดยตัวอย่างของไลบรารีที่สำคัญดังนี้

- System C library เป็นกลุ่มของมาตรฐานที่อยู่บนพื้นฐานของภาษา C ไลบรารี (libc)

- Media Libraries เป็นกลุ่มการทำงานมัลติมีเดีย เช่น ออดิโอ วีดีโอ รวมถึงรูปภาพต่างๆ เช่น ไฟล์สกุล MPEG4, H.264, MP3, ACC, AMR, JPG และ PNG

- Surface Manager เป็นกลุ่มการจัดการรูปแบบของหน้าจอ การวาดหน้าจอ

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

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

- SQLite เป็นกลุ่มของฐานข้อมูล ซึ่งเป็นระบบฐานข้อมูลที่ใช้เช่นเดียวกันกับซอฟต์แวร์ Firefox และ Apple iPhone ทั้งนี้นักพัฒนาสามารถใช้ฐานข้อมูลนี้เก็บข้อมูลของโปรแกรมประยุกต์ต่างๆ ได้

- Browser Engine เป็นกลุ่มของการแสดงผลบนเว็บเบราว์เซอร์โดยอยู่บนพื้นฐานของ Webkit

สำหรับการเรียกใช้โปรแกรมประยุกต์ต่างๆ ในชั้นไลบรารี จะไม่สามารถเรียกใช้โปรแกรมประยุกต์ในระดับเดียวกันได้ โดยจะต้องเรียกใช้โปรแกรมประยุกต์ในชั้นที่สูงกว่าเท่านั้นจึงจะสามารถเรียกใช้ได้ นอกจากนี้ในชั้นไลบรารี แอนดรอยด์ยังแบ่งเป็นชั้นย่อยที่เรียกว่า Android Runtime ซึ่งประกอบไปด้วย 2 ส่วนหลักคือ

- Dalvik VM (Virtual Machine) ส่วนนี้ถูกเขียนด้วยภาษาจาวา เพื่อใช้เฉพาะการใช้งานอุปกรณ์พกพาเคลื่อนที่ อย่างไรก็ตามสิ่งที่แตกต่างจาก Java VM (Virtual Machine) คือ Dalvik VM ซึ่งจะรันไฟล์ .dex ที่แปลโปรแกรมมาจากไฟล์ .class และ .jar โดยมีเครื่องมือ (Tool) ที่ชื่อว่า dx ทำหน้าที่ในการบีบอัด (Compress) คลาสของภาษาจาวา ทั้งนี้ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์พกพามากกว่า .class โดยเฉพาะอย่างยิ่งประสิทธิภาพในการใช้พลังงานจากแบตเตอรี่

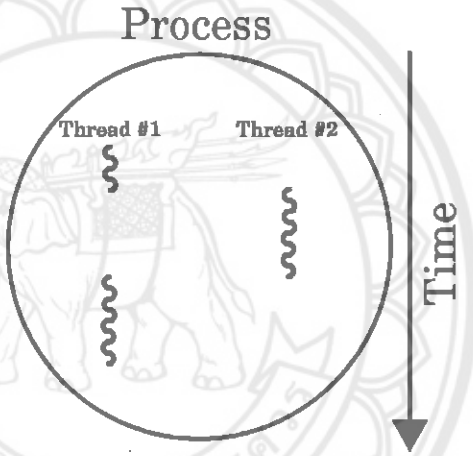
- Core Java Library ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็จะมีความแตกต่างจากไลบรารีของ Java SE (Java Standard Edition) และ Java ME (Java Mobile Edition)

รูปที่ 2.6 สถาปัตยกรรมของการออกแบบแอนดรอยด์ชั้นชั้นลีนุกซ์เคอร์เนล [6]

จากรูปที่ 2.6 ชั้นลีนุกซ์เคอร์เนล (Linux Kernal) ระบบแอนดรอยด์อยู่บนพื้นฐานของระบบปฏิบัติการ Linux โดยชั้น Linux Kernal มีฟังก์ชั่นการทำงานหลายๆ ส่วน ซึ่งแต่ละส่วนถูกพัฒนาขึ้นด้วยภาษา C เช่น การจัดการหน่วยความจำ (Memory Management) การจัดการกระบวนการ (Process Management) ทั้งนี้นักพัฒนาจะไม่มีสิทธิเข้าถึงส่วนนี้ได้โดยตรง อย่างไรก็ตามนักพัฒนาสามารถเข้าถึงระบบปฏิบัติการ Linux ได้จากชุดคำสั่ง Command Prompt เช่น adb shell ซึ่งจะสามารถใช้เครื่องมือต่างๆ ได้ เช่น การเข้าดูระบบไฟล์ (File System), กระบวนการคัดลอกไฟล์ (Copy File) เป็นต้น

2.2.4 กระบวนการและเทรด (Processes and Threads)

เมื่อโปรแกรมประยุกต์เริ่มต้นการทำงาน จะทำให้ส่วนประกอบต่างๆ ของโปรแกรมประยุกต์ไม่สามารถทำงานได้ เนื่องจากระบบปฏิบัติการแอนดรอยด์จะเริ่มทำงานแบบลินุกซ์โดยโปรแกรมประยุกต์จะทำงานแบบเทรดเดี่ยว (Single Thread) [8] ซึ่งเป็นการทำงานโดยปริยายที่กำหนดโดยระบบปฏิบัติการ ดังนั้นเมื่อต้องการกระบวนการที่ทำงานได้พร้อมๆ กัน เพื่อทำงานมากกว่า 1 งานในเวลาเดียวจะสามารถทำได้ด้วยการเขียนโปรแกรมประยุกต์ในรูปแบบมัลติเทรด (Multithreaded) [9] ดังรูปที่ 2.7



รูปที่ 2.7 การทำงานของกระบวนการที่มี 2 เทรดบน 1 โพรเซสเซอร์ [10]

การประมวลผลโปรแกรมประยุกต์พร้อมกันหลายๆ โปรแกรม จะมีการแบ่งการทำงานออกเป็นช่วงๆ และช่วงที่กำลังทำงานอยู่เรียกว่า งาน หรือ กระบวนการ (Process) [11] เป็นส่วนหนึ่งของระบบ ดังนั้นจึงมีการแบ่งเวลาในการเข้าใช้งานทรัพยากรของระบบเพื่อให้ดำเนินงานสำเร็จ เช่นเวลาซีพียู (CPU – Control Processing Unit) หน่วยความจำ ไฟล์ และอุปกรณ์อินพุต/เอาต์พุต เป็นต้น ระบบปฏิบัติการมีหน้าที่ในการจัดการกระบวนการทั้งหมด เช่นการสร้างกระบวนการ การลบกระบวนการ การจัดเวลาของกระบวนการ, การจัดการซิงโครไนซ์ และการติดต่อสื่อสารในกระบวนการ

เทรด (Threads) [12] ส่วนประกอบย่อยของกระบวนการ หรืออาจเรียกว่า Lightweight Process (LWP) โดย 1 กระบวนการประกอบไปด้วยเทรดเดี่ยวหรือมัลติเทรด ซึ่งเทรดประกอบด้วยส่วนต่าง ๆ ดังนี้

- Thread ID หมายเลขเทรดในกระบวนการ
- ตัวนับ เพื่อติดตามให้ทราบคำสั่งต่อไปที่จะประมวลผล
- ชุดของรีจิสเตอร์ เพื่อเก็บค่าตัวแปรที่ทำงานอยู่
- เรียงทับซ้อน (Stack) เพื่อเก็บประวัติการประมวลผล

ภายในกระบวนการประกอบด้วยเทรดซึ่งมีการแชร์รหัสคำสั่ง, ข้อมูล และทรัพยากร เช่น ไฟล์ อุปกรณ์ต่างๆ ถ้าเป็นกระบวนการที่มีเทรดเดี่ยว จะทำงานได้ทีละงาน แต่กระบวนการที่มีมัลติเทรดจะทำงานหลายๆ งานได้พร้อมกัน

ข้อดีของการทำงานมัลติเทรด คือ การตอบสนอง ในการทำงานหลายๆ งานพร้อมกันและหากมีการทำงานเป็นเวลานานในระบบมัลติเทรดนั้นจะทำให้งานอื่นๆ ทำงานต่อไปได้ เช่นการเปิดเว็บหลายๆ หน้าพร้อมกัน เว็บจะโหลดข้อมูลเข้ามาได้เรื่อยๆ แม้จะมีบางหน้าที่กำลังเปิดภาพอยู่ก็ตาม การแชร์ทรัพยากร โดยปกติเทรดจะมีการแชร์ทรัพยากรของกระบวนการ ทำให้โปรแกรมประยุกต์สามารถมีกิจกรรมหลายๆ อย่างของเทรด ภายในแอดเดรสเดียวกัน ด้านความประหยัดเนื่องจากระบบมัลติเทรด มีการใช้งานหน่วยความจำและทรัพยากรต่างๆ ร่วมกัน ทำให้เกิดความประหยัดค่าใช้จ่ายในการทำงาน การเอื้อประโยชน์ของสถาปัตยกรรมมัลติโพรเซสเซอร์ การที่มีหลายเทรดในกระบวนการเดียวทำให้สามารถแบ่งเทรดให้ CPU หลายตัวช่วยในการประมวลผล ทำให้ประมวลผลกระบวนการได้รวดเร็วขึ้น ซึ่งหากเป็นกระบวนการที่มีเทรดเดี่ยวจะใช้งาน CPU เพียง 1 หน่วยประมวลผลเท่านั้น

2.3 เอพีไอ (Application programming interface)

เอพีไอ (Application programming interface) ส่วนต่อประสานโปรแกรมประยุกต์คือวิธีการสำหรับการเรียกใช้งานระบบปฏิบัติการหรือโปรแกรมประยุกต์อื่นๆ ซึ่งทำหน้าที่เชื่อมต่อการทำงานระหว่างโปรแกรมประยุกต์กับระบบปฏิบัติการ ทั้งนี้การที่โปรแกรมประยุกต์จะเชื่อมต่อการ

ทำงานกับระบบปฏิบัติการได้นั้น จำเป็นต้องมีเอพีไอเป็นตัวเชื่อม ซึ่งหากไม่มีการเปิดเผยเอพีไอของระบบปฏิบัติการ ผู้ที่พัฒนาโปรแกรมประยุกต์จะมีความลำบากเมื่อต้องการพัฒนาโปรแกรมประยุกต์ให้มีความเข้ากันได้กับระบบปฏิบัติการได้อย่างมีประสิทธิภาพ

ช่องทางการติดต่อระหว่างระบบปฏิบัติการกับซอฟต์แวร์แอพพลิเคชั่นคือ ไลบรารี (library) ที่รวบรวมกระบวนการและฟังก์ชันย่อย (subroutine) ที่พัฒนาขึ้นมาเพื่อให้นักพัฒนาสามารถเรียกใช้งานลักษณะการนำไปใช้ สำหรับเอพีไอที่นำมาใช้งานดังนี้

2.3.1 Google Maps API

บริการของ Google ที่สามารถให้นักพัฒนาสามารถนำข้อมูลต่างๆ เช่น แผนที่, ถนน, แม่น้ำ, สถานที่สำคัญ, ร้านค้า เป็นต้น ไปใช้งานได้ในหลายแพลตฟอร์ม (Platform) ยกตัวอย่าง เช่น เว็บไซต์ หรือ สมาร์ทโฟน โดยตัวอย่างตำแหน่งมหาวิทยาลัยนเรศวรบน Google Maps API [13] ดังรูปที่ 2.8

รูปที่ 2.8 แผนที่ Google Maps บนเว็บไซต์ [14]

2.3.2 OpenStreetMap API

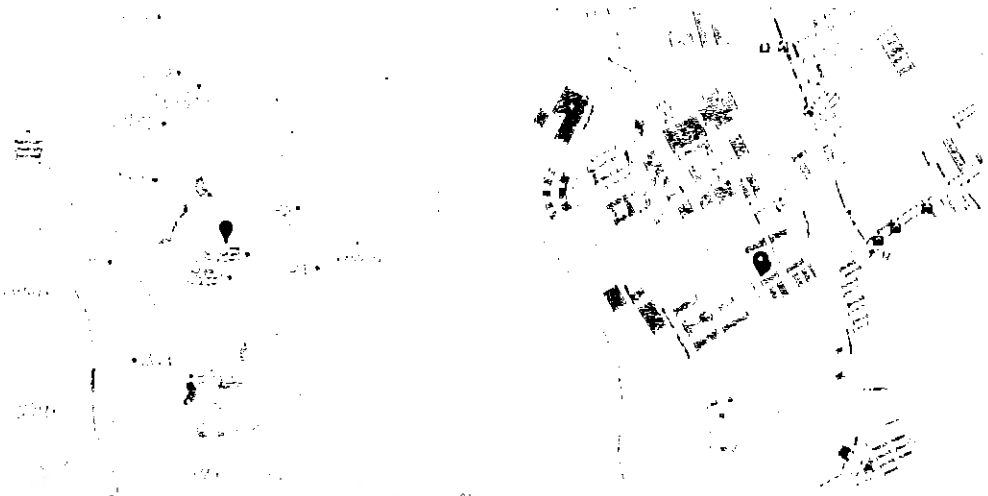
โครงการแบบเดียวกับ Google Maps ต่างกันตรงที่เป็นบริการแผนที่เสรี ที่ให้ผู้ใช้สามารถนำไปใช้งานได้โดยไม่มีกฎหมายลิขสิทธิ์คุ้มครอง โดยอาศัยการเก็บข้อมูลผ่านอาสาสมัครทั่วโลกช่วยวาดแผนที่ในบริเวณท้องถิ่นของตนผ่าน GPS เนื่องจาก OpenStreetMap API [15] เป็นบริการแผนที่เสรี ดังนั้นจึงมีการพัฒนาเอพีไอหลากหลาย ซึ่งเอพีไอที่มีความสามารถลักษณะเดียวกันกับ Google Maps คือ Leaflet ซึ่งแสดงผลดังรูปที่ 2.9

รูปที่ 2.9 แผนที่ OpenStreetMap บนเว็บไซต์ โดย Leaflet API [16]

2.3.3 เอพีไอพื้นฐานที่เกี่ยวข้องกับแผนที่

นอกจากการทำงานในส่วนแสดงแผนที่แล้ว Google Maps และ OpenStreetMap สามารถที่จะเพิ่มรายละเอียดต่างๆ บนแผนที่ได้ โดยเอพีไอที่นำมาใช้งาน ดังนี้

- การจัดการ Marker สามารถเพิ่มหรือลบ ซึ่งผู้ใช้สามารถที่จะกำหนดพิกัดและสัญรูป (Icon) โดยการกำหนด Marker ณ ตำแหน่งมหาวิทยาลัยนเรศวร ดังรูปที่ 2.10



รูปที่ 2.10 การเพิ่ม Marker บน Google Maps (ซ้าย) [14] และบน OpenStreetMap (ขวา) [16]

- การกำหนดรายละเอียดของ Marker เพื่อดูข้อมูลเพิ่มเติม ซึ่งสามารถกำหนดได้พร้อมกับการกำหนด Marker ดังตัวอย่างรูปที่ 2.11

รูปที่ 2.11 การเพิ่ม Marker พร้อมคำอธิบาย บน Google Maps (ซ้าย) [14]
และบน OpenStreetMap (ขวา) [16]

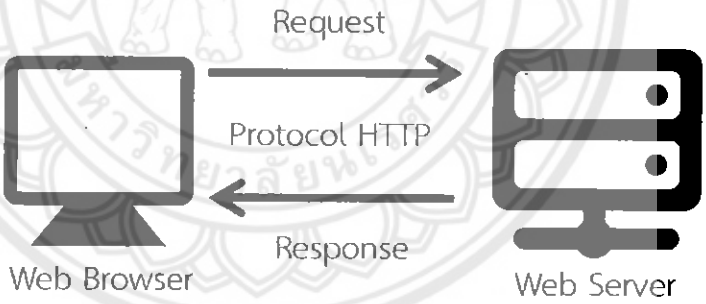
- การเพิ่มรูปร่างต่างๆ บนแผนที่ ซึ่งสามารถ เพิ่มหรือลบ โดยมีรูปร่างที่สามารถใช้งานได้ เช่น เส้น (Polylines) สี่เหลี่ยม (Rectangles) วงกลม (Circles) และรูปร่างที่ผู้ใช้สามารถกำหนดได้ (User-editable)

เอพีไอ ทั้ง Google Maps และ OpenStreetMap ยังสามารถดำเนินการต่างๆ กับแผนที่ได้อีกมากมาย เพื่อให้นักพัฒนาสามารถนำเอพีไอต่างๆ ไปประยุกต์กับแผนที่ให้มีความเหมาะสม ทั้งนี้เอพีไอบน Google Maps และ OpenStreetMap มีการพัฒนาที่แตกต่างกัน ดังนั้นการนำไปใช้งานจึงขึ้นอยู่กับความต้องการของระบบเป็นหลัก เนื่องจากความแตกต่างของเอพีไอและข้อกำจัดเรื่องลิขสิทธิ์คุ้มครอง

2.4 เครื่องบริการเว็บ (Web Server)

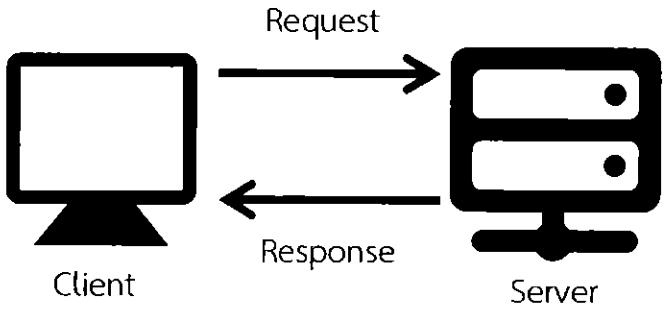
เครื่องคอมพิวเตอร์ที่ให้บริการพื้นที่จัดเก็บเว็บไซต์ให้ผู้ใช้เรียกใช้เว็บไซต์ได้โดยใช้โพรโทคอล (Protocol) เอชทีทีพี (HTTP) ผ่านทางเว็บบราวเซอร์ อีกหน้าที่หลักคือ เป็นสื่อกลางระหว่างการเชื่อมต่อโปรแกรมกับฐานข้อมูล

2.4.1 หลักการทำงานของเวิลด์ไวด์เว็บ (WWW)



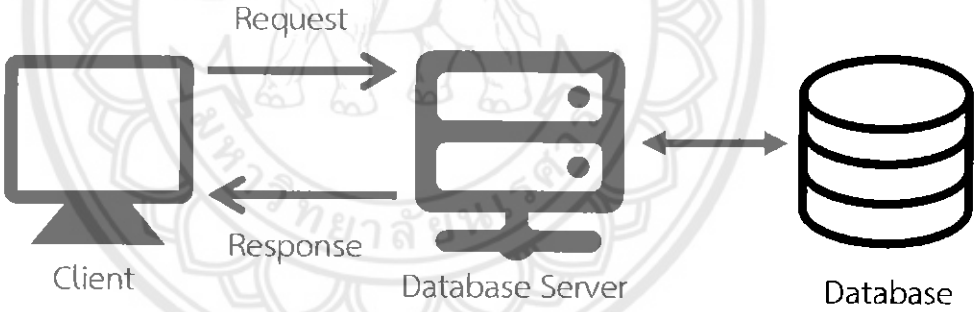
รูปที่ 2.12 การทำงานของ WWW [17]

จากรูปที่ 2.12 การทำงานของ WWW - World Wide Web [17] มีลักษณะใกล้เคียงกับการทำงานแบบ ผู้รับบริการ - เครื่องบริการ (Client-Server) คือ ลักษณะของการเชื่อมต่อของเครื่องผู้ให้บริการ (Server) และเครื่องผู้ใช้บริการ (Client) ดังรูปที่ 2.13



รูปที่ 2.13 การทำงานแบบ ผู้ใช้บริการ - เครื่องบริการ [17]

จากรูปที่ 2.13 ผู้รับบริการ (Client) จะร้องขอ (Request) ข้อมูลจากเครื่องบริการ (Server) หลังจากให้บริการข้อมูลตามที่ร้องขอจะทำการตอบกลับ (Response) ไปยังผู้รับบริการ ซึ่งเครื่องบริการจะมีเพียงเครื่องเดียว ส่วนผู้รับบริการอาจจะมีได้หลายเครื่อง ดังนั้นผู้รับบริการหลายเครื่องสามารถร้องขอจากเครื่องบริการพร้อมกันได้



รูปที่ 2.14 การทำงานของ ผู้รับบริการ – เครื่องบริการ
ในลักษณะของการให้บริการฐานข้อมูล [17]

จากรูปที่ 2.14 จะมีลักษณะคล้ายการทำงานแบบ ผู้รับบริการ – เครื่องบริการ โดยเครื่องบริการจะทำหน้าที่จัดเก็บข้อมูลทั้งหมดไว้ในฐานข้อมูล ซึ่งจะเรียกว่า เครื่องบริการข้อมูล (Database Server) การทำงานจะเริ่มจากผู้ใช้บริการ คือ ผู้ใช้บริการสามารถเพิ่ม ลบ แก้ไข หรือเลือกดูข้อมูลที่จัดเก็บไว้ในฐานข้อมูล ข้อดีของการเชื่อมต่อลักษณะนี้คือ จัดการข้อมูลต่างๆ ไว้เพียงที่เดียว ทำให้ข้อมูลมีความเป็นได้อย่างเดียว (Uniqueness) คือข้อมูลไม่กระจัดกระจาย

2.4.2 HTTP - Hypertext Transfer Protocol

โพรโทคอลในระดับชั้นโปรแกรมประยุกต์ (Application layer) สำหรับการแจกจ่ายและการทำงานร่วมกันกับสารสนเทศของสื่อหลายมิติ (Hypermedia) ใช้สำหรับการสื่อสารข้อมูลสำหรับเวิลด์ไวด์เว็บ

ฟังก์ชันเอชทีทีพี [18] คือการร้องขอและการตอบกลับระหว่างผู้รับบริการกับเครื่องบริการ ซึ่งผู้รับบริการคือผู้ใช้ปลายทางและเครื่องบริการคือเว็บไซต์ ผู้รับบริการจะการร้องขอข้อมูลจากเอชทีทีพีผ่านทางเว็บเบราว์เซอร์ ส่วนเครื่องบริการจะตอบกลับ โดยกำหนดทรัพยากร (resource) ในรูปแบบ เช่น ไฟล์เอชทีเอ็มแอล (HTML) และเนื้อหาต่างๆ หรือทำหน้าที่อื่นๆ ในนามของผู้รับบริการเพื่อส่งข้อความตอบกลับไปยังผู้รับบริการ ซึ่งข้อความตอบกลับจะประกอบไปด้วยสถานะการร้องขอและเนื้อหาต่างๆ

ผู้รับบริการจะเริ่มสร้างการร้องขอก่อน โดยเปิดการเชื่อมต่อด้วยเกณฑ์วิธีควบคุมการขนส่งข้อมูล (TCP) ไปยังพอร์ตเฉพาะของเครื่องบริการ (พอร์ต 80 เป็นค่าปริยาย) เครื่องบริการที่เปิดพอร์ตนั้น จะรอให้ผู้รับบริการส่งข้อความร้องขอเข้ามา เมื่อได้รับการร้องขอแล้ว เครื่องบริการจะตอบกลับด้วยข้อความสถานะ ตัวอย่างเช่น "HTTP/1.1 200 OK" ตามด้วยเนื้อหาข้อมูลที่ร้องขอ ข้อความแสดงข้อผิดพลาด หรือข้อมูลอย่างอื่น เป็นต้น

ทรัพยากรที่ถูกเข้าถึงด้วยเอชทีทีพีจะถูกระบุโดยใช้ตัวระบุแหล่งทรัพยากรสากล (URI) หรือเจาะจงคือ ตัวชี้แหล่งในอินเทอร์เน็ต (URL) โดยใช้ http: หรือ https: เป็นแผนของตัวระบุ (URI scheme)

ตารางที่ 2.1 วิธีของเอชทีทีพี ในการติดต่อระหว่างผู้รับบริการกับเครื่องบริการ [19]

วิธี (Method)	การทำงาน (Action)
GET	ร้องขอเอกสารจากเครื่องบริการ
HEAD	ร้องขอข้อมูลเกี่ยวกับเอกสารแต่ไม่รวมในส่วนของเนื้อหา
PUT	ส่งเอกสารจากผู้รับบริการไปเครื่องบริการ
POST	ส่งข้อมูลบางอย่างจากผู้รับบริการไปเครื่องบริการ
TRACE	ส่งข้อมูลร้องขอกลับมา

วิธี (Method)	การทำงาน (Action)
DELETE	ลบเว็บเพจ
CONNECT	สงวน
OPTIONS	สอบถามข้อมูลเกี่ยวกับตัวเลือกที่ใช้ได้

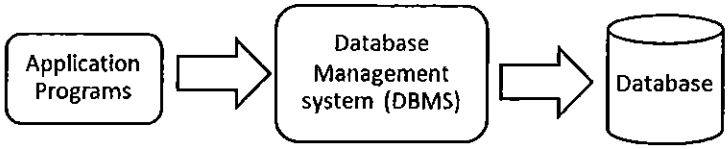
จากตารางที่ 2.1 เอชทีทีพีจะมีการกำหนดวิธี (Method) เพื่อแสดงให้เห็นการกระทำที่ต้องการที่จะดำเนินการกับทรัพยากรที่ระบุ สิ่งที่ทรัพยากรนั้นนำเสนอไม่ว่าเป็นข้อมูลที่มีอยู่ก่อนหรือสร้างขึ้นมาแบบพลวัต จะขึ้นอยู่กับการนำไปใช้ของเครื่องบริการ ซึ่งทรัพยากรมักจะสอดคล้องกับไฟล์ในเครื่องบริการนั้น โดยวิธีการดำเนินการต่างๆ

เครื่องบริการเอชทีทีพีจะต้องสามารถใช้วิธี GET และ HEAD ได้เป็นอย่างน้อย ดังนั้นเพื่อความเหมาะสมในการสื่อสารระหว่างสมาร์ทโฟนกับเครื่องบริการ จึงเลือกใช้วิธี GET ในร้องขอข้อมูลจากเครื่องบริการและวิธี POST ในการส่งข้อมูลให้กับเครื่องบริการ

2.4.3 ความหมายของฐานข้อมูล (database)

ฐานข้อมูล (database) [17] หมายถึง กลุ่มของข้อมูลที่ถูกรวบรวมไว้ในศูนย์กลาง โดยข้อมูลมีความสัมพันธ์กัน และลดความซ้ำซ้อนออกจากข้อมูล เพื่อนำข้อมูลมาใช้ร่วมกัน การจัดการการใช้ฐานข้อมูลจะดำเนินการผ่านระบบจัดการฐานข้อมูล (DBMS - Data Base Management system)

ระบบจัดการฐานข้อมูล คือ ซอฟต์แวร์ที่ทำหน้าที่สื่อกลางระหว่างผู้ใช้และโปรแกรมต่างๆ ที่เกี่ยวข้องกับฐานข้อมูล มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่าย สะดวกและรวดเร็ว โดยผู้ใช้สามารถสร้างฐานข้อมูล แก้ไขฐานข้อมูล หรือเลือกดูฐานข้อมูลตามเงื่อนไขต่างๆ ดังนั้นหากต้องการที่จะใช้งาน DBMS จะต้องดำเนินการผ่านระบบหรือโปรแกรมที่เรียกใช้ข้อมูล (Application Programs) ดังรูปที่ 2.14

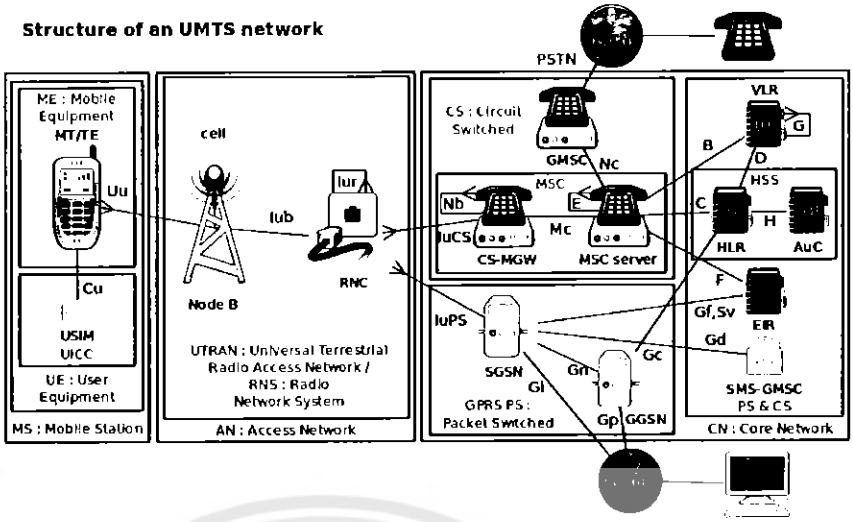


รูปที่ 2.14 องค์ประกอบของระบบฐานข้อมูล และความสัมพันธ์ [17]

MySQL คือระบบจัดการฐานข้อมูลด้วยภาษา SQL โดย MySQL เป็นฐานข้อมูลแบบโอเพนซอร์สที่ได้รับความนิยมมาก เนื่องจากประสิทธิภาพในการทำงานที่รวดเร็ว และการใช้งานง่าย ดังนั้นจึงเป็นนิยมในฐานะโอเพนซอร์ส อีกทั้ง MySQL สร้างบน LAMP Stack (Linux, Apache, MySQL, PHP/Perl/Python) และยังทำงานได้มากกว่า 20 แพลตฟอร์ม เช่น Linux, Windows, OS X, HP-UX, AIX, Netware เป็นต้น

2.5 3G (3rd generation mobile telecommunications)

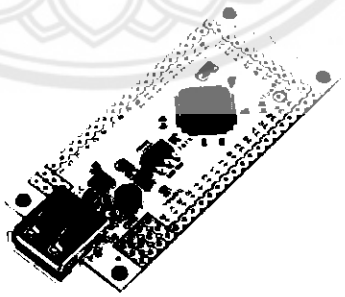
มาตรฐานโทรศัพท์มือถือยุคที่ 3 [20] พัฒนาเพื่อแทนที่ระบบโทรศัพท์ 2G เพื่อให้ผู้ใช้บริการสามารถใช้งานอินเทอร์เน็ตบนสมาร์ทโฟนในการรับส่งข้อมูลได้รวดเร็ว สามารถรองรับจำนวนผู้ใช้งานที่เพิ่มขึ้น โดยใช้คลื่นความถี่ตั้งแต่ 850, 900, 1800, 1900 และ 2100 MHz โดยเน้นความสามารถในการใช้งานด้านข้อมูลภาพและเสียงในระบบไร้สายด้วยความเร็วสูง (ความเร็วของสัญญาณขึ้นอยู่กับหลายปัจจัยหลักของระบบ) ลักษณะการทำงานของ 3G เมื่อเปรียบเทียบเทคโนโลยี 2G กับ 3G แล้ว 3G มีช่องสัญญาณความถี่ และความจุในการรับส่งข้อมูลที่มากกว่า ทำให้ประสิทธิภาพในการรับส่งข้อมูล รวมทั้งบริการระบบเสียงดีขึ้น สามารถใช้บริการมัลติมีเดียได้เต็มที่ และสมบูรณ์แบบขึ้นเช่น โทรศัพท์ต่างประเทศ รับส่งข้อความที่มีขนาดใหญ่ ประชุมทางไกลผ่านหน้าจออุปกรณ์สื่อสาร โดยโครงสร้างเครือข่าย 3G มาตรฐาน UMTS [21] ดังรูปที่ 2.15



รูปที่ 2.15 โครงสร้างเครือข่าย 3G มาตรฐาน UMTS [22]

2.6 IOIO Board

IOIO Board (โย-โย่ บอร์ด) [23] เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ทำหน้าที่เป็นบอร์ดอินพุต/เอาต์พุต เพื่อทำให้อุปกรณ์ที่ติดตั้งระบบปฏิบัติการแอนดรอยด์ (Android OS) สามารถเชื่อมต่ออุปกรณ์ภายนอกได้ผ่านพอร์ต USB ดังรูปที่ 2.16



รูปที่ 2.16 IOIO บอร์ด

2.6.1 ความสามารถของ IOIO บอร์ด

IOIO บอร์ด ทำหน้าที่อินพุต/เอาต์พุต โดยมีสัญญาณดิจิทัล (Digital) และแอนะล็อก (Analog) โดยรูปแบบการสื่อสารกับแอนดรอยด์ผ่าน adb (Android Debug Bridge) เพื่อเชื่อมต่อกับอุปกรณ์แอนดรอยด์ด้วยพอร์ต USB ซึ่งรองรับแอนดรอยด์เวอร์ชั่น 1.5 ขึ้นไป ในการพัฒนาโปรแกรมประยุกต์สามารถใช้ Android SDK และเขียนบนภาษาจาวา ดังนั้นในส่วนของฮาร์ดแวร์ (Hardware) ของบอร์ดไม่จำเป็นต้องมีการติดตั้งเฟิร์มแวร์ใหม่ ซึ่งจะเน้นการพัฒนาในส่วนของภาษาจาวา โดยเรียกการใช้งานต่างๆ ผ่านเอพีไอ

2.6.2 ชิป (Chip) ภายใน IOIO บอร์ด

IOIO บอร์ด เป็นไมโครคอนโทรลเลอร์ (Microcontroller) เบอร์ PIC24FJ256DA210 อยู่ภายในซึ่งมีคุณสมบัติดังนี้

- เป็นสถาปัตยกรรมชนิด 16-bit
- CPU มีความเร็ว 16 MIPS - Million Instruction Per Second
- Flash Program Memory ขนาด 128 กิโลไบต์
- RAM - Random Access Memory ขนาด 98,304 ไบต์
- ทำงานได้ในช่วงแรงดัน 2.2 ถึง 3.6 โวลต์ (Voltage)
- มี I/O จำนวน 52 พิน (Pins) (4-UART, 3-SPI, 3-I2C, 1-A/D 16x10-bit)
- มี Graphics Hardware Acceleration จำนวน 3 ตัว
- Color Look-up Table (CLUT) ที่มีได้สูงสุด 256 รายการ
- มีอินเตอร์เฟสโดยตรงกับ Monochrome, C-STN, TFT, OLED
- ความละเอียด 640x480 @30Hz, 16 bpp, หรือ 480x272 @60Hz, 16 bpp
- มี USB v2.0 On-The-Go (OTG)

- สามารถเป็นได้ทั้งแม่ข่าย (Host) และอุปกรณ์รอบข้าง (Peripheral)

- สามารถใช้งาน Capacitive touch sensing ได้ถึง 24 ตัว

นอกจากคุณสมบัติข้างต้น IOIO บอร์ด ได้ถูกพัฒนาขึ้นด้วยหลักการโอเพ่นซอร์ส คล้ายกับแอนดรอยด์ ดังนั้นจึงมีการสร้างไลบรารี เพื่อให้นักพัฒนาสามารถเรียกใช้งานเอพีไอได้ง่ายขึ้น ซึ่งมีไลบรารีดังตารางที่ 2.2

ตารางที่ 2.2 IOIO Useful Libraries Chart [23]

Library name	หน้าที่
IOIO	จัดการอินเตอร์เฟสทุกอย่างใน IOIO บอร์ด โดยทุกฟังก์ชั่นจะต้องผ่านไลบรารี
IOIO Factory	จัดการเชื่อมต่อช่องทาง TCP ของ IOIO บอร์ด
Digital Input	อินพุตจากพินต่างๆ และเก็บค่าเป็น Low หรือ High
Digital Output	เอาต์พุตจากพินต่างๆ และเก็บค่าเป็น Low หรือ High
Analog Input	อินพุตจากพินต่างๆ โดยเก็บค่าเป็น Voltage ที่ป้อนเข้ามา หรืออัตราส่วน
PWM Output	เอาต์พุตจากพินต่างๆ เป็นสัญญาณพัลส์ (Pulse) โดยสามารถกำหนดความกว้างได้
SPI Master	จัดการเชื่อมต่อช่องทาง SPI BUS ของบอร์ด IOIO
TWI Master	จัดการเชื่อมต่อช่องทาง Two Wire Interface ของบอร์ด IOIO
UART	จัดการเชื่อมต่อช่องทาง UART ทั้ง Input steam และ Output steam ของบอร์ด IOIO
Connection Lose Exception	ตรวจสอบการเชื่อมต่อระหว่าง IOIO กับ Android ว่าปกติหรือไม่
Out Of Recourse Exception	ตรวจสอบการใช้ทรัพยากรของ IOIO ว่าเกินขีดจำกัดหรือไม่
Incompatibility Exception	ตรวจสอบเมื่ออุปกรณ์ที่ไม่รองรับเชื่อมต่อเข้ามา

การพัฒนาโปรแกรมประยุกต์กับ IOIO บอร์ด จะเน้นการพัฒนาโปรแกรมบนอุปกรณ์แอนดรอยด์เป็นหลัก โดยมีการติดตั้งไลบรารีและเอพีไอ ที่ใช้ในการควบคุมบอร์ด IOIO เพื่อติดต่อกับ

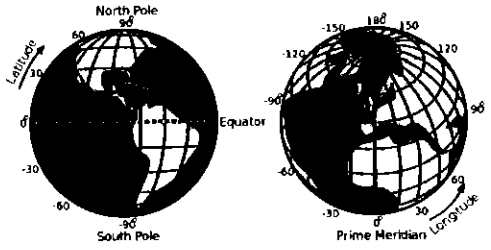
พอร์ตต่างๆ ให้สามารถใช้งานได้ด้วยการจัดการการเชื่อมต่อต่างๆ ผ่านพอร์ต USB บนอุปกรณ์แอนดรอยด์ ดังนั้นในการพัฒนาจึงมุ่งเน้นไปที่การเขียนโปรแกรมประยุกต์บนอุปกรณ์แอนดรอยด์ ด้วยการเรียกใช้งานเอพีไอเพื่อสั่งให้ IOIO บอร์ด ทำงานตามที่เขียนโปรแกรมไว้

2.7 การคำนวณหาระยะทางระหว่างจุด 2 จุดบนพิกัดภูมิศาสตร์

การคำนวณระยะทางระหว่างจุดนั้น จะไม่สามารถคำนวณในรูปแบบสมการเชิงเส้นได้ เนื่องจากการคำนวณนั้นมีพารามิเตอร์ (Parameter) เป็นตำแหน่ง GPS (Global Positioning System) แสดงหน่วยในระบบพิกัดแบบค่าตัวเลขทศนิยม (DD : Decimal Degree) เพื่อให้คอมพิวเตอร์สามารถคำนวณได้ง่ายขึ้น โดยจำเป็นต้องมีการศึกษาข้อมูลเกี่ยวกับระบบพิกัดภูมิศาสตร์และการคำนวณด้วย Haversine formula ดังนี้

2.7.1 ระบบพิกัดภูมิศาสตร์ (Geographic Coordinates System) [19-20]

พิกัดภูมิศาสตร์ (Geographic Coordinates System) [24] คือระบบที่บอกให้ทราบตำแหน่งของสถานที่ต่าง ๆ บนพื้นผิวโลก โดยยึดตามระบบพิกัดทรงกลม (spherical coordinate system) ซึ่งเป็นระบบค่าพิกัด [25] ที่อ้างอิงละติจูด (Latitude) และลองจิจูด (Longitude) ดังรูปที่ 2.17



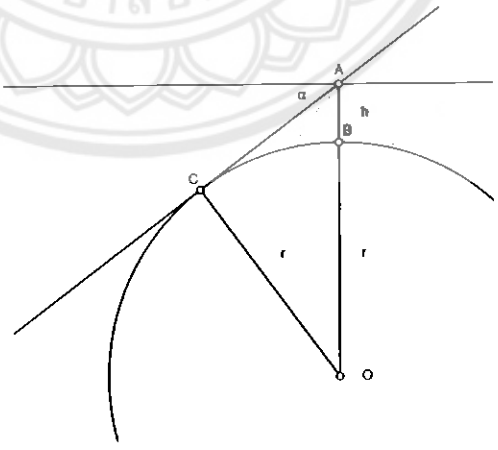
รูปที่ 2.17 ละติจูดและลองจิจูดบนโลก [26]

- ละติจูด คือเส้นที่ลากจากขั้วโลกเหนือมายังขั้วโลกใต้ ละติจูดที่ลากผ่านเส้นศูนย์สูตร (Equartor) จะมีค่า 0 องศา ละติจูดทางทิศเหนือของเส้นศูนย์สูตร มีค่าระหว่าง 0 ถึง 90 องศา และทางทิศใต้ของเส้นศูนย์สูตร มีค่าระหว่าง 0 ถึง -90 องศา โดยขั้วโลกแต่ละขั้วจะมีค่าละติจูดเป็น 90 องศา เช่น ขั้วโลกเหนือมีละติจูด 90 องศาเหนือ เป็นต้น

- ลองจิจูด คือเส้นที่ลากจากทิศทางตะวันออกหรือทิศทางตะวันตกผ่านเมืองกรีนิช ลอนดอน สหราชอาณาจักร ลองจิจูด 0 องศา จะเรียกว่าเส้นไพรม์เมริเดียน (Prime Meridian) ลองจิจูดทางทิศตะวันออก จะมีค่าระหว่าง 0 ถึง 180 องศา และทางทิศตะวันตกของไพรม์เมริเดียน จะมีค่า 0 ถึง -180 องศา เส้นที่ลากต่อเชื่อมทุกจุดที่มีลองจิจูดเท่ากันจะเรียกว่า เส้นเมริเดียน (meridian)

2.7.2 Haversine formula

Harversine formula [27] สมการในการคำนวณหาระยะหางระหว่างจุด 2 จุดบนพิกัดภูมิศาสตร์ โดยใช้ละติจูด และลองจิจูดมาคำนวณหาระยะทาง ซึ่งมีการใช้รัศมีของโลกในการคำนวณด้วย ดังรูปที่ 2.18



รูปที่ 2.18 รัศมีของโลก [28]

การคำนวณหาระยะทางสามารถคำนวณได้จากสมการ 2.1 นั้นคือ Haversine formula โดยอาศัยการหาระยะทางบนวงกลม ซึ่งโลกมีลักษณะเป็นวงกลมจึงมีรัศมี ดังนั้นจึงมีการนำรัศมีดังกล่าวมาคำนวณเพื่อหาระยะทางระหว่างจุด 2 จุดบนโลก ดังสมการที่ 2.1

$$d = 2r \arcsin\left(\sqrt{sin^2\left(\frac{\phi 2-\phi 1}{2}\right) + \cos(\phi 1)\cos(\phi 2)sin^2\left(\frac{\lambda 2-\lambda 1}{2}\right)}\right) \qquad (2.1)$$

โดยที่ d คือ ระยะห่าง (กิโลเมตร)

r คือ รัศมีโลก (รัศมีโลกเฉลี่ย = 6,371 กิโลเมตร)

$\phi 1, \phi 2$ คือ ละติจูดของจุดที่ 1 และจุดที่ 2

$\lambda 1, \lambda 2$ คือ ลองจิจูดของจุดที่ 1 และจุดที่ 2

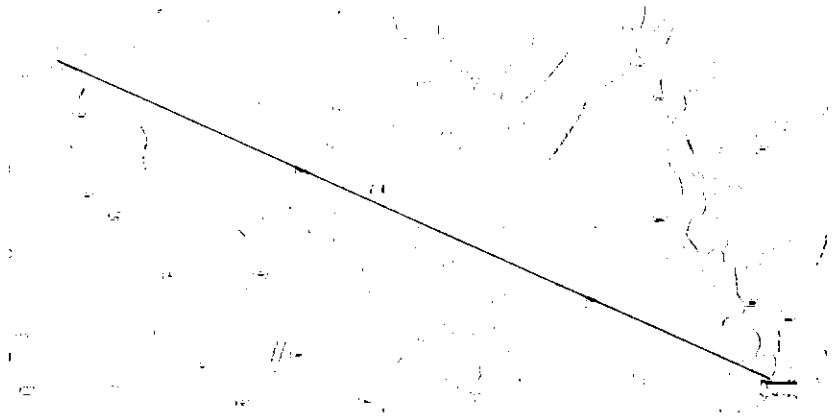
ละติจูดและลองจิจูดจะต้องถูกแปลงจากองศา (degrees) เป็นเรเดียน (radians) ด้วยการคูณ TT /180 ก่อนนำมาคำนวณ

ตัวอย่างการคำนวณหาระยะห่างระหว่างจังหวัดพิษณุโลกและสุโขทัย

การคำนวณระยะห่างระหว่างจุด 2 จุด จำเป็นต้องมีข้อมูลในการคำนวณ 2 ตำแหน่ง โดยแต่ละตำแหน่งจะประกอบไปด้วยละติจูดและลองจิจูด ซึ่งสมการในการคำนวณนั้นจะต้องมีการแปลงละติจูดและลองจิจูดให้อยู่ในหน่วยเรเดียนเสียก่อน ตามตารางที่ 2.3 จึงสามารถคำนวณหาระยะทางได้ โดยหากทำการวาดเส้นระหว่างตำแหน่งจะพบว่าเป็นเส้นตรงสีแดง ดังรูปที่ 2.19 เนื่องจาก Haversine formula ไม่ได้คำนวณตามเส้นทางถนน ดังนั้นระยะทางที่ได้จะไม่ตรงกับระยะทางบน Google maps

ตารางที่ 2.3 ข้อมูลตำแหน่งของจังหวัดพิษณุโลกและสุโขทัย [29]

จังหวัด	ละติจูด (องศาทศนิยม)	ละติจูด (เรเดียน)	ลองจิจูด (องศาทศนิยม)	ลองจิจูด (เรเดียน)
พิษณุโลก	16.823427	0.293624193	100.259870	1.749864839
สุโขทัย	17.010064	0.296881623	99.819844	1.742184937



รูปที่ 2.19 แสดงระยะห่างระหว่างจังหวัดพิษณุโลกและสุโขทัยด้วย Haversine formula [14]

จากสมการ 2.1

$$d = 2r \arcsin\left(\sqrt{sin^2\left(\frac{\phi 2 - \phi 1}{2}\right) + \cos(\phi 1)\cos(\phi 2)sin^2\left(\frac{\lambda 2 - \lambda 1}{2}\right)}\right)$$

โดยที่

$\Delta Latitude = \phi 2 - \phi 1 = 0.296881623 - 0.293624193 = 0.00325743$

$\Delta Longitude = \lambda 2 - \lambda 1 = 1.742184937 - 1.749864839 = -0.007679902$

$M = cos(0.293624193) = 0.95720123857$

$N = cos(0.296881623) = 0.95625338607$

$r = 6371$

ดังนั้น

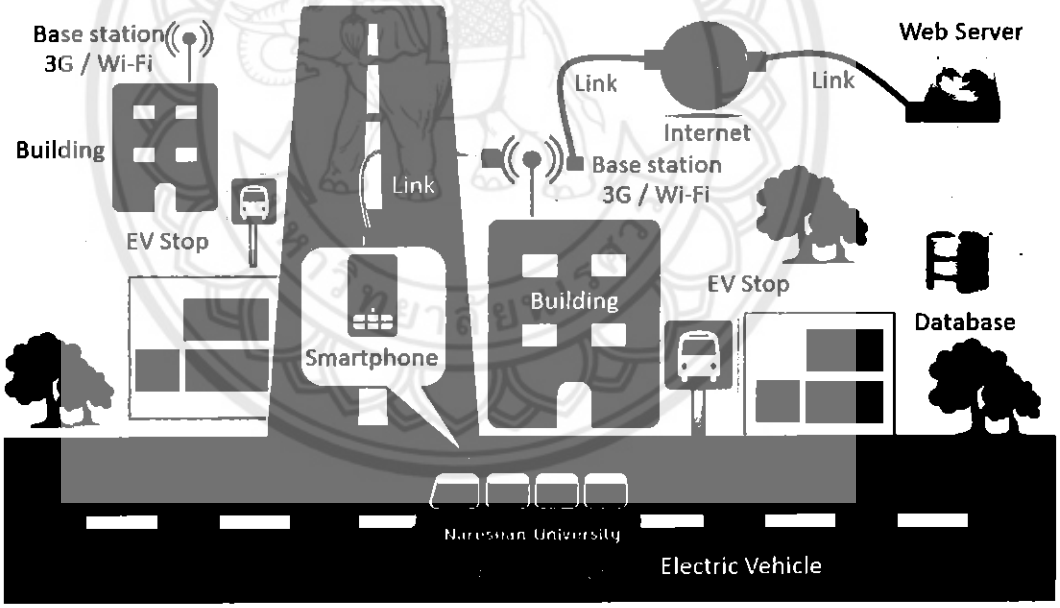
$$d = 2r * \arcsin\left(\sqrt{sin^2\left(\frac{\Delta Latitude}{2}\right) + M * N * sin^2\left(\frac{\Delta Longitude}{2}\right)}\right)$$

ระยะทาง (d) โดยประมาณ คือ 51.205 กิโลเมตร

จากหลักการและทฤษฎีต่างๆ ที่ได้อธิบายไว้ข้างต้น ได้แก่ อุปกรณ์ที่ทำงานบนระบบปฏิบัติการแอนดรอยด์, ระบบเอพีไอบนเครื่องบริการเว็บ และการคำนวณหาระยะทางระหว่างจุด 2 จุดบนพิกัดภูมิศาสตร์ เป็นหนึ่งในแผนการดำเนินโครงงาน ในหัวข้อศึกษาข้อมูล เพื่อให้เข้าใจและนำไปปฏิบัติ โดยการศึกษาข้อมูลดังกล่าวคือการนำเสนอหลักการที่สำคัญ โดยสามารถศึกษาเพิ่มเติมได้จากแหล่งอ้างอิง ในส่วนของการดำเนินการโครงงานจะอธิบายไว้ในบทถัดไป เพื่อให้เข้าใจหลักการทำงานของระบบในหัวข้อต่างๆ

บทที่ 3
วิธีดำเนินโครงงาน

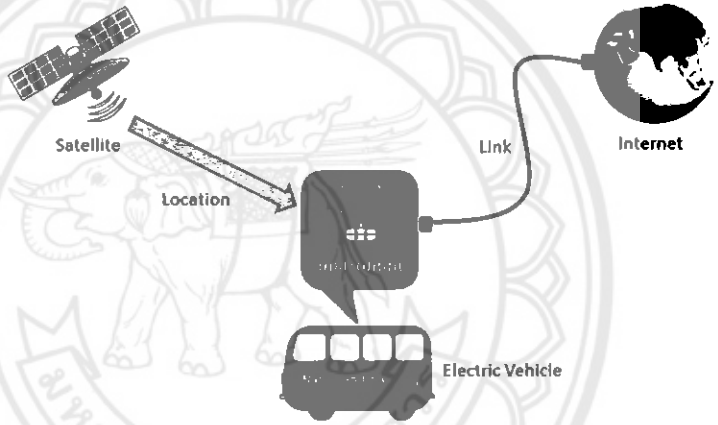
ในบทนี้เป็นการนำหลักการและทฤษฎีที่ได้ศึกษามาใช้ในขั้นตอนของการดำเนินโครงการโดยจะต้องวิเคราะห์ ออกแบบและประยุกต์การใช้งานของระบบ โดยระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G แบ่งการทำงานออกเป็น 3 ส่วน คือ ส่วนโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้าตามหัวข้อ 3.1 ส่วนคำนวณและจัดเก็บฐานข้อมูลตามหัวข้อ 3.2 ส่วนติดต่อกับผู้ใช้ตามหัวข้อ 3.3 โดยโครงสร้างโดยรวมของระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G ดังรูปที่ 3.1



รูปที่ 3.1 โครงสร้างโดยรวมของระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G

3.1 โปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า

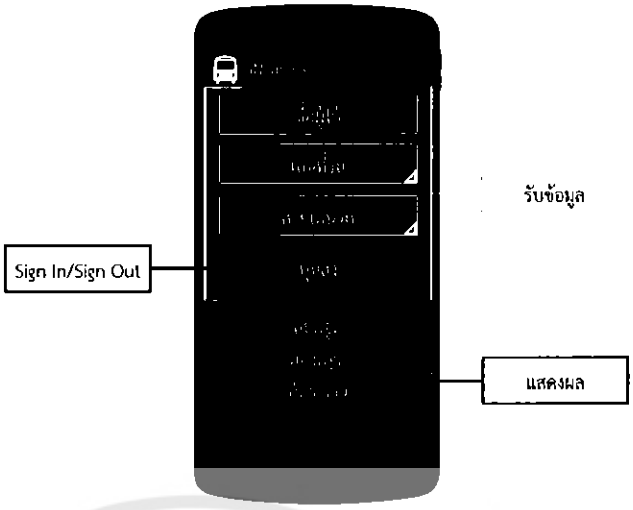
สมาร์ทโฟนเป็นอุปกรณ์ที่มีความสามารถในการระบุตำแหน่งผ่าน GPS โดยระบบปฏิบัติการแอนดรอยด์นั้นถูกออกแบบให้เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อน (Stack) ซึ่งหากต้องการใช้งาน GPS จะมีการเรียกใช้งานผ่าน API ดังนั้นในการทำงานของส่วนโปรแกรมประยุกต์นั้นจึงมีความสะดวกในการเขียนโปรแกรมเพื่อรับค่าจาก GPS ซึ่งสามารถรับค่าได้โดยตรงผ่าน API ได้ดังนี้ อัตราเร็ว, ละติจูดและลองจิจูด และสมาร์ทโฟนยังมีการเชื่อมต่ออินเทอร์เน็ต ดังนั้นข้อมูลที่ได้รับมาจึงจะต้องมีการส่งข้อมูลไปยังเครื่องบริการเว็บเพื่อบันทึกเป็นศูนย์ข้อมูล (Data center) ดังรูปที่ 3.2



รูปที่ 3.2 หลักการทำงานของโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า

3.1.1 การทำงานของโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า

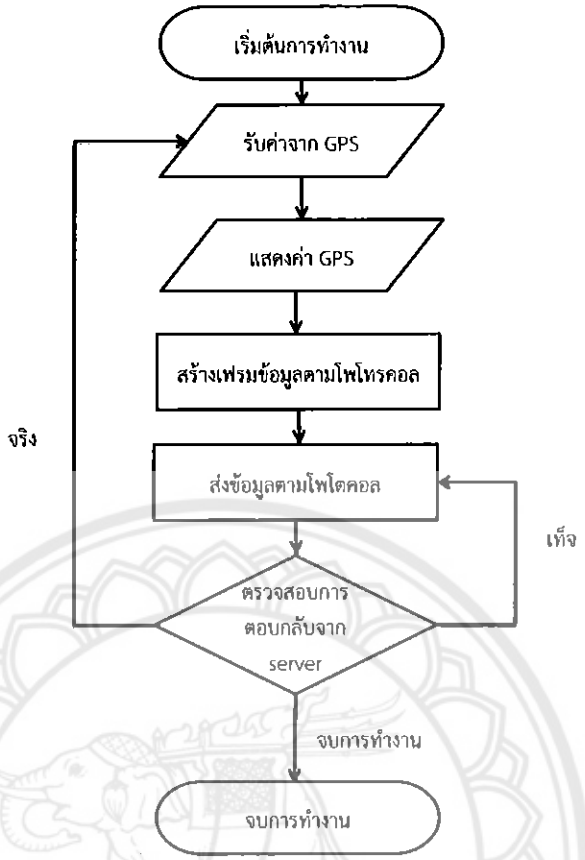
สมาร์ทโฟนที่นำไปติดตั้งบนรถไฟฟ้าของมหาวิทยาลัยต้องมีการติดตั้งโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า โดยแบ่งหน้าจอออกเป็น 2 ส่วน ดังรูปที่ 3.3 คือส่วนในการรับข้อมูล (กรอบสีเหลือง) คือส่วนรับข้อมูลจากผู้ใช้งาน ได้แก่ ชื่อผู้ใช้งาน เลขที่รถ และสายการเดินรถ สำหรับส่วนแสดงผล (กรอบสีเขียว) คือส่วนที่ได้รับมาจากสมาร์ทโฟน ประกอบไปด้วย อัตราเร็วรถ ละติจูดและลองจิจูด หลังจากนั้นสมาร์ทโฟนจะส่งข้อมูลไปยังเครื่องบริการเว็บ (Web Server) ผ่านระบบเครือข่ายอินเทอร์เน็ต ดังรูปที่ 3.2 ส่วนปุ่มส่ง (กรอบสีแดง) คือปุ่มสำหรับให้ผู้ใช้กดปุ่มเพื่อส่งข้อมูลหรือยกเลิกการส่งข้อมูล



รูปที่ 3.3 โปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า (Sender)

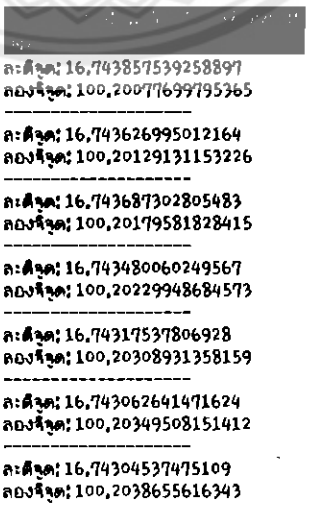
การทำงานของโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้าคือ คนขับรถไฟฟ้าต้องทำการเปิดโปรแกรมประยุกต์โดยให้ระบุหมายเลขรถไฟฟ้า สายเดินรถและชื่อผู้ขับ จากนั้นกดปุ่ม "Sign In" ปุ่มก็จะเปลี่ยนสถานะชื่อปุ่มจาก "Sign In" เป็น "Sign Out" เพื่อส่งข้อมูลไปยังเครื่องบริการเว็บ ข้อมูลที่ส่งประกอบด้วย วันที่ เวลา เลขที่รถ สายการเดินรถ ละติจูด ลองจิจูด อัตราเร็วของรถไฟฟ้า หากเครื่องบริการเว็บได้รับข้อมูลจะตอบกลับข้อความ ACK (Acknowledge) ไปยังสมาร์ทโฟนที่ส่งข้อมูลและแสดงข้อความบนหน้าจอเพื่อให้ทราบสถานะของการส่ง โดยการส่งข้อมูลนั้นจะส่งผ่านโพรโทคอลเอชทีทีพี (HTTP - HyperText Transport Protocol)

จากรูปที่ 3.4 เมื่อเปิดโปรแกรมประยุกต์ขึ้นมา โปรแกรมจะทำการร้องขอค่า GPS จากระบบแล้วนำค่าที่ได้ไปแสดงผลออกทางหน้าจอบนสมาร์ทโฟนและส่งค่าไปยังเครื่องบริการเว็บ หลังจากนั้นเครื่องบริการเว็บจะตรวจสอบข้อมูลที่ส่งมาจากสมาร์ทโฟน แล้วตอบกลับข้อความว่าส่งข้อมูลสำเร็จหรือไม่ หากข้อมูลที่ส่งไปนั้นถูกต้องตามรูปแบบโพรโทคอล สมาร์ทโฟนจะดำเนินการร้องขอค่า GPS ใหม่อีกครั้งและส่งข้อมูลเช่นเดิม แต่ถ้าข้อมูลที่ส่งไปนั้นไม่ถูกต้องตามรูปแบบโพรโทคอล สมาร์ทโฟนจะต้องทำการส่งข้อมูลใหม่อีกครั้ง



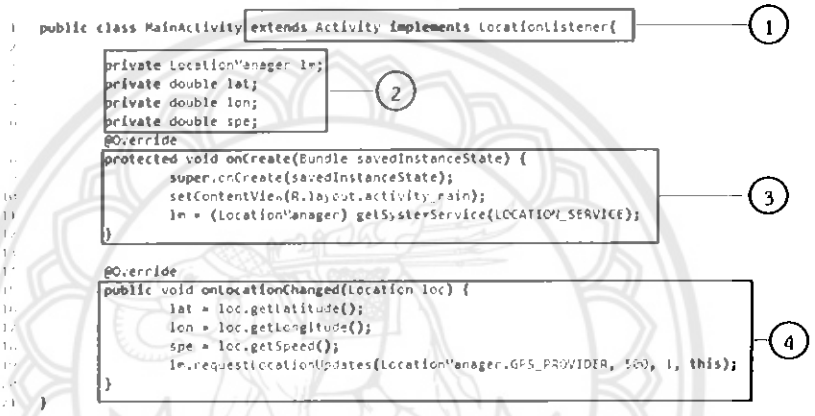
รูปที่ 3.4 แสดงขั้นตอนการทำงานของโปรแกรมประยุกต์ตัวระบุตำแหน่งรถไฟฟ้า

3.1.2 การทำงานของ Location Manager



รูปที่ 3.5 ตัวอย่างการอ่านค่า GPS ของสมาร์ทโฟน

จากรูปที่ 3.5 GPS ทำงานโดยดาวเทียมจะส่งสัญญาณแบบกระจาย (broadcast) ออกมาเป็นค่า GPS ซึ่งตัวรับ GPS (GPS receiver) ที่อยู่ในสมาร์ทโฟน จะหาระยะห่างระหว่างตัวสมาร์ทโฟนกับดาวเทียม 3 หรือ 4 ดวง แล้วนำมาหาตำแหน่งของตัวสมาร์ทโฟนบนพื้นโลก โดยวิธีการทางคณิตศาสตร์ที่เรียกว่า Triangulation นอกจาก GPS แล้วสมาร์ทโฟนยังสามารถบอกตำแหน่งของตนเองผ่านเสาสัญญาณโทรศัพท์เคลื่อนที่ หรือจุดเชื่อมต่ออินเทอร์เน็ต เช่น Wi-Fi (Wi-Fi Access Point) ได้อีกด้วย

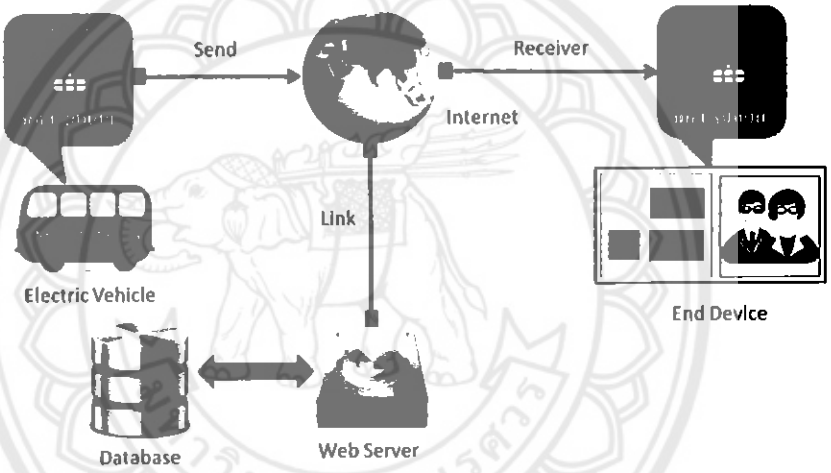


รูปที่ 3.6 รหัสเทียม (pseudo code) การอ่านค่า GPS ของสมาร์ทโฟน

จากรูปที่ 3.6 คือรหัสเทียม (pseudo code) ที่ใช้สำหรับการระบุตำแหน่ง ผ่านโปรแกรมประยุกต์เฟรมเวิร์ค Location Manager เพื่อจัดการกับตำแหน่งปัจจุบันของสมาร์ทโฟน โดยในรหัสเทียมบรรทัดแรก คือการประกาศการเรียกใช้งาน Locationlistener (1) เพื่อใช้งานฟังก์ชันสำหรับการระบุตำแหน่งเช่น การเปลี่ยนแปลงตำแหน่ง เป็นต้น โดยที่จะต้องมีการประกาศตัวแปรเพื่อเก็บข้อมูลต่างๆ (2) โดยโปรแกรมประยุกต์บนระบบปฏิบัติแอนดรอยด์จะเริ่มทำงานด้วยฟังก์ชัน onCreate (3) จึงมีการประกาศตัวแปรไว้ในฟังก์ชันเพื่อรับจาก GPS ดังนั้นหากมีการเปลี่ยนแปลงตำแหน่งบนสมาร์ทโฟนฟังก์ชัน onLocationChanged (4) จะถูกเรียกใช้งานและปรับค่าตัวแปรตามตำแหน่งที่มีการเปลี่ยนแปลง

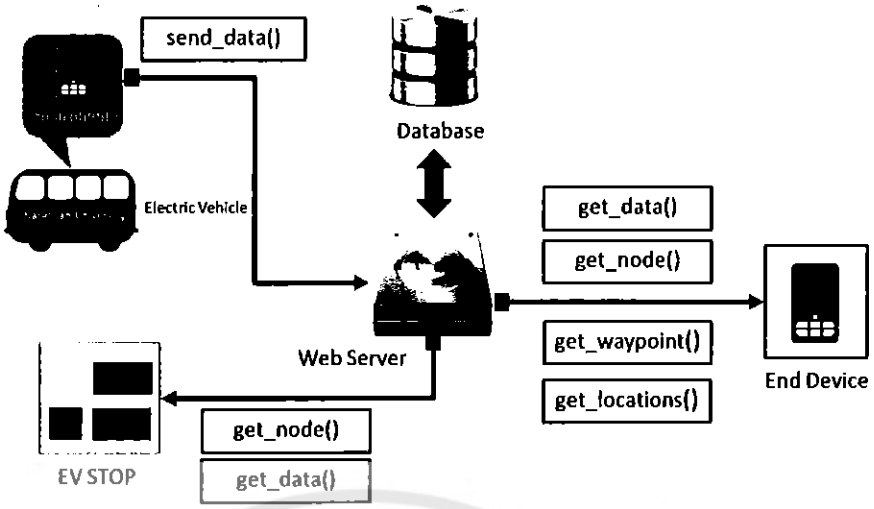
3.2 ส่วนคำนวณและจัดเก็บฐานข้อมูล

ส่วนคำนวณและจัดเก็บฐานข้อมูล จะทำงานด้วยเครื่องบริการเว็บ (Web Server) ที่มีการให้บริการอยู่ตลอดเวลาผ่านอินเทอร์เน็ต โดยแบ่งการทำงานออกเป็น 2 หน้าที่หลักคือ หน้าที่แรก คำนวณระยะทางเพื่อหาเวลาจากละติจูด (Latitude) และลองจิจูด (Longitude) ที่ได้รับมาจากโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า และอีกหน้าที่ จัดเก็บข้อมูลที่มีการรับ/ส่ง ลงในฐานข้อมูล (Database) เพื่อให้ผู้ใช้ (End Device) สามารถร้องขอข้อมูลในการแสดงผลได้ตลอดเวลา ซึ่งหลักการการทำงานของส่วนคำนวณและจัดเก็บฐานข้อมูล ดังรูปที่ 3.7



รูปที่ 3.7 โครงสร้างของส่วนคำนวณและจัดเก็บฐานข้อมูล

การติดต่อสื่อสารระหว่างผู้ใช้กับเครื่องบริการเว็บ เชื่อมต่อโพรโทคอล HTTP ผ่านอินเทอร์เน็ต โดยเรียกใช้งาน API ตามหน้าที่ของผู้ใช้ เช่น โปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้าส่งข้อมูลไปให้เครื่องบริการเว็บ (Send Data API) และผู้ใช้สามารถเลือกที่รับข้อมูลประมาณเวลาของการเดินรถไฟฟ้า (Get Data API), ข้อมูลป้ายจอดรถ (Get Node API), ข้อมูลตำแหน่งรถไฟฟ้าปัจจุบัน (Get Locations API) และข้อมูลตำแหน่งจุดอ้างอิง (Get Waypoint API), จากเครื่องบริการเว็บ ดังรูปที่ 3.8 โดยรายละเอียดของ API จะกล่าวในหัวข้อ 3.2.2



รูปที่ 3.8 การติดต่อสื่อสารระหว่างผู้ใช้กับเครื่องบริการเว็บผ่าน API

3.2.1 ส่วนคำนวณระยะทางและเวลา

การคำนวณหาระยะทางและเวลานั้นสามารถคำนวณได้จากระยะทางระหว่างตำแหน่งจุดอ้างอิง (Way Point) โดยจะต้องมีการกำหนดตำแหน่งจุดอ้างอิงและคำนวณหาระยะทางระหว่างจุดอ้างอิงด้วยสูตร Haversine formula ดังสมการที่ 3.1

$$d = 2r \arcsin\left(\sqrt{sin^2\left(\frac{\phi 2-\phi 1}{2}\right) + \cos(\phi 1)\cos(\phi 2)sin^2\left(\frac{\lambda 2-\lambda 1}{2}\right)}\right) \qquad (3.1)$$

โดยที่แปลง Haversine formula เป็นรหัสเทียม (pseudo code) เพื่อให้คอมพิวเตอร์สามารถคำนวณได้ง่ายและสามารถนำไปใช้งานได้ทันทีที่มีการเรียกใช้งาน โดยรหัสเทียมมีการอ้างอิงรูปแบบภาษาจาวาสคริปต์ (Javascript) เนื่องจากไม่จำเป็นต้องมีคอมไพเลอร์ และสามารถรันบนเว็บเบราว์เซอร์ของผู้ใช้ได้ทันทีที่มีการร้องขอบริการจากเครื่องบริการเว็บ ดังรูปที่ 3.9

```
function Haversine(lat1, lon1, lat2, lon2) {
    var r = 6371;
    lat1 = lat1 * (Math.PI / 180);
    lon1 = lon1 * (Math.PI / 180);
    lat2 = lat2 * (Math.PI / 180);
    lon2 = lon2 * (Math.PI / 180);

    var dlat = lat2 - lat1;
    var dlon = lon2 - lon1;

    var a = Math.pow(Math.sin(dlat / 2), 2) + Math.cos(lat1) * Math.cos(lat2) * Math.pow(Math.sin(dlon / 2), 2);
    var c = Math.asin(Math.sqrt(a));
    var d = 2 * r * c;
    return d;
}
```

รูปที่ 3.9 รหัสเทียมสำหรับคำนวณระยะทางระหว่างจุด 2 จุด

การคำนวณหาระยะทางระหว่างจุดอ้างอิง จะต้องทำการคำนวณจุดอ้างอิงทีละตำแหน่งตามลำดับจนถึงตำแหน่งสุดท้าย เพื่อนำระยะทางระหว่างจุดอ้างอิงไปคำนวณหาระยะทางระหว่างรถไฟฟ้ากับป้ายจอดรถ โดยข้อมูลของตำแหน่งจุดอ้างอิงจะประกอบไปด้วย ลำดับที่ตำแหน่งจุดอ้างอิง ละติจูด ลองจิจูด และระยะทางระหว่างจุด ดังตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างข้อมูลตำแหน่งจุดอ้างอิง

ลำดับที่ตำแหน่งจุดอ้างอิง (i)	ละติจูด (Latitude)	ลองจิจูด (Longitude)	ระยะทางระหว่างจุด distance(จุดที่ i, จุดที่ i+1) (กิโลเมตร)
1	16.7365340	100.1983940	0.106
2	16.7371250	100.1991740	0.105
3	16.7380140	100.1995020	0.108
4	16.7389800	100.1994100	0.105
5	16.7399100	100.1992630	0.106
6	16.7408520	100.1991070	0.109
7	16.7417950	100.1988310	

จากตารางที่ 3.1 การคำนวณระยะทางระหว่างจุด พบว่าไม่สามารถคำนวณระยะทางสำหรับลำดับสุดท้าย (ตำแหน่งที่ 7) ได้เนื่องจากตำแหน่งที่ต้องการคำนวณมีเพียงจุดเดียว ดังนั้นจึงสงวนไว้

สำหรับเพื่อตั้งใหม่ (reset) ให้กับตำแหน่งย้อนกลับไปที่ตำแหน่งที่ 1 ซึ่งสามารถคำนวณได้จาก รหัสเทียมสำหรับคำนวณระยะทางระหว่างจุด 2 จุด ดังรูปที่ 3.9 โดยมีการคำนวณดังนี้

ระยะทาง = distance(Latitude1, Longitude1, Latitude2, Longitude2)

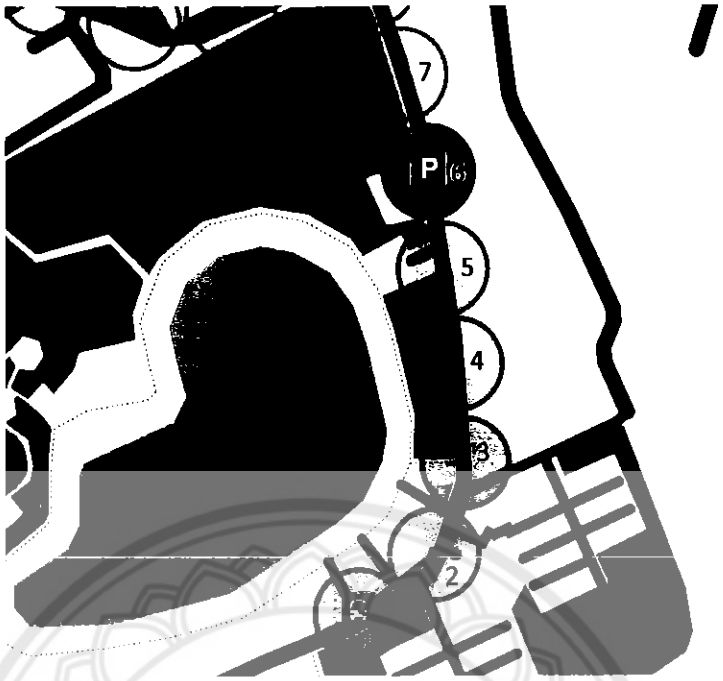
แทนค่า ละติจูด ลองจิจูดของจุดที่ 1 และ 2 ตามลำดับ จะได้

ระยะทาง = distance (16.7365340,100.1983940,16.7371250,100.1991740)

ระยะทาง = 0.106 กิโลเมตร

รูปที่ 3.10 ตำแหน่งจุดอ้างอิงโดยมีการกำหนดรัศมี 50 เมตร

จากรูปที่ 3.10 แสดงให้เห็นว่าตำแหน่งจุดอ้างอิงมีการกำหนดรัศมี 50 เมตร เพื่อตรวจสอบว่ามีรถไฟฟ้าอยู่ในบริเวณภายในรัศมีหรือไม่ หากรถไฟฟ้าอยู่ภายในรัศมีแสดงว่ารถไฟฟ้าอยู่บนเส้นทางที่ถูกกำหนดไว้ตามตำแหน่งจุดอ้างอิง โดยมีการวาดเส้นทางจากตำแหน่งจุดอ้างอิงเพื่อแสดงเส้นทางดังรูปที่ 3.11



รูปที่ 3.11 แสดงเส้นทางจากการวาดเส้นตรงของตำแหน่งแต่ละจุดอ้างอิง

จากรูปที่ 3.11 การคำนวณหาระยะทางระหว่างรถไฟฟ้า (ตำแหน่งที่ 1) กับป้ายจอดรถ (ตำแหน่งที่ 6) สามารถคำนวณได้จากการนำตำแหน่งปัจจุบันของรถไฟฟ้าว่าอยู่ลำดับใดของตำแหน่งจุดอ้างอิง (รถไฟฟ้าอยู่ภายในตำแหน่งที่ 1) จากนั้นทำการรวมระยะทางตั้งแต่ตำแหน่งจุดอ้างอิงของรถไฟฟ้า (ตำแหน่งที่ 1) จนถึงป้ายจอดรถ (ตำแหน่งที่ 6) ซึ่งจะได้ผลรวมของระยะทางทั้งหมดระหว่างรถไฟฟ้าและป้ายจอดรถจาก ($\sum_{i=1}^{6} distance[i]$) โดยระยะทางของแต่ละตำแหน่งจุดอ้างอิงมาจาก ตารางที่ 3.1 เนื่องจากผลลัพธ์ที่ต้องการคือเวลา ดังนั้นสามารถคำนวณได้จากสูตร ดังสมการที่ 3.2

$$t = \frac{s}{v} \tag{3.2}$$

$$t = \frac{\sum_{i=m}^{n} distance[i]}{v} \tag{3.3}$$

โดยที่ t คือ เวลาที่รถไฟฟ้าวิ่งถึงป้ายจอดรถ (วินาที)

s คือ ระยะทางระหว่างรถไฟฟ้ากับป้ายจอดรถ (เมตร)

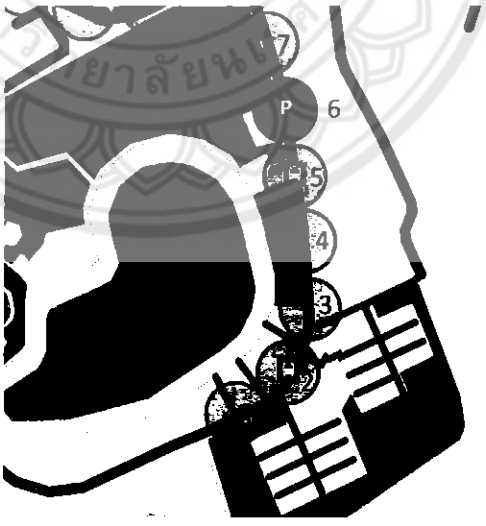
v คือ อัตราเร็วของรถไฟฟ้า (เมตรต่อวินาที)

m คือ ตำแหน่งจุดอ้างอิงของรถไฟฟ้า

n คือ ตำแหน่งจุดอ้างอิงของป้ายจอดรถไฟฟ้า

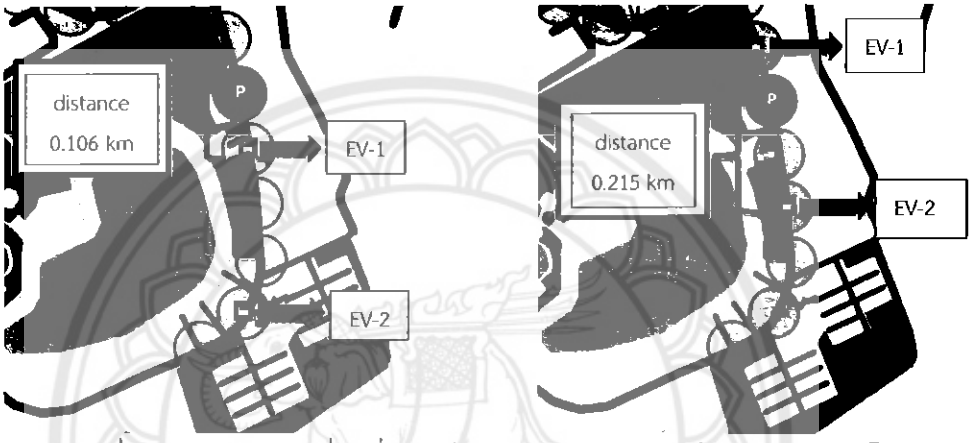
การคำนวณหาระยะทางระหว่างรถไฟฟ้าและป้ายจอดรถ สามารถคำนวณในกรณีที่มีป้ายจอดรถมากกว่า 1 ป้าย เนื่องจากตำแหน่งรถไฟฟ้าและตำแหน่งป้ายจอดรถไฟฟ้าอยู่ภายในรัศมีของตำแหน่งจุดอ้างอิงซึ่งสามารถคำนวณได้ จากสมการที่ 3.3

กรณีที่รถไฟฟ้ามีมากกว่า 1 คัน จะต้องมีการตรวจสอบว่ามีรถไฟฟ้าคันใดอยู่บนตำแหน่งจุดอ้างอิงที่ใกล้ที่สุดกับป้ายจอดรถจึงจะทำการคำนวณหาระยะทาง หากรถไฟฟ้าวิ่งผ่านป้ายจอดรถแล้ว จะต้องทำการตรวจสอบว่ารถไฟฟ้าที่กำลังจะมาถึงป้ายจอดรถนั้นเช่นเดียวกับการตรวจสอบครั้งแรก เพื่อหาตำแหน่งที่ใกล้ที่สุด ดังรูปที่ 3.12



รูปที่ 3.12 แสดงการเดินรถไฟฟ้าในกรณีที่รถไฟฟ้ามีมากกว่า 1 คัน

จากรูปที่ 3.12 รถไฟฟ้าคันที่ 1 (ตำแหน่งที่ 5) รถไฟฟ้าคันที่ 2 (ตำแหน่งที่ 2) และป้ายจอดรถ (ตำแหน่งที่ 6) จะพบว่าตำแหน่งรถไฟฟ้าคันที่ 1 จะมีระยะทางใกล้ป้ายจอดรถมากกว่ารถไฟฟ้าคันที่ 2 ดังนั้นจึงมีการคำนวณระยะทางระหว่างรถไฟฟ้าคันที่ 1 กับป้ายจอดรถ หลังจากรถไฟฟ้าคันที่ 1 วิ่งผ่านป้ายจอดรถ จะต้องมีการคำนวณระยะทางระหว่างรถไฟฟ้าคันที่ 2 กับป้ายจอดรถ ดังรูปที่ 3.13



รูปที่ 3.13 แสดงการเคลื่อนที่ของรถไฟฟ้า โดยป้ายจอดรถเลือกคำนวณเวลาโดยประมาณ

3.2.2 จัดการเกี่ยวกับฐานข้อมูล

เครื่องบริการเว็บ จะทำหน้าที่จัดการเกี่ยวกับฐานข้อมูล 2 ส่วน คือส่วนแรก เครื่องบริการเว็บจะรับข้อมูลจากสมาร์ทโฟนที่ระบุตำแหน่งปัจจุบันของรถไฟฟ้า เพื่อบันทึกตำแหน่งปัจจุบันของรถไฟฟ้าและรายละเอียดต่างๆ ลงในฐานข้อมูล โดยนำข้อมูลต่างๆ ไปใช้สำหรับเป็นศูนย์ข้อมูลในการติดต่อสื่อสารผ่านเอพีไอ (API) โดยโครงสร้างของฐานข้อมูล ดังรูปที่ 3.14 ซึ่งประกอบไปด้วย 7 ตารางดังต่อไปนี้

3.2.2.1 ตาราง data สำหรับเก็บข้อมูลของเวลาและระยะทางแต่ละป้ายจอดรถ

ประกอบด้วย 5 คอลัมน์

คอลัมน์ที่ 1 id_node สำหรับการเชื่อมโยงรหัสของป้ายจอดรถกับตาราง node

คอลัมน์ที่ 2 distance สำหรับเก็บระยะทางระหว่างรถไฟฟ้าและป้ายจอดรถ

คอลัมน์ที่ 3 date_now สำหรับเก็บวันที่ของข้อมูล

คอลัมน์ที่ 4 time สำหรับเก็บเวลาระหว่างรถไฟฟ้าและป้ายจอดรถ

คอลัมน์ที่ 5 time_now สำหรับเก็บเวลาของข้อมูล

3.2.2.2 ตาราง node สำหรับเก็บข้อมูลรายชื่อป้ายจอดรถทั้งหมด

ประกอบด้วย 5 คอลัมน์

คอลัมน์ที่ 1 id_node สำหรับการเชื่อมโยงรหัสของป้ายจอดรถกับตาราง data และเป็นคอลัมน์ที่เป็นคีย์หลัก

คอลัมน์ที่ 2 id_waypoint สำหรับการเชื่อมโยงรหัสของตำแหน่งจุดอ้างอิงกับตาราง waypoint

คอลัมน์ที่ 3 name_node สำหรับเก็บข้อมูลชื่อของป้ายจอดรถ

คอลัมน์ที่ 4 latitude_node สำหรับเก็บข้อมูลละติจูดของป้ายจอดรถ

คอลัมน์ที่ 5 longitude_node สำหรับเก็บข้อมูลลองจิจูดของป้ายจอดรถ

3.2.2.3 ตาราง waypoint สำหรับเก็บข้อมูลตำแหน่งจุดอ้างอิง

ประกอบด้วย 3 คอลัมน์

คอลัมน์ที่ 1 id_waypoint สำหรับการเชื่อมโยงรหัสของตำแหน่งจุดอ้างอิงกับตารางอื่นๆ และเป็นคอลัมน์ที่เป็นคีย์หลัก

คอลัมน์ที่ 2 latitude_waypoint สำหรับเก็บข้อมูลละติจูดของตำแหน่งจุดอ้างอิง

คอลัมน์ที่ 3 longitude_waypoint สำหรับเก็บข้อมูลลองจิจูดของตำแหน่งจุดอ้างอิง

3.2.2.4 ตาราง dis_waypoint สำหรับเก็บข้อมูลระยะทางระหว่างตำแหน่งจุดอ้างอิง

ประกอบด้วย 2 คอลัมน์

คอลัมน์ที่ 1 id_waypoint สำหรับการเชื่อมโยงรหัสของตำแหน่งจุดอ้างอิงกับตาราง waypoint

คอลัมน์ที่ 2 distance_waypoint สำหรับเก็บข้อมูลระยะทางระหว่างตำแหน่งจุดอ้างอิง

3.2.2.5 ตาราง locationsnow สำหรับเก็บข้อมูลรถไฟฟ้าที่เดินรถอยู่ ณ เวลาปัจจุบัน

ประกอบด้วย 8 คอลัมน์

คอลัมน์ที่ 1 date สำหรับเก็บวันที่ของข้อมูล

คอลัมน์ที่ 2 time สำหรับเก็บเวลาของข้อมูล

คอลัมน์ที่ 3 no สำหรับเก็บลำดับที่ของรถไฟฟ้า

คอลัมน์ที่ 4 line สำหรับเก็บสายเดินรถของรถไฟฟ้า

คอลัมน์ที่ 5 latitude สำหรับเก็บข้อมูลละติจูดของรถไฟฟ้า

คอลัมน์ที่ 6 longitude สำหรับเก็บข้อมูลลองจิจูดของรถไฟฟ้า

คอลัมน์ที่ 7 description สำหรับเก็บข้อมูลรายละเอียดต่างๆ ของรถไฟฟ้า

คอลัมน์ที่ 8 level สำหรับเก็บข้อมูลระดับตำแหน่งจุดอ้างอิงของรถไฟฟ้า

3.2.2.6 ตาราง users สำหรับเก็บข้อมูลของสมาชิกบนเว็บไซต์

ประกอบด้วย 5 คอลัมน์

คอลัมน์ที่ 1 id สำหรับการเชื่อมโยงรหัสของผู้ใช้ และเป็นคอลัมน์ที่เป็นคีย์หลัก

คอลัมน์ที่ 2 username สำหรับเก็บชื่อผู้ใช้งาน

คอลัมน์ที่ 3 password สำหรับเก็บรหัสผ่านของผู้ใช้งานโดยมีการเข้ารหัส

คอลัมน์ที่ 4 salt สำหรับเก็บข้อมูลที่ใช้ในการเพิ่มความปลอดภัยของรหัสผ่าน

คอลัมน์ที่ 5 email สำหรับเก็บข้อมูลอีเมล์ของผู้ใช้งาน

3.2.2.7 ตาราง locations สำหรับเก็บข้อมูลรถไฟฟ้าที่เดินรถย้อนหลัง

ประกอบด้วย 8 คอลัมน์

คอลัมน์ที่ 1 date สำหรับเก็บวันที่ของข้อมูล

คอลัมน์ที่ 2 time สำหรับเก็บเวลาของข้อมูล

คอลัมน์ที่ 3 no สำหรับเก็บลำดับที่ของรถไฟฟ้า

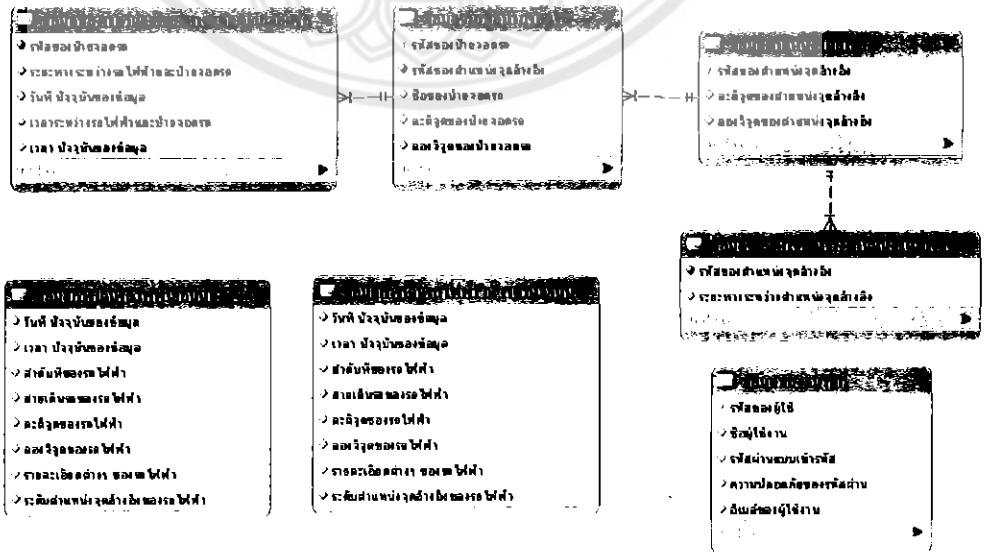
คอลัมน์ที่ 4 line สำหรับเก็บสายเดินรถของรถไฟฟ้า

คอลัมน์ที่ 5 latitude สำหรับเก็บข้อมูลละติจูดของรถไฟฟ้า

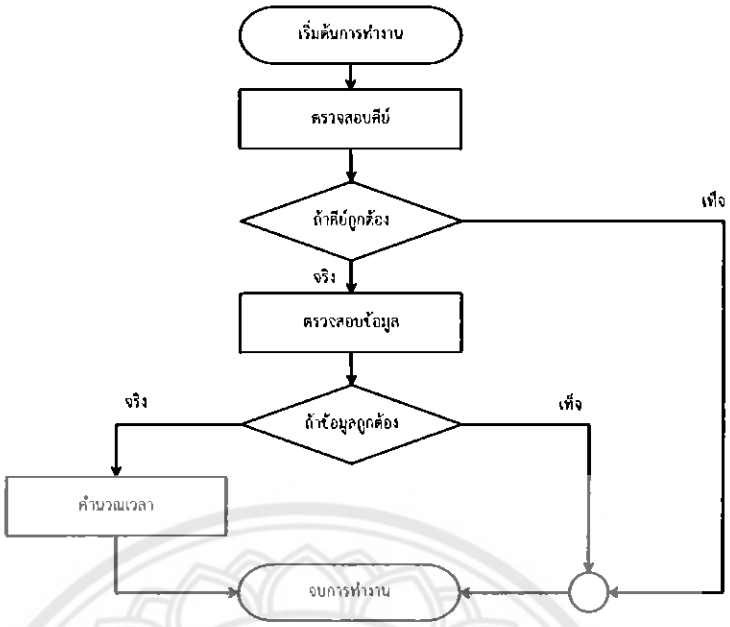
คอลัมน์ที่ 6 longitude สำหรับเก็บข้อมูลลองจิจูดของรถไฟฟ้า

คอลัมน์ที่ 7 description สำหรับเก็บข้อมูลรายละเอียดต่างๆ ของรถไฟฟ้า

คอลัมน์ที่ 8 level สำหรับเก็บข้อมูลระดับตำแหน่งจุดอ้างอิงของรถไฟฟ้า

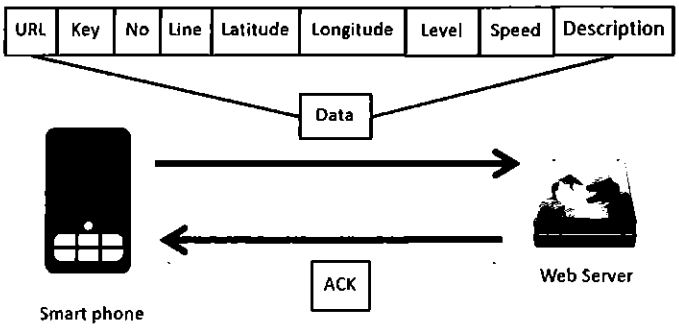


รูปที่ 3.14 โครงสร้างของฐานข้อมูล



รูปที่ 3.15 แสดงขั้นตอนการทำงานของเครื่องบริการบน API ส่งข้อมูล

จากรูปที่ 3.15 เอพีไอในการส่งข้อมูล (Send) ผู้ใช้สามารถส่งข้อมูลต่างๆ ไปยังเครื่องบริการเว็บได้ เช่น ข้อมูลตำแหน่งปัจจุบัน ข้อมูลอัตราเร็ว ข้อมูลระดับตำแหน่งจุดอ้างอิง และข้อมูลรายละเอียดของรถไฟฟ้า ซึ่งในการส่งจะต้องใส่คีย์ของผู้ใช้ ดังนั้นหากคีย์ไม่ถูกต้องเครื่องบริการเว็บจะจบการทำงาน แต่ถ้าคีย์ถูกต้องเครื่องบริการเว็บจะตรวจสอบข้อมูลที่ได้รับมา โดยจะทำการคำนวณหาเวลาโดยประมาณก่อนจบการทำงาน ในการข้อความตอบกลับผู้ใช้จะได้รับในรูปแบบ JSON - JavaScript Object Notation เนื่องจากมนุษย์และโปรแกรมประยุกต์สามารถอ่านเข้าใจได้โดยหลักการทำงานในการส่งข้อมูลพร้อมทั้งรับข้อมูลตอบกลับ ดังรูปที่ 3.16



รูปที่ 3.16 ตัวอย่างการส่งข้อมูลพร้อมได้รับข้อมูลตอบกลับ

เอพีไอในส่งข้อมูล (Send Data API)

สำหรับส่งข้อมูลไปยังเครื่องบริการเว็บ จะประกอบไปด้วย ลำดับที่ สายเดินรถ ตำแหน่งละติจูด ลองจิจูด รายละเอียดต่างๆ และระดับตำแหน่งจุดอ้างอิงของรถไฟฟ้า ตามรูปแบบโพรโทคอลในตารางที่ 3.2 หากมีการส่งข้อมูลเสร็จสิ้นจะมีข้อมูลตอบกลับ ดังรูปที่ 3.17 ซึ่งคำอธิบายสำหรับข้อมูลตอบกลับอธิบายไว้ ดังตารางที่ 3.3

การร้องขอ

ตารางที่ 3.2 รูปแบบโพรโทคอลในการส่งข้อมูลไปยังเครื่องบริการเว็บ

	http://inear3g.embeddednu.com/apis/send_data/
	GET
	• Key • No • Line • Latitude • Longitude • Level • Speed • Description
	JSON

ข้อมูลตอบกลับ

```
{
        "code":0,
        "msg":"Send data - OK"
}
```

รูปที่ 3.17 ข้อมูลตอบกลับในกรณีที่ส่งข้อมูลถูกต้อง โดยแสดงในรูปแบบ JSON

จากรูปที่ 3.17 จะพบว่าในส่วนของ “code” คือ 0 หมายถึง ข้อมูลที่ถูกส่งไปมีความถูกต้องและจัดเก็บลงฐานข้อมูลเรียบร้อย โดยแสดงข้อความตอบกลับในส่วนของ “msg” คือ Send data – OK หากผู้ใช้ส่งข้อมูลที่ไม่ครบหรือผิดพลาด ในข้อความตอบกลับจะแสดงในส่วนของ “code” คือ -1 ถึง -9 หมายถึง ข้อมูลที่ถูกส่งไปมีความผิดพลาดและจะไม่ถูกจัดเก็บลงฐานข้อมูล ซึ่งจะแสดงข้อความที่ผิดพลาดในส่วนของ “msg” คือ No description parameter หมายถึง ผู้ใช้ลืมส่งคำอธิบายของผู้ใช้มาด้วยตอนส่งข้อมูล ดังรูปที่ 3.18 โดยสามารถดูคำอธิบายได้จากตารางที่ 3.3

```
{
    "code": -9,
    "msg": "No description parameter"
}
```

รูปที่ 3.18 ข้อมูลตอบกลับในกรณีที่ส่งข้อมูลไม่ครบ โดยแสดงในรูปแบบ JSON

ตารางที่ 3.3 คำอธิบายข้อมูลตอบกลับในการส่งข้อมูลไปยังเครื่องบริการเว็บ

ชื่อ	คำอธิบาย
code	0 หมายถึง success -1 หมายถึง ไม่มีการเรียก API -2 หมายถึง คีย์ไม่ถูกต้อง -3 หมายถึง No ไม่มีข้อมูล -4 หมายถึง Lina ไม่มีข้อมูล -5 หมายถึง Latitude ไม่มีข้อมูล -6 หมายถึง Longitude ไม่มีข้อมูล -7 หมายถึง Level ไม่มีข้อมูล -8 หมายถึง Speed ไม่มีข้อมูล -9 หมายถึง Description ไม่มีข้อมูล
msg	ข้อความอธิบาย code

ตัวอย่างการใช้งาน

- http://inear3g.embeddednu.com/apis/send_data/evnu2013/1/RED/16.7365340/100.1983940/0/10/Test

  Key = evnu2013

  No = 1

  Line = RED

  Latitude = 16.7365340

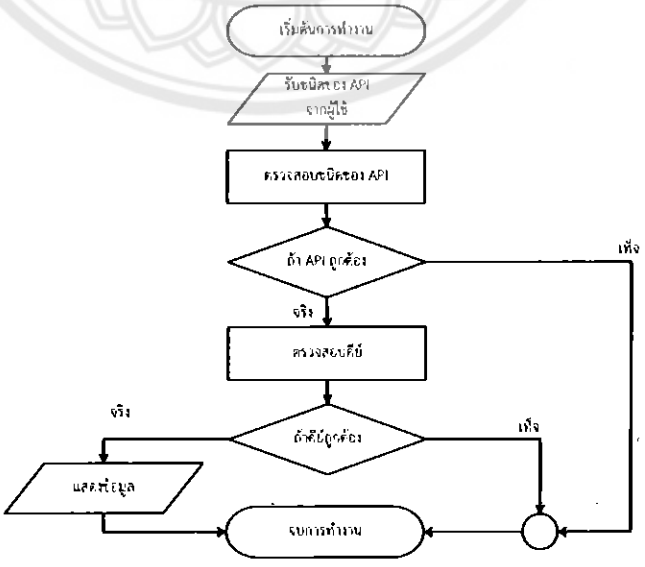
  Longitude = 100.1983940

  Level = 0

  Speed = 10

  Description = Test

ส่วนที่สองเครื่องบริการเว็บจะเปิดเอพีไอสำหรับส่งข้อมูลต่างๆ ไปยังสมาร์ทโฟนที่มีการร้องขอข้อมูล เพื่อให้ผู้ใช้สามารถนำข้อมูลไปพัฒนาในส่วนโปรแกรมประยุกต์บนสมาร์ทโฟน โดยจะต้องมีคีย์สำหรับเปิดใช้งาน ซึ่งมีการตั้งค่าไว้ในสมาร์ทโฟนเท่านั้นเพื่อป้องกันความปลอดภัยของข้อมูล ดังนั้นเฉพาะผู้ที่คีย์เท่านั้นถึงจะสามารถเข้าถึงข้อมูลได้



รูปที่ 3.19 แสดงขั้นตอนการทำงานของเครื่องบริการบน API ขอข้อมูล

จากรูป 3.19 เอพีไอในการขอข้อมูล (Get) ผู้ใช้สามารถร้องขอข้อมูลต่างๆ ของเครื่องบริการเว็บได้ เช่น ข้อมูลของป้ายจอดรถ ข้อมูลตำแหน่งปัจจุบัน ข้อมูลตำแหน่งจุดอ้างอิงและข้อมูลเวลากับระยะทาง ซึ่งในแต่ละครั้งจะต้องมีการใส่คีย์ของผู้ใช้ ผลลัพธ์การตอบกลับที่ได้จะแสดงในรูปแบบ JSON (กรณีที่ต้องการร้องขอข้อมูล) ดังนั้นหากไม่มีคีย์ของผู้ใช้ จะไม่สามารถร้องขอข้อมูลได้ทุกฟังก์ชันซึ่งเครื่องบริการเว็บจะแสดงข้อความผิดพลาดในรูปแบบ JSON โดยหน้าที่และรายละเอียดของแต่ละฟังก์ชัน ดังนี้

เอพีไอขอรายชื่อป้ายจอดรถ (Get Node API)

สำหรับขอรายชื่อป้ายจอดรถทั้งหมด ที่มีอยู่ในเครื่องบริการเว็บ ใช้สำหรับให้โปรแกรมประยุกต์สามารถแสดงรายชื่อป้ายจอดรถทั้งหมดที่มีอยู่ในเครื่องบริการเว็บ โดยทันทีที่ผู้ใช้ทำการเปลี่ยนแปลง เพิ่มหรือลบ ป้ายจอดรถในเครื่องบริการเว็บ ก็ไม่จำเป็นต้องทำการแก้ไขการทำงานที่โปรแกรมประยุกต์ ตามรูปแบบโพรโทคอลในตารางที่ 3.4 หากมีการร้องขอข้อมูลเสร็จสิ้นจะมีข้อมูลตอบกลับ ดังรูปที่ 3.20 ซึ่งคำอธิบายสำหรับข้อมูลตอบกลับอธิบายไว้ ดังตารางที่ 3.5

การร้องขอ

ตารางที่ 3.4 รูปแบบโพรโทคอลในการขอรายชื่อป้ายจอดรถจากเครื่องบริการเว็บ

URL	http://inear3g.embeddednu.com/apis/get_node/
Method	GET
Parameter	• Key O คีย์ของผู้ใช้
Response format	JSON

ข้อมูลตอบกลับ


```
{
    "code": 0,
    "msg": "Get node - OK",
    "node_list": [{
        "id": 3,
        "id_waypoint": 29,
        "name": "Faculty of Medicine",
        "lat": "16.7451670",
        "lon": "100.1907250"
    }, {
        "id": 2,
        "id_waypoint": 11,
        "name": "Faculty of Engineering",
        "lat": "16.7469960",
        "lon": "100.1964820"
    }, {
        "id": 1,
        "id_waypoint": 6,
        "name": "Bus stop",
        "lat": "16.7408520",
        "lon": "100.1991070"
    }]
}
```


รูปที่ 3.20 ข้อมูลตอบกลับในการขอรายชื่อป้ายจอดรถจากเครื่องบริการเว็บ รูปแบบ JSON

ตารางที่ 3.5 คำอธิบายข้อมูลตอบกลับในการขอรายชื่อป้ายจอดรถจากเครื่องบริการเว็บ

ชื่อ	คำอธิบาย
code	0 หมายถึง success -1 หมายถึง ไม่มีการเลือก API -2 หมายถึง คีย์ไม่ถูกต้อง
msg	ข้อความอธิบาย code
node_list	รายชื่ออุปกรณ์
Id	หมายเลขป้ายจอดรถ
Id_waypoint	หมายเลขตำแหน่งจุดอ้างอิง
name	ชื่อป้ายจอดรถ
lat	ตำแหน่ง latitude ของป้ายจอดรถ
lon	ตำแหน่ง longitude ของป้ายจอดรถ

ตัวอย่างการใช้งาน

http://inear3g.embeddednu.com/apis/get_node/evnu2013

- key = evnu2013

เอพีไอขอรายชื่อตำแหน่งจุดอ้างอิง (Get Waypoint API)

สำหรับขอรายชื่อตำแหน่งจุดอ้างอิงทั้งหมด โดยตำแหน่งจุดอ้างอิงนั้นคือ การวางตำแหน่งหลายจุดต่อเนื่องกันให้เป็นเส้นทางในการเดินรถไฟฟ้า เพื่อให้รถไฟฟ้าเดินรถตามเส้นทางที่กำหนดไว้ ตามรูปแบบโพรโทคอลในตารางที่ 3.6 หากมีการร้องขอข้อมูลเสร็จสิ้นจะมีข้อมูลตอบกลับ ดังรูปที่ 3.21 ซึ่งคำอธิบายสำหรับข้อมูลตอบกลับอธิบายไว้ ดังตารางที่ 3.7

การร้องขอ

ตารางที่ 3.6 รูปแบบโพรโทคอลในการขอรายชื่อตำแหน่งจุดอ้างอิงจากเครื่องบริการเว็บ

	http://inear3g.embeddednu.com/apis/get_waypoint/
	GET
	• Key ○ คีย์ของผู้ใช้
	JSON

ข้อมูลตอบกลับ


```
{
    "code": 0,
    "msg": "Get way point - OK",
    "waypoint_list":
    [
      { "id_waypoint": "1",  "lat": "16.7365340", "lon": "100.1983940" },
      { "id_waypoint": "2",  "lat": "16.7371250", "lon": "100.1991740" },
      { "id_waypoint": "3",  "lat": "16.7380140", "lon": "100.1995020" },
      { "id_waypoint": "4",  "lat": "16.7389800", "lon": "100.1994100" },
      { "id_waypoint": "5",  "lat": "16.7399100", "lon": "100.1992630" },
      { "id_waypoint": "6",  "lat": "16.7408520", "lon": "100.1991070" },
      { "id_waypoint": "7",  "lat": "16.7417950", "lon": "100.1988310" },
      { "id_waypoint": "8",  "lat": "16.7427140", "lon": "100.1985200" },
      { "id_waypoint": "9",  "lat": "16.7435440", "lon": "100.1980430" },
      { "id_waypoint": "10", "lat": "16.7443840", "lon": "100.1976210" },
      { "id_waypoint": "11", "lat": "16.7453390", "lon": "100.1974260" },
      { "id_waypoint": "12", "lat": "16.7461640", "lon": "100.1969670" },
      { "id_waypoint": "13", "lat": "16.7469960", "lon": "100.1964820" },
      { "id_waypoint": "14", "lat": "16.7478410", "lon": "100.1959990" },
      { "id_waypoint": "15", "lat": "16.7486830", "lon": "100.1955160" },
      { "id_waypoint": "16", "lat": "16.7494980", "lon": "100.1949470" },
      { "id_waypoint": "17", "lat": "16.7499030", "lon": "100.1940670" },
      { "id_waypoint": "18", "lat": "16.7497750", "lon": "100.1930590" },
      { "id_waypoint": "19", "lat": "16.7497290", "lon": "100.1920930" },
      { "id_waypoint": "20", "lat": "16.7500160", "lon": "100.1911700" },
      { "id_waypoint": "21", "lat": "16.7504940", "lon": "100.1903170" },
    ]
}
```


รูปที่ 3.21 ข้อมูลตอบกลับในการขอรายชื่อตำแหน่งจุดอ้างอิงจากเครื่องบริการเว็บ รูปแบบ JSON

ตารางที่ 3.7 คำอธิบายข้อมูลตอบกลับในการขอรายชื่อตำแหน่งจุดอ้างอิงจากเครื่องบริการเว็บ

ชื่อ	คำอธิบาย
code	0 หมายถึง success -1 หมายถึง ไม่มีการเลือก API -2 หมายถึง คีย์ไม่ถูกต้อง
msg	ข้อความอธิบาย code
waypoint_list	รายชื่อตำแหน่งจุดอ้างอิง
id_ waypoint	หมายเลขตำแหน่งจุดอ้างอิง
lat	ตำแหน่ง latitude ของตำแหน่งจุดอ้างอิง
lon	ตำแหน่ง longitude ของตำแหน่งจุดอ้างอิง

ตัวอย่างการใช้งาน

http://inear3g.embeddednu.com/apis/get_waypoint/evnu2013

- key = evnu2013

เอพีไอขอรายชื่อข้อมูล (Get Data API)

สำหรับขอข้อมูลเวลาและระยะทางสำหรับแต่ละป้ายจอดรถ โดยข้อมูลที่ได้จะประกอบไปด้วย ข้อมูลของป้ายจอดรถ เช่น เวลาที่รถไฟฟ้ากำลังจะมาถึง ระยะทางที่รถไฟฟ้ากำลังจะมาถึง เพื่อนำข้อมูลที่ได้ไปแสดงให้ในส่วนติดต่อกับผู้ใช้ ตามรูปแบบโพรโทคอลในตารางที่ 3.8 หากมีการร้องขอข้อมูลเสร็จสิ้นจะมีข้อมูลตอบกลับ ดังรูปที่ 3.22 ซึ่งคำอธิบายสำหรับข้อมูลตอบกลับอธิบายไว้ ดังตารางที่ 3.9

การร้องขอ

ตารางที่ 3.8 รูปแบบโพรโทคอลในการขอรายชื่อข้อมูลจากเครื่องบริการเว็บ

	http://inear3g.embeddednu.com/apis/get_data/
	GET
	• Key ○ คีย์ของผู้ใช้
	JSON

ข้อมูลตอบกลับ


```
{
    "code": 0,
    "msg": "Get data - OK",
    "data_list": [{
        "distance": "0.639",
        "date": "2013-04-14",
        "time": "35.5",
        "timenow": "00:01:35",
        "label": "Bus stop"
    }, {
        "distance": "0.529",
        "date": "2013-05-08",
        "time": "29.3888888889",
        "timenow": "00:01:35",
        "label": "Faculty of Engineering"
    }, {
        "distance": "2.202",
        "date": "2013-05-08",
        "time": "122.333333333",
        "timenow": "00:01:35",
        "label": "Faculty of Medicine"
    }]
}
```


รูปที่ 3.22 ข้อมูลตอบกลับในการขอรายชื่อข้อมูลจากเครื่องบริการเว็บ รูปแบบ JSON

ตารางที่ 3.9 คำอธิบายข้อมูลตอบกลับในการขอรายชื่อข้อมูลจากเครื่องบริการเว็บ

ชื่อ	คำอธิบาย
code	0 หมายถึง สมัครสำเร็จ -1 หมายถึง ไม่มีการเรียก API -2 หมายถึง ไม่ถูกต้อง
msg	ข้อความอธิบาย code
data_list	รายชื่อข้อมูล
distance	ระยะทางระหว่างรถไฟฟ้ากับป้ายจอดรถ
date	วันที่ ณ ปัจจุบันที่บันทึกข้อมูล
time	เวลาระหว่างรถไฟฟ้ากับป้ายจอดรถ
timenow	เวลา ณ ปัจจุบันที่บันทึกข้อมูล
label	ชื่อป้ายจอดรถ

ตัวอย่างการใช้งาน

http://inear3g.embeddednu.com/apis/get_data/evnu2013

- key = evnu2013

เอพีไอขอรายชื่อตำแหน่ง (Get Location API)

สำหรับขอรายชื่อตำแหน่งปัจจุบันของรถไฟฟ้าที่กำลังเดินรถ โดยข้อมูลจะประกอบด้วย วันที่ เวลา หมายเลข สายเดินรถ ตำแหน่ง ละติจูด ลองจิจูด รายละเอียดต่างๆ และระดับตำแหน่งจุดอ้างอิงของรถไฟฟ้า ตามรูปแบบโพรโทคอลในตารางที่ 3.10 หากมีการร้องขอข้อมูลเสร็จสิ้นจะมีข้อมูลตอบกลับ ดังรูปที่ 3.23 ซึ่งคำอธิบายสำหรับข้อมูลตอบกลับอธิบายไว้ ดังตารางที่ 3.11

การร้องขอ

ตารางที่ 3.10 รูปแบบโพรโทคอลในการขอรายชื่อตำแหน่งจากเครื่องบริการเว็บ

	http://inear3g.embeddednu.com/apis/get_locations/evnu2013
	GET
	• Key O **คีย์ของผู้ใช้**
	JSON

ข้อมูลตอบกลับ


```
{
    "code": 0,
    "msg": "Get location - OK",
    "location_list": [{
        "date": "2013-10-28",
        "time": "00:01:03",
        "no": "2",
        "line": "Red",
        "lat": "16.7380140",
        "lon": "100.1995020",
        "description": "Got",
        "level": "3"
    }, {
        "date": "2013-10-28",
        "time": "00:01:12",
        "no": "1",
        "line": "Red",
        "lat": "16.7499030",
        "lon": "100.1940670",
        "description": "Got",
        "level": "17"
    }]
}
```


รูปที่ 3.23 ข้อมูลตอบกลับในการขอรายชื่อตำแหน่งจากเครื่องบริการเว็บ รูปแบบ JSON

ตารางที่ 3.11 คำอธิบายโพรโทคอลในการขอรายชื่อตำแหน่งจากเครื่องบริการเว็บ

ชื่อ	คำอธิบาย
code	0 หมายถึง success 1 หมายถึง ไม่มีการเรียก API 2 หมายถึง มีข้อมูลผิดพลาด
msg	ข้อความอธิบาย code
location_list	รายชื่อข้อมูล
date	วันที่ (ณ ปัจจุบันที่บันทึกข้อมูล)
time	เวลา (ณ ปัจจุบันที่บันทึกข้อมูล)
no	หมายเลขรถไฟฟ้า
line	สายเดินรถไฟฟ้า
lat	ตำแหน่ง latitude
lon	ตำแหน่ง longitude
description	รายละเอียดเกี่ยวกับรถไฟฟ้า
level	ระดับตำแหน่งจุดอ้างอิง

ตัวอย่างการใช้งาน

http://inear3g.embeddednu.com/apis/get_locations/evnu2013

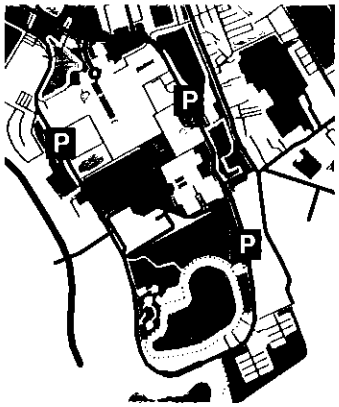
- key = evnu2013

3.2.3 ขั้นตอนการตั้งค่าเครื่องบริการเว็บ

- ตั้งค่าตำแหน่งจุดอ้างอิง (Waypoint) ของแผนที่เพื่อกำหนดเส้นทางของการเดินรถไฟฟ้า โดยจะต้องมีการกำหนดจุดอ้างอิง แต่ละจุดด้วยตนเอง ซึ่งสามารถหาตำแหน่งของแผนที่ได้จากบริการแผนที่ออนไลน์ต่างๆ และกำหนดรัศมีเป็น 50 เมตร เนื่องจากเป็นระยะทางที่สามารถมองเห็นได้ด้วยระดับสายตาและกำหนดรัศมีเพื่อการตรวจสอบรถไฟฟ้าว่าอยู่ในรัศมีของตำแหน่งจุดอ้างอิงใด ดังรูปที่ 3.24

รูปที่ 3.24 ตั้งค่าตำแหน่งจุดอ้างอิงของแผนที่เพื่อกำหนดเส้นทางของการเดินรถไฟฟ้า

- ตั้งค่าป้ายจอดรถ (Node) ของแผนที่เพื่อกำหนดป้ายจอดรถในการแสดงเวลา ดังรูปที่ 3.25 โดยจะต้องมีการกำหนดพิกัดตำแหน่งด้วยตนเอง ซึ่งสามารถหาตำแหน่งของแผนที่ได้จากบริการแผนที่ออนไลน์ต่างๆ

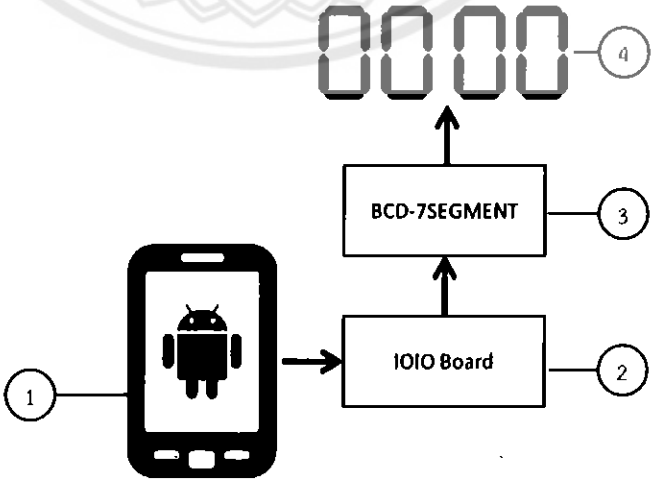


รูปที่ 3.25 ตั้งค่าป้ายจอดรถของแผนที่เพื่อกำหนดป้ายจอดรถในการแสดงเวลา

หลังจากมีการตั้งค่าเรียบร้อย ระบบเริ่มทำงานอัตโนมัติในกรณีที่มีผู้ใช้ส่งข้อมูลตำแหน่งมายังเครื่องบริการเว็บ โดยเครื่องบริการเว็บจะคำนวณระยะทางระหว่างรถไฟฟ้ากับป้ายจอดรถทุกป้ายที่มีการตั้งค่า เพื่อนำผลการคำนวณไปแสดงผลในส่วนติดต่อกับผู้ใช้

3.3 ส่วนติดต่อกับผู้ใช้

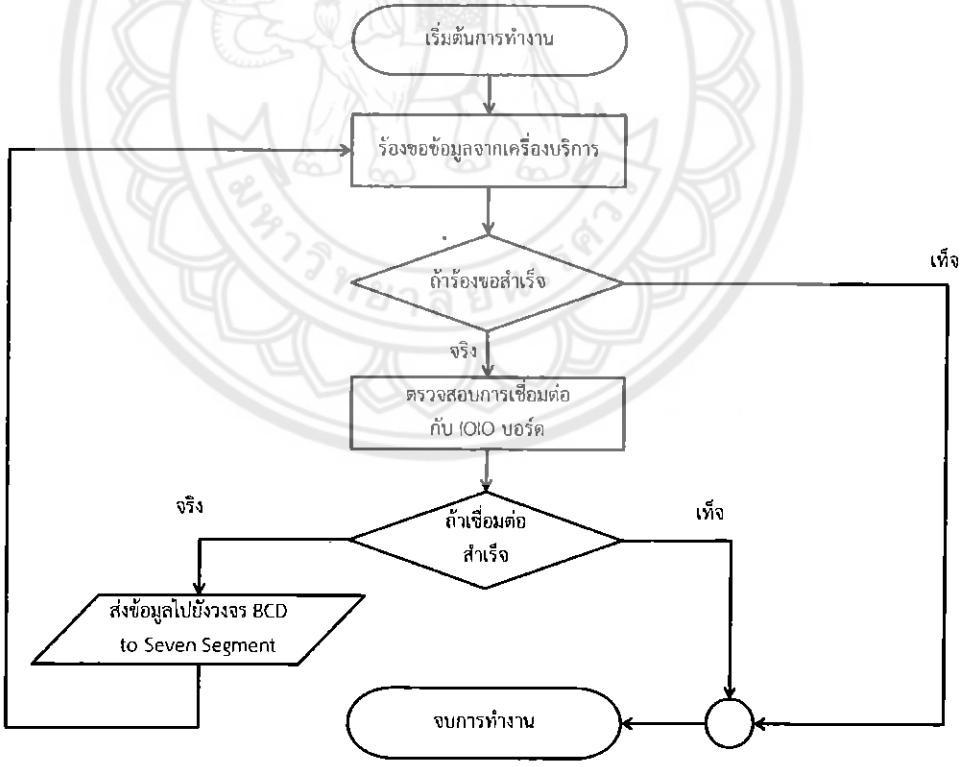
3.3.1 ส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ (User Interface on EV STOP)



รูปที่ 3.26 โครงสร้างของส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ

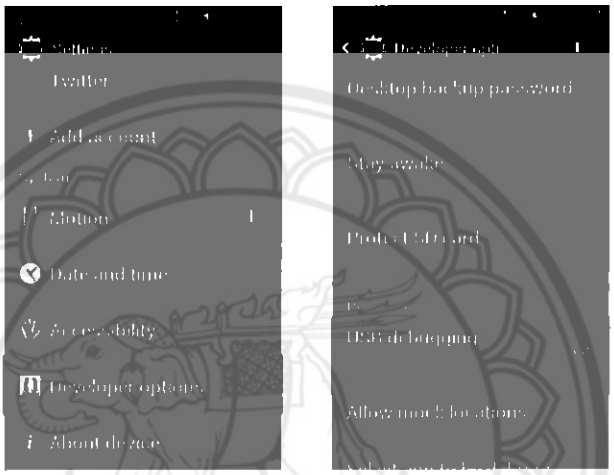
จากรูปที่ 3.26 แสดงการทำงานของส่วนติดต่อกับผู้ใช้บนป้ายจอดรถประกอบไปด้วยสมาร์ทโฟนแอนดรอยด์ (1) จะทำหน้าที่ร้องขอข้อมูลจากเครื่องบริการเว็บ โดยติดต่อสื่อสารผ่านเอพีไอ (Get Data API) ซึ่งจะแสดงข้อมูลในรูปแบบ JSON หลังจากนั้นทำการตรวจสอบข้อมูลที่ได้รับหากข้อมูลมีความถูกต้อง จะแปลงข้อมูลในรูปแบบ JSON ให้เป็นข้อมูลในรูปแบบตัวแปร (variable) แล้วทำการแปลงข้อมูลให้เป็นรูปแบบ BCD เพื่อส่งไป IOIO บอร์ด (2) ซึ่งมีการส่งค่าไปยัง IOIO บอร์ดตลอดเวลา เพื่อนำข้อมูลที่ได้รับเข้าวงจร BCD - Seven Segment (3) ไปแสดงบน Seven Segment (4) ให้กับผู้ใช้ แต่ถ้าข้อมูลมีความผิดพลาดสมาร์ทโฟนจะทำการร้องขอข้อมูลใหม่อีกครั้ง ดังนั้นเพื่อการพัฒนาให้สามารถแสดงเวลาได้จำเป็นต้องมีส่วนประกอบ ดังนี้

3.3.1.1 สมาร์ทโฟน



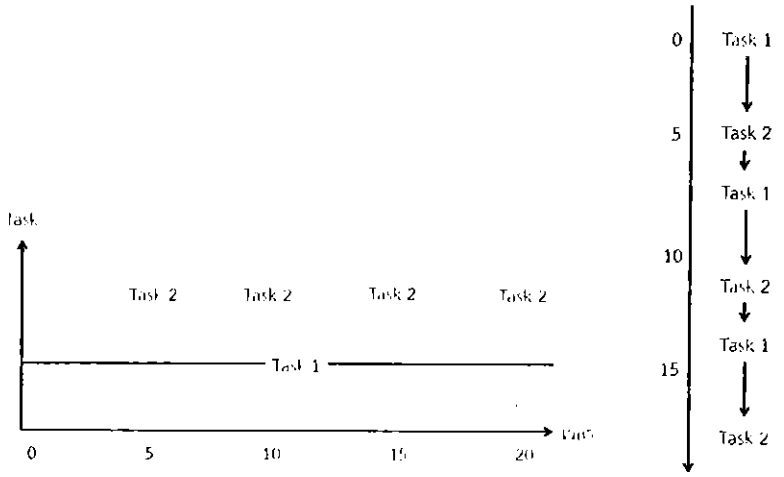
รูปที่ 3.27 แสดงการทำงานของโปรแกรมประยุกต์ในส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ

จากรูปที่ 3.27 เมื่อทำการเปิดโปรแกรมประยุกต์ มีการร้องขอข้อมูลจากเครื่องบริการเว็บเมื่อเครื่องบริการเว็บทราบถึงการร้องขอจะทำตอบกลับข้อมูลมายังสมาร์ทโฟน ถ้าข้อมูลถูกต้อง จะมีการตรวจสอบการเชื่อมต่อกับ IOIO บอร์ด ถ้ามีการเชื่อมต่อจะทำการส่งข้อมูลไปยังวงจร BCD - Seven Segment เพื่อแสดงผลให้กับผู้ใช้ หลังจากนั้นจะทำการร้องขอข้อมูลอีกครั้ง หากการร้องขอหรือเชื่อมต่อไม่สำเร็จก็ทำการจบการทำงานของโปรแกรมประยุกต์



รูปที่ 3.28 การตั้งค่า USB Debugging

สมาร์ทโฟนแอนดรอยด์ไม่อนุญาตให้มีการติดต่อกับอุปกรณ์ภายนอกได้โดยตรง โดยจำเป็นต้องเปิดโหมด USB Debugging ดังรูปที่ 3.28 ซึ่งการติดต่อนั้นต้องมีการตั้งค่าอุปกรณ์โดยเข้าไปที่หน้าเมนู Setting เลือก Developer options จากนั้นทำการเปิด USB Debugging ก็จะสามารถติดต่อกับอุปกรณ์ภายนอกได้ ในการทำงานของ IOIO บอร์ดจะรับข้อมูลผ่านสาย USB จากนั้นทำการเขียนค่าดิจิทัลออกทางขาพิน (pins) ต่างๆ ที่กำหนดไว้ เพื่อส่งข้อมูลดังกล่าวไปยังวงจรแปลง BCD เป็น Seven Segment ในวงจรจะมีไอซีเบอร์ 7447 ซึ่งวงจรจะมีตัวถอดรหัส (decoder) จากเลข BCD แสดงผลออกทาง LED Seven Segment จำนวน 4 หลัก

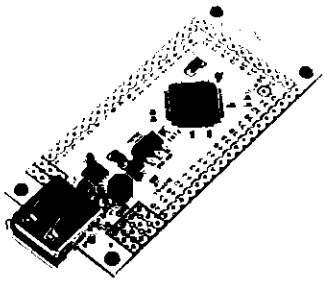


รูปที่ 3.29 แสดงการทำงานตัวโปรแกรมประยุกต์ระบุตำแหน่ง ในการร้องขอเวลาจากเครื่องบริการเว็บ และการนับเวลาของสมาร์ทโฟนแอนดรอยด์ในรูปแบบมัลติเทรด (ซ้าย) และแบบ 1 เทรด (ขวา)

จากรูปที่ 3.29 (ซ้าย) แสดงการร้องขอข้อมูลเวลาจากเครื่องบริการเว็บ โดย Task 1 จะทำงานตลอดเวลา ซึ่งหมายถึงการนับเวลาถอยหลังอยู่ตลอดเมื่อมีการทำงานอยู่ ส่วน Task 2 จะทำหน้าที่ร้องขอข้อมูลเวลาจากเครื่องบริการเว็บที่มีการเปลี่ยนแปลง แล้วทำการส่งข้อมูลเวลาไปยัง Task 1 เพื่อให้นับเวลาถอยหลัง เพื่อส่งค่าให้กับ IOIO บอร์ด ในการแสดงผลให้กับ Seven Segment

รูปที่ 3.29 (ขวา) การทำงานเพียง 1 เทรด โดย Task 1 ทำการนับเวลาถอยหลังเสร็จสิ้น Task 2 จะทำหน้าที่ร้องขอข้อมูลจากเครื่องบริการเว็บ เมื่อได้เวลาแล้วจะส่งข้อมูลให้ Task 1 ทำงานต่อ ซึ่งเสียเวลาและทำให้ Task 1 ไม่สามารถนับเวลาถอยหลังได้ ในขณะที่ Task 2 ไปร้องขอเวลาจากเครื่องบริการเว็บ

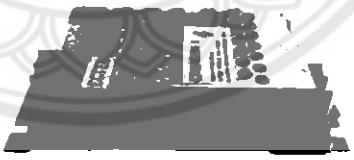
3.3.1.2 IOIO บอร์ด



รูปที่ 3.30 IOIO บอร์ด

จากรูปที่ 3.30 คือ IOIO บอร์ดทำหน้าที่อินพุตโดยรับข้อมูลเวลามาจากสมาร์ทโฟน จากนั้นทำหน้าที่เอาต์พุตในการส่งข้อมูลเวลาที่ถูกแปลงเป็น BCD ไปยังวงจร BCD - Seven Segment เพื่อไปแสดงผล โดยการพัฒนาโปรแกรมประยุกต์ให้ทำงานร่วมกับ IOIO บอร์ดจะใช้วิธีการเรียกใช้งาน API ของ IOIO บอร์ดในการสั่งงาน

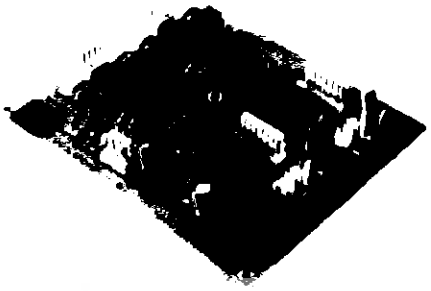
3.3.1.3 Switching Power Supply



รูปที่ 3.31 แสดง Switching Power Supply Output 12VDC 2A [4]

จากรูปที่ 3.31 คือ Switching Power Supply Input 220 VAC 50 Hz มี Output 12VDC 2A เพื่อใช้เป็นแหล่งจ่ายไฟให้กับวงจรทั้งหมดภายในป้ายจอดรถไฟฟ้ามหาวิทยาลัยนเรศวร และใช้สำหรับเป็นแหล่งจ่ายไฟให้กับ IOIO บอร์ด

3.3.1.4 วงจรแปลง BCD - Seven Segment



รูปที่ 3.32 แสดงวงจรแปลง BCD เป็น Seven Segment [4]

จากรูปที่ 3.32 คือวงจรแปลง BCD เป็น Seven Segment ทำหน้าที่รับข้อมูลมาจาก IOIO บอร์ด โดยจะรับมาเป็น BCD และมีไอซีเบอร์ 7447 ทำการแปลง BCD เป็นตัวเลขสำหรับแสดงบน Seven Segment โดยใช้ไฟเลี้ยงวงจรจาก Switching Power Supply 12VDC 2A ต่อเข้ามาที่จุดจ่ายไฟของวงจร (21) และที่จุดเดียวกันจะนำไปต่อกับบอร์ดที่ต่อใช้งานร่วมกับ ไมโครคอนโทรลเลอร์ เพื่อเป็นไฟเลี้ยงให้กับ ไมโครคอนโทรลเลอร์ และมีวงจรแปลงไฟสองส่วนได้แก่วงจรแปลงไฟเหลือ 5VDC และ 3.3VDC โดยวงจรแปลงไฟให้เหลือ 5VDC ใช้ในการเลี้ยงวงจร และไอซีต่างๆ และมีเอาต์พุตให้สามารถต่ออุปกรณ์ต่อไฟเลี้ยงใช้แรงดัน 5VDC (22) และแรงดัน 3.3VDC (23) นอกจากนี้แล้วยังมีส่วนที่สำคัญอีกส่วนหนึ่งได้แก่ส่วนรับข้อมูล BCD มาจาก IOIO บอร์ด (24) จากนั้นก็จะส่งออกไปที่ Seven Segment (25) ดังรูปที่ 3.33

รูปที่ 3.33 แสดงแผงวงจรแปลง BCD เป็น Seven Segment [4]

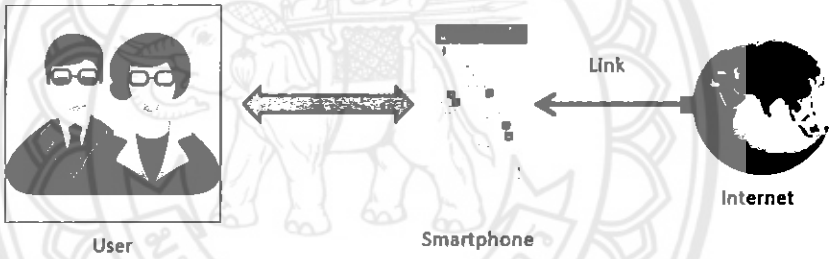
3.3.1.5 Seven Segment แสดงผล 4 หลัก

รูปที่ 3.34 แสดง Seven Segment 4 หลัก [4]

จากรูปที่ 3.34 คือ Seven Segment แสดงผล 4 หลัก ทำหน้าที่แสดงเวลาของรถไฟฟ้าที่จะมาถึงป้ายจอดรถ โดยรับข้อมูลที่ถูกถอดรหัสสำหรับแสดงผลบน Seven Segment จากวงจรแปลง BCD เป็น Seven Segment เข้ามาที่วงจรสำหรับใช้งานร่วมกับ Seven Segment ที่ Connector 8 Pin (26) ซึ่งวงจรนี้จะทำหน้าที่เรียงขาของ Seven Segment ให้เป็นตามลำดับเพื่อง่ายต่อการใช้งานกับการสั่งงาน และเพื่อนำไปยึดติดกับป้ายจอดรถไฟฟ้าได้ง่าย ดังรูปที่ 3.35

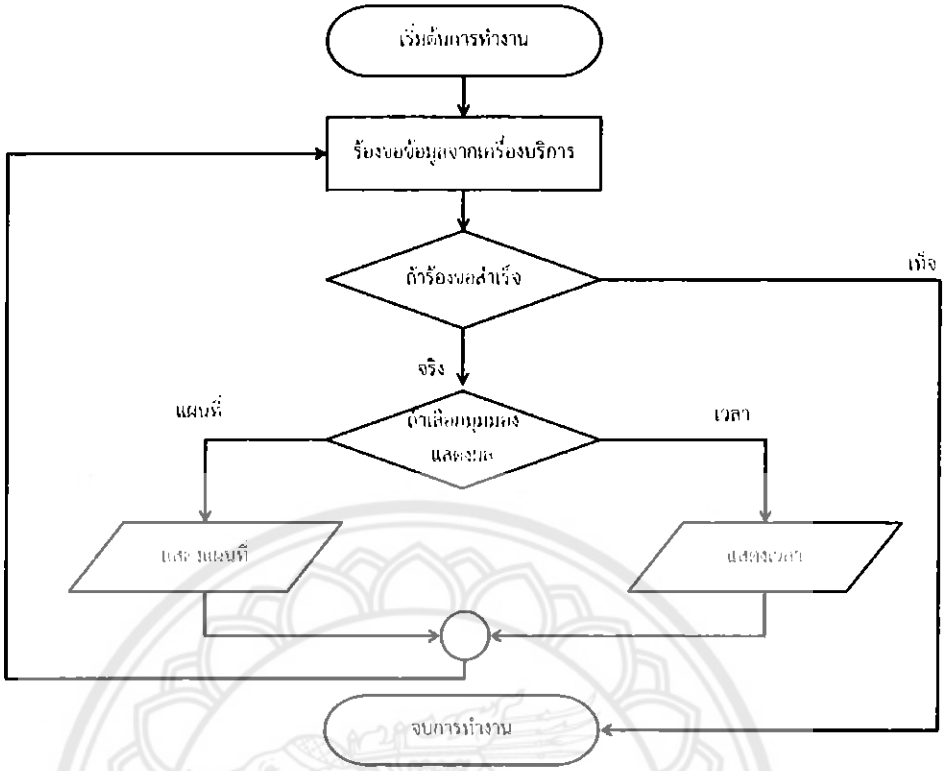
รูปที่ 3.35 แสดงวงจรสำหรับใช้งานร่วมกับ Seven Segment [4]

3.3.2 ส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน (User Interface on Smart Phone)



รูปที่ 3.36 โครงสร้างของส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน

จากรูปที่ 3.36 ส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน ต้องมีการเชื่อมต่ออินเทอร์เน็ตตลอดการใช้งาน เนื่องจากจะต้องมีการร้องขอข้อมูลและรับข้อมูลเพื่อนำมาแสดงในรูปแบบของโปรแกรมประยุกต์บนสมาร์ทโฟน โดยแบ่งการทำงานออกเป็น 2 รูปแบบ คือ รูปแบบที่หนึ่ง แสดงผลแผนที่ ประกอบไปด้วยตำแหน่งปัจจุบันของรถไฟฟ้าและตำแหน่งป้ายจอดรถ รูปแบบที่สอง แสดงเวลาโดยประมาณระหว่างรถไฟฟ้ากับป้ายจอดรถ



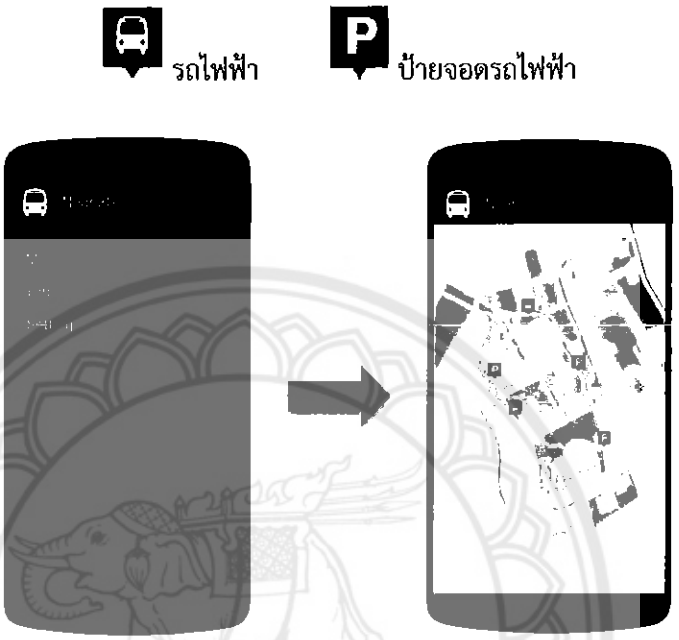
รูปที่ 3.37 แสดงขั้นตอนการทำงานของส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน

จากรูปที่ 3.37 เมื่อทำการเปิดโปรแกรมประยุกต์ มีการร้องขอข้อมูลจากเครื่องบริการเว็บ เมื่อเครื่องบริการเว็บทราบถึงการร้องขอจะทำตอบกลับข้อมูลมายังสมาร์ทโฟน ถ้าการร้องขอสำเร็จจะแสดงเมนูให้ผู้ใช้สามารถเลือกได้ว่าต้องการแสดงข้อมูลในรูปแบบใดระหว่างแผนที่กับการประมาณเวลา หลักจากนั้นจะทำการร้องขอข้อมูลอีกครั้ง ดังนั้นหากการร้องขอไม่สำเร็จจะทำการจบการทำงานของโปรแกรมประยุกต์

ผู้ใช้สามารถเลือกที่จะแสดงข้อมูลเฉพาะบางส่วนได้ เนื่องจากข้อมูลของการแสดงผลในส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน จำเป็นต้องมีการแสดงให้ผู้ใช้ระบุได้ว่าต้องการข้อมูลรูปแบบใด โดยมีแถบเมนูแสดงผลด้านซ้าย ซึ่งแบ่งออกเป็น 2 รูปแบบกับการตั้งค่า ดังนี้

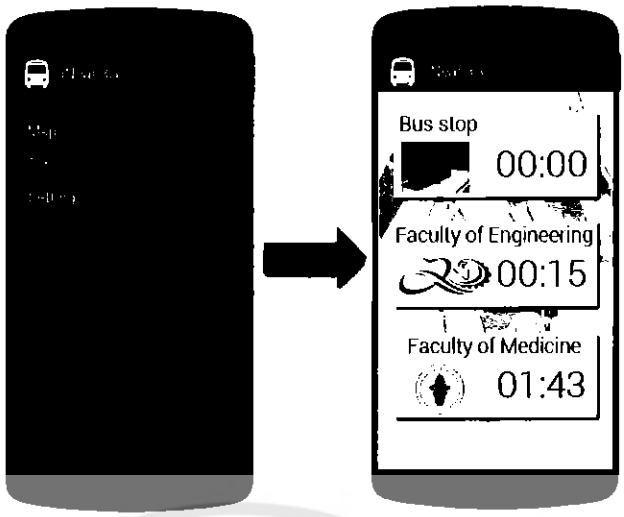
รูปแบบที่หนึ่ง แสดงผลเฉพาะแผนที่ (Map) ซึ่งพัฒนาด้วย Google Maps API บนระบบปฏิบัติการแอนดรอยด์ โดยมีการเลือกใช้งานฟังก์ชันแสดงแผนที่, Marker และแสดงตำแหน่งปัจจุบันของสมาร์ทโฟน ซึ่งนำข้อมูลต่างๆ มาจากการร้องขอจากเอพีไอ เช่น การนำข้อมูลมาแสดง

ตำแหน่งของรถไฟฟ้า (Get Locations API) และตำแหน่งของป้ายจอดรถ (Get Node API) ดังรูปที่ 3.38 โดยสัญลักษณ์แทนสถานที่ดังนี้



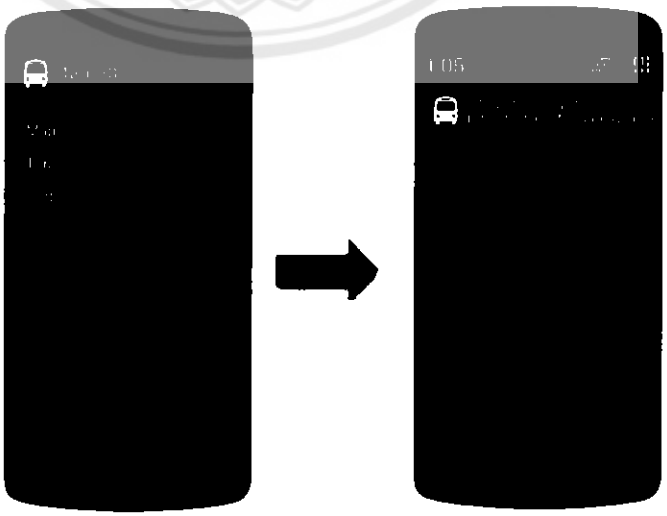
รูปที่ 3.38 ส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟนแสดงผลเฉพาะแผนที่

รูปแบบที่สอง แสดงผลเฉพาะการประมาณเวลา (Time Approximation) ในรูปแบบตัวเลขดิจิทัลในหน่วย (นาที:วินาที) ซึ่งการแสดงข้อมูลต่างๆ มาจากการร้องขอจาก API โดยข้อมูลที่ปรากฏจะแสดงเฉพาะชื่อป้ายจอดรถ ตราสัญลักษณ์และเวลาโดยประมาณ (Get Data API) ซึ่งจะมีการเปลี่ยนแปลงตัวเลขอยู่ตลอดเวลา ถ้าตำแหน่งรถไฟฟ้ามีการเคลื่อนที่ ดังรูปที่ 3.39



รูปที่ 3.39 ส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟนแสดงผลเฉพาะการประมาณเวลา

แถบเมนูการตั้งค่า สำหรับให้มีการตั้งค่าเพื่อเปิดปิดระบบแจ้งเตือน ในกรณีที่รถไฟฟ้าใกล้จะถึงป้ายจอดรถ ให้มีระบบแจ้งเตือนเวลาอัตโนมัติ (Notifications) เพื่อให้ผู้ใช้ทราบล่วงหน้า ซึ่งผู้ใช้สามารถเลือกได้ว่าจะแจ้งเตือนเฉพาะป้ายจอดรถ เพื่อรองรับหากผู้ใช้ต้องการใช้งานคุณสมบัติอื่นๆ ของสมาร์ทโฟน ระบบจะมีการแจ้งเตือนเพื่อให้ผู้ใช้สามารถรับรู้ได้ว่า ณ เวลาปัจจุบัน รถไฟฟ้ากำลังจะมาถึงป้ายจอดรถ อีกภายในระยะเวลาเท่าใด โดยจะมีการแจ้งเตือนเวลาอัตโนมัติ พร้อมกับระบบเสียง สั่น และแสงแอลอีดี (LED) (แสงแอลอีดีมีเฉพาะสมาร์ทโฟนบางรุ่น) ดังรูปที่ 3.40



รูปที่ 3.40 ส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟนในส่วนการตั้งค่าและระบบแจ้งเตือนเวลาอัตโนมัติ

บทที่ 4

ผลการทดสอบ

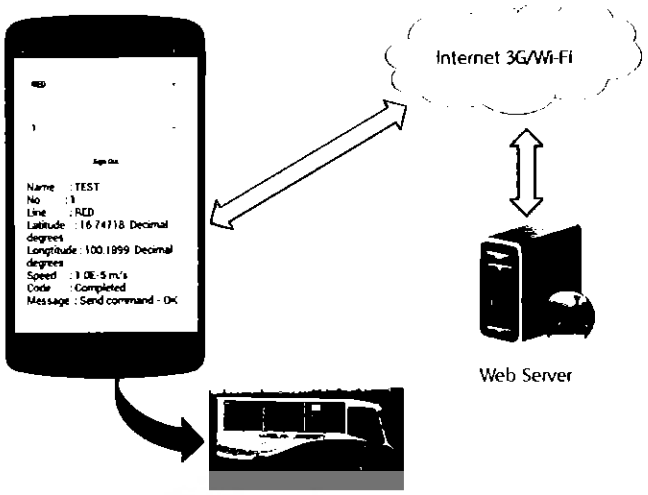
ในบทนี้จะกล่าวถึงการทดสอบต่างๆ หลังจากที่ออกแบบและพัฒนาระบบเพื่อทดสอบและเก็บรวบรวมข้อมูลไว้ตั้งแต่ การทดสอบเก็บตำแหน่งของถนนเส้นทางมหาวิทยาลัยนเรศวร การทดสอบความแม่นยำในการระบุตำแหน่งในกรณีรถไฟฟ้าเคลื่อนที่ การทดสอบความแม่นยำของโปรแกรมประยุกต์ตัวระบุตำแหน่งปัจจุบันของรถไฟฟ้า การทดสอบการคำนวณเวลาของเครื่องบริการ การวัดประสิทธิภาพด้านความเร็วการส่งข้อมูลไปยังเครื่องบริการเว็บ การทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ การทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน การทดสอบระบบโดยรวม และอัตราค่าใช้จ่ายทั้งหมดของระบบ

4.1 การทดสอบเก็บตำแหน่งของถนนเส้นทางมหาวิทยาลัยนเรศวร

การทดสอบเก็บตำแหน่งเส้นทางภายในมหาวิทยาลัยนเรศวรโดยใช้สมาร์ทโฟนแอนดรอยด์ติดตั้งโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า ซึ่งจะอ่านค่า GPS ได้ 3 คอลัมน์ ดังรูปที่ 4.1 ประกอบด้วย คอลัมน์ที่ 1 คือ ละติจูด คอลัมน์ที่ 2 คือ ลองจิจูด และคอลัมน์ที่ 3 คือ อัตราเร็ว (เมตรต่อวินาที) หลังจากอ่านค่าเรียบร้อย จะทำการส่งข้อมูลไปยังเครื่องบริการเว็บผ่านอินเทอร์เน็ต ด้วยเครือข่าย 3G เพื่อบันทึกลงฐานข้อมูล

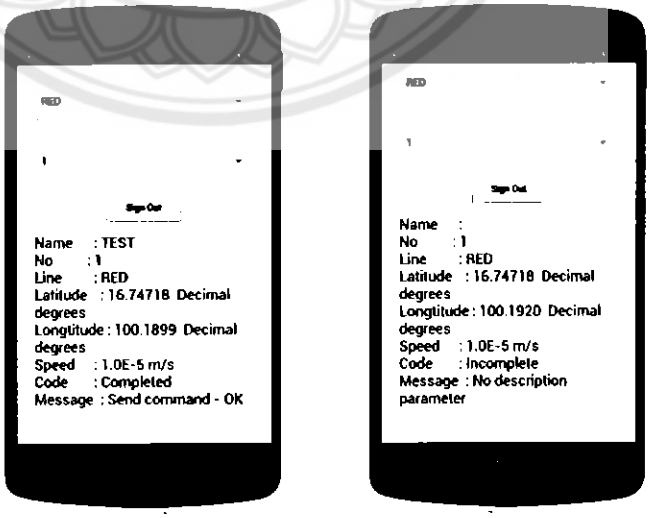
16.744862403720617	100.19636082462966	1.2510010467300414
16.74478265945952	100.19663415849209	2.3769009576568605
16.744704907760024	100.19668486900628	2.502001093460083
16.744672930799425	100.19672912545502	2.689651058746338
16.74464858137071	100.19677053205669	2.689651058746338
16.744608096778393	100.19682593643665	3.064950989318848

รูปที่ 4.1 ตัวอย่างค่า GPS บนสมาร์ทโฟน



รูปที่ 4.2 การสื่อสารระหว่างสมาร์ทโฟนบนรถไฟฟ้าและเครื่องบริการเว็บ

จากรูปที่ 4.2 ในการระบุตำแหน่งปัจจุบันของรถไฟฟ้าจำเป็นต้องมีสมาร์ทโฟนอยู่บนรถไฟฟ้า เพื่อการระบุตำแหน่งปัจจุบันไปยังเครื่องบริการเว็บ ซึ่งตำแหน่งของรถไฟฟ้าที่อ่านจาก GPS มีการตั้งค่าให้รับค่าทุกๆ 2 วินาที และแสดงค่าออกทางหน้าจอของสมาร์ทโฟน เมื่อมีการส่งข้อมูลไปยังเครื่องบริการเว็บ ถ้าข้อมูลถูกต้องตามโพรโทคอลแล้วเครื่องบริการเว็บจะส่งข้อมูลตอบกลับมายังสมาร์ทโฟน โดยมีข้อความรูปที่ 4.3



รูปที่ 4.3 การส่งข้อมูลถูกต้อง (ซ้าย) การส่งข้อมูลผิดพลาด (ขวา)

ถ้าข้อมูลผิดพลาดเครื่องบริการเว็บจะตอบกลับด้วย Code : -1 ถึง -9 และแสดงข้อมูลที่ผิดพลาดใน Msg ดังนั้นหลังจากสมาร์ทโฟนส่งข้อมูลเสร็จจะต้องมีการตรวจสอบข้อมูลตอบกลับว่า Code เป็น Completed หรือไม่ ถ้า Code มีข้อมูลเป็น Completed ให้สมาร์ทโฟนรับค่า GPS ค่าใหม่แล้วส่งข้อมูลใหม่อีกครั้ง ทำซ้ำแบบนี้ไปจนกว่าจะปิดโปรแกรมประยุกต์ แต่ Code มีข้อมูลเป็น Incomplete โปรแกรมประยุกต์จะส่งข้อมูลใหม่อีกครั้งจนกระทั่งส่งข้อมูลสำเร็จหรือจนกว่าจะปิดโปรแกรมประยุกต์

4.2 การทดสอบความแม่นยำในการระบุตำแหน่งในกรณีรถไฟฟ้าเคลื่อนที่

การทดสอบนี้จะแสดงให้เห็นว่าอัตราเร็วมีผลต่อการส่งข้อมูลไปยังเครื่องบริการเว็บ ซึ่งการทดสอบนั้นจะให้รถไฟฟ้าวิ่งด้วยอัตราเร็วที่แตกต่างกัน แล้วเริ่มนับเวลาตั้งแต่ขั้นตอนการรับค่าจาก GPS จนกระทั่งส่งข้อมูลไปยังเครื่องบริการเว็บและได้รับข้อความตอบกลับ โดยทำการส่งข้อมูล 15 ครั้ง ในแต่ละอัตราเร็ว ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงความสัมพันธ์ระหว่างเวลาและความเร็วในการรับส่งข้อมูลจากสมาร์ทโฟนไปยังเครื่องบริการเว็บ

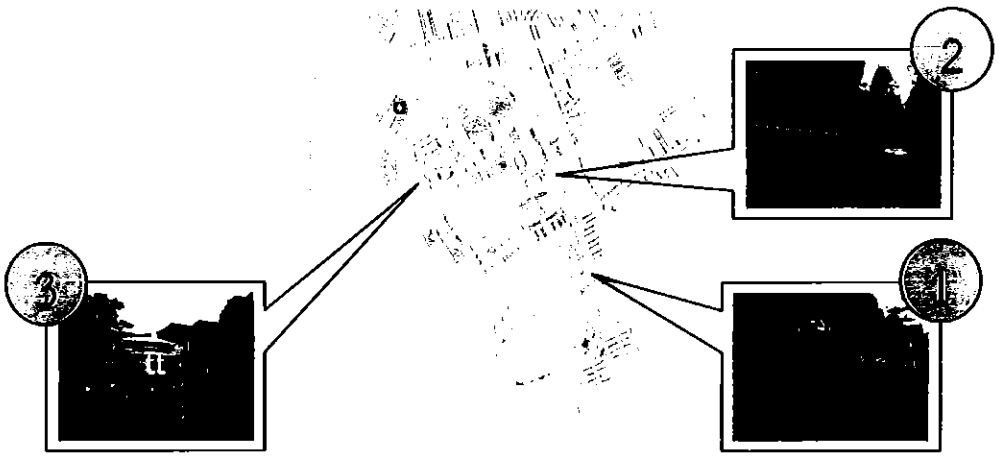
ครั้ง	20 km/hr		40 km/hr		60 km/hr	
	เวลาในการส่ง (ms)	อัตราเร็วจาก GPS	เวลาในการส่ง (ms)	อัตราเร็วจาก GPS	เวลาในการส่ง (ms)	อัตราเร็วจาก GPS
1	222	18.014	768	37.830	1,375	51.566
2	371	18.240	351	35.353	1,252	51.791
3	232	18.690	449	39.300	1,076	51.791
4	306	18.690	608	39.194	1,725	50.666
5	310	18.915	663	38.419	1,018	52.242
6	245	18.915	785	35.942	1,180	50.440
7	224	19.140	823	35.671	1,116	51.116
8	236	19.816	448	37.101	1,348	51.566
9	202	19.816	222	35.804	1,460	50.666

ครั้ง	20 km/hr		40 km/hr		60 km/hr	
	เวลาในการส่ง (ms)	อัตราเร็วจาก GPS	เวลาในการส่ง (ms)	อัตราเร็วจาก GPS	เวลาในการส่ง (ms)	อัตราเร็วจาก GPS
10	273	20.041	871	36.704	1,328	53.368
11	219	20.266	232	38.281	1,120	52.692
12	201	20.491	606	35.804	1,082	51.791
13	208	20.455	310	36.930	1,264	52.242
14	178	19.140	985	39.530	1,203	52.917
15	196	18.115	537	36.125	1,107	54.242
เฉลี่ย	241.533	19.250	577.200	37.199	1,243.600	51.940
Max	371.000	20.491	985.000	39.530	1,725.000	54.242
Min	178.000	18.014	222.000	35.353	1,018.000	50.440

จากตารางที่ 4.1 แสดงผลการทดสอบทำให้ทราบว่าความเร็วที่ 20 กิโลเมตรต่อชั่วโมง โปรแกรมประยุกต์ตัวระบุตำแหน่งปัจจุบันของรถไฟฟ้า สามารถรับอัตราเร็วได้เฉลี่ยประมาณ 19.250 กิโลเมตรต่อชั่วโมง ใช้เวลาในการรับส่งข้อมูลไปยังเครื่องบริการเว็บเป็นเวลาในหน่วยมิลลิวินาที เมื่ออัตราเร็วเพิ่มขึ้นระยะเวลาในการรับส่งข้อมูลก็จะมากขึ้น (ระยะเวลาในการรับส่งข้อมูลขึ้นอยู่กับเครื่องบริการเว็บและความเร็วของอินเทอร์เน็ต)

4.3 การทดสอบความแม่นยำของโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า

การทำลองนำสมาร์ทโฟนที่ติดตั้งโปรแกรมประยุกต์ตัวระบุตำแหน่งปัจจุบันของรถไฟฟ้าไปเก็บตำแหน่ง ที่จุดต่างๆ ที่วางแผนไว้ ดังรูปที่ 4.4 ทั้งหมด 3 จุด ตามตารางที่ 4.2 โดยมีการกำหนดตำแหน่งจากบริการแผนที่ออนไลน์ เพื่อทราบค่าตำแหน่งละติจูดและลองจิจูดของแต่ละจุด เพื่อนำไปเปรียบเทียบกับตำแหน่งที่สมาร์ทโฟนที่ได้รับ ตามตารางที่ 4.3 เพื่อหาความผิดพลาดของการระบุตำแหน่ง



รูปที่ 4.4 ตำแหน่งที่เก็บจากโปรแกรมประยุกต์ตัวระบุตำแหน่งปัจจุบันของรถไฟฟ้า

ตารางที่ 4.2 ค่าพิกัดตำแหน่งได้จากการกำหนดจุดบนบริการแผนที่ออนไลน์

ลำดับ	ชื่อตำแหน่ง	ตำแหน่งได้จากบริการแผนที่ออนไลน์	
		ละติจูด	ลองจิจูด
1.	ป้ายจอดรถเมล์	16.7408520	100.1991070
2.	ป้ายจอดรถไฟฟ้าสามแยกตึก EE	16.7453390	100.1974260
3.	ป้ายจอดรถไฟฟ้าคณะแพทยศาสตร์	16.7451670	100.1907250

ตารางที่ 4.3 ค่าพิกัดตำแหน่งได้จากการสมาร์ทโฟน

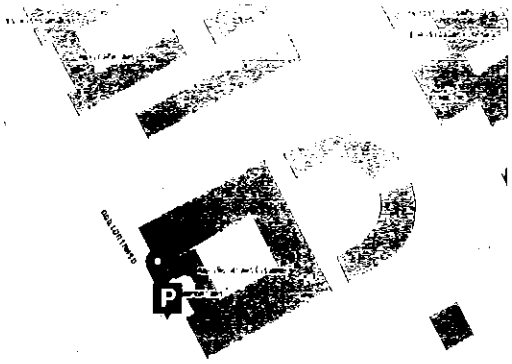
ลำดับ	ชื่อตำแหน่ง	ตำแหน่งได้จากสมาร์ทโฟน	
		ละติจูด	ลองจิจูด
1.	ป้ายจอดรถเมล์	16.7408390	100.1991260
2.	ป้ายจอดรถไฟฟ้าสามแยกตึก EE	16.7453750	100.1974020
3.	ป้ายจอดรถไฟฟ้าคณะแพทยศาสตร์	16.74514372	100.1907455

นำข้อมูลจากตารางที่ 4.3 และตารางที่ 4.4 ไปแสดงผลในแผนที่ ดังรูปที่ 4.5 ถึงรูปที่ 4.7 โดยสัญลักษณ์แทนตำแหน่งดังนี้

ตำแหน่งได้จากการสมาร์ทโฟน ตำแหน่งได้จากบริการแผนที่ออนไลน์

รูปที่ 4.5 ความคลาดเคลื่อนของตำแหน่งป้ายจอดรถเมล์ประมาณ 2.486 เมตร

รูปที่ 4.6 ความคลาดเคลื่อนของตำแหน่งป้ายจอดรถไฟฟ้าสามแยกตึก EE ประมาณ 4.749 เมตร



รูปที่ 4.7 ความคลาดเคลื่อนของตำแหน่งป้ายจอดรถไฟฟ้าคณะแพทยศาสตร์ ประมาณ 3.386 เมตร

จากการนำข้อมูลทั้งสองแหล่ง มาหาค่าความคลาดเคลื่อนเพื่อแสดงให้เห็นว่าค่าที่อ่านได้จากสมาร์ทโฟนมีความคลาดเคลื่อนไปจากค่าที่กำหนดไว้กี่เมตร เริ่มจากหาระยะห่างระหว่าง 2 จุดตามสมการที่ 4.1

$$d = 2r \arcsin\left(\sqrt{sin^2\left(\frac{\phi 2-\phi 1}{2}\right) + \cos(\phi 1)\cos(\phi 2)sin^2\left(\frac{\lambda 2-\lambda 1}{2}\right)}\right) \qquad (4.1)$$

ตารางที่ 4.4 ผลการทดสอบหาค่าประสิทธิภาพความแม่นยำของการระบุตำแหน่งของสมาร์ทโฟน

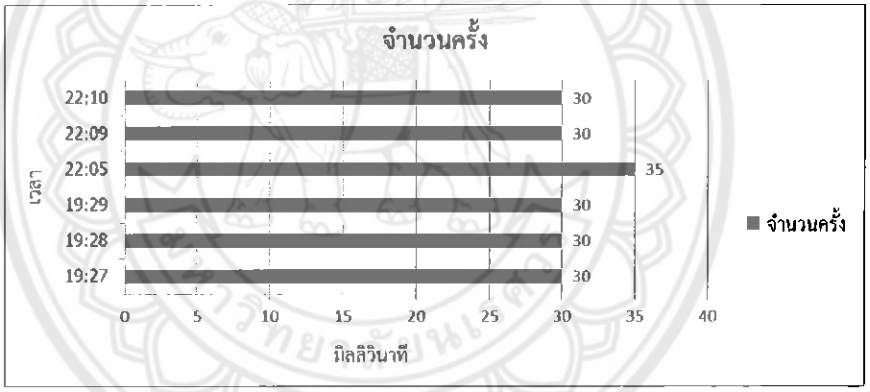
ลำดับ	ชื่อตำแหน่ง	ค่าความแตกต่าง		ค่าความผิดพลาด (เมตร)
		ละติจูด	ลองจิจูด	
1.	ป้ายจอดรถเมล์	0.00001300	0.00001900	2.4865140353829290
2.	ป้ายจอดรถไฟฟ้าสามแยกตึก EE	0.00003600	0.00002400	4.7491883959520160
3.	ป้ายจอดรถไฟฟ้าคณะแพทย์	0.00002328	0.00002050	3.3861060159189305

4.4 การวัดประสิทธิภาพด้านความเร็วการส่งข้อมูลไปยังเครื่องบริการเว็บ

การทดสอบนี้เพื่อการทดสอบประสิทธิภาพของความเร็วในการอัปโหลด (Upload) ข้อมูลของโปรแกรมประยุกต์ตัวระบุตำแหน่งปัจจุบันของรถไฟฟ้าบนสมาร์ทโฟนไปยังเครื่องบริการเว็บ โดยอัปโหลดข้อมูลภายใน 1 นาที สามารถสรุปผลการทดสอบได้ดังตารางที่ 4.5

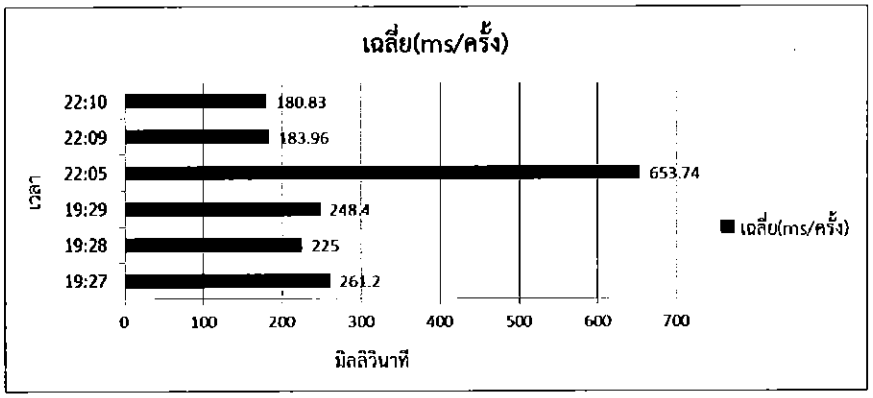
ตารางที่ 4.5 สรุปผลการทดสอบอัปโหลดข้อมูลไปยังเครื่องบริการเว็บ

เวลาที่ทดสอบ	19:27	19:28	19:29	22:05	22:09	22:10	22:11
จำนวนครั้ง	30	30	30	35	30	30	30
เฉลี่ย(ms/ครั้ง)	261.2	225	248.4	653.74	183.96	180.83	180.83



รูปที่ 4.8 แสดงจำนวนครั้งในแต่ละช่วงเวลาที่มีการอัปโหลดข้อมูล

จากรูปที่ 4.8 แสดงให้เห็นว่าข้อมูลจะมีการเปลี่ยนแปลงค่า GPS ทุกๆ 2 วินาที หรือส่งข้อมูล 30 ครั้งต่อนาที แล้วส่งไปยังเครื่องบริการเว็บ ค่ามากที่สุดคือ 35 ครั้งต่อนาที ซึ่งหมายความว่ามีการส่งข้อมูลไม่สมบูรณ์จึงมีการส่งข้อมูลซ้ำค่าเดิมไปยังเครื่องบริการเว็บ ค่าปกติจะอยู่ที่ 30 ครั้งต่อนาที ดังนั้นการส่งข้อมูลไปยังเครื่องบริการเว็บไม่มีการผิดพลาด



รูปที่ 4.9 แสดงอัตราเฉลี่ยของเวลาในการส่งข้อมูลไปยังเครื่องบริการเว็บ

จากรูป 4.9 แสดงอัตราเฉลี่ยของเวลาตั้งแต่รับค่า GPS จนกระทั่งเครื่องบริการเว็บส่งข้อความตอบกลับมายังสมาร์ทโฟน ซึ่งเวลาในการส่งที่ใช้เวลาน้อยที่สุด คือ 180.83 มิลลิวินาที แต่หากนำค่าเวลามาเฉลี่ยทั้งหมดจะอยู่ที่ 276.28 มิลลิวินาทีต่อครั้ง

4.5 การทดสอบการคำนวณเวลาของเครื่องบริการ

การทดสอบการคำนวณเวลาของเครื่องบริการ สามารถทดสอบด้วยการส่งข้อมูลตำแหน่งปัจจุบันของรถไฟฟ้าไปยังเครื่องบริการเว็บเพื่อให้คำนวณหาเวลาโดยประมาณ จากการนำระยะทางระหว่างตำแหน่งปัจจุบันของรถไฟฟ้าและป้ายจอดรถมาหารด้วยอัตราเร็วที่ได้รับจาก GPS ซึ่งจะได้ผลลัพธ์คือเวลาที่รถไฟฟ้าจะเดินทางถึงป้ายจอดรถ

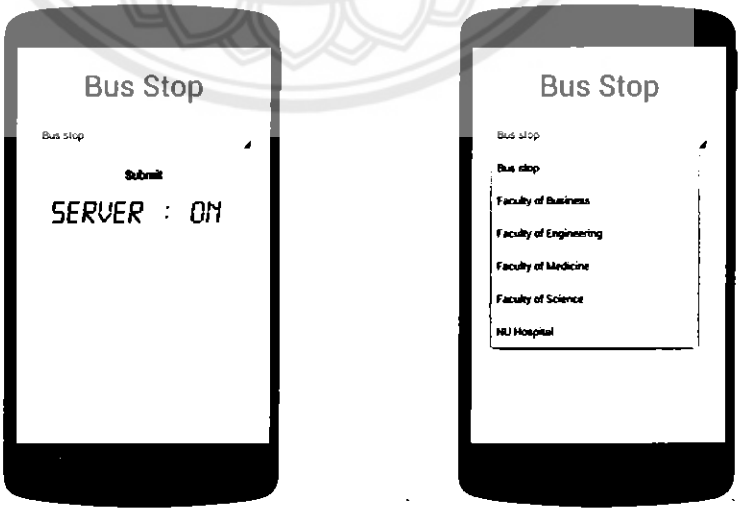
จากตารางที่ 4.6 จะเห็นได้ว่าอัตราเร็วที่ได้จาก GPS จะมีอัตราเร็วที่ช้าก็ว่าอัตราเร็วของรถไฟฟ้า ซึ่งจะทำให้การหาค่าเวลาคลาดเคลื่อนจากการคำนวณหลายวินาที แต่ค่าความคลาดเคลื่อนถือว่าเป็นค่าที่สามารถยอมรับได้ เนื่องจากในขณะที่รถไฟฟ้ากำลังจะถึงป้ายจอดรถนั้น ผู้ใช้สามารถมองเห็นรถไฟฟ้าได้ด้วยระดับสายตา ดังนั้นจึงทำให้การแสดงเวลาบนป้ายจอดรถหรือสมาร์ทโฟนไม่ส่งผลกระทบต่อผู้ใช้รถไฟฟ้า

ตารางที่ 4.6 การคำนวณเวลาของเครื่องบริการเว็บ (เวลามีหน่วยเป็นวินาที)

ลำดับ	อัตราเร็วของรถไฟฟ้า	ระยะทาง (เมตร)	เวลาที่ได้จากการคำนวณ	อัตราเร็วที่ได้จาก GPS	เวลาจากเครื่องบริการเว็บ
1	20	200	36.000	19.140	37.618
2	20	200	36.000	18.240	39.474
3	40	200	18.000	18.240	39.474
4	40	200	18.000	38.419	18.741
5	60	200	12.000	50.44	14.274
6	60	200	12.000	54.242	13.274

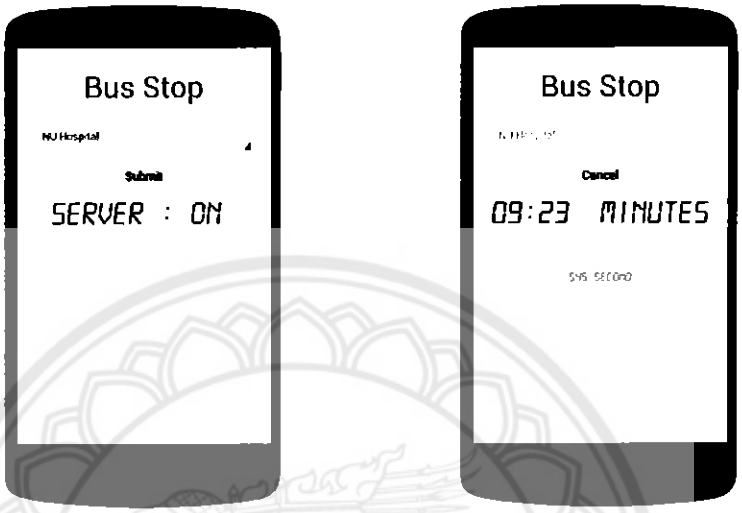
4.6 การทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ

การทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ จำเป็นต้องมีการเชื่อมต่ออินเทอร์เน็ตเพื่อร้องขอข้อมูลจากเครื่องบริการเว็บ เพื่อนำมาส่งให้กับ IOIO บอร์ด เพื่อแสดงผลบน Seven Segment ให้กับผู้ใช้งาน ในการทดสอบนี้จะทำการตรวจสอบข้อมูลที่ได้รับจากเครื่องบริการแล้วตรวจสอบว่าข้อมูลที่สมาร์ทโฟนส่งไปยัง Seven Segment ตรงกับข้อมูลที่ได้รับ โดยขั้นตอนการทำงานของส่วนแสดงผลส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ ดังรูปที่ 4.10 และรูปที่ 4.11



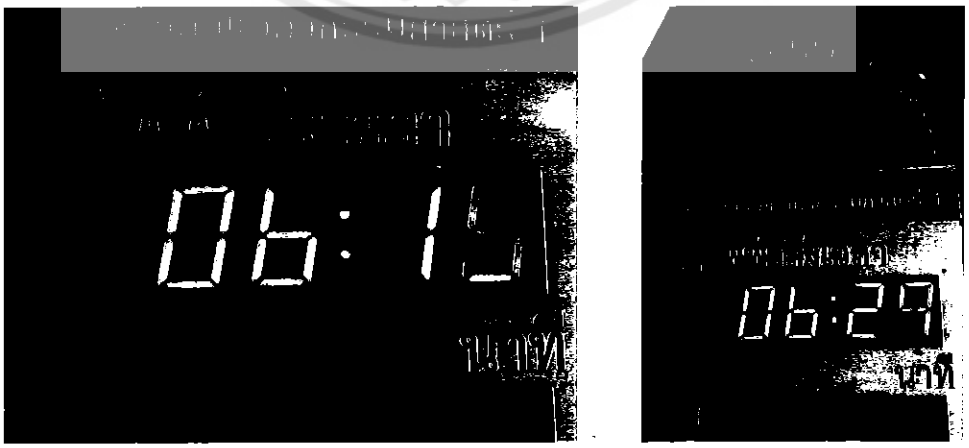
รูปที่ 4.10 ขั้นตอนการเลือกป้ายจอดรถของโปรแกรมประยุกต์ที่ใช้ติดต่อกับผู้ใช้บนป้ายจอดรถ

จากรูปที่ 4.10 โปรแกรมประยุกต์ที่ใช้ติดต่อกับผู้ใช้บนป้ายจอดรถเมื่อเปิดโปรแกรมประยุกต์ขึ้นมาแล้วโปรแกรมจะให้ทำการเลือกป้ายจอดรถที่ต้องการแสดงเวลาและส่งข้อมูลไปยัง Seven Segment



รูปที่ 4.11 ขั้นตอนการแสดงเวลาของโปรแกรมประยุกต์ที่ใช้ติดต่อกับผู้ใช้บนป้ายจอดรถ

จากรูปที่ 4.11 โปรแกรมประยุกต์ที่ใช้ติดต่อกับผู้ใช้บนป้ายจอดรถ เมื่อทำการเลือกป้ายจอดรถไฟฟ้าแล้วให้กดปุ่ม Submit จากนั้นโปรแกรมประยุกต์ก็จะทำการนับเวลาถอยหลังและส่งข้อมูลไปยัง IOIO บอร์ดเพื่อแสดงผลที่ Seven Segment ดังรูปที่ 4.12



รูปที่ 4.12 การแสดงเวลาบนสมาร์ทโฟนกับป้ายจอดรถไฟฟ้า

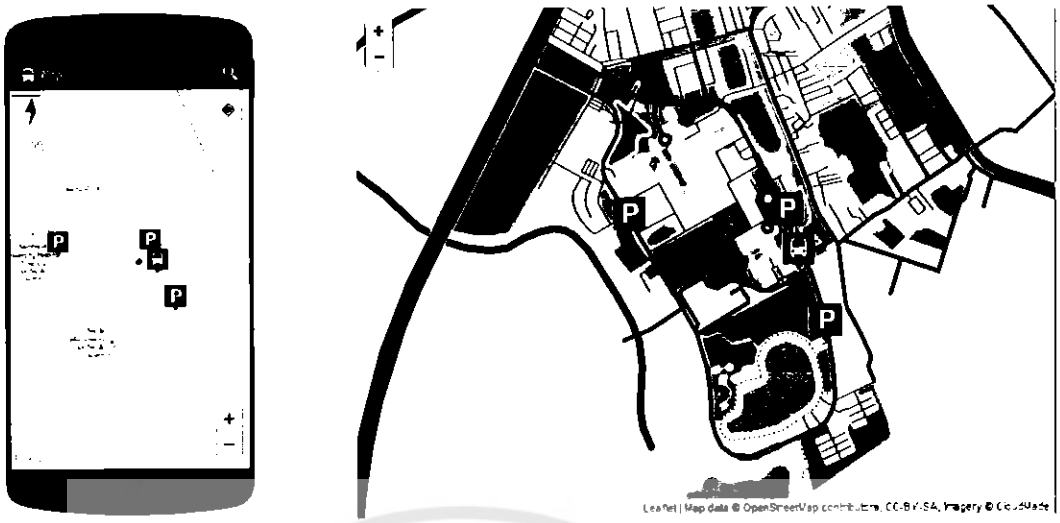
จากรูปที่ 4.12 การแสดงเวลาบนสมาร์ทโฟนกับป้ายจอดรถไฟฟ้า โปรแกรมประยุกต์บนสมาร์ทโฟนสามารถส่งเวลาไปยัง Seven Segment ได้แม่นยำและตรงกับเวลาที่แสดงบนหน้าจอสมาร์ทโฟน ดังนั้นหากมีการนับเวลาลดลงบนสมาร์ทโฟน ในส่วนของ Seven Segment จะมีการเปลี่ยนแปลงตามสมาร์ทโฟน

4.7 การทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน

การทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน นั้นจะต้องมีการติดตั้งโปรแกรมประยุกต์ติดต่อกับผู้ใช้และมีการเชื่อมต่ออินเทอร์เน็ตผ่านเครือข่าย 3G โดยหลังจากเปิดโปรแกรมประยุกต์ จะมีการร้องข้อมูลจากเครื่องบริการเว็บผ่านเอพีไอ เพื่อนำข้อมูลมาแสดงผลให้กับผู้ใช้ ซึ่งผู้ใช้สามารถเลือกได้ว่าจะแสดงเฉพาะแผนที่หรือประมาณเวลา และมีการตั้งค่าระบบแจ้งเตือนอัตโนมัติ ดังนั้นจึงแบ่งการทดสอบในหัวข้อนี้ออกเป็น 3 ส่วน ดังนี้

4.7.1 ทดสอบแสดงผลในส่วนของแผนที่

การทดสอบแสดงผลในส่วนของแผนที่เพื่อติดตามตำแหน่งของรถไฟฟ้าที่มีการเคลื่อนที่ ซึ่งแสดงผลเฉพาะแผนที่ โดยข้อมูลที่ปรากฏจะแสดงเฉพาะตำแหน่งของผู้ใช้ ตำแหน่งของรถไฟฟ้า ตำแหน่งของป้ายจอดรถ และแผนที่ ดังนั้นหากรถไฟฟ้ามีการเคลื่อนที่ตำแหน่งของรถไฟฟ้าบนแผนที่จะมีการเปลี่ยนแปลงตามที่รถไฟฟ้าเคลื่อนที่จริง หากรถไฟฟ้ามีการหยุดการเคลื่อนที่ ตำแหน่งของรถไฟฟ้าบนแผนที่จะหยุดนิ่งตามรถไฟฟ้า ดังรูปที่ 4.13

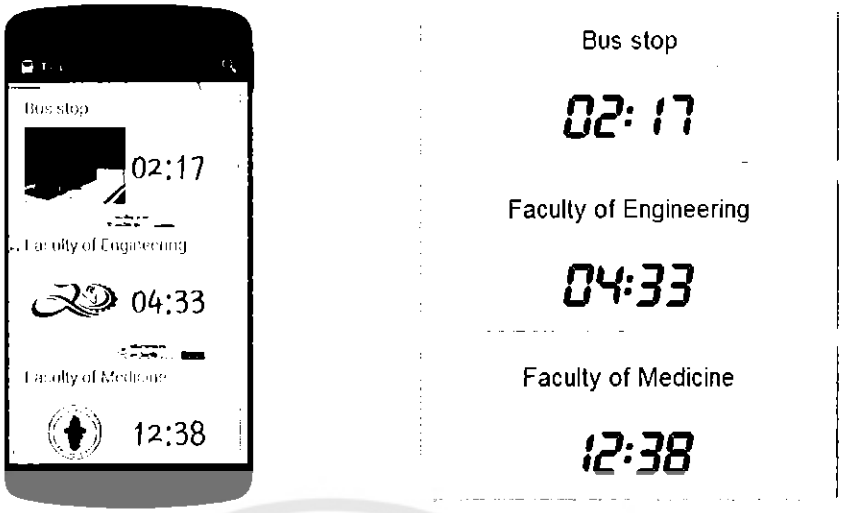


รูปที่ 4.13 การแสดงผลในส่วนของแผนที่บนสมาร์ทโฟน (ซ้าย) แสดงแผนที่บนเว็บไซต์ (ขวา)

จากรูปที่ 4.13 จะพบว่าตำแหน่งของรถไฟฟ้าที่แสดงบนสมาร์ทโฟนกับเว็บไซต์เป็นตำแหน่งที่ใกล้เคียงกัน เนื่องจากสมาร์ทโฟนจะต้องมีการร้องขอข้อมูลจากเครื่องบริการเว็บ ในทางกลับกันแผนที่ที่แสดงบนเว็บไซต์มีการร้องขอข้อมูลได้ไวกว่าสมาร์ทโฟน เนื่องจากฐานข้อมูลทั้งหมดอยู่บนเครื่องบริการเว็บ

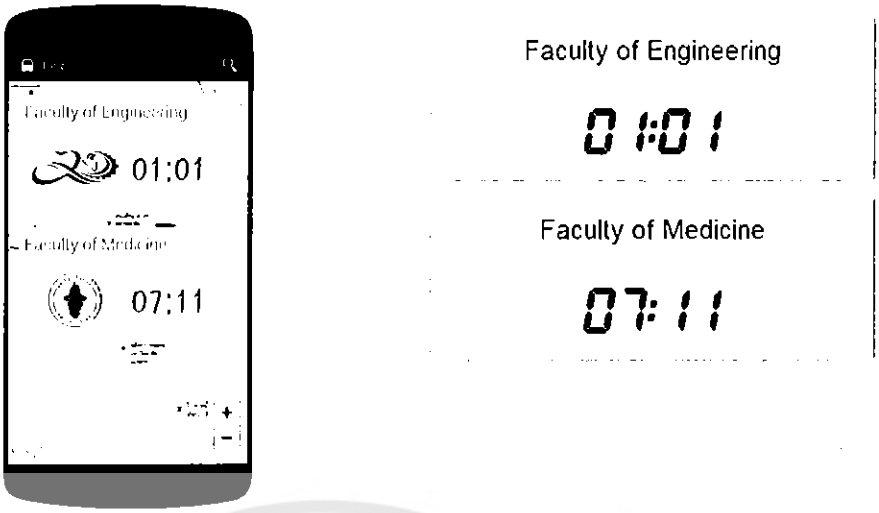
4.7.2 ทดสอบแสดงผลในส่วนของประมาณเวลา

การทดสอบแสดงผลในส่วนของประมาณเวลาเพื่อแสดงเวลาโดยประมาณระหว่างรถไฟฟ้ากับป้ายจอดรถ โดยข้อมูลที่ปรากฏจะแสดงเฉพาะชื่อป้ายจอดรถ ตราสัญลักษณ์และเวลาโดยประมาณ ดังนั้นหากรถไฟฟ้ายังเดินทางไม่ถึงป้ายจอดรถ จะแสดงเวลาโดยประมาณระหว่างรถไฟฟ้ากับป้ายจอดรถ หากรถไฟฟ้าเดินทางผ่านป้ายจอดรถแล้วจะไม่แสดงเวลาโดยประมาณของป้ายจอดรถนั้น แต่แสดงเวลาโดยประมาณของป้ายจอดรถที่กำลังจะเดินทางถึง ดังรูปที่ 4.14 และ 4.15



รูปที่ 4.14 การแสดงผลในส่วนของประมาณเวลาบนสมาร์ทโฟน (ซ้าย) แสดงเวลาบนเว็บไซต์ (ขวา)

จากรูปที่ 4.14 การทดสอบใช้งานการแสดงผลของการประมาณเวลาบนสมาร์ทโฟน ภาพซ้าย จะพบว่าโปรแกรมประยุกต์จะมีการแสดงเวลาโดยประมาณของแต่ละป้ายจอดรถ โดยเมื่อเปรียบเทียบกับเวลาบนเว็บไซต์จะมีเวลาที่ใกล้เคียงกันในบางช่วงเวลา และบางช่วงเวลาจะมีเวลาที่ห่างกัน เนื่องจากบนสมาร์ทโฟนเสียเวลาในการเชื่อมต่อกับเครื่องบริการ แต่เวลาที่ห่างนั้นเป็นค่าที่สามารถมองเห็นรถไฟฟ้าได้ด้วยสายตา ดังนั้นเมื่อเทียบเวลาที่ห่างกับเวลาจริง พบว่าใกล้เคียงกับความเป็นจริง

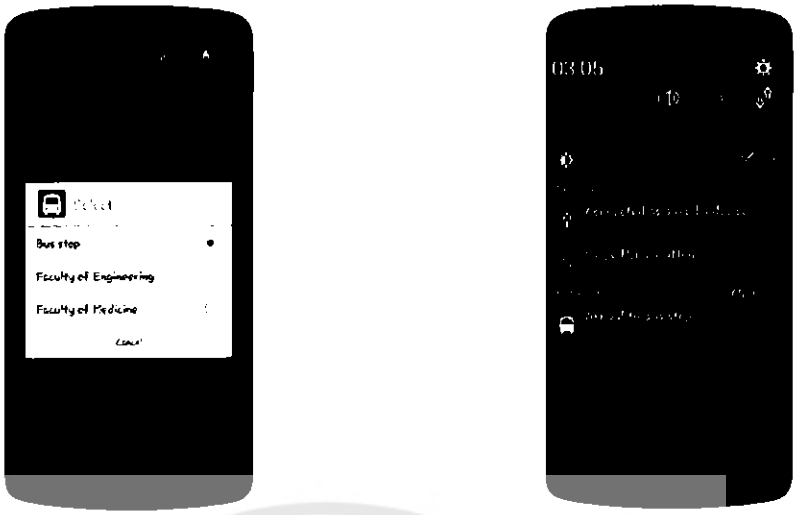


รูปที่ 4.15 การแสดงผลในส่วนของประมาณเวลาบนสมาร์ทโฟน (ซ้าย) แสดงเวลาบนเว็บไซต์ (ขวา) ในกรณีที่รถไฟฟ้าวิ่งป้ายจอดรถ Bus stop

จากรูปที่ 4.15 จะพบว่าป้ายจอดรถ Bus stop ไม่แสดงข้อมูลการประมาณเวลา เนื่องจากรถไฟฟ้าวิ่งผ่านป้ายจอดรถ Bus stop และไม่มีรถไฟฟ้าคันอื่นที่กำลังจะวิ่งมาป้าย Bus stop จึงไม่มีการแสดงเวลา โดยโปรแกรมประยุกต์และเว็บไซต์จะแสดงเวลาของป้ายจอดรถอัตโนมัติ ในกรณีที่มีรถไฟฟ้าที่กำลังจะวิ่งถึงป้ายจอดรถ

4.7.3 ทดสอบระบบแจ้งเตือนอัตโนมัติ

ในการทดสอบระบบแจ้งเตือนอัตโนมัตินั้น ในโปรแกรมประยุกต์ได้มีการเพิ่มการตั้งค่าสำหรับเปิดระบบแจ้งเตือนอัตโนมัติไว้ให้กับผู้ใช้งาน สามารถเลือกป้ายจอดรถได้ว่าต้องการข้อมูลของป้ายจอดรถใด มาแจ้งเตือนในสมาร์ทโฟน โดยเงื่อนไขของระบบแจ้งเตือนอัตโนมัตินั้น จะทำการตรวจสอบว่าเวลาที่รถไฟฟ้าจะมาถึงป้ายจอดรถ น้อยกว่า 60 วินาที ให้มีการแจ้งเตือนผ่านระบบแจ้งเตือนอัตโนมัติ ดังรูปที่ 4.16



รูปที่ 4.16 การตั้งค่าระบบแจ้งเตือนอัตโนมัติโดยเลือกป้ายจอดรถ (ซ้าย)
และการแจ้งเตือนอัตโนมัติของป้ายจอดรถ Bus stop (ขวา)

4.7.4 การทดสอบนำโปรแกรมประยุกต์แสดงผลส่วนติดต่อกับผู้ใช้ไปใช้งาน

จากการทดสอบนำโปรแกรมประยุกต์ไปใช้งานกับกลุ่มตัวอย่างที่มีสมาร์ทโฟน จำนวน 20 คนและ กลุ่มตัวอย่างที่ไม่มีสมาร์ทโฟนจำนวน 20 คน ดังตารางที่ 4.7 พบว่า

ตารางที่ 4.7 ข้อมูลการเปรียบเทียบการใช้งานโปรแกรมประยุกต์แสดงผลส่วนติดต่อกับผู้ใช้

	กลุ่มตัวอย่างที่มีสมาร์ทโฟน (20 คน)	กลุ่มตัวอย่างที่ไม่มีสมาร์ทโฟน (20 คน)
วางแผนการขึ้นรถไฟฟ้าทันเวลา	18 คน	9 คน
ขึ้นรถไฟฟ้าไม่ทันเวลา	2 คน	11 คน

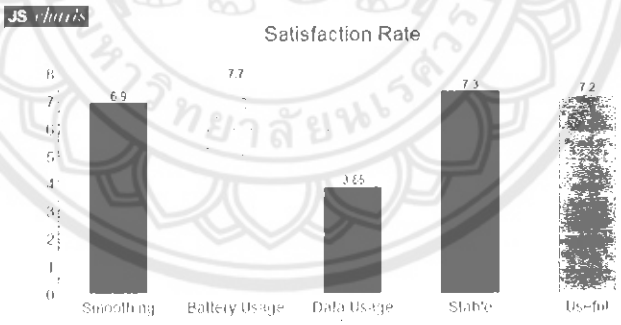
กลุ่มตัวอย่างที่มีสมาร์ทโฟนจำนวน 20 คน สามารถวางแผนการขึ้นรถไฟฟ้าทันเวลาจำนวน 18 คน ซึ่งอีก 2 คนไม่สามารถขึ้นรถไฟฟ้าทันเวลา เนื่องจากอุปกรณ์ที่กลุ่มตัวอย่างทดสอบมีข้อกำหนด (Specification) ที่แตกต่างกัน ยกตัวอย่างเช่น แรม (RAM) น้อยกว่า 512 MB , ซีพียู

(CPU) น้อยกว่า 1 GHz และความเร็วในการเชื่อมต่ออินเทอร์เน็ต ซึ่งอาจจะส่งผลให้โปรแกรมประยุกต์หยุดการทำงาน

กลุ่มตัวอย่างไม่มีสมาร์ทโฟนจำนวน 20 คน สามารถวางแผนขึ้นรถไฟฟ้าทันเวลาจำนวน 9 คน โดยมีการรอรถไฟฟ้าที่ป้ายจอดรถอยู่ตลอดเวลา ซึ่งอีก 11 คน ไม่สามารถขึ้นรถไฟฟ้าทันเวลาเนื่องจากอยู่บนตึกหรืออาคาร จึงทำให้ไม่สามารถเดินทางมาทันรถไฟฟ้า

4.7.5 ความพึงพอใจสำหรับผู้ใช้ในกลุ่มตัวอย่างที่มีสมาร์ทโฟน

จากการสำรวจในกลุ่มตัวอย่างที่มีสมาร์ทโฟนพบว่า ความพึงพอใจ และความตอบสนองของโปรแกรมประยุกต์ มีอยู่ 5 ส่วน คือส่วนที่แรก ด้านความราบเรียบ ใช้งานและไม่รู้สึกสะดุดหรือกระตุก (Smooth) ของโปรแกรมประยุกต์ ส่วนที่สอง การใช้แบตเตอรี่ของโปรแกรมประยุกต์ ส่วนที่สาม การใช้อินเทอร์เน็ตของโปรแกรมประยุกต์ ส่วนที่สี่ ความเสถียร (Stable) ของโปรแกรมประยุกต์ และส่วนสุดท้าย ความมีประโยชน์ (Useful) ในการใช้โปรแกรมประยุกต์

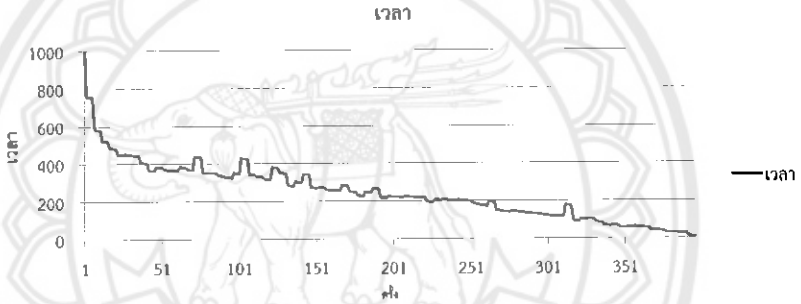


รูปที่ 4.17 กราฟความพึงพอใจสำหรับผู้ใช้ในกลุ่มตัวอย่างที่มีสมาร์ทโฟน

จากรูปที่ 4.17 ผลการสำรวจในกลุ่มตัวอย่างที่มีสมาร์ทโฟนพบว่า คะแนนเฉลี่ยของแต่ละด้านมีค่าสูงและต่ำต่างกัน เนื่องจาก ในกลุ่ม Data usage นั้นค่าจะมีค่าต่ำ โดยค่าที่ได้คือค่าที่ใช้วัดการใช้งานอินเทอร์เน็ต และด้านที่เหลือจะมีค่าสูง โดยค่าที่ได้คือค่าที่ใช้วัดประสิทธิภาพ, พลังงานแบตเตอรี่, เสถียรและประโยชน์ในโปรแกรมประยุกต์

4.8 การทดสอบระบบโดยรวม

นำสมาร์ทโฟนสำหรับระบุตำแหน่งปัจจุบันของรถไฟฟ้าขึ้นบนรถไฟฟ้ามหาวิทยาลัยนเรศวร ขณะวิ่งบริการ และมีการแสดงผลส่วนติดต่อกับผู้ใช้ทั้งบนป้ายจอดรถและสมาร์ทโฟน เพื่อให้ผู้ใช้สามารถติดตามตำแหน่งของรถไฟฟ้า (สำหรับสมาร์ทโฟน) และประมาณเวลาระหว่างรถไฟฟ้ากับป้ายจอดรถ เริ่มขึ้นรถไฟฟ้าบริเวณจุดจอดรถไฟฟ้า (หอพักนิสิต) โดยสมาร์ทโฟนสำหรับระบุตำแหน่งจะเริ่มทำงานทันทีหลังจากมีการเปิดใช้งาน ดังนั้นในส่วนของการติดตามตำแหน่งและประมาณเวลาของรถไฟฟ้า จะทำงานทันทีที่รถไฟฟ้ามีการเคลื่อนที่ โดยในการทดสอบนี้แสดงการประมาณเวลาที่ป้ายจอดรถไฟฟ้าคณะวิศวกรรมศาสตร์ (สามแยกตึกวิศวกรรมไฟฟ้าและคอมพิวเตอร์) ดังรูปที่ 4.18



รูปที่ 4.18 กราฟการแสดงเวลาที่ป้ายจอดรถในขณะที่รถไฟฟ้าเคลื่อนที่

จากการทดสอบพบว่าในขณะที่รถไฟฟ้าเคลื่อนที่เข้าใกล้ป้ายจอดรถไฟฟ้ามากขึ้นเวลาในการแสดงผลที่ป้ายจอดรถจะลดลงเรื่อยๆ ตามอัตราเร็วของรถไฟฟ้า จนกระทั่งรถไฟฟ้ามาถึงป้ายจอดรถเวลาจะแสดงเป็น 00:00 เช่นเดียวกัน ในส่วนของการติดตามตำแหน่งผ่านสมาร์ทโฟนจะแสดงตำแหน่งตามที่รถไฟฟ้ามีการเคลื่อนที่ ทั้งนี้โดยรวมระบบสามารถที่จะติดตามและประมาณเวลาการเดินรถไฟฟ้าได้ใกล้เคียงกับความเป็นจริง การนำสมาร์ทโฟนไว้ภายในรถไฟฟ้าไม่มีผลใดๆ กับการสื่อสารผ่านเครือข่าย 3G หรือดาวเทียมเพื่อระบุตำแหน่งปัจจุบันแต่อย่างใด เว้นแต่เครือข่ายผู้ให้บริการ 3G มีปัญหาหรือมีผู้ใช้งานเครือข่ายจำนวนมาก

4.9 อัตราค่าใช้จ่ายทั้งหมดของระบบ

ในการรับส่งข้อมูลผ่านอินเทอร์เน็ตด้วยเครือข่าย 3G นั้นจำเป็นต้องมีค่าบริการอินเทอร์เน็ต เนื่องจากโปรแกรมประยุกต์จำเป็นจะต้องมีการเชื่อมต่ออินเทอร์เน็ตตลอดเวลา เพื่อติดต่อสื่อสารระหว่างสมาร์ทโฟนกับเครื่องบริการเว็บ ในการส่งตำแหน่งปัจจุบันของโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้าและรับข้อมูลต่างๆ ในโปรแกรมประยุกต์บนป้ายจอดรถไฟฟ้าและสมาร์ทโฟน โดยมีรายละเอียดของค่าบริการดังต่อไปนี้

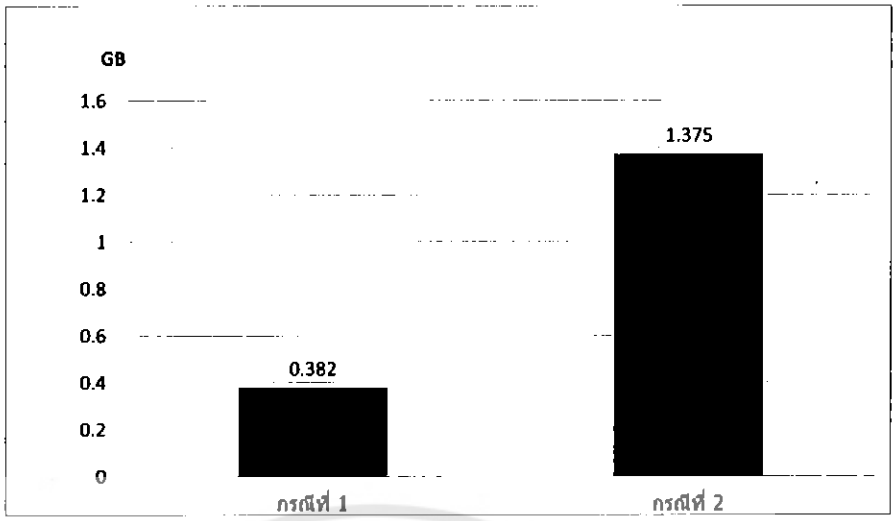
อัตราค่าบริการสำหรับโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้าอาศัยการใช้โปรโมชั่น 3G แบบคิดตามปริมาณซึ่งมีการจ่ายค่าบริการตามปริมาณที่ใช้งานจริง ตามตารางที่ 4.8 และโปรโมชั่นที่ไม่จำกัดการใช้งาน แต่กำหนดความเร็วในการรับส่งข้อมูล 128 Kbps ซึ่งเหมาะสำหรับการรับส่งข้อมูลในโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้าตลอดเวลา ตามตารางที่ 4.9 และโปรแกรมประยุกต์บนป้ายจอดรถไฟฟ้า ตารางที่ 4.10

ตารางที่ 4.8 เปิด/ปิดโปรแกรมประยุกต์ระบุตำแหน่งตามเวลาที่ใช้จริง

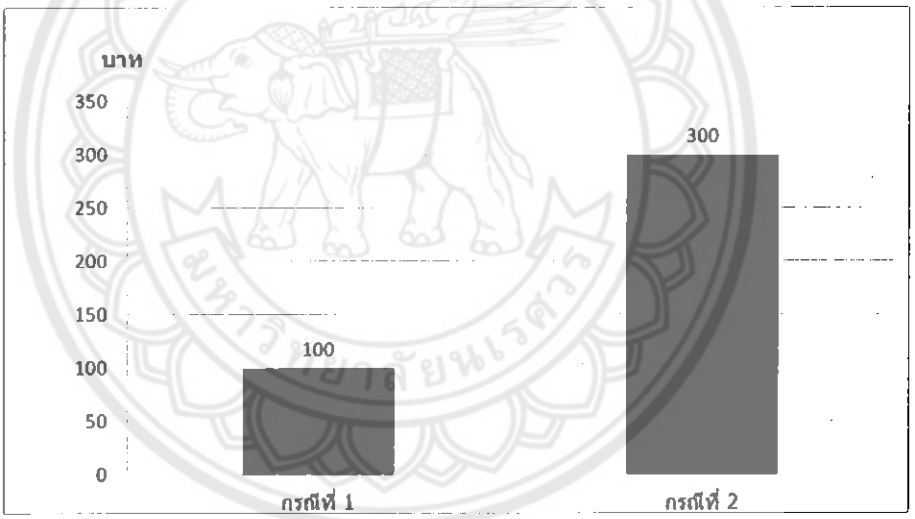
จำนวนการส่งข้อมูล	ปริมาณที่ใช้	หน่วย
1 ครั้ง	1.780	KB
30 ครั้งต่อนาที	53.400	KB
25 นาทีต่อ 1 รอบ	1.304	MB
10 รอบต่อ 1 วัน	13.037	MB
30 วัน	0.382	GB

ตารางที่ 4.9 กรณีที่ 2 เปิดโปรแกรมประยุกต์ระบุตำแหน่งตลอดเวลา

จำนวนการส่งข้อมูล	จำนวนที่ใช้	หน่วย
1 ครั้ง	1.780	KB
30 ครั้งต่อนาที	53.400	KB
1 วันใช้ 15 ชั่วโมง (07:00 - 22:00 น.)	46.934	MB
30 วัน	1.375	GB



รูปที่ 4.19 กราฟเปรียบเทียบปริมาณการใช้ 3G กรณีที่ 1 และ 2



รูปที่ 4.20 กราฟเปรียบเทียบค่าใช้จ่ายกรณีที่ 1 และ 2

จากรูปที่ 4.19 และรูปที่ 4.20 พบว่า ค่าใช้จ่ายทั้ง 2 กรณีมีความแตกต่างกันมาก เนื่องจากกรณีที่ 1 เปิด/ปิดโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้าตามเวลาที่ใช้จริง โดยปิดโปรแกรมประยุกต์ในขณะที่ไม่ได้ใช้งานจึงทำให้ปริมาณการใช้งานอินเทอร์เน็ตรายเดือนปริมาณ 0.382 GB จึงเหมาะสำหรับใช้โปรโมชั่น 3G แบบคิดตามปริมาณซึ่งมีการจ่ายค่าบริการตามปริมาณที่ใช้งานจริงโดยมีค่าใช้จ่าย 100 บาท (500 MB) และกรณีที่ 2 เปิดโปรแกรมประยุกต์ระบุตำแหน่ง

ปัจจุบันของรถไฟฟ้าตลอดเวลา พบว่าปริมาณการใช้งานอินเทอร์เน็ตรายเดือนมีเพิ่มขึ้นเป็น 1.375 GB จึงเหมาะสำหรับใช้โปรโมชั่นที่ไม่จำกัดการใช้งาน แต่กำหนดความเร็วในการรับส่งข้อมูล 128 Kbps โดยมีค่าใช้จ่าย 300 บาท

ตารางที่ 4.10 อัตราค่าบริการสำหรับโปรแกรมประยุกต์ส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ 1 ป้าย

จำนวนการรับข้อมูล	จำนวนที่ใช้	หน่วย
1 ครั้ง	6.320	KB
12 ครั้งต่อนาที	75.840	KB
1 วัน (07:00 - 22:00 น.)	66.656	MB
30 วัน	1.953	GB

จากตารางที่ 4.8 ถึงตารางที่ 4.10 พบปริมาณการใช้งานอินเทอร์เน็ตรายเดือนปริมาณขนาดมาก ซึ่งค่าบริการอินเทอร์เน็ต 3G ปัจจุบันมีค่าบริการที่ราคาถูกและใช้งานได้ไม่จำกัด ดังนั้นจึงเหมาะสำหรับนำมาใช้ในโปรแกรมยุกต์ เพื่อลดต้นทุนของค่าบริการและยังสามารถใช้งานได้เช่นเดียวกับโปรโมชั่นอื่นๆ ดังนั้นค่าบริการรายเดือนสามารถจ่ายในราคา 300 บาท ต่อ 1 ซิม และยังสามารถใช้งานอินเทอร์เน็ตได้ไม่จำกัดโดยมีการกำหนดความเร็วไว้ที่ 128 Kbps

4.9.1 ค่าใช้จ่ายสำหรับสมาร์ทโฟนระบุตำแหน่งปัจจุบันของรถไฟฟ้า

1. สมาร์ทโฟนแอนดรอยด์	ราคา	1,300	บาท
รวม	ราคา	1,300	บาท

4.9.2 ค่าอุปกรณ์ ณ ป้ายจอดรถไฟฟ้า

1. กล่องใส่อุปกรณ์	ราคา	70	บาท
2. ส่วนแสดงผล Seven Segment 4 ชุด	ราคา	100	บาท
3. วงจรแปลงเลข BCD - Seven Segment	ราคา	300	บาท

4. สมาร์ทโฟนแอนดรอยด์ ราคา 1,300 บาท
5. IOIO บอร์ด ราคา 1,950 บาท
6. ตู้ป้ายจอดรถไฟฟ้า ราคา 8,280 บาท

รวม ราคา 12,000 บาท

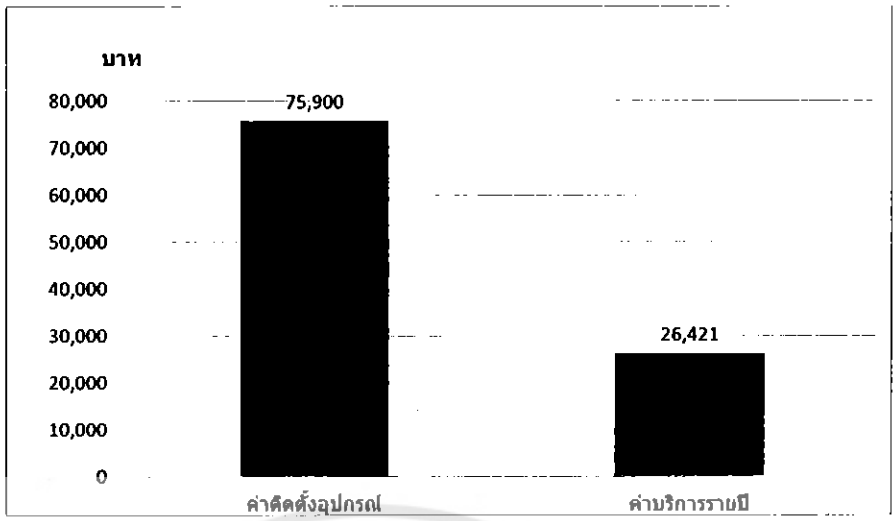
ค่าใช้จ่ายทั้งหมดของระบบนั้นมาจาก 2 ส่วนคือ ส่วนแรก ค่าใช้จ่ายสำหรับติดตั้งอุปกรณ์ในครั้งแรก ซึ่งประกอบไปด้วย สมาร์ทโฟนระบุตำแหน่งปัจจุบันของรถไฟฟ้า สมาร์ทโฟนแสดงเวลา ณ ป้ายจอดรถไฟฟ้าและอุปกรณ์ต่างๆ ส่วนที่สอง ค่าบริการอินเทอร์เน็ต 3G รายเดือน และค่าเช่าเครื่องบริการเว็บ ดังตารางที่ 4.11 และ 4.12 ตามลำดับ

ตารางที่ 4.11 ค่าใช้จ่ายสำหรับติดตั้งอุปกรณ์ในครั้งแรก

รายการ	ราคา (บาท)	จำนวน	ราคารวม (บาท)
ค่าใช้จ่ายสำหรับสมาร์ทโฟนระบุตำแหน่งปัจจุบันของรถไฟฟ้า	1,300	3 เครื่อง	3,900
ค่าใช้จ่ายสำหรับอุปกรณ์ ณ ป้ายจอดรถไฟฟ้า	12,000	6 ป้าย	72,000
รวม	75,900 บาท		

ตารางที่ 4.12 ค่าใช้จ่ายสำหรับอินเทอร์เน็ต 3G และเครื่องบริการเว็บ**รายปี**

รายการ	ราคารายเดือน (บาท)	จำนวน	ราคารายปี (บาท)
ค่าบริการอินเทอร์เน็ต 3G สำหรับป้ายจอดรถไฟฟ้า	300	6 หมายเลข	21,600
ค่าบริการอินเทอร์เน็ต 3G สำหรับระบุตำแหน่งปัจจุบัน	100	3 หมายเลข	3,600
ค่าเช่าเครื่องบริการเว็บ	-	1 เครื่อง	1,221
รวม	26,421 บาท		



รูปที่ 4.21 กราฟค่าใช้จ่ายสำหรับการติดตั้งอุปกรณ์และค่าบริการรายปี

จากรูปที่ 4.21 ค่าใช้จ่ายสำหรับการติดตั้งอุปกรณ์โดยมีสมาร์ทโฟนระบุตำแหน่งปัจจุบันของรถไฟฟ้า 3 เครื่อง ป้ายจอดรถไฟฟ้า 6 ป้าย ซึ่งจะต้องเสียค่าติดตั้งในครั้งแรกราคา 75,900 บาทและค่าบริการรายปี ประกอบด้วยค่าบริการอินเทอร์เน็ต 3G และค่าเช่าเครื่องบริการเว็บ ซึ่งจะมีค่าใช้จ่ายรายปี ปีละ 26,421 บาท

จากการทดสอบระบบในแต่ละส่วนตั้งแต่ระบบระบุตำแหน่งปัจจุบันของรถไฟฟ้าและระบบติดตามตำแหน่งและประมาณเวลาของรถไฟฟ้า พบว่าการทำงานโดยรวมของระบบทั้งหมด มีการทำงานตามที่ได้ออกแบบและทำงานร่วมกับระบบอื่นๆ ได้อย่างมีประสิทธิภาพ ทั้งนี้การทดสอบทำให้ทราบถึงปัญญา แนวทางการแก้ไขปัญหาและแนวทางในการพัฒนาต่อ โดยจะอธิบายไว้ในบทถัดไป

บทที่ 5

สรุปผลและข้อเสนอแนะ

การพัฒนาระบบติดตามและประมาณเวลาการเดินรถไฟฟ้าด้วยสมาร์ทโฟนผ่านเครือข่าย 3G มาใช้แทนเครือข่าย Zigbee เพื่อลดการติดตั้งของเราเตอร์ของระบบเครือข่าย Zigbee โดยสาเหตุที่เลือกใช้ระบบเครือข่าย 3G เนื่องจากมีสัญญาณครอบคลุมทั่วประเทศและสามารถใช้บริการได้ทันทีที่มีสัญญาณและใช้งานสมาร์ทโฟนให้เกิดประโยชน์มากขึ้น

การศึกษาหลักการและทฤษฎีที่เกี่ยวข้องนั้น เพื่อให้เข้าใจหลักการทำงานเบื้องต้นและนำไปปฏิบัติ โดยสามารถศึกษาเพิ่มเติมจากแหล่งอ้างอิง เพื่อให้เข้าใจมากขึ้นและนำไปประยุกต์ในการพัฒนาระบบได้

การนำหลักการและทฤษฎีต่างๆ ที่เกี่ยวข้องมาวิเคราะห์ ออกแบบและประยุกต์การใช้งานของระบบ เพื่อให้ระบบโดยรวมมีความสามารถในการระบุตำแหน่งและประมาณเวลา ทั้งนี้การวิเคราะห์และออกแบบระบบให้มีความสามารถดังกล่าว จำเป็นต้องมีการแบ่งการทำงานออกเป็นส่วนย่อย เพื่อพัฒนาในแต่ละส่วนให้สามารถทำงานร่วมกันได้

ในการทดสอบระบบโดยรวม พบว่ากลุ่มตัวอย่าง 20 คน สามารถขึ้นรถไฟฟ้าได้ทันตามที่โปรแกรมประยุกต์ระบุ 18 คน โดยที่ผู้ใช้สามารถติดตามตำแหน่งและเวลาโดยประมาณของรถไฟฟ้ากับป้ายจอดรถ ดังนั้นสรุปได้ว่าการมีระบบดังกล่าว สามารถทำให้ผู้ใช้งานรถไฟฟ้าสามารถวางแผนการขึ้นรถไฟฟ้าได้ทันเวลา

5.1 สรุปผลการทดสอบ

จากการทดสอบพบว่าในขณะที่รถไฟฟ้าเคลื่อนที่เข้าใกล้ป้ายจอดรถไฟฟ้ามากขึ้นเวลาในการแสดงผลที่ป้ายจอดรถจะลดลงเรื่อยๆ ตามอัตราเร็วของรถไฟฟ้า จนกระทั่งรถไฟฟ้ามาถึงป้ายจอดรถเวลาจะแสดงเป็น 00:00 เช่นเดียวกับในส่วนของการติดตามตำแหน่งผ่านสมาร์ทโฟนจะแสดงตำแหน่งตามที่รถไฟฟ้ามีการเคลื่อนที่ ทั้งนี้โดยรวมระบบสามารถที่จะติดตามและประมาณเวลาการ

เดินรถไฟฟ้าได้ใกล้เคียงกับความเป็นจริง การนำสมาร์ทโฟนไว้ภายในรถไฟฟ้าไม่มีผลใดๆ กับการสื่อสารผ่านเครือข่าย 3G หรือดาวเทียมเพื่อระบุตำแหน่งปัจจุบันแต่อย่างใด เว้นแต่เครือข่ายผู้ให้บริการ 3G มีปัญหาหรือมีผู้ใช้งานเครือข่ายจำนวนมาก

จากการทดสอบโปรแกรมประยุกต์ระบุตำแหน่งปัจจุบันของรถไฟฟ้า ความแม่นยำในการระบุตำแหน่งนั้นมีค่าความผิดพลาดอยู่ในระดับที่สามารถมองเห็นได้ด้วยสายตา ปัจจัยที่ก่อให้เกิดความผิดพลาด เช่น เครือข่ายอินเทอร์เน็ตมีผู้ใช้บริการจำนวนมาก โปรแกรมประยุกต์รับค่า GPS จากเสาสัญญาณของเครือข่ายอินเทอร์เน็ต ดังนั้นค่าความผิดพลาดของแต่ละสถานที่จะแตกต่างกันขึ้นอยู่กับสภาพแวดล้อมของแต่ละสถานที่

จากการทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนป้ายจอดรถ พบว่าการแสดงผลบน Seven Segment นั้นได้รับข้อมูลมาจากสมาร์ทโฟนโดยตรง ดังนั้นการแสดงผลจะมีความผิดพลาดอาจจะเกิดจากสมาร์ทโฟนหรือ IOIO บอร์ดหรือวงจร Seven Segment เท่านั้น

จากการทดสอบแสดงผลส่วนติดต่อกับผู้ใช้บนสมาร์ทโฟน พบว่าการแสดงผลในรูปแบบแผนที่ มีความแม่นยำ และในส่วนแสดงผลในรูปแบบประมาณเวลามีค่าใกล้เคียงกับความเป็นจริง โดยสมาร์ทโฟนมีการร้องขอข้อมูลทุกๆ 5 วินาที โดยทำการนับถอยหลังในช่วงเวลาที่รอการร้องขอ

5.2 เปรียบเทียบระบบติดตามรถไฟฟ้ามหาวิทยาลัยนเรศวรแต่ละรุ่น

ในอดีตระบบติดตามรถไฟฟ้ามหาวิทยาลัยนเรศวรนั้นมีการพัฒนาระบบมาแล้ว 3 รุ่น โดยประกอบไปด้วย ระบบเครือข่าย ALOHA Network, ระบบเครือข่าย Zigbee และปัจจุบันระบบเครือข่าย 3G โดยมีการเปรียบเทียบคุณลักษณะของการทำงานในแต่ละด้าน ดังตารางที่ 5.1

ตารางที่ 5.1 เปรียบเทียบระบบติดตามรถไฟฟ้ามหาวิทยาลัยนเรศวรแต่ละรุ่น

รุ่นที่	1	2	3
รูปแบบการสื่อสาร	ALOHA Network	Zigbee	3G
ช่องสัญญาณ	Distributed	WPAN	WWAN
ความถี่	2.4 GHz	2.4 GHz	850/900/2100 MHz
ระบบติดตามรถไฟฟ้า	ป้ายจอดรถ	เว็บไซต์	เว็บไซต์/สมาร์ทโฟน
ระบบประมาณเวลา	ป้ายจอดรถ	ป้ายจอดรถ	เว็บไซต์/ป้ายจอดรถ/สมาร์ทโฟน
ระบบแจ้งเตือนอัตโนมัติ	-	-	สมาร์ทโฟน
ระยะทางการสื่อสาร	0.2 – 0.5 กิโลเมตร	0.6 – 0.9 กิโลเมตร	5 - 8 กิโลเมตร
จำนวนโหนด	19	11	0
ค่าใช้จ่าย 3 ปี *	217,492 บาท	210,671 บาท	183,521 บาท **

* อ้างอิงจากภาคผนวก ค

** อ้างอิงราคาของ 3G จาก บริษัท ไอโมบาย-พลัส จำกัด

ข้อมูล ณ วันที่ 3 ธันวาคม 2556

จากตารางที่ 5.1 พบว่าระบบที่ใช้เครือข่าย 3G มีระบบที่รองรับการทำงานได้มากกว่า 1 รูปแบบ เช่น ระบบติดตามรถไฟฟ้า สามารถติดตามได้ทั้งเว็บไซต์และสมาร์ทโฟน ระบบประมาณเวลา สามารถตรวจสอบเวลาได้จากเว็บไซต์, ป้ายจอดรถและสมาร์ทโฟน และระบบแจ้งเตือนอัตโนมัติ ซึ่งไม่มีในระบบเดิม โดยรองรับการทำงานบนสมาร์ทโฟน ดังนั้นจะเห็นว่าเครือข่าย 3G สามารถส่งสัญญาณได้ระยะทางที่ไกลถึง 5-8 กิโลเมตร และไม่มีการวางโหนด เพื่อลดค่าใช้จ่ายในส่วนของการวางโหนด ซึ่งจะเห็นว่ามีค่าใช้จ่ายต่ำกว่าเครือข่ายอื่น

5.3 ปัญหาในการดำเนินงานและแนวทางการแก้ไข

จากผลการทดสอบ ทำให้พบปัญหาในการดำเนินงานซึ่งปัญหาที่พบนั้นทำให้ระบบมีความผิดพลาดหรือทำงานได้ไม่เต็มประสิทธิภาพตามที่วางแผนไว้ ดังนั้นจึงเสนอแนวทางการแก้ไขปัญหาไว้เพื่อเป็นแนวทางในการแก้ไขและปรับปรุงระบบให้มีประสิทธิภาพ ดังตารางที่ 5.2

ตารางที่ 5.2 ปัญหาในการดำเนินงานและแนวทางการแก้ไข

ปัญหาที่พบ	แนวทางแก้ไขปัญหา
1. ระบบจะไม่สามารถใช้งานได้ในกรณีที่เครื่องบริการเว็บมีปัญหา เนื่องจากข้อมูลทั้งหมดถูกจัดเก็บไว้ในเครื่องบริการเว็บ และการสื่อสารของแต่ส่วนทำงานผ่านเครื่องบริการเว็บ	1. เลือกใช้งานเครื่องบริการเว็บที่มีความเสถียรและมีการจัดเก็บข้อมูลสำรอง
2. สมาร์ทโฟนมีใช้งานแบตเตอรี่จำนวนมากหากมีการใช้งานหน้าจอ จึงทำส่งผลกระทบกับระบบเนื่องจากสมาร์ทโฟนเป็นอุปกรณ์ที่ดำเนินการรับส่งข้อมูล	2. พัฒนาโปรแกรมประยุกต์ให้มีการทำงานเป็น Background Service โดยไม่จำเป็นต้องเปิดหน้าจอ
3. การรับสัญญาณจาก GPS จำเป็นต้องมีการเชื่อมต่ออินเทอร์เน็ต	3. เลือกใช้สมาร์ทโฟนที่มี GPS ในการรับสัญญาณจากดาวเทียมได้โดยตรง
4. สภาพแวดล้อมมีผลต่อการรับสัญญาณ GPS จากดาวเทียม	4. เลือกรับสัญญาณ GPS จากเสาสัญญาณผู้ให้บริการเครือข่าย 3G

5.4 ข้อเสนอแนะ

ในการพัฒนาระบบให้มีประสิทธิภาพหรือเปลี่ยนรูปแบบการสื่อสารเพื่อให้ระบบมีประสิทธิภาพนั้น ได้มีการเสนอแนะและวิธีการ เพื่อให้ระบบมีประสิทธิภาพและแก้ไขปัญหาที่พบจากตารางที่ 5.2 ดังนั้นจึงมีการเสนอแนะและวิธีการไว้เพื่อให้ผู้ที่มาพัฒนาระบบในอนาคตทราบถึงปัญหาของระบบ และพัฒนาแก้ไขปัญหาดังกล่าวให้หมดไป ดังตารางที่ 5.3

ตารางที่ 5.3 ข้อเสนอแนะและวิธีการ

ข้อเสนอแนะ	วิธีการ
1. พัฒนาระบบติดต่อกับผู้ใช้บนสมาร์ทโฟนให้รองรับระบบปฏิบัติการอื่นๆ	1. พัฒนาโปรแกรมประยุกต์ด้วย PhoneGap ผ่าน API ที่มีการพัฒนาขึ้นมา
2. พัฒนาการสื่อสารข้อมูลให้สามารถทำงานร่วมกันได้ระหว่างระบบเครือข่ายไร้สาย Zigbee และ 3G	2. ใช้เครือข่ายไร้สาย Zigbee ในการแสดงเวลาระหว่างรถไฟฟ้าและป้ายจอดรถ โดยใช้เครือข่าย 3G ในการแสดงผลติดตามตำแหน่งและส่งตำแหน่งปัจจุบัน
3. เพิ่มการเข้ารหัสในการรับส่งข้อมูล แทนการใช้คีย์ในการตรวจสอบข้อมูล	3. ใช้อัลกอริทึมการเข้ารหัสกับข้อมูลที่รับส่งระหว่างสมาร์ทโฟนกับเครื่องบริการเว็บ
4. พัฒนาระบบบนเว็บไซต์ เพื่อจัดการ เพิ่ม ลบ ป้ายจอดรถ	4. ใช้เว็บไซต์ในการจัดการเพิ่ม ลบ ป้ายจอดรถ โดยรองรับการเพิ่มป้ายจอดรถได้ง่าย

5.5 ความรู้ที่จำเป็นต่อการพัฒนาต่อ

1. พื้นฐานในการพัฒนาโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์
2. พื้นฐานในการพัฒนา IOIO บอร์ด
3. การใช้วงจรพื้นฐาน เช่น BCD - Seven Segment
4. พื้นฐานในการพัฒนาเว็บไซต์ เช่น PHP, MySQL, HTML, Javascript
5. พื้นฐานในการเข้ารหัสและถอดรหัสของ JSON

บรรณานุกรม

[1] Manuel Baumann, Bálint Simon, Hanna Dura and Marcel Weil (2012). The contribution of electric vehicles to the changes of airborne emissions. IEEE International Energy Conference and Exhibition (ENERGYCON)

[2] **จุฬาทิพย์ สุพรมมา.,ยุทธชัย อยู่บำรุง. (2554). การติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย. พิษณุโลก: มหาวิทยาลัยนเรศวร**

[3] เหมวิจักษณ์ บุญมาก.,วิชิต ธรรมพิทักษ์. (2554). ระบบเครือข่ายและแสดง**ผลสำหรับติดตาม**รถไฟฟ้า. พิษณุโลก: มหาวิทยาลัยนเรศวร

[4] วิศวะ นามวงษ์.,วันเฉลิม จันทร์ทรง. (2555). ระบบติดตามและประ**มาณเวลาการเดิน**รถไฟฟ้าด้วยเครือข่าย Zigbee. พิษณุโลก: มหาวิทยาลัยนเรศวร

[5] วิกิพีเดีย. Smartphone. สืบค้นเมื่อ 26 มีนาคม 2556, จาก http://en.wikipedia.org/wiki/Smartphone

[6] พร้อมเลิศ หล่อวิจิตร., (2555). คู่มือเขียนแอพ Android สำหรับผู้เริ่ม**ต้น. กรุงเทพฯ : โปร**วิชั่น

[7] Nexus 5, Retrieved November 5, 2013, สืบค้นเมื่อ 27 มีนาคม 2556, จาก http://en.wikipedia.org/wiki/File:Nexus_5_Front_View.png

[8] Processes and Threads, สืบค้นเมื่อ 30 มีนาคม 2556, จาก http://developer.android.com/guide/components/processes-and-threads.html

[9] Process and Threads, สืบค้นเมื่อ 2 เมษายน 2556, จาก http://bit.ly/1bG68nc

[10] วิกิพีเดีย. Multithreaded process, สืบค้นเมื่อ 14 เมษายน 2556, จาก http://en.wikipedia.org/wiki/File:Multithreaded_process.svg

[11] วิกิพีเดีย. Process (computing), สืบค้นเมื่อ 15 เมษายน 2556, จาก http://en.wikipedia.org/wiki/Process_(computing)

[12] วิกิพีเดีย. Thread (computing), สืบค้นเมื่อ 15 เมษายน 2556, จาก http://en.wikipedia.org/wiki/Thread_(computing)

ภาคผนวก ก

ผลงานวิจัยเผยแพร่ ประเภทวารสาร

ที่	ชื่อ	ชื่อบทความ	ชื่อวารสาร	Year
1	เศรษฐา ตั้งค้าวานิช วันเฉลิม จันทร์ทรง วิศวะ นามวงษ์ จักรกฤษ เติมฤทธิกุล กฤษณะ วทานิยานนท์ สุวิทย์ กิระวิทยา	Arrival Time Approximation and Tracking of NU-EV via Wireless Network	Naresuan University Engineering Journal [วารสารระดับชาติ]	2013

ผลงานวิจัยเผยแพร่ ประเภทนำเสนอในที่ประชุมวิชาการ

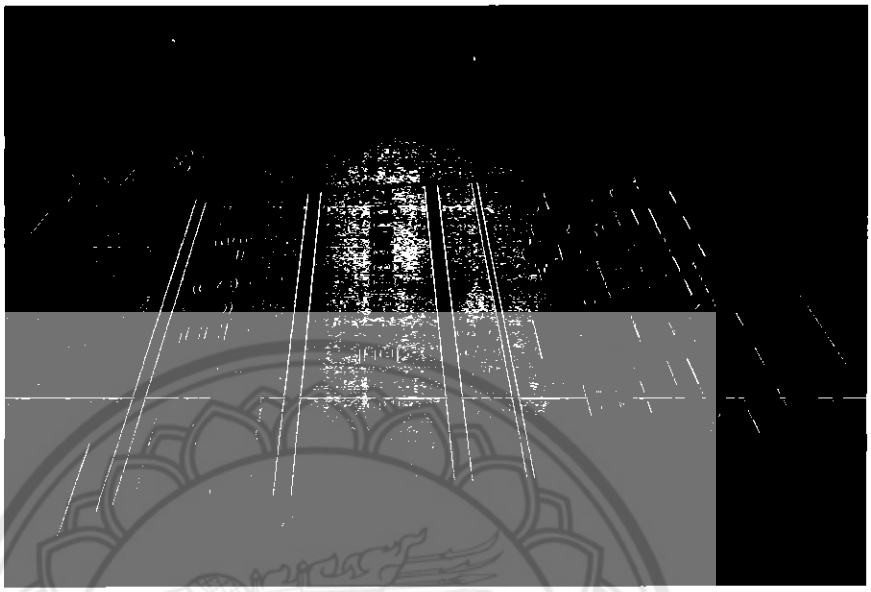
ที่	ชื่อ	ชื่อบทความ	ชื่อการประชุม	ปี	สถานที่จัด
1	Settha Tangkawanit, Krisana Wataniyanon, and Surachet Kanprachar .	"An Improvement of EV Data Updating Time using Fuzzy Logic", Special Sessions: Smart Technology in Campus Electric Vehicle System, The International Conference On Communications	The International Conference On Communications, IEECON 2014 [ระดับนานาชาติ]	2014	Pattaya City, Thailand
2	Settha Tangkawanit, Chakkrit Termrittikun, and Surachet Kanprachar	. "Electric Vehicle Tracking and Notification Application for Smart Phones", Special Sessions: Smart Technology in Campus Electric Vehicle System	Communications, IEECON 2014 [ระดับนานาชาติ]	2014	Pattaya City, Thailand

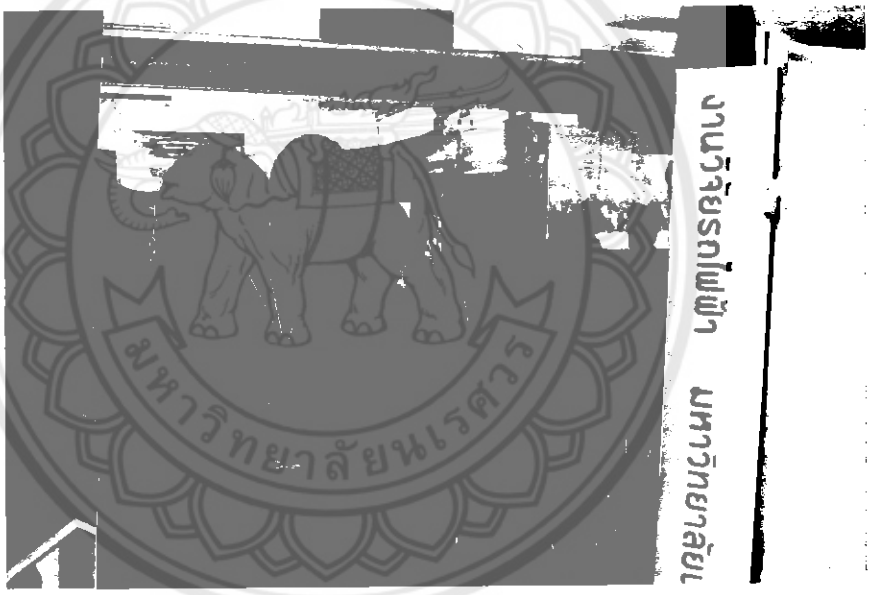
ภาคผนวก ข

การแสดงผลงานด้านวิชาการ

1. OPEN HOUSE 20 ปี คณะวิศวกรรมศาสตร์ 21-23 พฤษภาคม 2556 ณ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

2. งานเปิดโลกวิศวกรรมศาสตร์แห่งประเทศไทยครั้งที่ 6 ณ มหาวิทยาลัยนเรศวร 11 พฤศจิกายน 2556





มหาวิทยาลัยนเรศวร

ภาคผนวก ค

ราคาซื้อเริ่มต้นของระบบ ALOHA Network

	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม
[redacted]									
EVStopStation	12,000	6	72,000						
[redacted]	4,000	6	24,000						
[redacted]	6,600	3	19,800						
[redacted]	500	3	1,500						
[redacted]	3,000	1	3,000						
[redacted]	31,000	1	31,000						
[redacted]	1,221	1	1,221						
รวม			152,521						

ราคาบำรุงรักษาของระบบ ALOHA Network

	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม
[redacted]									
EVStopStation(MCU)				550	2	1,100	550	3	1,650
EVStopStation(7-Seg)				850	6	5,100	850	6	5,100
EVStopStation(Transceiver1)				1,800	2	3,600	1,800	3	5,400
EVStopStation(Transceiver2)				1,800	2	3,600	1,800	3	5,400
[redacted]				1,800	2	3,600	1,800	2	3,600
[redacted]				550	3	1,650	550	3	1,650
[redacted]				1,800	3	5,400	1,800	3	5,400
[redacted]				250	3	750	250	3	750
[redacted]				500	3	1,500	500	3	1,500
[redacted]				1,800	1	1,800	1,800	1	1,800
[redacted]				1,000	1	1,000	1,000	1	1,000
[redacted]				1,221	1	1,221	1,400	1	1,400
รวม						฿30,321.00			฿34,650.00

ราคาซื้อเริ่มต้นของระบบ Zigbee

	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม
EVStopStation	10,000	6	60,000						
	2,800	6	16,800						
	6,600	3	19,800						
	500	3	1,500						
	3,000	1	3,000						
	31,000	1	31,000						
	1,221	1	1,221						
รวม			133,321						

ราคาบำรุงรักษาของระบบ Zigbee

	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม
EVStopStation(MCU)				550	2	1,100	550	3	1,650
EVStopStation(7-Seg)				850	6	5,100	850	6	5,100
EVStopStation(Xbee)				2,000	2	4,000	2,000	2	4,000
				2,000	2	4,000	2,000	2	4,000
				550	6	3,300	550	6	3,300
				2,000	6	12,000	2,000	6	12,000
				250	6	1,500	250	6	1,500
				500	6	3,000	500	6	3,000
				2,000	1	2,000	2,000	1	2,000
				1,000	1	1,000	1,000	1	1,000
				1,400	1	1,400	1,400	1	1,400
รวม						฿38,400.00			฿38,950.00

ราคาซื้อเริ่มต้นของระบบ 3G

	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม
EVStopStation	12,000	6	72,000						
EVStopStation:Top-upmoney	3,600	6	21,600						
	1,300	3	3,900						
	1,200	3	3,600						
	1,221	1	1,221						
รวม			102,321						

ราคาบำรุงรักษาของระบบ 3G

	[redacted]			[redacted]			[redacted]		
	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม	ราคาต่อหน่วย	จำนวน	รวม
[redacted]									
EVStopStation(Smartphone)				2,000	2	4,000	2,000	3	6,000
EVStopStation(7-Seg)				850	6	5,100	850	6	5,100
EVStopStation:Top-upmoney				3,600	6	21,600	3,600	6	21,600
[redacted]				1,300	3	3,900	1,300	3	3,900
[redacted]				1,200	3	3,600	1,200	3	3,600
[redacted]				1,400	1	1,400	1,400	1	1,400
รวม						฿39,600.00			฿41,600.00