ระบบติดตามรถไฟสายตะวันออกอัจฉริยะ เวอร์ชัน 4.0
INTELLIGENT TRAIN TRACKING SYSTEM FOR EASTERN RAIL ROUTE VERSION 4.0

โดย

นางสาวทักษณี เจียรวิบูลย์สุข

นางสาวธัญญพร เล้าจิรเชฐกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ระบบติดตามรถไฟสายตะวันออกอัจฉริยะ เวอร์ชัน 4.0

INTELLIGENT TRAIN TRACKING SYSTEM FOR EASTERN RAIL ROUTE VERSION 4.0

โดย

นางสาวทักษณี	เจียรวิบูลย์สุข	57010516
นางสาวธัญญพร	เล้าจิรเชฐกุล	57010607

อาจารย์ที่ปรึกษา

ศ.ดร.พรชัย ทรัพย์นิธิ

ดร.เวธิต ภาคย์พิสุทธิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

10/5/61

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

21/5/61

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบติดตามรถไฟสายตะวันออกอัจฉริยะ เวอร์ชัน 4.0
INTELLIGENT TRAIN TRACKING SYSTEM FOR EASTERN RAIL ROUTE VERSION 4.0

ผู้จัดทำ

1.	นางสาวทักษณี	เจียรวิบูลย์สุข	57010516
2.	นางสาวธัญญพร	เล้าจิรเชฐกุล	57010607

..... อาจารย์ที่ปรึกษา

(ศ.ดร.พรชัย ทรัพย์นิธิ)

..... อาจารย์ที่ปรึกษาร่วม

(ดร.เวธิต ภาคย์พิสุทธิ์)

กิตติกรรมประกาศ

ปริญญานิพนธ์ “ระบบติดตามรถไฟสายตะวันออกอัจฉริยะ เวอร์ชัน 4.0” สำเร็จลุล่วงได้ด้วยความกรุณาและความช่วยเหลือจากอาจารย์ที่ปรึกษา ศ.ดร. พรชัย ทรัพย์นิธิ และ ดร. เวธิต ภาคย์พิสุทธิ์ รวมทั้งนักศึกษาปริญญาเอก คุณศิรวิชญ์ กิตติวิชญกุล ที่ได้ให้คำปรึกษา คำแนะนำ และตรวจสอบแก้ไขข้อบกพร่องในการจัดทำปริญญานิพนธ์ รวมทั้งนักศึกษาปริญญาโทและปริญญาเอกภาควิชาวิศวกรรมโทรคมนาคมของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และการรถไฟแห่งประเทศไทย รวมถึงพนักงานรักษารถไฟในการสนับสนุนในการทำปริญญานิพนธ์ ทางคณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ บิดา มารดา เพื่อนนักศึกษา ตลอดจนผู้ที่เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ ที่ได้ให้กำลังใจและมีส่วนช่วยเหลือให้ปริญญานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี

คณะผู้จัดทำหวังว่าปริญญานิพนธ์นี้จะเป็นประโยชน์กับผู้สนใจไม่มากก็น้อย

นางสาวทักษณี เจียรวิบูลย์สุข
นางสาวธัญญพร เล้าจิรเชฐกุล
ผู้จัดทำ

ระบบติดตามรถไฟสายตะวันออกอัจฉริยะ เวอร์ชัน 4.0
INTELLIGENT TRAIN TRACKING SYSTEM FOR
EASTERN RAIL ROUTE VERSION 4.0

โดย นางสาวทักษณี เจียรวิบูลย์สุข 57010516
นางสาวธัญญพร เล้าจิรเชฐกุล 57010607

อาจารย์ที่ปรึกษา ศ.ดร.พรชัย ทรัพย์นิธิ
ดร.เวธิต ภาคย์พิสุทธิ์

บทคัดย่อ

ระบบติดตามรถไฟได้ถูกสร้างขึ้นเพื่อแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นกับการเดินทางด้วยรถไฟในสายตะวันออก ยกตัวอย่างเช่น บ่อยครั้งที่รถไฟในสายนี้มักมาถึงสถานีต่าง ๆ ไม่ตรงตามตารางเวลารถไฟ ส่งผลให้ผู้ที่ใช้บริการเสียเวลาหรือพลาดการใช้บริการรถไฟได้ ฉะนั้นในงานก่อนหน้า ได้มีการพัฒนาระบบติดตามรถไฟสายตะวันออกเวอร์ชัน 4.0 เพื่อแสดงผลตำแหน่งรถไฟบนเว็บไซต์และแอปพลิเคชันบนระบบปฏิบัติการไอโอเอส อีกทั้งยังมีการสร้างระบบในการวัดปริมาณแบตเตอรีคงเหลือของอุปกรณ์ติดตามรถไฟ เพื่อแจ้งเตือนสถานะแบตเตอรีแก่ผู้ดูแลระบบ อีกหนึ่งปัญหาที่มักเกิดขึ้นของระบบติดตามรถไฟเวอร์ชันก่อนหน้าคือผู้ใช้งานส่วนใหญ่ไม่ทราบเวลาที่รถไฟถึงสถานีต่าง ๆ เนื่องจากในแอปพลิเคชันจะแสดงเฉพาะตำแหน่งของรถไฟโดยคร่าวเท่านั้น ดังนั้นในงานนี้ ผู้วิจัยจึงได้พัฒนาระบบประมาณเวลาถึงสถานีของรถไฟ เพื่อให้ผู้ใช้งานได้ทราบเวลาที่รถไฟจะถึงสถานีต่าง ๆ และเป็นประโยชน์สำหรับการวางแผนเดินทาง นอกจากนี้ ผู้วิจัยยังพัฒนาระบบการคำนวณตำแหน่งของรถไฟบนระบบฐานข้อมูล เพื่อลดภาระการทำงานของแอปพลิเคชันและเว็บไซต์ ในส่วนสุดท้าย ผู้วิจัยยังได้มีการพัฒนาและปรับปรุงส่วนการแสดงผลบนเว็บไซต์และแอปพลิเคชันบนระบบปฏิบัติการไอโอเอสเพิ่มเติม เพิ่มฟังก์ชันการทำงานของระบบติดตามรถไฟให้มีความหลากหลายและสะดวกสบายมากขึ้น

ABSTRACT

The train tracking system has been built for the trains in eastern rail route to solve the existing problems. For example, the trains are often not on time bringing about the wast of time and the missed train service. In this work, we developed the train tracking systems version 4.0 to display the train positions on web and iOS applications, and create the battery notification for administration. One more problem is that many users do not know the arrival times of train for any station since the applications show only the train positions. Therefore, in this work, we develop the time estimation system to predict the arrival times of train for any station, useful for travel planning. Moreover, the calculation of train positions are performed on the database instead of the web and mobile applications to mitigate the application load. Finally, we further improve and develop the web and iOS applications to increase the usability functions of train tracking system.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		IV
สารบัญรูป		VII
สารบัญตาราง		XII
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	2
	1.3 ขอบเขตของปริญญานิพนธ์	2
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**4**
	2.1 GPS	4
	2.2 บอร์ด Arduino	6
	2.3 GPS Module Ublox NEO 6M	8
	2.4 3G Module	9
	2.5 Voltage Sensor Module	12
	2.6 LED	13
	2.7 LINE Notify	13
	2.8 ระบบปฏิบัติการ iOS	13
	2.9 ภาษา Swift	14
	2.10 ภาษา HTML	15
	2.11 ภาษา JavaScript	15
	2.12 ภาษา CSS	16
	2.13 โปรแกรม Xcode	17
	2.14 ภาษาพีเอชพี PHP	18

สารบัญ (ต่อ)

		หน้า
	2.15 ระบบฐานข้อมูล	19
	2.16 Firebase	20
	2.17 K-Nearest Neighbors	20
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**22**
	3.1 การออกแบบ	22
	3.2 เครื่องมือที่ใช้ในการทดลอง	57
	3.3 การจัดเก็บผลการทดลอง	60
บทที่ 4	**ผลการทดลอง**	**61**
	4.1 การวัดความต่างศักย์ของแบตเตอรี	61
	4.2 การแสดงสถานะของแบตเตอรีด้วย LED	64
	4.3 การแจ้งเตือนบนแอปพลิเคชัน LINE	67
	4.4 การทดสอบแอปพลิเคชัน	70
	4.5 ทดสอบการติดตั้งแอปพลิเคชันบน App Store	83
	4.6 การทดสอบการทำงานของเว็บไซต์	84
	4.7 การทดสอบของระบบการคำนวณตำแหน่งของรถไฟเพื่อลดภาระการทำงานของแอปพลิเคชัน	89
	4.8 ระบบการประมาณเวลาถึงสถานีของรถไฟ	90
	4.9 การส่งค่าพิกัดตำแหน่งรถไฟ และค่าเปอร์เซ็นต์ของแบตเตอรี	103
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**106**
	5.1 สรุปผล	106
	5.2 ข้อเสนอแนะ	107

สารบัญ (ต่อ)

		หน้า
บรรณานุกรม		108
ภาคผนวก	โค้ดคำสั่ง	110

สารบัญรูป

รูปที่		หน้า
2.1	บอร์ด Arduino ต่อกับคอมพิวเตอร์	6
2.2	Layout & Pin out Arduino Board (Model: Arduino UNO R3)	7
2.3	GPS Module Ublox NEO 6M	8
2.4	3G Module (UC15-T)	9
2.5	ส่วนประกอบของ 3G Module (UC15-T)	11
2.6	Voltage Sensor Module	13
2.7	โปรแกรม Xcode	18
3.1	บล็อกไดอะแกรมของระบบติดตามรถไฟสายตะวันออกอัจฉริยะ	22
3.2	โปรแกรม Arduino IDE	23
3.3	โปรแกรม Notepad++	24
3.4	การวัดความต่างศักย์ของแบตเตอรี	24
3.5	แผนผังการทำงานของการแสดงสถานะแบตเตอรี	26
3.6	แผนผังการทำงานของระบบแจ้งเตือนปริมาณแบตเตอรี	27
3.7	แผนผังการทำงานของระบบแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 เกิดปัญหา	28
3.8	แผนผังการทำงานของระบบแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 ให้บริการ	30
3.9	แผนผังการทำงานของระบบแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 งดให้บริการ	30
3.10	แผนผัง User Interface ของแอปพลิเคชัน	32
3.11	โปรแกรม Xcode	32
3.12	การสร้างโปรเจคงานแบบ single view application	33
3.13	หน้าต่างที่ใช้สำหรับเขียนคำสั่ง	33
3.14	หน้าต่างที่ใช้ในการสร้าง User Interface	34
3.15	แผนผังการทำงานในหน้าแสดงตำแหน่งรถไฟ	36

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.16	แผนผังการทำงานในหน้าแสดงสถานีที่ใกล้ตัวผู้ใช้งานที่สุด	38
3.17	แผนผังการทำงานในหน้าแสดงขบวนรถไฟที่กำลังเข้าสถานี	39
3.18	แผนผังการคำนวณสถานะของรถไฟขบวนที่ 275 เทียบกับสถานีที่ผู้ใช้เลือก	40
3.19	บล็อกไดอะแกรมระบบแจ้งเตือนบนแอปพลิเคชัน	41
3.20	ข้อมูลต่าง ๆ ที่เก็บในฐานข้อมูลของ Firebase	42
3.21	แผนผังการทำงานในการตัดสินใจเพื่อส่งการแจ้งเตือนไปยังแอปพลิเคชัน	43
3.22	การออกแบบ App Information ใน App Store	44
3.23	การออกแบบรายละเอียดของแอปพลิเคชันใน App Store (1)	45
3.24	การออกแบบรายละเอียดของแอปพลิเคชันใน App Store (2)	46
3.25	หน้าหลักภาษาอังกฤษของหน้าเว็บไซต์แสดงผล	47
3.26	หน้าภาษาอังกฤษแสดงตำแหน่งขบวน 275-276 กรุงเทพฯ-อรัญประเทศ	47
3.27	หน้าภาษาอังกฤษแสดงตำแหน่งขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง	48
3.28	หน้าภาษาอังกฤษแสดงตำแหน่งขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี	48
3.29	หน้าภาษาอังกฤษแสดงตำแหน่งขบวน ขบวน 367-368 กรุงเทพ-ฉะเชิงเทรา	49
3.30	แผนผังการทำงานของการระบุตำแหน่งของรถไฟบนเว็บไซต์	50
3.31	แผนผังการคำนวณตำแหน่งของรถไฟ	51
3.32	แผนผังการทำงานของหลักการ K-Nearest Neighbors (K-NN)	53
3.33	แผนผังการทำงานของระบบการประมาณเวลาถึงสถานีของรถไฟขบวน 275-276	54
3.34	แผนผังการทำงานของระบบการประมาณเวลาถึงสถานีของรถไฟขบวน 283-284	55
3.35	แผนผังการทำงานของระบบการติดตามรถไฟ	56

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.36	บอร์ด Arduino Uno R3	57
3.37	GPS Module Ublox NEO 6M	58
3.38	3G Module (UC15-T)	58
3.39	Voltage Sensor Module	59
4.1	การทดลองวัดความต่างศักย์ของแบตเตอรี	61
4.2	การทดลองวัดเปอร์เซ็นต์ของแบตเตอรี	62
4.3	การทำ curve fitting ในช่วงที่แบตเตอรีลดลงแบบเชิงเส้น (Linear)	62
4.4	การทำ curve fitting ของช่วงที่แบตเตอรีลดลงแบบเอกซ์โพเนนเชียล (Exponential)	62
4.5	กราฟอัตราการลดลงของแบตเตอรี	63
4.6	อุปกรณ์ภายในกล่องอุปกรณ์ติดตามรถไฟ	65
4.7	กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 81 - 100 เปอร์เซ็นต์	65
4.8	กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 61 - 80 เปอร์เซ็นต์	65
4.9	กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 41 – 60 เปอร์เซ็นต์	66
4.10	กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 21 – 40 เปอร์เซ็นต์	66
4.11	กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 11 – 20 เปอร์เซ็นต์	66
4.12	กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 1 – 10 เปอร์เซ็นต์	67
4.13	กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 0 เปอร์เซ็นต์	67

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.14	การแจ้งเตือนเมื่อปริมาณแบตเตอรีคงเหลือ 11 เปอร์เซ็นต์	68
4.15	การแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 เกิดปัญหา	68
4.16	การแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 ให้บริการ	69
4.17	การแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 283 งดให้บริการ	69
4.18	การแสดงผลตำแหน่งปัจจุบันของรถไฟขบวนที่ 275-276 และ 283-284 บนโปรแกรมจำลอง Xcode	70
4.19	การแสดงผลตำแหน่งปัจจุบันของรถไฟขบวนที่ 281-282 และ 367-368 บนโปรแกรมจำลอง Xcode	71
4.20	ไอคอนของแอปพลิเคชัน	72
4.21	หน้าจอเริ่มต้นของแอปพลิเคชัน	72
4.22	หน้าต่างเพื่อขออนุญาตส่งการแจ้งเตือน	73
4.23	หน้าหลักของแอปพลิเคชัน	73
4.24	หน้าเมนูขบวนรถไฟ	74
4.25	ปุ่ม setting เมื่อกดแล้วจะนำไปสู่หน้าการตั้งค่า	74
4.26	หน้าแสดงตำแหน่งของรถไฟ	75
4.27	หน้าต่างแสดงขบวนรถไฟและสถานีรถไฟที่ต้องการให้แจ้งเตือน	76
4.28	หน้าต่างแสดงการทำรายการสำเร็จ	76
4.29	หน้าต่างแสดงรายการการแจ้งเตือน เมื่อตั้งค่าการแจ้งเตือนแล้ว	77
4.30	หน้าแสดงการแจ้งเตือนของแอปพลิเคชัน	77
4.31	หน้าต่างยืนยันการยกเลิกการแจ้งเตือน	78
4.32	หน้าต่างแสดงรายการแจ้งเตือน เมื่อยกเลิกการแจ้งเตือน 1 รายการ	78
4.33	หน้าแสดงสถานีรถไฟ	79
4.34	หน้าต่างแจ้งเตือนเพื่อเรียกร้องพิกัดของผู้ใช้งาน	80
4.35	ปุ่มการตั้งค่าเมื่อกดแล้วจะนำไปสู่หน้าการตั้งค่าตำแหน่งที่ตั้ง	80
4.36	หน้าแสดงสถานีที่ใกล้ตัวผู้ใช้งานที่สุด	81

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.37	หน้าแสดงขบวนรถไฟที่กำลังเข้าสถานีรถไฟ	82
4.38	คำสั่งในการดึงข้อความให้เร็วขึ้น	82
4.39	แอปพลิเคชันบน App Store	83
4.40	หน้า Sales and Trend	84
4.41	หน้าเว็บไซต์หลัก	84
4.42	หน้าเว็บไซต์แสดงตำแหน่งรถไฟ	86
4.43	หน้าเว็บไซต์เมื่อค่า s มีค่าเป็น 0	86
4.44	หน้าเว็บไซต์ในระบบปฏิบัติการ android และระบบปฏิบัติการ iOS ที่ใช้งานผ่านสมาร์ตโฟนเมื่อค่า s มีค่าเป็น 0	87
4.45	หน้าดาวน์โหลดแอปพลิเคชันจาก Google Play Store	88
4.46	หน้าดาวน์โหลดแอปพลิเคชันจากจาก App Store	88
4.47	ตารางเวลาตามกำหนดการของการรถไฟแห่งประเทศไทย	89
4.48	ตำแหน่งของรถไฟที่คำนวณโดยใช้ระบบฐานข้อมูล	89
4.49	หน้าเว็บไซต์การประมาณเวลาถึงสถานีของรถไฟขบวน 275-276	102
4.50	หน้าเว็บไซต์การประมาณเวลาถึงสถานีของรถไฟขบวน 283-284	103
4.51	กล่องอุปกรณ์ติดตามรถไฟ ขบวนที่ 281-282 กรุงเทพฯ-กบินทร์บุรี	103
4.52	กล่องอุปกรณ์ติดตามรถไฟ ขบวนที่ 367-368 กรุงเทพ-ฉะเชิงเทรา	104
4.53	ค่าพิกัดตำแหน่งรถไฟ และค่าเปอร์เซ็นต์ของแบตเตอรีในฐานข้อมูล	104

สารบัญตาราง

ตารางที่		หน้า
3.1	รายละเอียดการต่ออุปกรณ์ GPS Module กับบอร์ด Arduino	58
3.2	รายละเอียดการต่ออุปกรณ์ 3G Module กับบอร์ด Arduino	59
3.3	รายละเอียดการต่ออุปกรณ์ Voltage Sensor Module กับบอร์ด Arduino	59
4.1	รายละเอียดการแสดงสถานะของแบตเตอรี	64
4.2	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟขบวน 275-276 กรุงเทพฯ-อรัญประเทศ เมื่อใช้ชุดข้อมูลแบบรวม และชุดข้อมูลแบบจำแนกวัน	90
4.3	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟขบวน 275-276 กรุงเทพฯ-อรัญประเทศ เมื่อใช้ชุดข้อมูลแบบมีความเร็วเฉลี่ยและชุดข้อมูลแบบไม่มีความเร็วเฉลี่ย	91
4.4	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีรถไฟในวันจันทร์ เมื่อใช้ชุดข้อมูลวันจันทร์	92
4.5	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันอังคาร เมื่อใช้ชุดข้อมูลวันอังคาร	94
4.6	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันพุธ เมื่อใช้ชุดข้อมูลวันพุธ	95
4.7	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันพฤหัสบดี เมื่อใช้ชุดข้อมูลวันพฤหัสบดี	96
4.8	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันศุกร์ เมื่อใช้ชุดข้อมูลวันศุกร์	97
4.9	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันเสาร์ เมื่อใช้ชุดข้อมูลวันเสาร์	99
4.10	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันอาทิตย์ เมื่อใช้ชุดข้อมูลวันอาทิตย์	100

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.11	ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง เมื่อใช้ชุดข้อมูลแบบจำแนกวัน	101

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อันเนื่องมาจากในปัจจุบันประชาชนนิยมการเดินทางด้วยระบบขนส่งสาธารณะแบบราง ไม่ว่าจะเป็น รถไฟฟ้าระบบขนส่งมวลชนกรุงเทพ (BTS) รถไฟฟ้ามหานคร (MRT) รวมถึงรถไฟมากขึ้น เนื่องจากสามารถช่วยประหยัดค่าใช้จ่ายในการเดินทาง และสามารถหลีกเลี่ยงปัญหารถติดได้ ซึ่งโครงการนี้เป็นโครงการที่พิจารณาถึงความสำคัญของการเดินทางด้วยรถไฟชนบทที่เดินทางระหว่างกรุงเทพเชื่อมต่อกับภาคตะวันออก โดยปัญหาที่มักเกิดขึ้นบ่อยครั้งเมื่อเดินทางด้วยรถไฟสายนี้ คือ ขบวนรถไฟมักเกิดความล่าช้า ไม่ตรงเวลาตามตารางรถไฟ ส่งผลให้ผู้ใช้บริการเสียเวลา และอาจจะเดินทางไม่ทันตามขบวนรถไฟที่ต้องการใช้บริการ จึงเป็นที่มาของโครงการระบบติดตามรถไฟสายตะวันออกอัจฉริยะที่ถูกพัฒนาขึ้นก่อนหน้า ซึ่งมีการพัฒนาส่วนของอุปกรณ์ติดตามรถไฟ จัดทำเว็บไซต์ และแอปพลิเคชันของสมาร์ตโฟนในระบบปฏิบัติการแอนดรอยด์ รวมถึงระบบปฏิบัติการไอโอเอส เพื่อแสดงผลตำแหน่งของรถไฟในสายตะวันออกแก่ผู้ใช้งาน เพื่อช่วยลดปัญหาจากการเดินทางด้วยรถไฟ ซึ่งในปัจจุบันได้มีการใช้ระบบติดตามรถไฟสายตะวันออกจำนวน 2 ขบวน ได้แก่ ขบวน 275-276 กรุงเทพฯ-อรัญประเทศ และขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง อีกทั้งยังมีรถไฟสายตะวันออกอีก 2 ขบวนที่กำลังจะมีการติดตามเกิดขึ้นในเร็ว ๆ นี้ ได้แก่ ขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และขบวน 367-368 กรุงเทพฯ-ฉะเชิงเทรา เป็นต้น ฉะนั้นจึงจำเป็นต้องมีการขยายระบบติดตามรถไฟ เพื่อรองรับผู้ใช้งานในเส้นทางดังกล่าวเพิ่มมากขึ้น ดังนั้น ในส่วนของแอปพลิเคชันบนระบบปฏิบัติการไอโอเอสรวมถึงเว็บไซต์นั้น จำเป็นจะต้องมีการพัฒนาใหม่ให้มีฟังก์ชันการใช้งานที่คลอบคลุมการใช้งานมากขึ้น และรองรับขบวนรถไฟทั้งสองขบวนดังกล่าวที่จะถูกเพิ่มเข้ามา อีกหนึ่งปัญหาที่มักเกิดขึ้นของระบบติดตามรถไฟเวอร์ชันก่อนหน้าคือผู้ใช้งานส่วนใหญ่ไม่ทราบเวลาที่รถไฟถึงสถานีต่าง ๆ ที่แน่นอน เนื่องจากในแอปพลิเคชันจะแสดงเฉพาะตำแหน่งของรถไฟโดยคร่าวเท่านั้น อีกทั้งอุปกรณ์ที่ใช้ติดตามตามตำแหน่งรถไฟนั้น บ่อยครั้งมักเกิดปัญหาแบตเตอรีหมดระหว่างทาง ซึ่งจะทำให้ไม่สามารถรับพิกัดตำแหน่งรถไฟ และส่งพิกัดมายังฐานข้อมูลได้ ซึ่งทั้งสองปัญหาดังกล่าวเป็นปัญหาสำคัญที่มักเกิดขึ้นในระบบติดตามรถไฟอยู่บ่อยครั้ง

จากความสำคัญที่กล่าวมาข้างต้นจึงเป็นที่มาของปริญญานิพนธ์ “ระบบติดตามรถไฟสายตะวันออกอัจฉริยะ เวอร์ชัน 4.0” ซึ่งแบ่งออกเป็น 3 ส่วนหลัก ๆ ในส่วนแรก คือการพัฒนาอุปกรณ์ติดตามรถไฟให้มีประสิทธิภาพมากขึ้น โดยเพิ่มระบบการวัดปริมาณแบตเตอรีคงเหลือ และแสดงสถานะแบตเตอรีของอุปกรณ์ติดตามรถไฟ เพื่อหลีกเลี่ยงปัญหาแบตเตอรีหมดระหว่างทาง อีกทั้งยังสร้างระบบการแจ้งเตือน เพื่อให้ผู้ดูแลระบบทราบเมื่อรถไฟเกิดปัญหา ในส่วนที่สอง คือการพัฒนา

ในส่วนการแสดงผลตำแหน่งรถไฟผ่านทางแอปพลิเคชันของสมาร์ตโฟนในระบบปฏิบัติการไอโอเอส ได้แก่ การเพิ่มหน้าต่างแสดงตำแหน่งของรถไฟสองขบวนที่จะเพิ่มขึ้น เพิ่มระบบแจ้งเตือนสถานีรถไฟ และเพิ่มหน้าต่างของจำนวนและตำแหน่งรถไฟที่กำลังเข้าสู่สถานี เพื่อรองรับการให้บริการแก่ผู้ใช้งานที่เพิ่มมากขึ้น รวมถึงเพื่อรองรับขบวนรถไฟที่จะถูกเพิ่มขึ้นมาอีกสองขบวนในอนาคต และในส่วนสุดท้าย ได้แก่การพัฒนาระบบติดตามรถไฟให้มีประสิทธิภาพมากขึ้น ได้แก่ พัฒนาระบบการคำนวณตำแหน่งของรถไฟบนฐานข้อมูล เพื่อลดภาระการทำงานของแอปพลิเคชันและเว็บไซต์ พัฒนาระบบประมาณเวลาถึงสถานีของรถไฟ เพื่อให้ผู้ใช้งานได้ทราบเวลาที่รถไฟจะถึงสถานีต่าง ๆ และเพิ่มระบบติดตามรถไฟสายตะวันออกขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และขบวน 367-368 กรุงเทพฯ-ฉะเชิงเทรา

1.2 วัตถุประสงค์

1) ออกแบบและสร้างระบบการวัดปริมาณแบตเตอรีคงเหลือของอุปกรณ์ติดตามรถไฟ

2) ออกแบบและสร้างระบบการแจ้งเตือนผู้ดูแลระบบเมื่อรถไฟเกิดปัญหา ยกตัวอย่าง เช่น แบตเตอรีหมดระหว่างทางหรือรถไฟเสีย เป็นต้น

3) พัฒนาแอปพลิเคชันสำหรับโทรศัพท์มือถือสมาร์ตโฟนเพื่อติดตามตำแหน่งรถไฟบนระบบปฏิบัติการไอโอเอส (iOS)

4) ออกแบบและพัฒนาระบบแจ้งเตือนสถานีรถไฟบนแอปพลิเคชันในระบบปฏิบัติการไอโอเอส

5) พัฒนากระบวนการคำนวณตำแหน่งของรถไฟบนฐานข้อมูล (Server) เพื่อลดภาระการทำงานของแอปพลิเคชัน

6) ออกแบบและสร้างระบบการประมาณเวลาถึงสถานีของรถไฟแต่ละขบวน

7) เพื่อขยายระบบติดตามรถไฟสายตะวันออกให้ครอบคลุมกลุ่มผู้ใช้งานมากขึ้น โดยเพิ่มขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และ 367-368 กรุงเทพฯ-ฉะเชิงเทรา

1.3 ขอบเขตของปริญญานิพนธ์

สำหรับระบบติดตามรถไฟ ผู้วิจัยออกแบบและพัฒนาอุปกรณ์ติดตามรถไฟขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และ 367-368 กรุงเทพฯ-ฉะเชิงเทรา โดยจะรับข้อมูลตำแหน่งรถไฟอันประกอบไปด้วย ละติจูด ลองจิจูด ความเร็ว และเพิ่มอุปกรณ์ Voltage Sensor สำหรับวัดเปอร์เซ็นต์แบตเตอรีคงเหลือ อีกทั้งมี LED แสดงสถานะเปอร์เซ็นต์แบตเตอรี จากนั้นไมโครคอนโทรลเลอร์ จะทำหน้าที่ประมวลผลข้อมูลเหล่านี้ เพื่อทำการส่งไปยังฐานข้อมูลโดยใช้ 3G Module และยังสร้างระบบการแจ้งเตือน เพื่อให้ผู้ดูแลระบบทราบเมื่อรถไฟเกิดปัญหา นอกจากนี้ ผู้วิจัยยังได้มีการพัฒนาและ

ปรับปรุงส่วนการแสดงผลบนเว็บไซต์และแอปพลิเคชันบนระบบปฏิบัติการไอโอเอส (iOS) เพื่อแสดงข้อมูลของรถไฟสายตะวันออกทั้ง 4 ขบวน รวมทั้งพัฒนาระบบแจ้งเตือนสถานีรถไฟ ในส่วนสุดท้ายผู้วิจัยยังมีการพัฒนาระบบการคำนวณตำแหน่งของรถไฟบนฐานข้อมูล เพื่อลดภาระการทำงานของแอปพลิเคชันและเว็บไซต์ และพัฒนาระบบประมาณเวลาถึงสถานีของรถไฟ เพื่อให้ผู้ใช้งานได้ทราบเวลาที่รถไฟจะถึงสถานีต่าง ๆ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 GPS

GPS (Global Positioning System) คือ ระบบระบุตำแหน่งบนพื้นโลก ซึ่งทำงานร่วมกับดาวเทียมบอกตำแหน่งทั้งหมด 24 ดวง ดาวเทียม GPS เป็นดาวเทียมที่มีวงโคจรระดับกลาง (Medium Earth Orbit) ที่ระดับความสูงประมาณ 20,200 กิโลเมตร จากพื้นผิวโลก โดยที่แนวคิดในการพัฒนาระบบ GPS เริ่มต้นตั้งแต่ปี ค.ศ. 1957 โดยนักวิทยาศาสตร์ของสหรัฐอเมริกา ต่อมาในปี ค.ศ. 1960 เริ่มทดสอบใช้งานจริงในกองทัพเรือสหรัฐอเมริกา ระบบ GPS เป็นเพียงระบบหนึ่งในระบบ Global Navigation Satellite System (GNSS) ที่ใช้งานอยู่ในปัจจุบัน โดยระบบอื่น ได้แก่ GLONASS เป็นระบบดาวเทียมของประเทศรัสเซีย Galileo เป็นระบบดาวเทียมของสหภาพยุโรป BeiDo เป็นระบบดาวเทียมของประเทศจีน IRNSS เป็นระบบดาวเทียมระดับภูมิภาคของประเทศอินเดีย และ QZSS เป็นระบบดาวเทียมระดับภูมิภาคของประเทศญี่ปุ่น [1]

2.1.1 องค์ประกอบหลักของ GPS

ระบบกำหนดตำแหน่งบนโลก ประกอบด้วย 3 ส่วนหลัก คือ ส่วนอวกาศ (Space segment) ส่วนสถานีควบคุม (Control segment) และส่วนผู้ใช้ (User segment)

2.1.1.1 ส่วนอวกาศ (Space segment)

เป็นส่วนที่อยู่บนอวกาศ ประกอบด้วยดาวเทียม 24 ดวง โดยดาวเทียม 21 ดวง ทำหน้าที่ส่งสัญญาณคลื่นวิทยุจากอวกาศ ส่วนดาวเทียมอีก 3 ดวง เป็นดาวเทียมปฏิบัติงานเสริม ซึ่งจะแบ่งดาวเทียมเป็น 6 วงโคจร วงโคจรละ 4 ดวง อยู่สูงจากพื้นดินประมาณ 20,200 กิโลเมตร

2.1.1.2 ส่วนสถานีควบคุม (Control segment)

ประกอบไปด้วยสถานีภาคพื้นดินที่ควบคุมระบบ ที่กระจายอยู่ตามส่วนต่าง ๆ ของโลก โดยแบ่งออกเป็นสถานีควบคุมหลัก และสถานีรับส่งสัญญาณ 3 แห่ง ซึ่งสถานีควบคุมหลักตั้งอยู่ที่ฐานทัพอากาศในเมืองโคโลราโดสปริงส์ (Colorado Springs) มลรัฐโคโรลาโดของสหรัฐอเมริกา โดยมีสถานีติดตามดาวเทียม 5 แห่ง ทำการรังวัดติดตามดาวเทียมตลอดเวลา และสถานีรับส่งสัญญาณ 3 แห่ง ได้แก่ หมู่เกาะควาจาเลียน หมู่เกาะดิเอโกการ์เซีย และหมู่เกาะแอสเซนชัน

2.1.1.3 ส่วนผู้ใช้ (User segment)

ประกอบด้วยเครื่องรับสัญญาณ GPS ซึ่งมีหลายขนาด สามารถพกพาได้ หรือติดไว้ในรถ เรือ หรือยานพาหนะอื่น ๆ เครื่อง GPS จะทำหน้าที่เปลี่ยนสัญญาณจากสัญญาณคลื่นวิทยุจากอวกาศ เป็นตำแหน่ง ความเร็ว และเวลาโดยประมาณ

2.1.2 หลักการทำงานของ GPS

ดาวเทียม GPS ประกอบด้วยดาวเทียม 24 ดวง โดยแบ่งวงโคจรเป็น 6 รอบ แต่ละวงโคจรมีดาวเทียม 4 ดวง การโคจรจะเอียงทำมุมเอียง 55 องศากับเส้นศูนย์สูตร ซึ่งดาวเทียมแต่ละดวงใช้เวลาในการโคจรรอบโลก 12 ชั่วโมง รัศมีวงโคจรจากพื้นโลก 20,162.81 กิโลเมตร หรือ 12,600 ไมล์

การทำงานของ GPS คือ รับสัญญาณจากดาวเทียมแต่ละดวง โดยสัญญาณดาวเทียมจะประกอบด้วยข้อมูลที่ระบุตำแหน่งและเวลาขณะส่งสัญญาณ ซึ่งเครื่องรับสัญญาณ GPS จะต้องประมวลผลเวลาที่รับสัญญาณดาวเทียมเทียบกับเวลาจริง ณ ปัจจุบัน เพื่อหาระยะทางระหว่างเครื่องรับสัญญาณ GPS กับดาวเทียมแต่ละดวง ในการวัดระยะห่างระหว่างดาวเทียมกับเครื่องรับ GPS สามารถทำได้โดยใช้สูตรคำนวณ ระยะทาง = ความเร็ว x ระยะเวลา นั่นคือวัดระยะเวลาที่คลื่นวิทยุส่งจากดาวเทียมมายังเครื่องรับ GPS คูณด้วยความเร็วของคลื่นวิทยุจะเท่ากับระยะทางที่เครื่องรับ GPS อยู่ห่างจากดาวเทียม นอกจากนี้ยังต้องใช้ดาวเทียมอย่างน้อย 4 ดวงในการค้นหาพิกัดตำแหน่งด้วยดาวเทียม เพื่อให้เกิดความแม่นยำในการบอกตำแหน่งบนผิวโลก ซึ่งความแม่นยำจะขึ้นอยู่กับตำแหน่งของดาวเทียมแต่ละดวงด้วย ดาวเทียมที่มีระยะห่างกันมากจะมีความแม่นยำในการระบุตำแหน่งมากกว่าดาวเทียมที่มีระยะห่างใกล้กัน และยิ่งมีจำนวนดาวเทียมที่รับสัญญาณได้มากก็ยิ่งให้ความแม่นยำมากขึ้น ความแปรปรวนของชั้นบรรยากาศก็มีผลต่อความแม่นยำในการระบุตำแหน่ง ซึ่งชั้นบรรยากาศประกอบด้วยประจุไฟฟ้า ความชื้น อุณหภูมิ และความหนาแน่นที่แปรปรวนตลอดเวลา ทำให้ค่าที่คำนวณได้จากเครื่องรับสัญญาณเพี้ยนไป ก่อนดาวเทียมจะถูกส่งขึ้นสู่อวกาศได้มีการกำหนดวงโคจรของดาวเทียม ทำให้ตำแหน่งของดาวเทียมแต่ละดวงที่ส่งสัญญาณมายังเครื่องรับ GPS ได้ถูกกำหนดไว้แล้ว สถานีควบคุมจะตรวจสอบการโคจรของดาวเทียมอยู่ตลอดเวลาเพื่อตรวจสอบความถูกต้อง [2]

2.1.3 ประเภทของ GPS

2.1.3.1 GPS Navigator

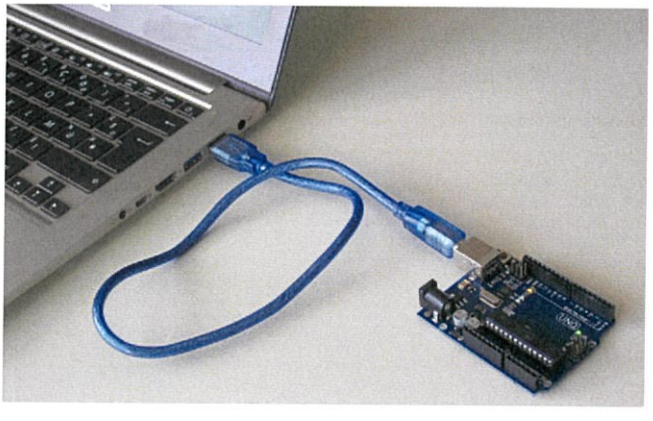
อุปกรณ์และระบบนำทางเป็น GPS ที่ใช้งานในรถยนต์ทั่วไป ซึ่งบอกแผนที่การเดินทาง โดยการป้อนข้อมูลของเป้าหมายที่ต้องการลงไปที่เครื่องนำทาง GPS

2.1.3.2 GPS Tracking System

อุปกรณ์และระบบติดตามยานพาหนะหรือสัตว์เลี้ยง ซึ่งเป็น GPS ที่สามารถติดตามการเดินทางและบอกพิกัดตำแหน่งของเครื่อง GPS ได้ โดยมีอุปกรณ์ติดตามรถแบบออฟไลน์ ซึ่งสามารถตรวจสอบประวัติการเดินทางได้ แต่ไม่สามารถตรวจสอบตำแหน่งที่อยู่ของเครื่อง GPS ได้ และอุปกรณ์ติดตามรถแบบกึ่งออฟไลน์ ซึ่งจะทำงานร่วมกับสมาร์ตโฟน สามารถดูประวัติการเดินทางพร้อมทั้งตำแหน่งปัจจุบันของอุปกรณ์ GPS ได้

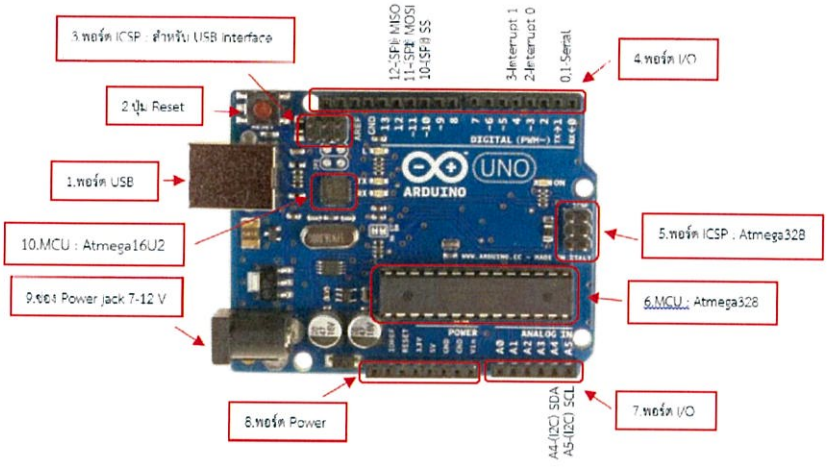
2.2 บอร์ด Arduino

Arduino เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ซึ่งตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานง่าย สามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือสามารถต่อกับบอร์ดเสริมเพื่อความสะดวกในการใช้งาน ผู้ใช้งานสามารถโปรแกรมเพิ่มเติมในตัวบอร์ดได้โดยการเสียบสาย USB เชื่อมคอมพิวเตอร์กับบอร์ด Arduino ดังรูปที่ 2.1 แล้วทำการเขียนโปรแกรม ซึ่งบอร์ด Arduino ที่ใช้คือ Arduino Uno R3



รูปที่ 2.1 บอร์ด Arduino ต่อกับคอมพิวเตอร์ [3]

2.2.1 ส่วนประกอบของ Arduino Uno R3



รูปที่ 2.2 Layout & Pin out Arduino Board (Model : Arduino UNO R3) [4]

1) USB Port : ใช้สำหรับต่อกับคอมพิวเตอร์ เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด

2) Reset Button : ปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

3) ICSP Port Atmega16U2 : เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

4) I/O Port : Digital I/O ตั้งแต่ขา D0 ถึง D13

5) ICSP Port : Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

6) MCU : Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

7) I/O Port : Analog I/O ตั้งแต่ขา A0 ถึง A5

8) Power Port : ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V +5V GND และ Vin

9) Power Jack : รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

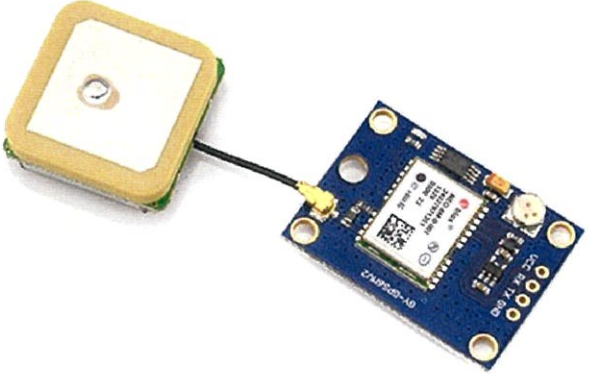
10) MCU Atmega16U2 : เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับคอมพิวเตอร์ผ่าน Atmega16U2

2.2.1.1 ข้อมูลจำเพาะของบอร์ด Arduino Uno R3

1) ชิปไอซีไมโครคอนโทรเลอร์	:	ATmega328
2) ใช้แรงดันไฟฟ้า	:	5V
3) รองรับการจ่ายแรงดันไฟฟ้า	:	6 – 20V
4) พอร์ต Digital I/O	:	14 พอร์ต
5) พอร์ต Analog Input	:	6 พอร์ต
6) กระแสไฟที่จ่ายได้ในแต่ละพอร์ต	:	40 mA
7) กระแสไฟที่จ่ายได้ในพอร์ต 3.3V	:	50 mA
8) พื้นที่โปรแกรมภายใน	:	32 KB
9) พื้นที่แรม	:	2 KB
10) พื้นที่หน่วยความจำถาวร (EEPROM)	:	1 KB
11) ความถี่คริสตัล	:	16 MHz
12) ขนาด	:	68.6x53.4 mm
13) น้ำหนัก	:	25 กรัม

2.3 GPS Module Ublox NEO 6M

GPS โมดูลขนาดกระทัดรัดที่มาพร้อมกับเสาอากาศแบบเซรามิก โดยการเชื่อมต่อเข้ากับบอร์ด Arduino จะใช้เพียง 4 ขา ได้แก่ Rx Tx Vcc และ GND แสดงดังรูปที่ 2.3



รูปที่ 2.3 GPS Module Ublox NEO 6M [5]

2.3.1 ข้อมูลจำเพาะของ GPS Module Ublox NEO 6M

1) แหล่งจ่ายไฟ 3V-5V
2) เสาอากาศแบบเซรามิกสามารถถอดออกได้
3) เวลาในการเริ่มต้นใช้งานจะใช้เวลา 1 วินาที
4) ความเร็วในการรับสัญญาณในที่ร่ม 162 dBm
5) มีเทคโนโลยีการป้องกันการรบกวน
6) รองรับ SBAS (WAAS, EGNOS, MSAS, GAGAN)
7) อัตราการอัพเดตตำแหน่ง 5Hz
8) ทำงานในช่วงอุณหภูมิ -40 ถึง 85 ° C
9) มี EEPROM บันทึกการตั้งค่าข้อมูลพารามิเตอร์ เมื่ออุปกรณ์ไม่มีพลังงาน
10) มี LED ระบุสถานะ
11) มีแบตเตอรีสำรองข้อมูล
12) อัตราการส่งข้อมูลเริ่มต้น 9600

2.4 3G Module

3G Module เป็นบอร์ดสื่อสารเครือข่ายโทรศัพท์เคลื่อนที่ระบบ 3G ใช้ชิพสื่อสารจาก Quectel รุ่น UC15-T รองรับคลื่นความถี่ 850/2100 MHz ในระบบ 3G UMTS และรองรับคลื่นความถี่ 850/900/1800/1900 MHz ในระบบ 2G GSM มีความเร็วในการสื่อสารข้อมูลสูงสุดระหว่างโมดูลกับเครือข่าย 3.6 Mbps (downlink) และ 384 Kbps (uplink) ซึ่งในการเชื่อมต่อเข้ากับบอร์ด Arduino จะใช้ขา VIN GND Rx และ Tx แสดงดังรูปที่ 2.4

รูปที่ 2.4 3G Module (UC15-T) [6]

2.4.1 คุณสมบัติของ 3G Module (UC15-T)

1) ใช้โมดูลสื่อสาร UC15-T จาก Quectel Wireless Solutions Co., Ltd.

2) รองรับความถี่ 850/2100 MHz ในระบบ UMTS 3G

3) รองรับความถี่ 850/900/1800/1900 MHz ในระบบ GSM 2G

4) รองรับความเร็วในการสื่อสาร HSPDA สูงสุดที่ 3.6 Mbps Downlink และ 384 Kbps Uplink

5) รองรับ HSPA Release 5 (Category 6)

6) รองรับ EDGE Downlink only

7) รองรับ GPRS Multi-slot class 12

8) รองรับ UMTS Release 99/5

9) รองรับ GSM Release 99/4

10) รองรับคำสั่ง AT Command 3GPP TS27.007 กับ 3GPP TS27.005 และ enhanced AT command ของ Quectel

11) รองรับโปรโตคอลต่าง ๆ ได้แก่ PPP/TCP/UDP/FTP/HTTP/FILE/MMS/SMTP/SSL

12) ออกแบบเป็น Module บอร์ดขนาดเล็ก

13) เชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านทาง Serial UART

14) มีพอร์ต Micro USB สำหรับเชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์ผ่านพอร์ต USB ได้

15) มีจัมพ์เปอร์สำหรับเริ่มต้นการทำงานทันทีเมื่อจ่ายไฟ (Auto start)

16) มีพินเฮดเดอร์สำหรับเริ่มต้นและปิดการทำงานด้วย Software โดยการทริกขา I/O

17) มีสวิตช์สำหรับเริ่มต้นและปิดการทำงานด้วยการกดปุ่ม (Power switch)

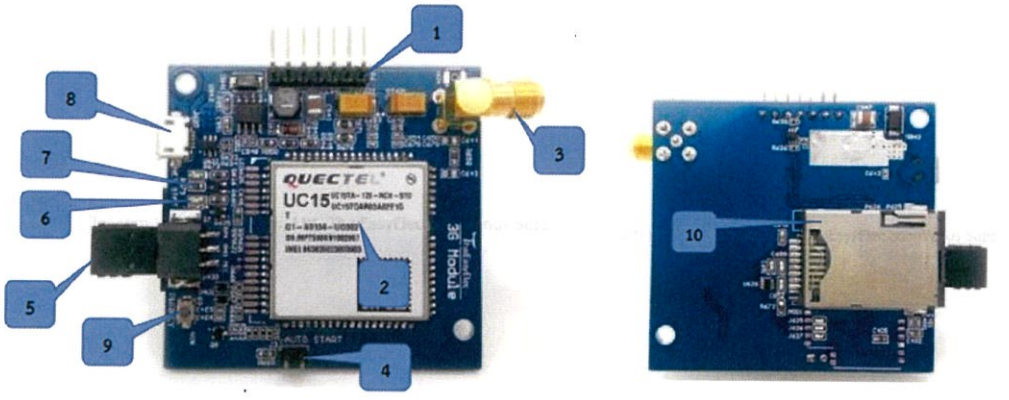
18) มีไฟแสดงสถานะเปิดการทำงาน (Status LED)

19) มีไฟแสดงสถานะเชื่อมต่อเครือข่าย (Network LED)

20) คอนเนคเตอร์สำหรับเสาอากาศสื่อสารหลัก (Main Antenna) แบบ RP-SMA

21) มีช่องเสียบ SIM Card แบบ Push-Pull พร้อมวงจรป้องกัน ESD

2.4.2 ส่วนประกอบของ 3G Module (UC15-T)



รูปที่ 2.5 ส่วนประกอบของ 3G Module (UC15-T) [6]

1) ขาพินเฮดเดอร์ P6 สำหรับเชื่อมต่อขาสัญญาณต่าง ๆ

2) UC15-T Module

3) Connector แบบ SMA สำหรับเสียบเสาอากาศสื่อสารหลัก (Main Antenna)

4) Auto Start Jumper J3 สำหรับให้โมดูลเริ่มต้นทำงานอัตโนมัติ

5) Connector เชื่อมต่อสัญญาณ Audio out และ Mic In

6) Status LED

7) Network LED

8) Micro USB สำหรับพอร์ต USB Modem / USB AT / USB DM

9) Power Key Switch S3 สำหรับเริ่มการทำงานของโมดูล

10) Sim Card Socket

2.4.3 การเปิด-ปิดการทำงาน 3G Module (UC15-T)

ผู้ใช้งานสามารถเลือกเปิด-ปิดการทำงานของโมดูลได้ 3 วิธี ดังนี้

1) เปิดอัตโนมัติ ผู้ใช้สามารถให้โมดูลเปิดอัตโนมัติได้โดยเชื่อมจัมพ์เปอร์ J3 (3G Module มี Jumper มาให้) เมื่อจ่ายไฟให้กับบอร์ด 3G Module โมดูล UC15 จะเริ่มทำงานทันที

2) เปิด-ปิดด้วยโปรแกรม ผู้ใช้สามารถสั่งเปิด-ปิดโมดูลโดยต่อขา PWRKEY บนคอนเน็คเตอร์ P6 เข้ากับไมโครคอนโทรลเลอร์ตามต้องการ ตัวอย่างเช่น เชื่อมต่อกับขา D4 ของ Arduino (อ้างอิงจาก Arduino 1.0 pinout บน Arduino Uno R3) แล้วเขียนโปรแกรมควบคุมให้ขา D4 เป็น Digital Output ดังนี้

- การเปิดโมดูลหลังจากจ่ายไฟให้กับบอร์ดในโปรแกรม ให้สั่งลอจิก Low ค้างไว้ 1 - 2 วินาทีจากนั้นสั่งลอจิก High ไฟแสดงสถานะ STAT จะติด

การปิดโมดูลในขณะที่โมดูลทำงานในโปรแกรม ให้สั่งลอจิก Low ค้างไว้ 1 – 2 วินาทีจากนั้นสั่งลอจิก High ไฟแสดงสถานะ STAT จะดับ

3) เปิด-ปิดด้วยตัวเอง ผู้ใช้สามารถสั่งเปิด-ปิดโมดูลโดยกดสวิตช์ S3 ดังนี้

- การเปิดโมดูลหลังจากจ่ายไฟให้กับบอร์ดในโปรแกรม ให้สั่งลอจิก Low ค้างไว้ 1 - 2 วินาทีจากนั้นสั่งลอจิก High ไฟแสดงสถานะ STAT จะติด

- การปิดโมดูลในขณะที่โมดูลทำงานในโปรแกรม ให้สั่งลอจิก Low ค้างไว้ 1 – 2 วินาทีจากนั้นสั่งลอจิก High ไฟแสดงสถานะ STAT จะดับ

2.4.4 การใช้งาน AT Command

AT Command คือ ชุดคำสั่งสำหรับใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารต่าง ๆ เช่น โมเด็ม หรืออุปกรณ์ DTE (Data Terminal Equipment) เพื่อสั่งงานอุปกรณ์เหล่านี้ให้ทำงานตามที่ต้องการ รวมถึง 3G Module ก็ใช้คำสั่ง AT COMMAND ในการติดต่อสื่อสารและสั่งงานผ่านบอร์ด Arduino โดยรูปแบบการใช้งานคำสั่ง AT Command มีดังนี้

1) AT+<x>=? คือ การทดสอบคำสั่งนั้น ๆ เมื่อทดสอบใช้คำสั่ง โมดูลจะตอบกลับเป็นพารามิเตอร์ และค่าต่าง ๆ ที่ต้องกำหนดในคำสั่งนั้น ๆ เพื่อใช้ในการเขียนสั่งงานโมดูล

2) AT+<x>? คือ การอ่านค่าพารามิเตอร์ต่าง ๆ ของคำสั่งที่ได้กำหนดไปแล้ว

3) AT+<x>=<...> คือ การเขียนคำสั่งเพื่อกำหนดค่าการทำงานให้กับโมดูลตามคำสั่งและค่าพารามิเตอร์ต่าง ๆ

4) AT<x> คือ คำสั่งในการสั่งงานโมดูลที่ไม่มีพารามิเตอร์ใด เช่น สั่งวางสาย ดูรหัสผู้ผลิตโมดูล

2.5 Voltage Sensor Module

โมดูลสำหรับวัดแรงดันไฟฟ้าช่วง 0-24 โวลต์ โดยใช้หลักการวัดแรงดันไฟฟ้าทางขาแอนะล็อกของบอร์ด Arduino ที่สามารถอ่านค่าแรงดันไฟฟ้าที่ 0-5 โวลต์ออกมาเป็นค่าดิจิทัล 0-1023 โมดูลนี้ใช้วงจรแบ่งแรงดันลดไฟจาก 0-24 โวลต์ให้อยู่ในช่วง 5 โวลต์ สามารถวัดแรงดันไฟฟ้าได้สูงสุด 24.9 โวลต์ โดยใช้ไฟเลี้ยง 5 โวลต์ หรือวัดแรงดันไฟฟ้าได้สูงสุด 16.5 โวลต์ โดยใช้ไฟเลี้ยง 3.3 โวลต์ แสดงดังรูปที่ 2.6

รูปที่ 2.6 Voltage Sensor Module [7]

2.6 LED

หลอด LED หรือไดโอดเปล่งแสง (Light Emitting Diode) คือ อุปกรณ์ไฟฟ้าที่เปล่งแสงออกมาเมื่อมีกระแสไหลผ่าน ซึ่งกระแสที่ไหลผ่านนั้นจะต้องไม่มากจนเกินไป เพราะอาจจะทำให้หลอด LED เสียหายได้ โดยในการจำกัดกระแสที่ไหลผ่านหลอด LED จะใช้ตัวต้านทานต่อเข้ากับวงจร เมื่อต่อหลอด LED เข้ากับแหล่งจ่าย Vcc โดยตรง หลอด LED อาจได้รับแรงดันตกคร่อมและกระแสมากเกินไป จึงต้องนำตัวต้านทานมาช่วย ในการคำนวณค่าตัวต้านทานสามารถทำได้ดังสมการที่ 2.1 ซึ่ง R คือค่าตัวต้านทาน V_{cc} คือแหล่งจ่ายไฟ V_F คือแรงดันตกคร่อม LED (Forward Voltage) I_F คือกระแสตกคร่อม LED (Forward Current) [8]

$$R = \frac{(V_{cc} - V_F)}{I_F} \tag{2.1}$$

2.7 LINE Notify

LINE Notify คือ บริการของทาง LINE สามารถส่งข้อความการแจ้งเตือนต่าง ๆ จากบัญชีทางการที่ให้บริการโดย LINE ที่ชื่อ "LINE Notify" ซึ่งจะมี Token ไว้สำหรับแจ้งเตือนไปยังแอคเคาท์ของผู้ขอใช้งาน แต่มีข้อจำกัดบางประการ คือ สามารถส่งแจ้งเตือนได้เฉพาะผู้ที่ขอใช้งานหรือกลุ่มที่ผู้ขอใช้งานเป็นสมาชิกเท่านั้น [9]

2.8 ระบบปฏิบัติการ iOS

ระบบปฏิบัติการ iOS คือ ระบบปฏิบัติการบนอุปกรณ์พกพา ได้แก่ สมาร์ตโฟน แท็บเล็ตคอมพิวเตอร์ เป็นต้น ซึ่งพัฒนาและจำหน่ายโดยบริษัท Apple เปิดตัวครั้งแรกในปี 2007 โดยเปิดใช้ระบบปฏิบัติการบน iPhone ต่อมาได้มีการพัฒนาเพิ่มเติมเพื่อใช้บนอุปกรณ์พกพาอื่น ๆ เช่น

iPhone iPod touch iPad iPad mini และ Apple TV 2 ซึ่งความแตกต่างของระบบปฏิบัติการ iOS และระบบปฏิบัติการอื่น ๆ เช่น วินโดวส์โฟนของไมโครซอฟท์ และระบบปฏิบัติการแอนดรอยด์ของกูเกิล คือ Apple ไม่อนุญาตให้นำ iOS ไปติดตั้งบนอุปกรณ์ที่ไม่ใช่อุปกรณ์ของ Apple ซึ่งข้อดีของ iOS คือ มีแอปพลิเคชันหลากหลาย โดยสามารถดาวน์โหลดได้ที่ App Store และมีโปรแกรม iTunes ที่สนับสนุนการจัดการอุปกรณ์ และโปรแกรมเว็บเบราว์เซอร์ (Safari) ตอบสนองได้รวดเร็ว ส่วนข้อเสียของ iOS คือ ผู้ใช้งานไม่สามารถปรับเปลี่ยนหน้าจอได้ตามความต้องการ แต่ละแอปพลิเคชันไม่สามารถทำงานได้พร้อม ๆ กัน [10]

2.8.1 คุณสมบัติของระบบปฏิบัติการ iOS [11]

2.8.1.1 หน้าจอหลัก (Home screen)

เป็นส่วนแสดงข้อมูลแอปพลิเคชันที่มีบนเครื่อง ซึ่งรองรับการทำงานในระบบปฏิบัติการ iOS สามารถดาวน์โหลดได้ที่ App store หรือผ่านเว็บไซต์ของ Apple ได้โดยตรง

2.8.1.2 แฟ้มข้อมูล (Folders)

ระบบโฟลเดอร์สามารถลากแอปพลิเคชันแต่ละอันมารวมกันเพื่อสร้างโฟลเดอร์ได้ ทำให้การจัดการแอปพลิเคชันอยู่เป็นหมวดหมู่ ซึ่งชื่อสำหรับโฟลเดอร์จะถูกตั้งอัตโนมัติตามประเภทของการใช้งานภายใน แต่สามารถแก้ไขในภายหลังได้

2.8.1.3 การแจ้งเตือน (Notification)

มีการปรับปรุงคุณลักษณะการแจ้งเตือนที่ถูกออกแบบใหม่อย่างสมบูรณ์ เพื่อให้การทำงานรวดเร็วและตอบสนองความต้องการของผู้ใช้ได้ง่ายขึ้น

2.9 ภาษา Swift

ภาษา Swift เป็นโอเพนซอร์ส ซึ่งเป็นภาษาที่ Apple ได้สร้างและออกแบบมาเพื่อให้นักพัฒนาใช้พัฒนาโปรแกรมบน Mac OS X และ iOS สำหรับใช้สร้างแอปพลิเคชันบน iOS เช่น iPhone iPod touch iPad iPad mini และ Apple TV ซึ่งภาษา Swift เป็นภาษาที่ออกแบบให้มีประสิทธิภาพสูงและง่ายต่อการพัฒนา โดยนำข้อดีของภาษาอื่น ๆ เข้ามามากมาย เช่น Type Inference Clean Syntax No semicolons Closures Generics อีกทั้งภาษา Swift ยังถูกออกแบบให้มีความปลอดภัยในการเขียนโปรแกรมมากขึ้น ทำให้ลดข้อผิดพลาดของโปรแกรมที่พัฒนา ยกตัวอย่างเช่น ไม่อนุญาตให้มีตัวแปรที่ไม่ได้ถูกกำหนดค่าในโปรแกรม ไม่ต้องเขียนสัญลักษณ์ * ขณะประกาศตัวแปร Pointer สามารถตรวจสอบการใช้งานค่าต่ำสุดและสูงสุดของ

ตัวเลขจำนวนเต็ม และจะต้องเขียนวงเล็บปีกกาครอบส่วนของโปรแกรมที่อยู่ภายใต้เงื่อนไขใด ๆ [12]

2.10 ภาษา HTML

HTML ย่อมาจากคำว่า Hypertext Markup Language เป็นภาษาหลักที่ใช้ในการเขียนเว็บเพจ ซึ่ง HTML มีโครงสร้างการเขียนโดยอาศัย Tag <> ในการควบคุมการแสดงผลของข้อความ หรือรูปภาพ การสร้างเว็บเพจโดยใช้ภาษา HTML สามารถทำได้โดยการเขียนโปรแกรมผ่าน Text Editor ต่าง ๆ เช่น Notepad EditPlus Microsoft FrontPage หรือ Dream Weaver การเรียกใช้งานเอกสาร HTML สามารถใช้โปรแกรมเว็บเบราว์เซอร์ เช่น Internet Explorer (IE) Mozilla Firefox Safari Opera และ Google Chrome [13]

2.10.1 โครงสร้างหลักของ HTML

โครงสร้างหลักของ HTML จะเริ่มด้วย <html> และจบด้วย </html> เสมอ ซึ่งชุดคำสั่งที่ใช้จะแยกเป็น 2 ส่วน

1) head คำสั่งในส่วนนี้จะใช้บรรยายรายละเอียดเกี่ยวกับเว็บเพจ เช่น ชื่อเรื่องของหน้าเว็บ (Title) ชื่อผู้จัดทำเว็บ (Author) ซึ่งจะไม่แสดงผลที่เว็บเพจโดยตรง

2) body คำสั่งในส่วนนี้เป็นเนื้อหาหลักของหน้าเว็บ ใช้ในการจัดรูปแบบตัวอักษร จัดหน้า ใส่รูปภาพ ซึ่งตัวอักษรในส่วนนี้จะแสดงที่เว็บเบราว์เซอร์โดยตรง

2.11 ภาษา JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรม ซึ่งใช้ในการสร้างและพัฒนาเว็บไซต์ร่วมกับภาษา HTML สามารถตอบสนองผู้ใช้งานได้มากขึ้น เช่น สร้างลูกเล่นต่าง ๆ ให้กับเว็บเพจ เนื่องจากภาษา HTML ไม่สามารถสร้างลูกเล่นได้ JavaScript สามารถตรวจสอบความถูกต้องของข้อมูลในแบบฟอร์มของผู้ใช้งาน ซึ่ง JavaScript เป็นภาษาประเภท Interpreted Language คือทำงานทีละคำสั่ง มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต และ JavaScript เป็น Client-side Script ซึ่งจะประมวลผลบนเครื่องคอมพิวเตอร์ของผู้ใช้งานโดยใช้โปรแกรมเว็บเบราว์เซอร์ การที่ JavaScript ไม่ได้ถูกประมวลผลบนเครื่องเว็บเซิร์ฟเวอร์ เป็นการช่วยแบ่งเบาภาระการทำงานของเซิร์ฟเวอร์ ทำให้ทำงานได้รวดเร็ว การเขียนคำสั่งของ JavaScript ต้องเขียนร่วมกับภาษา HTML โดยแทรกอยู่ระหว่างคำสั่ง <Head> กับ

</Head> หรือหลังจาก <Body> ซึ่ง JavaScript จะสนใจในเรื่องของตัวอักษรพิมพ์เล็กและตัวอักษรพิมพ์ใหญ่ หากใช้ไม่ตรงตามฟังก์ชันการใช้งานของ JavaScript ก็ไม่สามารถทำงานในส่วนนั้น ๆ ได้ [14]

2.11.1 คุณสมบัติที่สำคัญของ JavaScript

2.11.1.1 การเขียนโปรแกรมสามารถทำได้ด้วยโปรแกรม Text Editor ได้ทุกประเภท เช่น Notepad Q edit เป็นต้น

2.11.1.2 การเขียนโปรแกรมจะประมวลผลและแสดงผลได้กับเบราว์เซอร์เท่านั้น

2.11.1.3 โปรแกรมที่เขียนด้วย JavaScript จะโหลดพร้อมกับเว็บเพจที่ถูกเรียกใช้งานโดยเบราว์เซอร์

2.11.1.4 ภาษาโปรแกรมเป็นแบบ Interpreted Language สามารถแสดง Source code ให้เห็นได้ในเว็บเพจโดยตรง หรือจะซ่อนให้เรียกใช้งานก็ได้

2.11.2 ข้อดีของ JavaScript

JavaScript ทำงานบนเว็บเบราว์เซอร์ (Client-side Script) ทำงานเฉพาะในเครื่องคอมพิวเตอร์ของผู้ใช้งานเท่านั้น จึงไม่มีข้อจำกัดการทำงานเกี่ยวกับเซิร์ฟเวอร์ ซึ่งต่างกับภาษาสคริปต์อื่น ๆ เช่น PHP ASP JSP หรือ Perl ต้องประมวลผลและทำงานที่เครื่องเซิร์ฟเวอร์ (Server-side Script) จึงต้องใช้บนเซิร์ฟเวอร์ที่สนับสนุนภาษาเหล่านี้เท่านั้น

2.11.3 ข้อเสียของ JavaScript

JavaScript ไม่สามารถรับและส่งข้อมูลต่าง ๆ กับเซิร์ฟเวอร์ได้โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์เพื่อนำมาแสดงบนเว็บเพจ หรือการรับข้อมูลจากผู้ใช้งานเพื่อนำไปเก็บบนเซิร์ฟเวอร์ JavaScript จึงต้องอาศัยภาษา Server-side Script ในการทำงาน

2.12 ภาษา CSS

CSS ย่อมาจาก Cascading Style Sheet คือ ภาษาที่ใช้ในการจัดรูปแบบการแสดงผลเอกสารที่เขียนจากภาษา HTML ถูกกำหนดมาตรฐานโดย W3C (World Wide Web Consortium) เมื่อปีพ.ศ. 2539 ใน HTML 4.0 ในรูปแบบของ CSS level 1 Recommendations โดยที่ CSS กำหนดรูปแบบของเนื้อหาในเอกสาร ได้แก่ สีข้อความ สีพื้นหลัง ระยะห่างเส้นขอบ ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งเพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพธ์ของ

เอกสาร HTML จะทำการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล เมื่อมีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะเหมือนกันทุกหน้าภายในเว็บไซต์เดียวกัน ก็สามารถแก้ไขได้ง่าย ไม่ยุ่งยากซับซ้อน [15]

2.12.1 ประโยชน์ของภาษา CSS

2.12.1.1. ภาษา CSS จะช่วยในการจัดรูปแบบแสดงผลให้กับภาษา HTML ซึ่งจะช่วยลดการใช้ภาษา HTML ให้น้อยลง โดยเหลือเพียงแต่ส่วนที่เป็นเนื้อหาภายในเอกสาร HTML เท่านั้น ทำให้สามารถแก้ไขเอกสารได้ง่ายและรวดเร็วขึ้น

2.12.1.2 ทำให้ขนาดไฟล์ HTML เล็กลง เนื่องจากภาษา CSS ช่วยลดการใช้ภาษา HTML ขนาดไฟล์จึงเล็กลงไปด้วยเช่นกัน

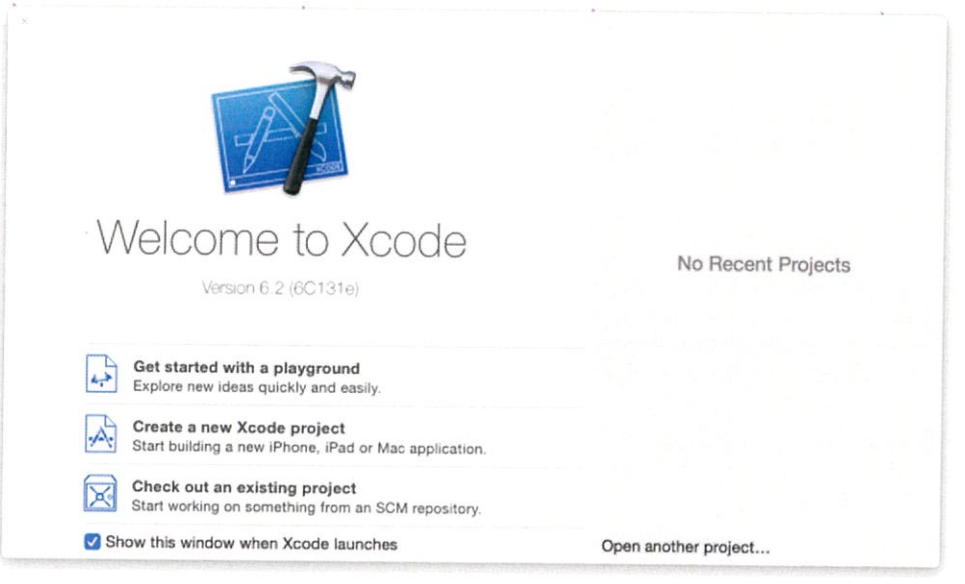
2.12.1.3 ภาษา CSS เป็นภาษา Style Sheets สามารถใช้กำหนดรูปแบบการแสดงผลให้เอกสาร HTML เหมือนกันทั้งหน้าหรือทุกหน้าได้ จึงทำให้เวลาแก้ไขหรือปรับปรุงทำได้ง่ายขึ้น เพียงแก้ไข Style Sheets ที่ใช้งานเพียงชุดเดียว

2.12.1.4 การใช้งาน CSS ทำให้การแสดงผลในสื่อต่าง ๆ ถูกปรับเปลี่ยนไปได้อย่างเหมาะสม เช่น การแสดงผลบนหน้าจอ และการแสดงผลในมือถือ

2.12.1.5 ภาษา CSS สามารถใช้งานได้หลากหลายเว็บเบราว์เซอร์ ทำให้การใช้งานนั้นสะดวกมากยิ่งขึ้น

2.13 โปรแกรม Xcode

โปรแกรม Xcode เป็นเครื่องมือสำหรับนักพัฒนาโปรแกรม และแอปพลิเคชันบนแพลตฟอร์ม OS X และ iOS บนสมาร์ตโฟนสำหรับเครื่อง Mac iPhone และ iPad ซึ่ง Xcode รองรับคำสั่งโปรแกรมภาษา C C++ Objective-C Objective-C++ Java Apple Script Python Ruby ResEdit (Rez) และ Swift โดยนักพัฒนาโปรแกรมที่ต้องการจะพัฒนาแอปพลิเคชันบน iOS จำเป็นต้องมี XCode IDE ติดตั้งในเครื่องคอมพิวเตอร์ก่อน ซึ่ง XCode IDE จะประกอบด้วยพื้นที่สำหรับเขียนโค้ด พื้นที่สำหรับออกแบบหน้าจอ user interface มีการตรวจสอบคอมไพเลอร์ที่มีประสิทธิภาพสูง ชื่อว่า LLVM Compiler 2.0 สามารถสร้างแอปพลิเคชันให้ทำงานได้เร็วขึ้น มีระบบตรวจสอบโค้ดที่มีประสิทธิภาพ สามารถแก้ไขโค้ดที่ผิดให้อัตโนมัติ มีตัวตรวจสอบความผิดพลาดของโค้ดในการเขียนโปรแกรมที่ชื่อว่า LLDB สามารถ ตรวจสอบโค้ดได้ขณะที่โปรแกรมกำลังทำการทดสอบอยู่



รูปที่ 2.7 โปรแกรม Xcode [16]

2.14 ภาษาพีเอชพี (PHP)

PHP (PHP Hypertext Preprocessor) คือภาษาคอมพิวเตอร์จำพวกภาษาสคริปต์ คำสั่งต่าง ๆ จะเก็บอยู่ในไฟล์ที่เรียกว่า script และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ซึ่ง PHP ได้รับการพัฒนาและออกแบบมาเพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถเพิ่มหรือแก้ไขเนื้อหาได้โดยอัตโนมัติลักษณะเด่นของ PHP คือแสดงผลในรูปแบบ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ทำให้ผู้อื่นไม่สามารถอ่านหรือคัดลอกคำสั่งที่เขียนไปใช้งานต่อได้เหมือนภาษาอื่น ๆ เช่น JavaScript และภาษา PHP สามารถเชื่อมต่อกับฐานข้อมูลได้ ทำให้สามารถส่งข้อมูลต่าง ๆ ไปเก็บไว้ในฐานข้อมูลได้โดยผ่านทางอินเทอร์เน็ต คำสั่งของ PHP สามารถสร้างและเขียนผ่านโปรแกรมแก้ไขข้อความทั่วไป เช่น Notepad ซึ่งทำให้การทำงานของ PHP สามารถทำงานได้ในหลายระบบปฏิบัติการ เช่น Unix Windows Mac OS โดยเมื่อเขียนคำสั่งแล้วนำมาประมวลผลในเซิร์ฟเวอร์ Apache Microsoft Internet Information Services (IIS) Personal Web Server Netscape และ iPlanet servers Oreilly Website Pro server Caudium Xitami OmniHTTPd [17]

2.15 ระบบฐานข้อมูล

ฐานข้อมูล (Database) คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้อย่างเป็นระบบ โดยมีความสัมพันธ์ซึ่งกันและกัน และไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูล

ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกัน ในระบบฐานข้อมูลจะประกอบด้วยหลายแฟ้มข้อมูล และเปิดโอกาสให้ผู้ใช้สามารถใช้งานและป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Database Management System) ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ ซึ่งการเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล หรือการแก้ไขฐานข้อมูล [18]

2.15.1 โครงสร้างของฐานข้อมูล

2.15.1.1 Character คือ ตัวอักขระแต่ละตัว ตัวเลข หรือเครื่องหมาย

2.15.1.2 Field คือ ชุดข้อมูลที่ใช้แทนโครงสร้างต่าง ๆ ซึ่งเป็นชุดข้อมูลที่แสดงในแนวคอลัมน์ เช่น ชื่อของบุคคล หมายเลขโทรศัพท์

2.15.1.3 Record คือ ระเบียนหรือรายการข้อมูล เป็นการนำ Field หลายๆ Field มารวมกัน เช่น ระเบียนของพนักงานแต่ละคน

2.15.1.4 Table /File คือ ตารางหรือแฟ้มข้อมูล ประกอบขึ้นด้วยระเบียนต่าง ๆ เช่น ตารางข้อมูลของบุคคล

2.15.1.5 Database คือ ฐานข้อมูล ประกอบด้วยตาราง และแฟ้มข้อมูลต่าง ๆ ที่เกี่ยวข้องหรือมีความสัมพันธ์กัน

2.15.2 ประโยชน์ของฐานข้อมูล

2.15.2.1 ลดความซ้ำซ้อนของข้อมูล

2.15.2.2 การรักษาความถูกต้องเชื่อถือได้ของข้อมูล

2.15.2.3 การป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้อย่างสะดวก ซึ่งจะให้เฉพาะผู้ที่เกี่ยวข้องเท่านั้น

2.16 Firebase

Firebase (ไฟร์เบส) คือ บริการ backend และแพลตฟอร์มครบวงจรสำหรับนักพัฒนาแอปพลิเคชัน และโปรแกรมประยุกต์บนเว็บแพลตฟอร์มที่มีเครื่องมือและโครงสร้างพื้นฐานที่ได้รับการออกแบบมาเพื่อช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันที่มีคุณภาพสูง Firebase ถูกสร้างขึ้นจากคุณสมบัติเสริมว่านักพัฒนาสามารถผสมและจับคู่เพื่อให้พอดีกับความต้องการของตน บริษัท Firebase ก่อตั้งขึ้นในปี 2011 โดยแอนดรูลีและเจมส์ เทมปลิน สินค้าเริ่มต้นของ Firebase เป็นฐานข้อมูลเรียลไทม์ซึ่งมี API ที่ช่วยให้นักพัฒนาในการจัดเก็บและซิงค์ข้อมูล

ต่อมา google ได้ซื้อกิจการ Firebase โดยใช้ชื่อว่า Google Firebase 2.0 และมีการพัฒนาจากบริการ backend เก็บข้อมูลอย่างเดียว มาเป็นแพลตฟอร์มครบวงจรสำหรับนักพัฒนาแอปพลิเคชัน (รองรับ iOS แอปพลิเคชัน Android แอปพลิเคชัน และ Web แอปพลิเคชัน) รองรับบริการแทบทุกอย่างที่นักพัฒนาแอปพลิเคชันต้องใช้งาน [19]

2.17 K-Nearest Neighbors

K-Nearest Neighbors หรือ K-NN เป็นวิธีที่ใช้ในการจัดแบ่งคลาส โดยการตรวจสอบจำนวนบางจำนวนของกรณีหรือเงื่อนไขที่เหมือนกันหรือใกล้เคียงกันมากที่สุด โดยจะหาผลรวมของจำนวนเงื่อนไขหรือกรณีต่าง ๆ สำหรับแต่ละคลาส และคำนวณหาคลาสที่ใกล้เคียงกับข้อมูลที่ต้องการพิจารณามากที่สุด ซึ่งวิธีการหาคลาสที่ใกล้เคียงกับข้อมูลที่ต้องการพิจารณามากที่สุดนั้น ขั้นแรกทำการสร้างกลุ่มข้อมูลตัวอย่าง จากนั้นกำหนดค่า K โดย K คือจำนวนกลุ่มข้อมูลที่ใกล้เคียงกับข้อมูลที่ต้องการพิจารณามากที่สุด ซึ่งค่า K ควรกำหนดให้เหมาะสมกับกลุ่มข้อมูล หากเลือกค่าน้อยไป ก็จะส่งผลให้ความแม่นยำลดลง เช่น มีจำนวนกลุ่มข้อมูลตัวอย่างหลายคลาส และกำหนดให้ค่า K เท่ากับ 1 ซึ่งหมายความว่าจะเลือกพิจารณากลุ่มข้อมูลตัวอย่างแค่กลุ่มเดียว ส่งผลให้การกำหนดกลุ่มข้อมูลให้กับข้อมูลที่ต้องการพิจารณามีความผิดพลาด เป็นต้น ซึ่งหากเลือกค่า K มากไปก็จะส่งผลให้โอกาสของกลุ่มข้อมูลที่ไม่เกี่ยวข้องมีเพิ่มมากขึ้นด้วย จากนั้นทำการหาระยะห่างระหว่างข้อมูลที่ต้องการพิจารณากับกลุ่มข้อมูลตัวอย่าง ซึ่งสามารถคำนวณระยะห่างได้โดยใช้สมการ Euclidean Distance ดังสมการที่ 2.2 โดย $d(x_i, y_i)$ คือระยะห่างระหว่างตำแหน่ง x และ y แต่ละตัวแปรในข้อมูล x คือจุดที่ต้องการพิจารณา และ y คือจุดของข้อมูลในแต่ละกลุ่มข้อมูล และ n คือจำนวนตัวแปรในกลุ่มข้อมูล

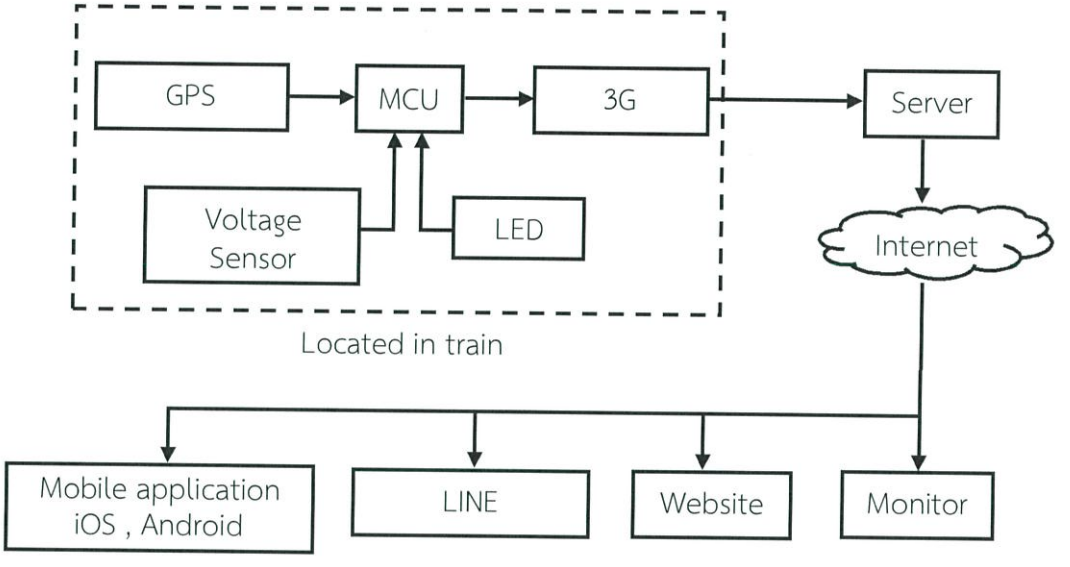
$$d(x_i, y_i) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (2.2)$$

เมื่อคำนวณระยะห่างระหว่างเงื่อนไขหรือกรณีต่าง ๆ ได้แล้ว จากนั้นก็เลือกกลุ่มข้อมูลที่ใกล้เคียงกับข้อมูลที่ต้องการพิจารณาตามจำนวน K ที่กำหนดไว้ เมื่อพิจารณาข้อมูลจำนวน K ชุด และสังเกตว่ากลุ่มข้อมูลใดใกล้เคียงกับข้อมูลที่ต้องการพิจารณามากที่สุด จากนั้นทำการกำหนดกลุ่มข้อมูลให้กับข้อมูลที่ต้องการพิจารณา เพื่อจัดกลุ่มว่าข้อมูลที่ต้องการพิจารณานั้นจะมีแนวโน้มเป็นอย่างไร [20]

บทที่ 3

การออกแบบและการจัดทำเปรียญญานิพนธ์

3.1 การออกแบบ



รูปที่ 3.1 บล็อกไดอะแกรมของระบบติดตามรถไฟสายตะวันออกอัจฉริยะ

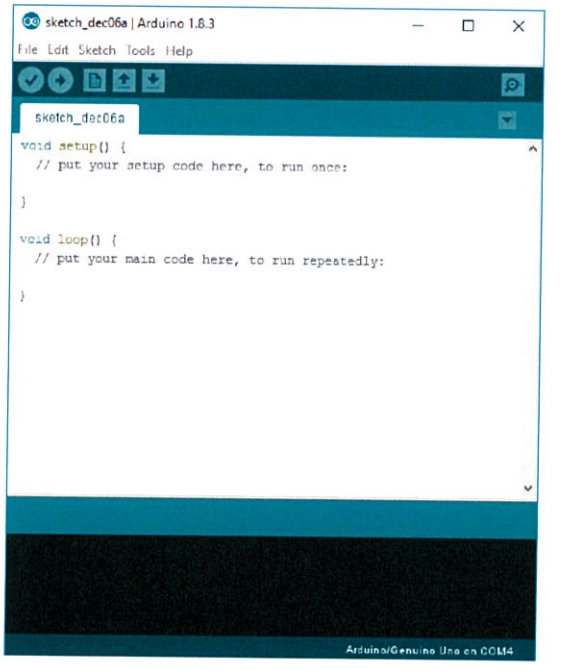
รูปที่ 3.1 แสดงบล็อกไดอะแกรมภาพรวมของระบบติดตามรถไฟสายตะวันออกอัจฉริยะ ในส่วนของอุปกรณ์ที่ถูกติดตั้งบนรถไฟ ประกอบไปด้วย GPS Module ซึ่งทำหน้าที่ในการรับข้อมูลตำแหน่งรถไฟอันประกอบไปด้วย ละติจูด ลองจิจูด ความเร็ว จากนั้นไมโครคอนโทรลเลอร์จะทำหน้าที่ประมวลผลข้อมูลเหล่านี้ เพื่อทำการส่งไปยังฐานข้อมูลโดยใช้ 3G Module อีกทั้งในโครงการนี้ยังมีการติดตั้ง Voltage Sensor สำหรับวัดแบตเตอรีคงเหลือของอุปกรณ์ เพื่อนำไปออกแบบระบบการแจ้งเตือนแบตเตอรีคงเหลือแก่ผู้ดูแลระบบ เพื่อให้สามารถแก้ไขปัญหาแบตเตอรีหมดระหว่างทางได้ทันท่วงที นอกเหนือจากระบบแจ้งเตือนแบตเตอรีคงเหลือแล้วนั้น ยังมีระบบระบบแจ้งเตือนเมื่ออุปกรณ์ติดตามรถไฟมีปัญหาอีกด้วย โดยการแจ้งเตือนนั้นจะแสดงผ่านแอปพลิเคชันไลน์ ในส่วนการแสดงผลตำแหน่งรถไฟนั้น ได้มีการแสดงผลตำแหน่งรถไฟผ่านหลายช่องทาง ยกตัวอย่างเช่น แอปพลิเคชันบนระบบปฏิบัติการไอโอเอส แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ เว็บไซต์ และหน้าจอมอนิเตอร์ เป็นต้น อย่างไรก็ตามในโครงการวิจัยนี้

จะเน้นไปที่การพัฒนาในการแสดงผลผ่านทางแอปพลิเคชันบนระบบปฏิบัติการไอโอเอส และบนเว็บไซต์เป็นหลัก เนื่องจากเป็นส่วนที่มีผู้ใช้งานอยู่เป็นจำนวนมาก

3.1.1 การออกแบบการวัดความต่างศักย์ของแบตเตอรี

3.1.1.1 การติดตั้งโปรแกรม Arduino IDE

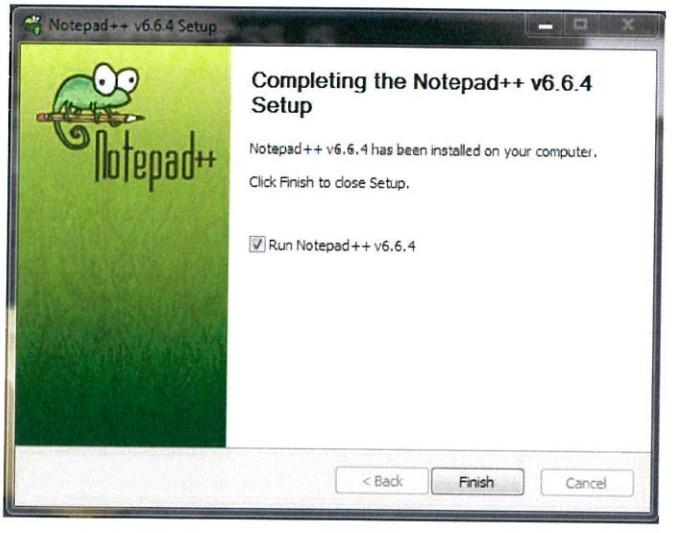
ทำการติดตั้งโปรแกรม Arduino IDE เพื่อเขียนโปรแกรมในการรับค่าพิกัดรถไฟ และโปรแกรมในการคำนวณเปอร์เซ็นต์แบตเตอรีคงเหลือของแบตเตอรีสำรอง ซึ่งถูกติดตั้งไว้เพื่อเป็นแหล่งจ่ายพลังงานให้แก่อุปกรณ์ทั้งหมดที่ถูกติดตั้งบนรถไฟ และส่งค่าดังกล่าวเข้าฐานข้อมูล แสดงดังรูปที่ 3.2



รูปที่ 3.2 โปรแกรม Arduino IDE

3.1.1.2 การติดตั้งโปรแกรม Notepad++

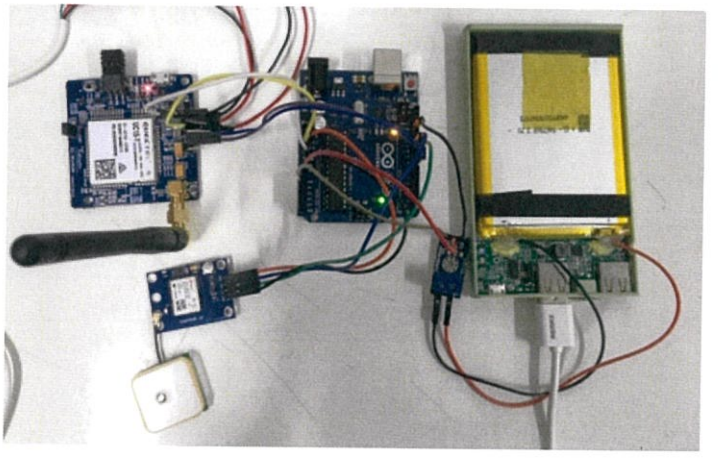
ทำการติดตั้งโปรแกรม Notepad++ เพื่อเขียนโปรแกรมในการรับค่าพิกัดรถไฟ และค่าเปอร์เซ็นต์ของแบตเตอรีที่ประมวลผลแล้วจากโปรแกรม Arduino จากนั้นได้ทำการเขียนโปรแกรมเพื่อเชื่อมต่อกับฐานข้อมูล และส่งข้อมูลไปยังฐานข้อมูล แสดงดังรูปที่ 3.3



รูปที่ 3.3 โปรแกรม Notepad++

3.1.1.3 การออกแบบการวัดความต่างศักย์ของแบตเตอรี

สำหรับการวัดความต่างศักย์ของแบตเตอรีสำรองนั้น จะทำการวัดโดยใช้ Voltage Sensor เพื่อหาสมการแนวโน้มการลดลงของแบตเตอรี โดยใช้อัลกอริทึม curve fitting ของโปรแกรม MATLAB ซึ่งจะวัดจากระบบการใช้อุปกรณ์ติดตามรถไฟจริง เพื่อให้เกิดความแม่นยำในการหาสมการแนวโน้มการลดลงของแบตเตอรี จากนั้นจะนำสมการแนวโน้มที่ได้ไปคำนวณเปอร์เซ็นต์ของแบตเตอรีคงเหลือ และส่งค่าดังกล่าวเข้าฐานข้อมูลต่อไป รูปที่ 3.4 แสดงการวัดความต่างศักย์ของแบตเตอรี



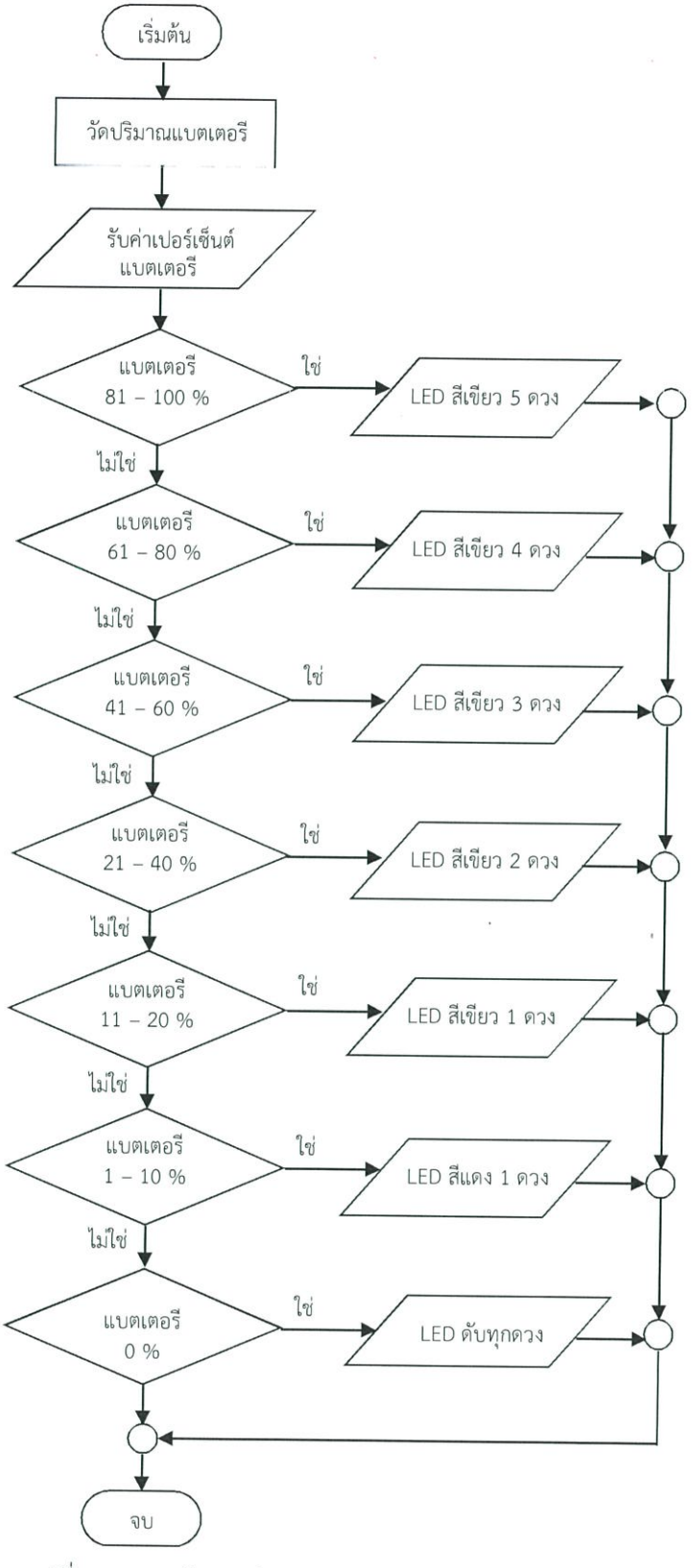
รูปที่ 3.4 การวัดความต่างศักย์ของแบตเตอรี

3.1.1.4 ออกแบบ LED แสดงสถานะของแบตเตอรี

เนื่องจากระบบติดตามรถไฟที่ใช้งานอยู่นั้น ผู้ที่ถืออุปกรณ์ติดตามรถไฟไม่สามารถทราบถึงปริมาณแบตเตอรีคงเหลือของอุปกรณ์ติดตามรถไฟ ซึ่งในบางครั้งอาจส่งผลให้เกิดปัญหาแบตเตอรีหมดกลางทาง เนื่องจากแบตเตอรีอาจถูกชาร์ตมาไม่เต็มหรือเสื่อม เป็นต้น ดังนั้นจึงได้มีการออกแบบการแสดงสถานะของแบตเตอรี โดยใช้หลอด LED หรือไดโอดเปล่งแสง ซึ่ง LED จะเปล่งแสงได้ก็ต่อเมื่อมีกระแสไหลผ่าน โดยกระแสที่ไหลผ่านนั้นจะต้องไม่มากจนเกินไป เพราะอาจจะทำให้หลอด LED เสียหายได้ จึงจำกัดกระแสที่ไหลผ่านหลอด LED ด้วยตัวต้านทาน ในการคำนวณค่าตัวต้านทานสามารถทำได้ดังสมการที่ 3.1 ซึ่ง R คือค่าตัวต้านทาน V_{cc} คือแหล่งจ่ายไฟ V_F คือแรงดันตกคร่อม LED (Forward Voltage) I_F คือกระแสตกคร่อม LED (Forward Current)

$$R = \frac{(V_{cc} - V_F)}{I_F} \tag{3.1}$$

จากสมการที่ 3.1 แหล่งจ่ายไฟเท่ากับ 5 โวลต์ กระแสตกคร่อมของหลอด LED เท่ากับ 20 มิลลิแอมป์ ซึ่งแรงดันตกคร่อมของ LED แต่ละสีมีค่าไม่เท่ากัน หลอด LED สีเขียว จะมีแรงดันตกคร่อมเท่ากับ 3 โวลต์ ส่วนหลอด LED สีแดง จะมีแรงดันตกคร่อมเท่ากับ 2 โวลต์ เมื่อต้องการหาค่าตัวต้านทานสำหรับจำกัดกระแสของหลอด LED สีเขียว ซึ่งคำนวณแล้วได้เท่ากับ 100 โอห์ม ส่วนค่าตัวต้านทานสำหรับจำกัดกระแสของหลอด LED สีแดง คำนวณแล้วได้เท่ากับ 150 โอห์ม โดยการออกแบบสถานะของแบตเตอรีจะใช้หลอด LED ทั้งหมด 5 ดวง ซึ่งเป็นสีเขียวอย่างเดียว 4 ดวง และเป็นสีเขียวและสีแดง 1 ดวง แสดงการทำงานดังรูปที่ 3.5

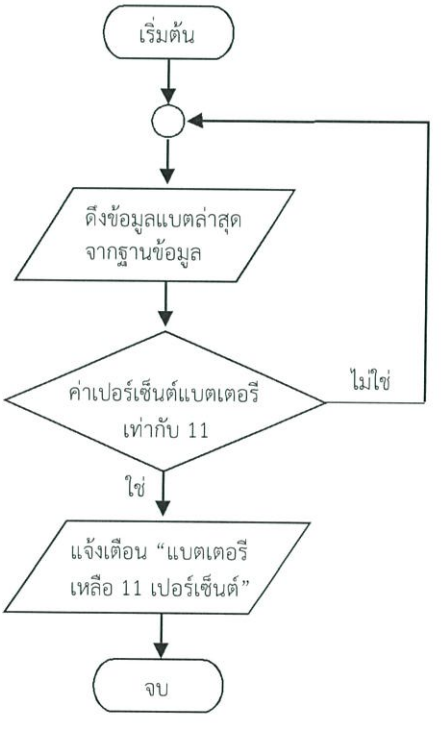


รูปที่ 3.5 แผนผังการทำงานของการแสดงสถานะแบตเตอรี

3.1.2 การออกแบบระบบแจ้งเตือน

3.1.2.1 การออกแบบระบบแจ้งเตือนปริมาณแบตเตอรี

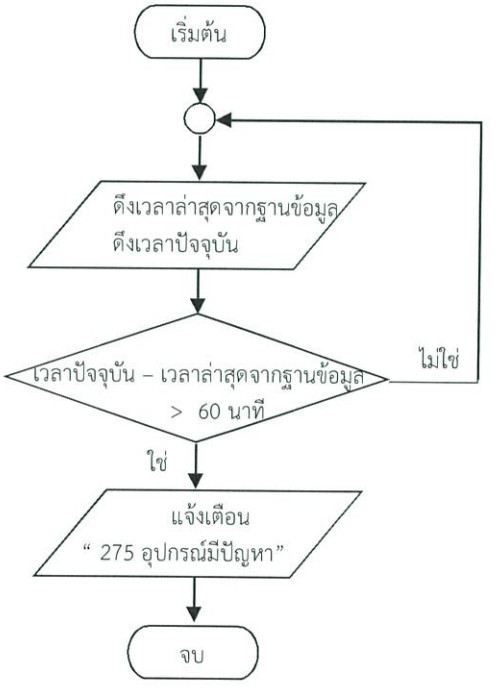
ในการแจ้งเตือนปริมาณแบตเตอรีคงเหลือนั้นจะมีการออกแบบระบบแจ้งเตือนข้อความผ่านแอปพลิเคชัน LINE โดยใช้บริการที่ชื่อว่า Line Notify ซึ่งเป็นบริการของทาง LINE สามารถส่งข้อความการแจ้งเตือนต่าง ๆ ไปยังแอคเคาท์ผู้ขอใช้งาน แต่มีข้อจำกัดบางประการ คือ สามารถส่งแจ้งเตือนได้เฉพาะผู้ที่ขอใช้งาน หรือกลุ่มที่ผู้ขอใช้งานเป็นสมาชิกเท่านั้น โดยผู้พัฒนาได้ทำการกำหนดให้มีการแจ้งเตือนเมื่อปริมาณแบตเตอรีคงเหลือใกล้จะหมด โดยทำการดึงค่าเปอร์เซ็นต์แบตเตอรีล่าสุดจากฐานข้อมูลมาทำการเขียนโปรแกรม โดยในโครงการนี้กำหนดให้ระบบจะมีการแจ้งเตือนเมื่อค่าเปอร์เซ็นต์แบตเตอรีล่าสุดในฐานข้อมูลเป็น 11 เปอร์เซ็นต์ เนื่องจากเมื่อค่าเปอร์เซ็นต์แบตเตอรีคงเหลือ 11 เปอร์เซ็นต์ อุปกรณ์จะสามารถใช้งานได้ประมาณ 3 ชั่วโมง ซึ่งจะให้ระบบส่งข้อความแจ้งเตือนว่า “แบตเตอรีเหลือ 11 เปอร์เซ็นต์” ผ่านแอปพลิเคชัน LINE เพื่อให้ผู้ดูแลระบบได้ทราบและทำการแก้ไขเบื้องต้น ไม่ว่าจะเป็น การชาร์ตแบตเตอรีเพิ่มเมื่อถึงปลายทาง หรือการปิดเครื่องชั่วคราวในช่วงสถานีที่มีผู้ใช้งานน้อย เป็นต้น ซึ่งการแก้ไขเบื้องต้นเหล่านี้เพื่อเป็นการป้องกันแบตเตอรีหมดระหว่างทาง ซึ่งแสดงการทำงานดังรูปที่ 3.6



รูปที่ 3.6 แผนผังการทำงานของระบบแจ้งเตือนปริมาณแบตเตอรี

3.1.2.2 การออกแบบระบบแจ้งเตือนเมื่ออุปกรณ์ที่ใช้ในการติดตามรถไฟเกิดปัญหา

เนื่องจากระบบติดตามรถไฟที่ใช้งานอยู่ในปัจจุบันนั้น เมื่ออุปกรณ์ที่ใช้ในการติดตามรถไฟมีปัญหา เช่น อุปกรณ์ติดตามรถไฟไม่สามารถส่งข้อมูลพิกัดตำแหน่งของรถไฟมายังฐานข้อมูล ซึ่งจะมีผลทำให้ผู้ใช้งานไม่สามารถทราบตำแหน่งของรถไฟผ่านแอปพลิเคชันและเว็บไซต์ได้ อีกทั้งผู้ดูแลระบบไม่ทราบถึงปัญหาของอุปกรณ์ที่เกิดขึ้นได้ ดังนั้นในโครงการนี้จึงได้มีการออกแบบระบบแจ้งเตือนข้อความให้ผู้ดูแลระบบทราบผ่านทางแอปพลิเคชัน LINE โดยใช้บริการ Line Notify เช่นเดียวกัน โดยหลักการทำงานจะทำการดึงค่าเวลาล่าสุดจากฐานข้อมูล และนำค่าเวลาในปัจจุบันมาทำการเขียนโปรแกรม เพื่อคำนวณหาระยะเวลาที่อุปกรณ์ไม่สามารถส่งค่ามายังฐานข้อมูลได้ โดยใช้วิธีนำค่าเวลาในปัจจุบันลบกับค่าเวลาล่าสุดจากฐานข้อมูล โดยโครงการนี้มีการกำหนดไว้ว่า หากนำเวลาทั้งสองดังกล่าวมาลบกันแล้วมีค่ามากกว่า 60 นาที (หมายความว่าข้อมูลพิกัดตำแหน่งของรถไฟได้หยุดส่งมายังฐานข้อมูลเป็นเวลา 1 ชั่วโมง) ให้ระบบทำการส่งข้อความแจ้งเตือนผ่านทางแอปพลิเคชัน LINE เพื่อให้ผู้ดูแลระบบได้ทราบ มีข้อความว่า “ 275 อุปกรณ์มีปัญหา ” เมื่ออุปกรณ์ติดตามรถไฟขบวน 275 มีปัญหา และแจ้งเตือน “ 283 อุปกรณ์มีปัญหา ” เมื่ออุปกรณ์ติดตามรถไฟขบวน 283 มีปัญหา ในที่นี้ระบบการทำงานของทั้งสองขบวนจะเหมือนกันทั้งขบวน 275 และ 283 แสดงดังรูปที่ 3.7

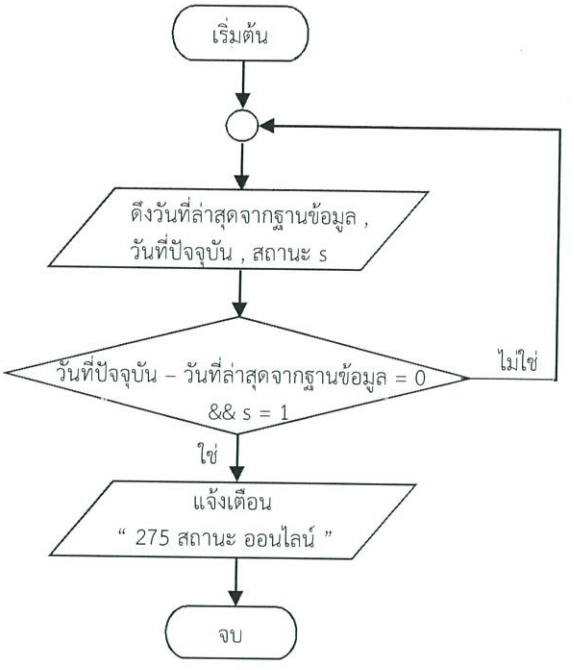


รูปที่ 3.7 แผนผังการทำงานของระบบแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 เกิดปัญหา

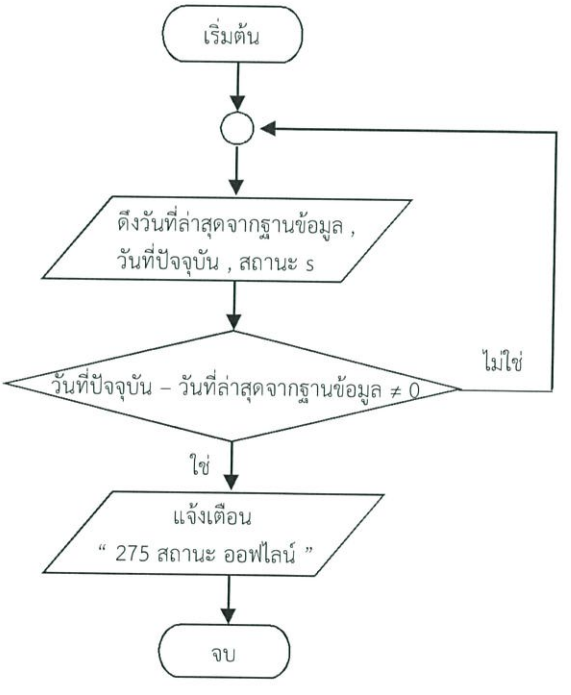
3.1.2.3 การออกแบบระบบแจ้งเตือนสถานะการให้บริการของอุปกรณ์ที่ใช้ในการติดตามรถไฟ

เนื่องจากระบบติดตามรถไฟที่ใช้งานอยู่ในปัจจุบันนั้น ผู้ดูแลระบบไม่สามารถทราบได้ว่าระบบติดตามรถไฟของทั้ง 2 ขบวน มีขบวนใดบ้างที่มีการเปิดให้บริการหรืองดให้บริการ ซึ่งทำให้ผู้ดูแลระบบต้องเข้าไปตรวจสอบในฐานข้อมูลเป็นประจำเพื่อทราบถึงสถานะการให้บริการของรถไฟแต่ละขบวน ดังนั้นจึงได้มีการออกแบบระบบแจ้งเตือนข้อความเพื่อให้ผู้ดูแลระบบทราบสถานะการให้บริการของระบบติดตามรถไฟผ่านทางแอปพลิเคชัน LINE โดยใช้บริการ Line Notify เช่นเดียวกัน โดยวิธีการคือ ทำการดึงค่าวันที่ล่าสุดจากฐานข้อมูล ค่าวันที่ในปัจจุบัน และค่าสถานะการให้บริการ มาทำการเขียนโปรแกรม เพื่อคำนวณหาสถานะการให้บริการของรถไฟแต่ละขบวน โดยใช้วิธีนำค่าวันที่ในปัจจุบันลบกับค่าวันที่ล่าสุดจากฐานข้อมูล หากลบกันแล้วมีค่าเท่ากับศูนย์ และมีสถานะการให้บริการ ซึ่งหมายความว่าอุปกรณ์ที่ใช้ในการติดตามรถไฟเปิดให้บริการ ให้ระบบทำการส่งข้อความแจ้งเตือนผ่านทางแอปพลิเคชัน LINE เพื่อให้ผู้ดูแลระบบได้ทราบ โดยแจ้งเตือน “ 275 สถานะ ออนไลน์ ” เมื่ออุปกรณ์ติดตามรถไฟขบวน 275 เปิดให้บริการ และแจ้งเตือน “ 283 สถานะ ออนไลน์ ” เมื่ออุปกรณ์ติดตามรถไฟขบวน 283 เปิดให้บริการ

อย่างไรก็ตาม หากค่าวันที่ในปัจจุบันลบกับค่าวันที่ล่าสุดจากฐานข้อมูลแล้วมีค่าไม่เท่ากับศูนย์ ให้ระบบทำการส่งข้อความแจ้งเตือน “ 275 สถานะ ออฟไลน์ ” เมื่ออุปกรณ์ติดตามรถไฟขบวน 275 งดให้บริการ และแจ้งเตือน “ 283 สถานะ ออฟไลน์ ” เมื่ออุปกรณ์ติดตามรถไฟขบวน 283 งดให้บริการ ซึ่งการทำงานของทั้งสองขบวนจะเหมือนกัน โดยตัวอย่างแผนผังการทำงานของรถไฟขบวน 275 เมื่ออุปกรณ์ให้บริการแสดงดังรูปที่ 3.8 แผนผังการทำงานของรถไฟขบวน 275 เมื่ออุปกรณ์งดให้บริการแสดงการทำงานดังรูปที่ 3.9



รูปที่ 3.8 แผนผังการทำงานของระบบแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 ให้บริการ



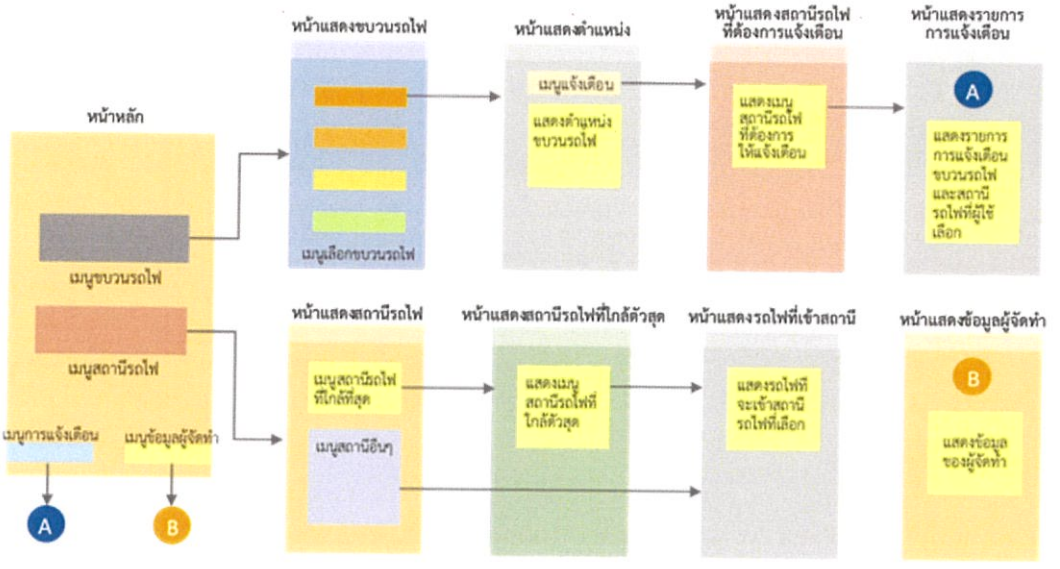
รูปที่ 3.9 แผนผังการทำงานของระบบแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 งดให้บริการ

3.1.3 การออกแบบแอปพลิเคชันบนสมาร์ตโฟนในระบบปฏิบัติการไอโอเอส (iOS)

3.1.3.1 การออกแบบ User Interface และการทำงานแอปพลิเคชัน

ในการออกแบบ User Interface ของแอปพลิเคชันบนสมาร์ตโฟนในระบบปฏิบัติการไอโอเอสนั้น ผู้วิจัยได้ออกแบบให้หน้าหลักประกอบไปด้วย 4 เมนู คือ เมนูขบวนรถไฟ เมนูสถานีรถไฟ เมนูการแจ้งเตือน และเมนูข้อมูลผู้จัดทำโครงการ โดยเมนูขบวนรถไฟจะแสดงขบวนรถไฟสายตะวันออกทั้ง 4 ขบวน คือ ขบวน 275-276 กรุงเทพฯ-อรัญประเทศ ขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง ขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และขบวน 367-368 กรุงเทพฯ-ฉะเชิงเทรา ซึ่งเมื่อผู้ใช้งานกดเลือกขบวนรถไฟขบวนใด ๆ แอปพลิเคชันจะแสดงตำแหน่งของขบวนรถไฟขบวนนั้น ๆ ในส่วนถัดมาคือส่วนเมนูสถานีรถไฟจะมีการแสดงสถานีรถไฟที่อยู่ใกล้ผู้ใช้งานมากที่สุด โดยจะมีการเรียกร้องพิกัดของผู้ใช้งานจาก GPS บนมือถือของผู้ใช้งาน โดยส่วนนี้มีเป้าหมายในการออกแบบมาสำหรับผู้ใช้งานที่ไม่ได้ใช้บริการรถไฟสายตะวันออกเป็นประจำ ซึ่งอาจทำให้ไม่สามารถทราบชื่อสถานีรถไฟได้ ทั้งนี้ในส่วนนี้ก็จะแสดงสถานีอื่น ๆ ที่เหลือให้เลือกด้วยเช่นกัน ซึ่งเมื่อผู้ใช้งานเลือกสถานีรถไฟที่ต้องการแล้ว แอปพลิเคชันจะแสดงขบวนรถไฟทั้งหมดและเวลาที่รถไฟเข้าสถานีของสถานีรถไฟที่ผู้ใช้งานเลือก

นอกจากนั้น ยังได้มีการออกแบบและพัฒนาระบบการแจ้งเตือนเมื่อรถไฟใกล้มาถึงสถานี ซึ่งในหน้าแสดงตำแหน่งรถไฟนั้น จะมีปุ่มกดเพื่อให้ผู้ใช้งานเลือกสถานีรถไฟที่ต้องการให้มีการแจ้งเตือน ซึ่งเมื่อเลือกสถานีเรียบร้อยแล้ว รายการการแจ้งเตือนถูกจะแสดงในหน้าแสดงรายการการแจ้งเตือน เพื่อให้ผู้ใช้งานสามารถยกเลิกการแจ้งเตือนได้ ซึ่งหน้าแสดงรายการการแจ้งเตือนนั้นผู้ใช้งานสามารถเข้าถึงได้จากเมนูการแจ้งเตือนในหน้าหลักได้เช่นกัน โดยผู้พัฒนาได้ทำการออกแบบให้ระบบแจ้งเตือนไปยังผู้ใช้งานเมื่อรถไฟอยู่ห่างจากสถานีที่ผู้ใช้งานเลือก 2 สถานี โดยแผนผังลำดับของแต่ละหน้าดังรูปที่ 3.10

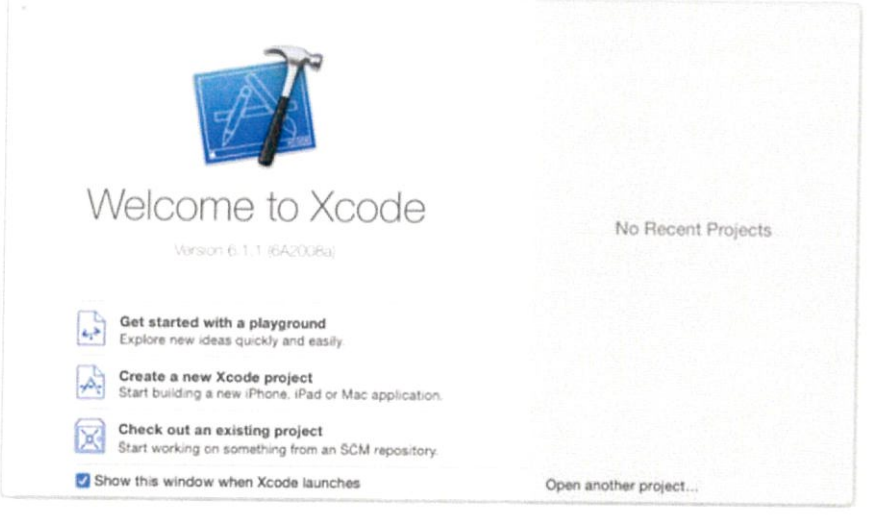


รูปที่ 3.10 แผนผัง User Interface ของแอปพลิเคชัน

3.1.3.2 การติดตั้งโปรแกรมเพื่อออกแบบแอปพลิเคชันสำหรับสมาร์ทโฟน

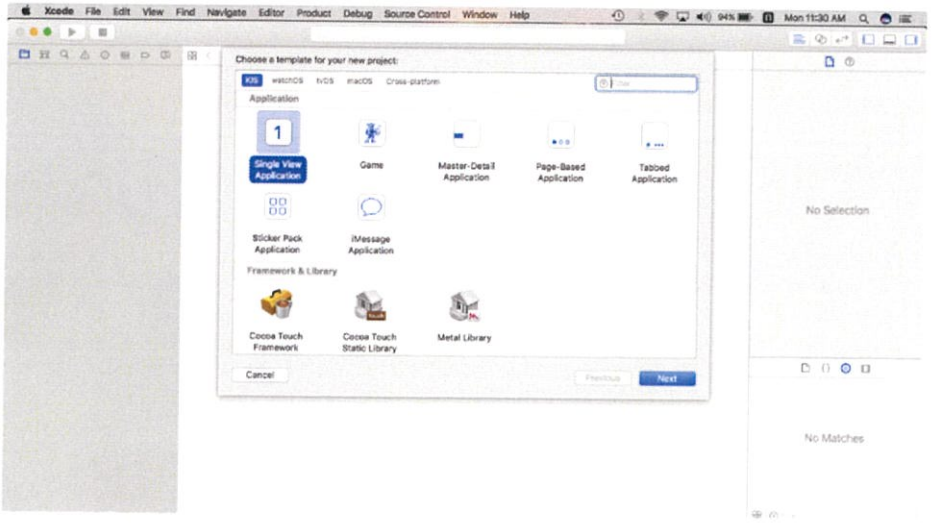
การสร้างแอปพลิเคชันสำหรับระบบปฏิบัติการไอโอเอสในปริญญานิพนธ์นี้จะใช้โปรแกรม Xcode ซึ่งภาษาที่ใช้ในการเขียนแอปพลิเคชันคือภาษา Swift โดยขั้นตอนในการสร้างแอปพลิเคชันสามารถแบ่งออกเป็น 3 ขั้นตอนดังนี้

1) ทำการดาวน์โหลดและติดตั้งโปรแกรม Xcode หลังจากนั้นทำการเปิดโปรแกรม ซึ่งจะแสดงได้ดังรูปที่ 3.11



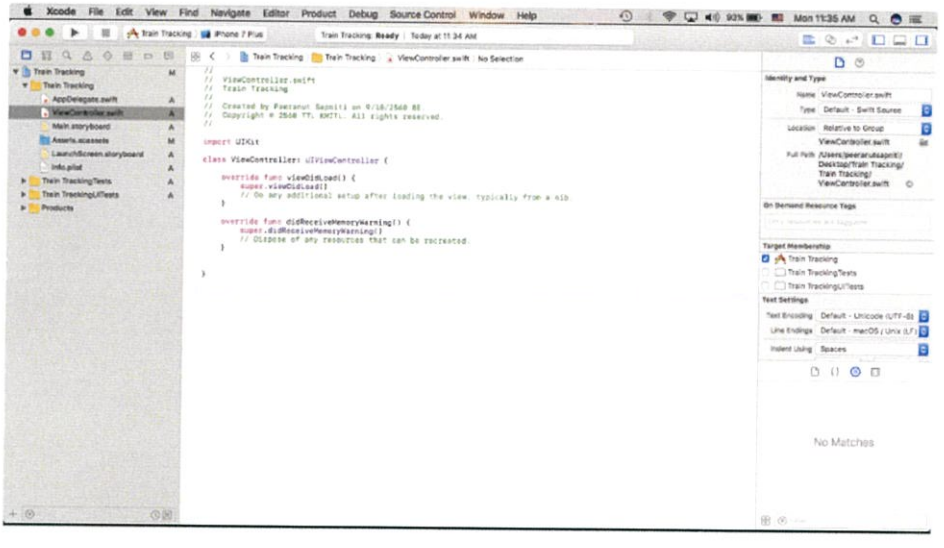
รูปที่ 3.11 โปรแกรม Xcode

2) เลือกเมนู Create a new Xcode project เพื่อเป็นการสร้างโปรเจคงานขึ้นมาใหม่ โดยเลือกเป็นแบบ single view application ซึ่งเป็นแม่แบบที่สร้างมาพร้อมกับหน้าเพียงหน้าเดียว สามารถนำมาประยุกต์ได้ง่าย ดังแสดงในรูปที่ 3.12

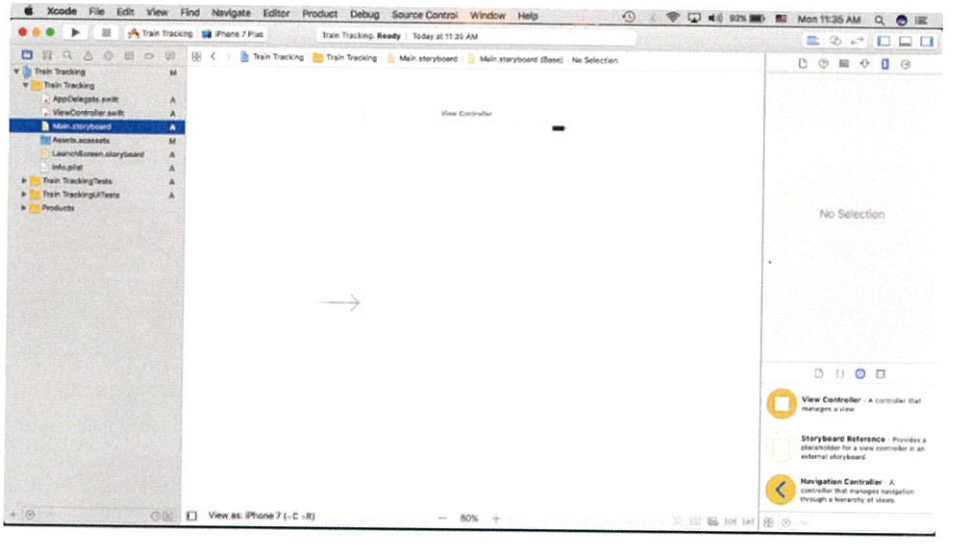


รูปที่ 3.12 การสร้างโปรเจคงานแบบ single view application

3) หลังจากทำการสร้างโปรเจคขึ้นมาใหม่แล้ว ตัวโปรแกรมจะมีทั้งหน้าต่างที่ไว้สำหรับเขียนคำสั่ง (ViewController.swift) และหน้าต่างที่ใช้สำหรับออกแบบตัวแอปพลิเคชัน (Main.storyboard) ดังแสดงในรูปที่ 3.13 และ รูปที่ 3.14



รูปที่ 3.13 หน้าต่างที่ใช้สำหรับเขียนคำสั่ง



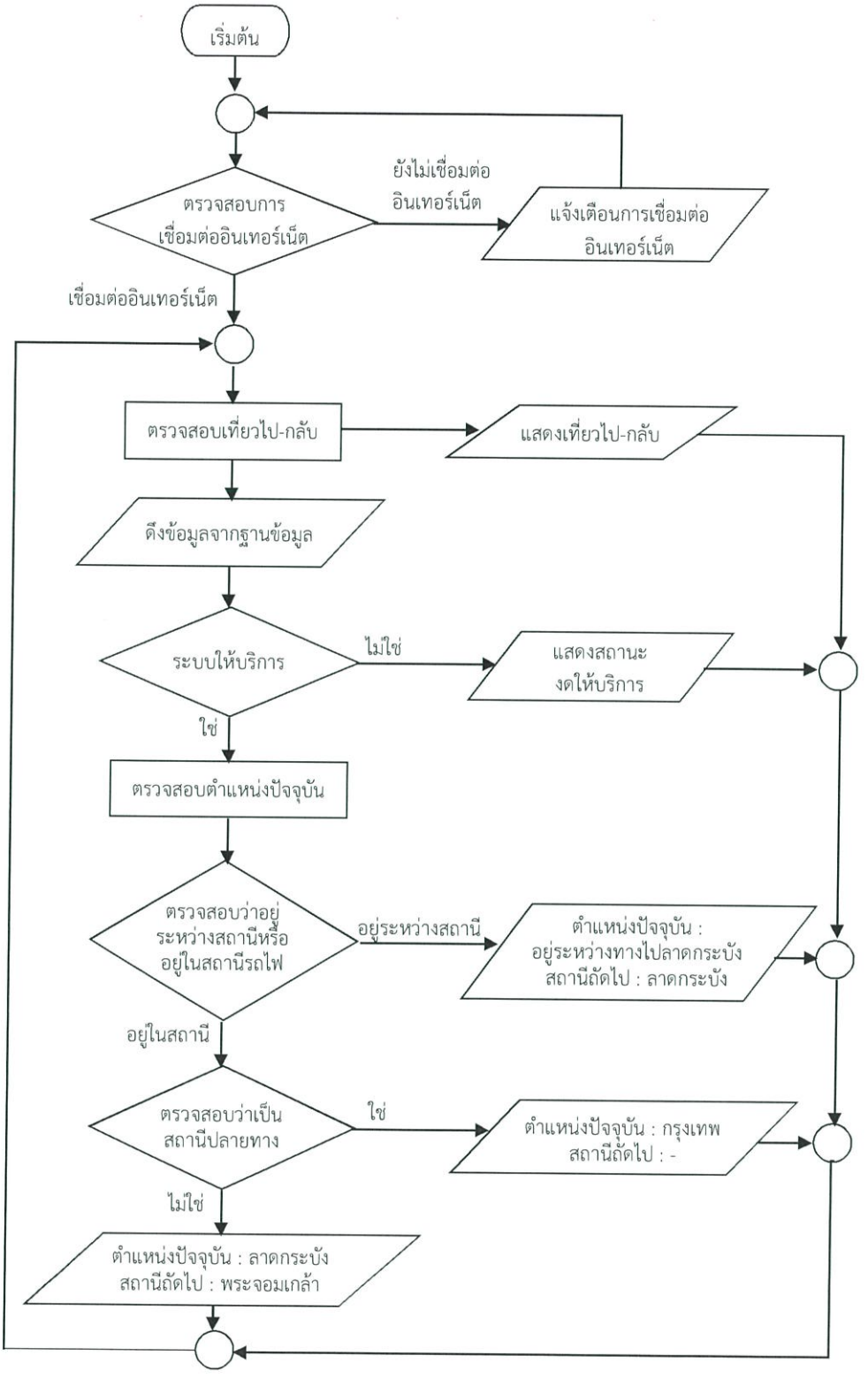
รูปที่ 3.14 หน้าต่างที่ใช้ในการสร้าง User Interface

3.1.3.3 การออกแบบการทำงานแอปพลิเคชันในหน้าแสดงตำแหน่งรถไฟ

สำหรับหน้าแสดงตำแหน่งรถไฟ ผู้วิจัยได้ทำการสร้างเงื่อนไขในการตัดสินใจเพื่อแสดงตำแหน่งของรถไฟในขณะนั้น โดยจะทำการตรวจสอบการเชื่อมต่ออินเทอร์เน็ตก่อน เมื่อเชื่อมต่ออินเทอร์เน็ตเรียบร้อยแล้ว แอปพลิเคชันจะดึงค่าพิกัดตำแหน่งปัจจุบันของรถไฟ สถานะการทำงาน วันที่ และเวลา มาจากฐานข้อมูล และนำมาเก็บไว้ในตัวแปรที่สร้างขึ้นเพื่อใช้ในการแสดงตำแหน่งของรถไฟ โดยจะทำการตรวจเช็คว่ารถไฟอยู่ในช่วงขาไปหรือขากลับโดยใช้เวลาและพิกัดเป็นตัวอ้างอิง จากนั้นจึงจะแสดงผลหมายเลขของรถไฟ ในที่นี้หากระบบติดตามรถไฟกำลังให้บริการจะเข้าเงื่อนไขการตรวจสอบตำแหน่งปัจจุบันของรถไฟ แต่ถ้าระบบติดตามรถไฟไม่ให้บริการจะแสดงข้อความสถานะงดให้บริการตามลำดับ

หลังจากตรวจสอบตำแหน่งปัจจุบันแล้ว จะตรวจสอบต่อว่าตำแหน่งนั้นเป็นตำแหน่งที่อยู่ในสถานีรถไฟหรือเป็นตำแหน่งที่อยู่ระหว่างสถานีรถไฟ ซึ่งหากเป็นตำแหน่งที่อยู่ระหว่างสถานีในช่องตำแหน่งปัจจุบันจะแสดงข้อความ “ อยู่ระหว่างทางไปสถานีถัดไป ” และช่องสถานีถัดไปจะแสดงชื่อของสถานีถัดไป ในตรงกันข้าม หากรถไฟอยู่ในสถานี จะมีการตรวจสอบอีกว่าเป็นสถานีปลายทางหรือไม่ ถ้าหากไม่ใช่สถานีปลายทาง ในช่องตำแหน่งปัจจุบันจะแสดงชื่อของสถานีปัจจุบันและช่องสถานีถัดไปจะแสดงชื่อของสถานีถัดไป แต่ถ้าหากเป็นสถานีปลายทาง ในช่องตำแหน่ง

ปัจจุบันจะแสดงชื่อของสถานีปัจจุบัน และช่องสถานีถัดไปจะแสดงเครื่องหมายขีดกลาง ซึ่งมีผังการทำงานดังรูปที่ 3.15 ซึ่งในหน้าแสดงตำแหน่งรถไฟของรถไฟขบวนอื่น ๆ จะมีการทำงานเช่นเดียวกับการแสดงตำแหน่งของรถไฟขบวนที่ 275



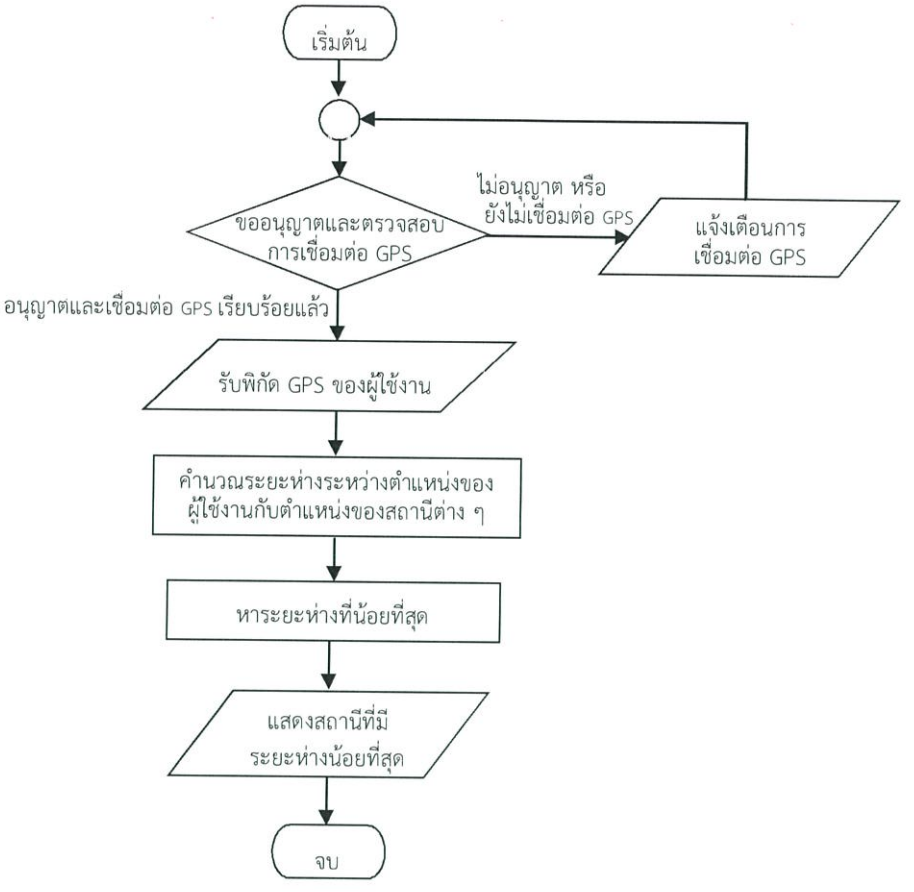
รูปที่ 3.15 แผนผังการทำงานในหน้าแสดงตำแหน่งรถไฟ

3.1.3.4 ออกแบบการทำงานแอปพลิเคชันในหน้าแสดงสถานีที่ใกล้ตัวผู้ใช้งานที่สุด

สำหรับหน้าแสดงตำแหน่งสถานีรถไฟที่อยู่ใกล้ผู้ใช้งานมากที่สุด ผู้วิจัยได้ทำการสร้างเงื่อนไขในการตัดสินใจเพื่อแสดงสถานีรถไฟที่ใกล้ตัวผู้ใช้งานที่สุดในขณะนั้น โดยจะทำการเรียกร้องพิกัดของผู้ใช้งานจาก GPS บนมือถือ เมื่อได้พิกัดของผู้ใช้งานเรียบร้อยแล้ว แอปพลิเคชันจะคำนวณระยะห่างระหว่างตำแหน่งของผู้ใช้งานกับตำแหน่งของสถานีต่าง ๆ โดยใช้สมการ Harversine ในการคำนวณระยะห่างระหว่างจุดสองจุด ตามสมการที่ 3.2 [21] โดย d คือระยะห่างระหว่างตำแหน่งของผู้ใช้งานกับตำแหน่งของสถานี r คือรัศมีของโลก ซึ่งมีค่าประมาณ 6,371 กิโลเมตร φ_1, λ_1 คือละติจูดและลองจิจูดของตำแหน่งผู้ใช้งานตามลำดับ φ_2, λ_2 คือละติจูดและลองจิจูดของตำแหน่งสถานีตามลำดับ

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (3.2)$$

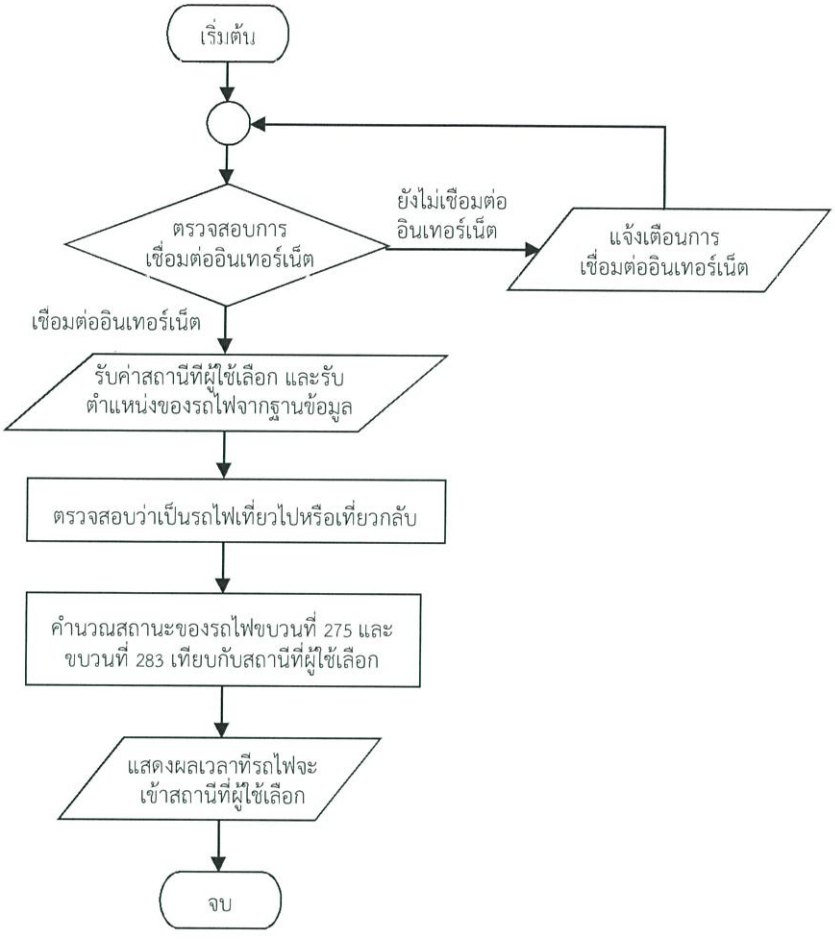
โดยเมื่อได้ระยะห่างระหว่างตำแหน่งของผู้ใช้งานกับตำแหน่งของสถานีต่าง ๆ แล้ว จึงนำมาหาระยะห่างที่น้อยที่สุดเพื่อตัดสินใจว่าผู้ใช้งานอยู่ใกล้สถานีใดมากที่สุด ซึ่งมีผังการทำงานแสดงดังรูปที่ 3.16



รูปที่ 3.16 แผนผังการทำงานในหน้าแสดงสถานีที่ใกล้ตัวผู้ใช้งานที่สุด

3.1.3.5 ออกแบบการทำงานแอปพลิเคชันในหน้าแสดงขบวนรถไฟที่กำลังเข้าสถานีรถไฟ

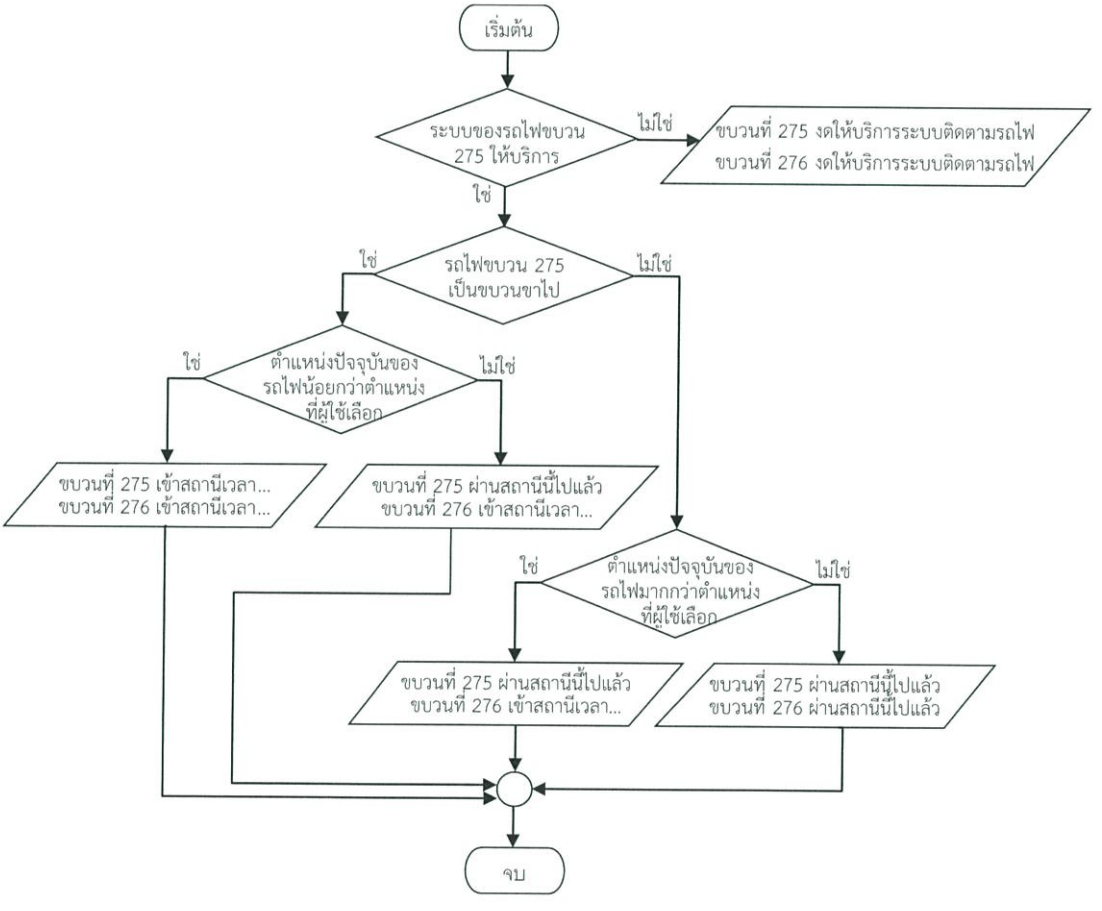
เมื่อผู้ใช้งานกดเลือกสถานีที่ต้องการทราบเวลาที่รถไฟจะมาถึง จากนั้นแอปพลิเคชันจะแสดงหมายเลขขบวนรถไฟทั้งหมดที่กำลังเข้าสู่สถานีดังกล่าว เงื่อนไขในการตัดสินใจเพื่อแสดงขบวนรถไฟและเวลาที่รถไฟจะเข้าสู่สถานีใด ๆ แสดงดังแผนผังการทำงานในรูปที่ 3.17 เริ่มจากทำการตรวจสอบการเชื่อมต่ออินเทอร์เน็ตก่อน เมื่อเชื่อมต่ออินเทอร์เน็ตเรียบร้อยแล้ว แอปพลิเคชันจะรับค่าสถานีที่ผู้ใช้เลือก พิกัดตำแหน่งปัจจุบันของรถไฟ และสถานะการทำงานมาจากฐานข้อมูลนำมาเก็บไว้ในตัวแปรที่สร้างขึ้น จากนั้นจะตรวจสอบว่ารถไฟเป็นขบวนขาไปหรือขากลับโดยใช้ตำแหน่งของรถไฟและเวลาปัจจุบันเป็นตัวอ้างอิง จากนั้นจะคำนวณสถานะของรถไฟขบวนที่ 275 และ 283 เทียบกับสถานีที่ผู้ใช้เลือก เมื่อคำนวณสถานะเรียบร้อยแล้ว จึงแสดงผลเวลาที่รถไฟจะเข้าสถานีที่ผู้ใช้เลือก



รูปที่ 3.17 แผนผังการทำงานในหน้าแสดงขบวนรถไฟที่กำลังเข้าสถานี

ในส่วนของการคำนวณสถานะของรถไฟขบวนที่ 275 เทียบกับสถานีที่ผู้ใช้เลือก จะมีการคำนวณสถานะเหมือนกันกับการคำนวณสถานะของรถไฟขบวนที่ 283 เทียบกับสถานีที่ผู้ใช้เลือก ซึ่งขบวน 275 จะเป็นขบวนขาไป เดินทางจากกรุงเทพฯไปยังอรัญประเทศ และขบวน 276 เป็นขบวนขากลับ เดินทางจากอรัญประเทศไปยังกรุงเทพฯ โดยจะเริ่มจากการตรวจสอบว่าระบบติดตามรถไฟขบวน 275 ให้บริการหรือไม่ ถ้าระบบติดตามรถไฟขบวน 275 ไม่ให้บริการ แอปพลิเคชันจะแสดงข้อความ “ขบวนที่ 275 งดให้บริการระบบติดตามรถไฟ ขบวนที่ 276 งดให้บริการระบบติดตามรถไฟ” ในทางตรงกันข้าม หากระบบติดตามรถไฟขบวน 275 ให้บริการ จะตรวจสอบว่ารถไฟขบวน 275 ที่วิ่งอยู่ในขณะนั้นเป็นขบวนขาไปหรือขบวนขากลับ โดยทำการตรวจสอบด้วยเงื่อนไขการเปรียบเทียบตำแหน่งปัจจุบันของรถไฟเทียบกับตำแหน่งของรถไฟเมื่อ 1 ชั่วโมงที่แล้ว

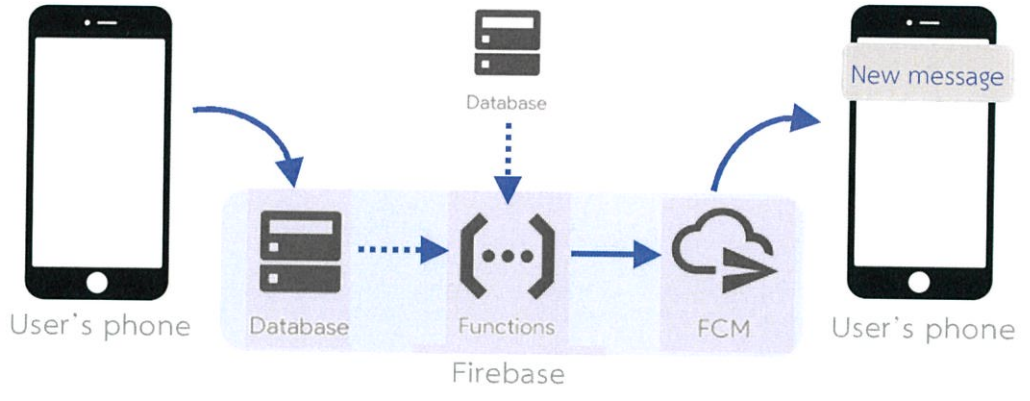
ถ้าหากตำแหน่งปัจจุบันของรถไฟมากกว่าตำแหน่งของรถไฟเมื่อ 1 ชั่วโมงที่แล้ว จะเป็นขบวนขาไป แต่ถ้าหากน้อยกว่า จะเป็นขบวนขากลับ ซึ่งหากเป็นขบวนขาไป จะตรวจสอบต่อไปว่าตำแหน่งปัจจุบันของรถไฟน้อยกว่าตำแหน่งที่ผู้ใช้เลือกหรือไม่ หากใช่แอปพลิเคชันจะแสดงข้อความ “ขบวนที่ 275 เข้าสถานีเวลา... ขบวนที่ 276 เข้าสถานีเวลา...” แต่ถ้าหากไม่ใช่ แอปพลิเคชันจะแสดงข้อความ “ขบวนที่ 275 ผ่านสถานีนี้ไปแล้ว ขบวนที่ 276 เข้าสถานีเวลา...” อย่างไรก็ตาม ถ้าหากรถไฟไม่ใช่ขบวนขาไป จะตรวจสอบต่อไปว่าตำแหน่งปัจจุบันของรถไฟมากกว่าตำแหน่งที่ผู้ใช้เลือกหรือไม่ หากใช่ แอปพลิเคชันจะแสดงข้อความ “ขบวนที่ 275 ผ่านสถานีนี้ไปแล้ว ขบวนที่ 276 เข้าสถานีเวลา...” หากไม่ใช่ แอปพลิเคชันจะแสดงข้อความ “ขบวนที่ 275 ผ่านสถานีนี้ไปแล้ว ขบวนที่ 276 ผ่านสถานีนี้ไปแล้ว” แสดงดังแผนผังการทำงานในรูปที่ 3.18



รูปที่ 3.18 แผนผังการคำนวณสถานะของรถไฟขบวนที่ 275 เทียบกับสถานีที่ผู้ใช้เลือก

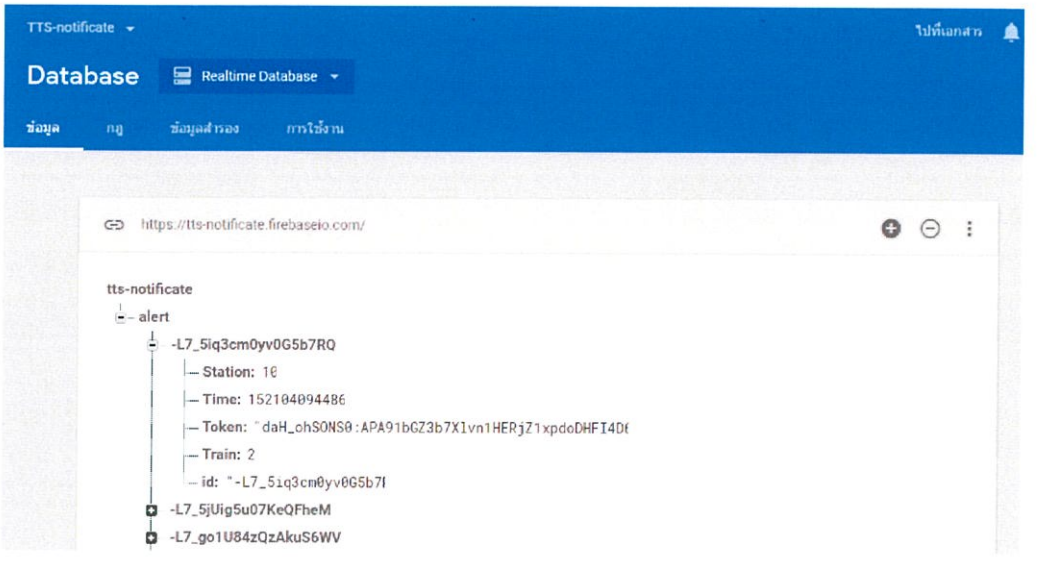
3.1.3.6 ออกแบบระบบแจ้งเตือนสถานีรถไฟบนแอปพลิเคชันในระบบปฏิบัติการไอโอเอส

ในการออกแบบระบบแจ้งเตือนบนแอปพลิเคชันนั้น จะใช้บริการ Firebase ซึ่งเป็นบริการของ google โดยจะใช้ Firebase database ซึ่งเป็น cloud database ในการเก็บข้อมูลของขบวนรถไฟและสถานีรถไฟที่ผู้ใช้เลือก และใช้ Cloud Functions for Firebase ในการเก็บฟังก์ชันการทำงานของโปรแกรม โดยเมื่อ deploy โค้ทขึ้นไปใน Cloud Functions for Firebase แล้ว Cloud Functions จะรอรับ trigger จากบริการของ Firebase และจะทำงานตามฟังก์ชันที่เขียนไว้ ซึ่งก็คือ ดึงข้อมูลตำแหน่งปัจจุบันของรถไฟจากดาต้าเบสของระบบติดตามรถไฟ และส่งการแจ้งเตือนไปยังผู้ใช้ โดยใช้ Firebase Cloud Messaging (FCM) โดยระบบจะมีบล็อกไดอะแกรมการทำงานดังรูปที่ 3.19



รูปที่ 3.19 บล็อกไดอะแกรมระบบแจ้งเตือนบนแอปพลิเคชัน

โดยเมื่อผู้ใช้กดเลือกขบวนรถไฟและสถานีที่ต้องการให้ระบบแจ้งเตือนในแอปพลิเคชันแล้ว แอปพลิเคชันจะส่งค่าไปยังฐานข้อมูลใน Firebase โดยข้อมูลใน Firebase จะแยกเป็นรายการเป็นชุด ๆ โดยในชุดของข้อมูลในฐานข้อมูลนั้นจะประกอบไปด้วย Station คือ หมายเลขสถานีรถไฟที่ผู้ใช้ต้องการให้แจ้งเตือน ในที่นี้ สถานีที่ 10 จะหมายถึงสถานีพระจอมเกล้า Time คือ เวลาที่ผู้ใช้งานส่งข้อมูลมายัง Firebase ซึ่งจะมีหน่วยเป็นมิลลิวินาที Token คือ รหัสของโทรศัพท์ของผู้ใช้ Train คือ หมายเลขขบวนรถไฟที่ผู้ใช้ต้องการให้แจ้งเตือน ในที่นี้หมายเลขขบวนรถไฟที่ 2 จะหมายถึงรถไฟขบวนที่ 283 กรุงเทพ-บ้านพลูตาหลวง และ id คือ รหัสที่สุ่มขึ้นมาเพื่อแยกชุดข้อมูลออกเป็นรายการต่าง ๆ ดังรูปที่ 3.20

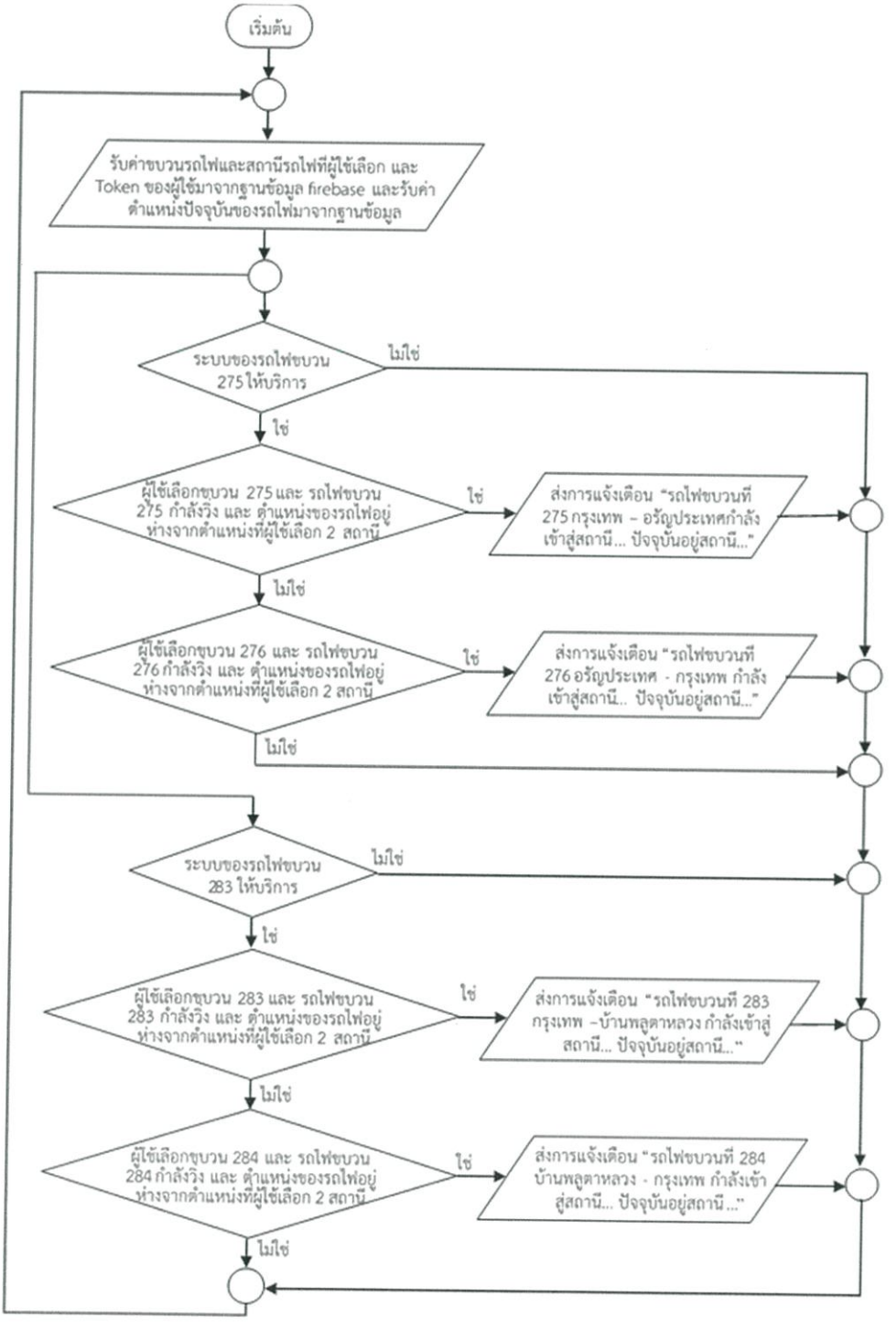


รูปที่ 3.20 ข้อมูลต่าง ๆ ที่เก็บในฐานข้อมูลของ Firebase

ในส่วนของ cloud function จะใช้ NodeJS ในการเขียน ซึ่งได้ทำการสร้างเงื่อนไขในการตัดสินใจเพื่อส่งการแจ้งเตือนไปยังแอปพลิเคชันในระบบปฏิบัติการไอโอเอส โดยเริ่มจากรับค่าขบวนรถไฟและสถานีรถไฟที่ผู้ใช้เลือก และ Token ของผู้ใช้มาจากฐานข้อมูล Firebase และรับค่าตำแหน่งปัจจุบันของรถไฟมาจากฐานข้อมูล โดยขั้นแรกจะมีการตรวจสอบว่าระบบติดตามรถไฟของรถไฟขบวน 275 ให้บริการหรือไม่ หากให้บริการ จะตรวจสอบต่อไปว่าผู้ใช้เลือกขบวน 275 และรถไฟขบวน 275 กำลังวิ่ง และตำแหน่งของรถไฟอยู่ห่างจากตำแหน่งที่ผู้ใช้เลือก 2 สถานีหรือไม่ ถ้าใช่ จะส่งการแจ้งเตือน “รถไฟขบวนที่ 275 กรุงเทพ – อรัญประเทศกำลังเข้าสู่สถานี... ปัจจุบันอยู่สถานี...” หากไม่ใช่ จะตรวจสอบต่อว่าผู้ใช้เลือกขบวน 276 และ รถไฟขบวน 276 กำลังวิ่ง และตำแหน่งของรถไฟอยู่ห่างจากตำแหน่งที่ผู้ใช้เลือก 2 สถานีหรือไม่ หากใช่จะส่งการแจ้งเตือน “รถไฟขบวนที่ 276 อรัญประเทศ - กรุงเทพ กำลังเข้าสู่สถานี... ปัจจุบันอยู่สถานี...”

ในขั้นตอนต่อไป จะเป็นการตรวจสอบขบวนรถไฟ 283-284 โดยเริ่มต้นจากการตรวจสอบว่า ระบบติดตามรถไฟของรถไฟขบวน 283 ให้บริการหรือไม่ หากให้บริการ จะตรวจสอบว่า ผู้ใช้เลือกขบวน 283 และ รถไฟขบวน 283 กำลังวิ่ง และ ตำแหน่งของรถไฟอยู่ห่างจากตำแหน่งที่ผู้ใช้เลือก 2 สถานีหรือไม่ หากใช่ จะส่งการแจ้งเตือน “รถไฟขบวนที่ 283 กรุงเทพ –บ้านพลูตาหลวงกำลังเข้าสู่สถานี... ปัจจุบันอยู่สถานี...” หากไม่ใช่จะตรวจสอบว่าผู้ใช้เลือกขบวน 284 และรถไฟขบวน 284 กำลังวิ่ง และ ตำแหน่งของรถไฟอยู่ห่างจากตำแหน่งที่ผู้ใช้เลือก 2 สถานีหรือไม่ หากใช่จะส่งการแจ้งเตือน “รถไฟขบวนที่ 284 บ้านพลูตาหลวง - กรุงเทพ กำลังเข้าสู่สถานี... ปัจจุบันอยู่สถานี...” เมื่อส่งข้อความแจ้งเตือนเรียบร้อยแล้ว หรือระบบติดตามมรถไฟขบวนที่ 275

และ 283 ไม่ให้บริการ หรือเมื่อตรวจสอบเงื่อนไขแล้วไม่ตรงตามเงื่อนไข ระบบจะวนกลับไปรับค่าจากฐานข้อมูลและตรวจสอบซ้ำ ซึ่งมีผังการทำงานดังรูปที่ 3.21



รูปที่ 3.21 แผนผังการทำงานในการตัดสินใจเพื่อส่งการแจ้งเตือนไปยังแอปพลิเคชัน

3.1.3.7 ออกแบบการติดตั้งแอปพลิเคชันบน App Store

การติดตั้งแอปพลิเคชันบน App Store จำเป็นต้องมีการสมัคร Developer Account เพื่อที่จะสามารถอัพไฟล์จาก Xcode ขึ้น App Store ได้ ข้อมูลรายละเอียดเกี่ยวกับแอปพลิเคชันจะแบ่งเป็นส่วนๆ โดยส่วนแรกสุดคือ ส่วน App Information ดังรูปที่ 3.22

Name: ชื่อแอปพลิเคชันที่จะแสดงใน App Store

Subtitle: สรุปแอปพลิเคชัน ที่จะปรากฏใช้ชื่อแอปพลิเคชันบน App Stpre

Privacy Policy URL: ลิงค์ URL ที่จะไปลิงค์กับนโยบายความเป็นส่วนตัวขององค์กร

Bundle ID: Bundle ID ต้องตรงกับ Bundle ID ในโปรแกรม Xcode

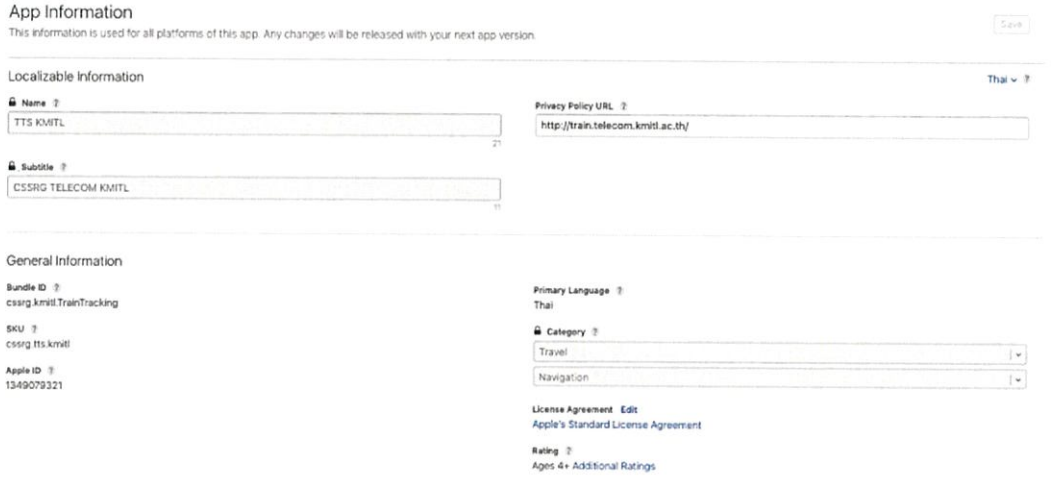
SKU: หมายเลขเฉพาะสำหรับแอปพลิเคชันนี้ ซึ่งจะไม่ปรากฏใน App store

Apple ID: หมายเลขที่สร้างขึ้นอัตโนมัติให้กับแอปพลิเคชัน

Primary Language: หากไม่มีข้อมูลแอปพิเคชันที่แปลเป็นภาษาท้องถิ่นใน App Store ระบบจะใช้ข้อมูลจากภาษาหลักแทน

Category: หมวดหมู่ที่อธิบายถึงแอปพลิเคชันนี้

Rating: การประเมินแอปพลิเคชันว่าเหมาะสำหรับผู้ใช้อายุเท่าใด



รูปที่ 3.22 การออกแบบ App Information ใน App Store

ต่อมาเป็นส่วนของรายละเอียดแอปพลิเคชัน เช่น App Previews and Screenshots, Promotional Text, Description, Keywords, Support URL, Marketing URL ดังรูปที่ 3.23

App Previews and Screenshots: ภาพตัวอย่างการใช้งานแอปพลิเคชัน โดยต้องใส่ภาพหน้าจอหรือสามารถทำภาพขึ้นมาเพื่อใส่ในส่วนนี้ได้ เช่น ภาพประกอบที่มีข้อความอธิบายวิธีการใช้งานแอปพลิเคชันบนหน้าจอเพื่อให้ผู้ใช้งานเข้าใจได้ง่ายขึ้น

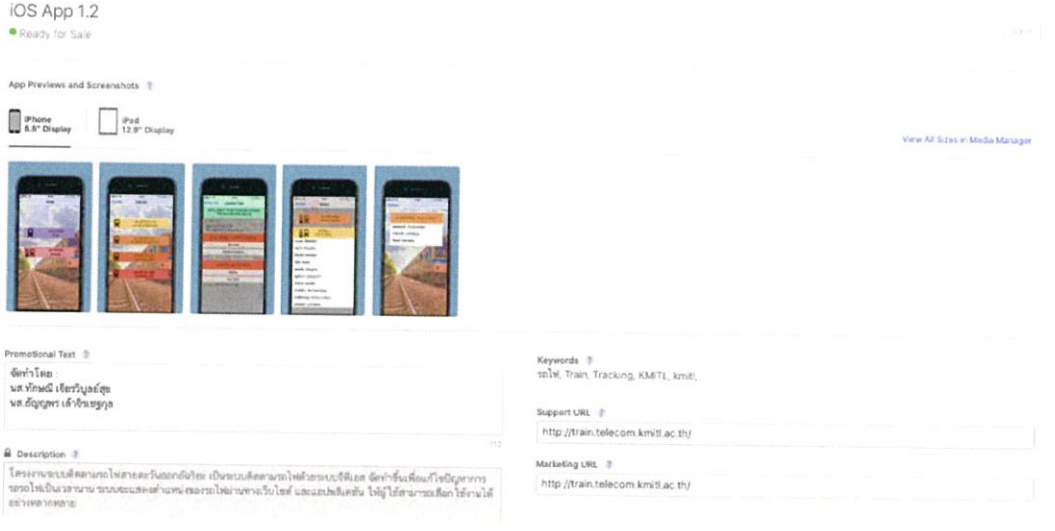
Promotional Text: ข้อความส่งเสริมการขายช่วยให้สามารถแจ้งผู้เข้าชมเกี่ยวกับคุณลักษณะของแอปพลิเคชันปัจจุบันได้โดยไม่ต้องส่งการอัพเดต

Description: คำอธิบายเกี่ยวกับรายละเอียดของแอปพลิเคชันที่มีรายละเอียดคุณลักษณะ และฟังก์ชันการทำงาน

Keywords: คำที่อธิบายถึงแอปพลิเคชัน keyword จะทำให้ผลการค้นหาแอปพลิเคชันมีความถูกต้องมากขึ้น

Support URL: URL ที่สนับสนุนการทำงานของแอปพลิเคชัน ซึ่ง URL นี้จะปรากฏบน App Store

Marketing URL: URL ที่มีข้อมูลทางการตลาดเกี่ยวกับแอปพลิเคชัน ซึ่ง URL นี้จะปรากฏบน App Store



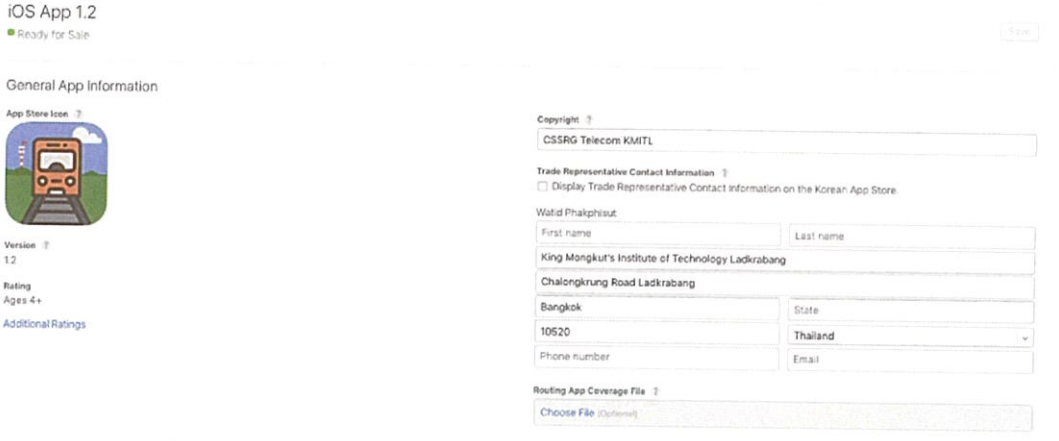
รูปที่ 3.23 การออกแบบรายละเอียดของแอปพลิเคชันใน App Store (1)

และยังมีส่วนของรายละเอียดของแอปพลิเคชันเพิ่มเติมอีก เช่น App Store Icon, version, Copyright ดังรูปที่ 3.24

App Store Icon: ภาพ Icon ของแอปพลิเคชันขนาด 1024 x 1024 PX เพื่อใช้แสดงใน App Store

Version: หมายเลขเวอร์ชันของแอปพลิเคชัน

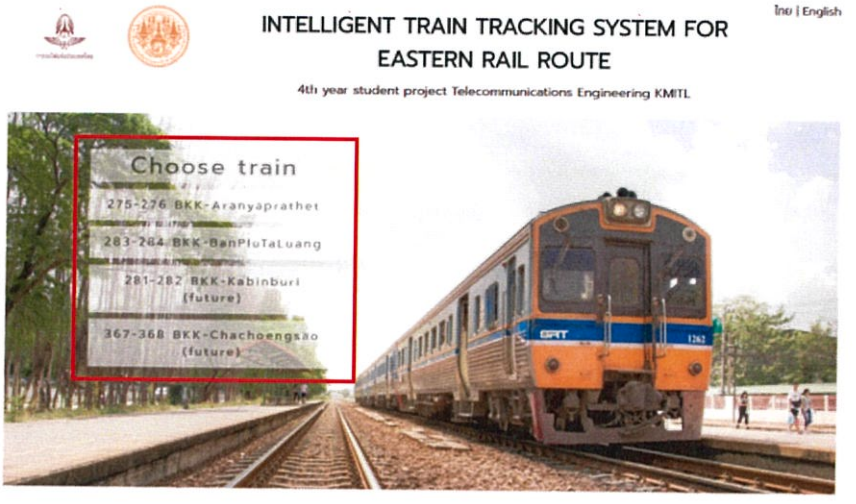
Copyright: ชื่อบุคคลหรือนิติบุคคลที่เป็นเจ้าของสิทธิ์แต่เพียงผู้เดียว



รูปที่ 3.24 การออกแบบรายละเอียดของแอปพลิเคชันใน App Store (2)

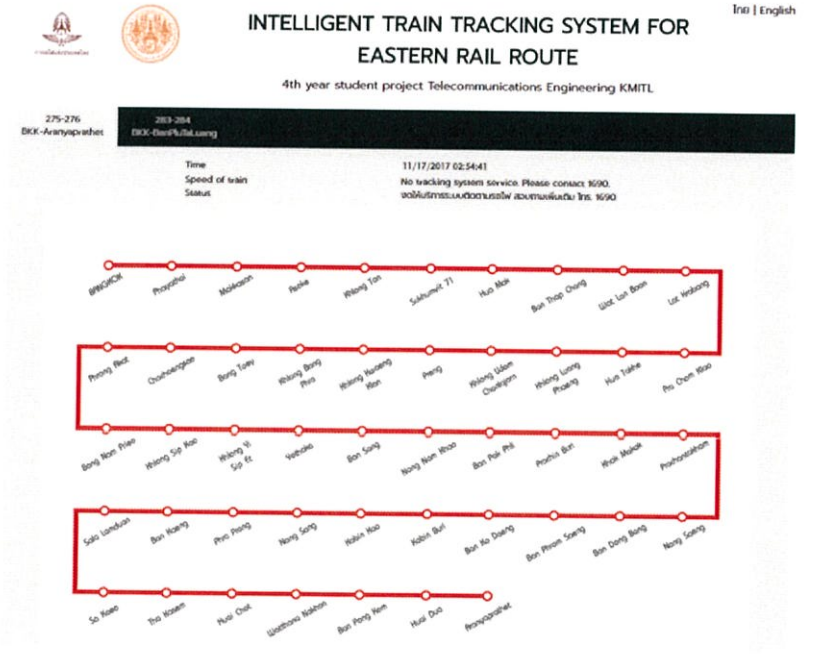
3.1.4 การออกแบบเว็บไซต์

ในที่นี้ ผู้วิจัยได้มีการออกแบบการแสดงตำแหน่งรถไฟบนเว็บไซต์ในหน้าภาษาอังกฤษ ซึ่งจะมีการบอกตำแหน่งปัจจุบันของรถไฟขบวนต่าง ๆ โดยผู้ใช้งานสามารถเข้าไปเลือกได้ที่หน้าหลักของเว็บไซต์ ดังแสดงในรูปที่ 3.25

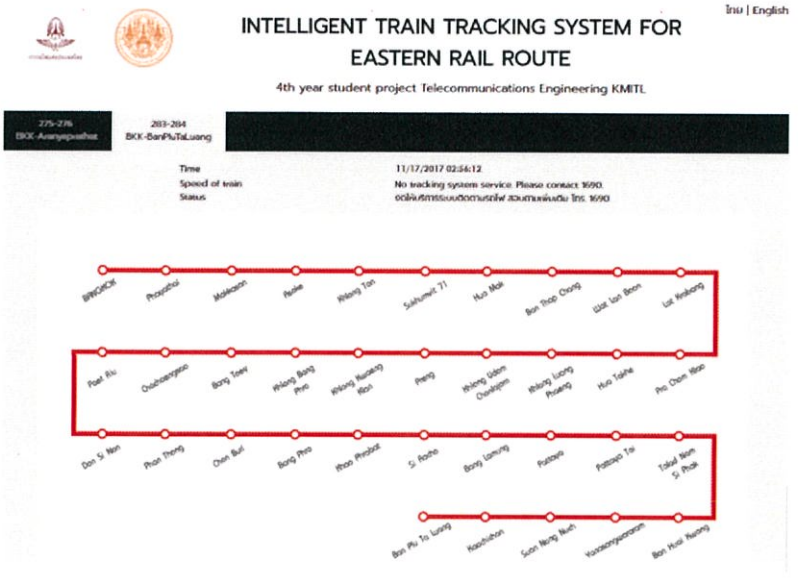


รูปที่ 3.25 หน้าหลักภาษาอังกฤษของหน้าเว็บไซต์แสดงผล

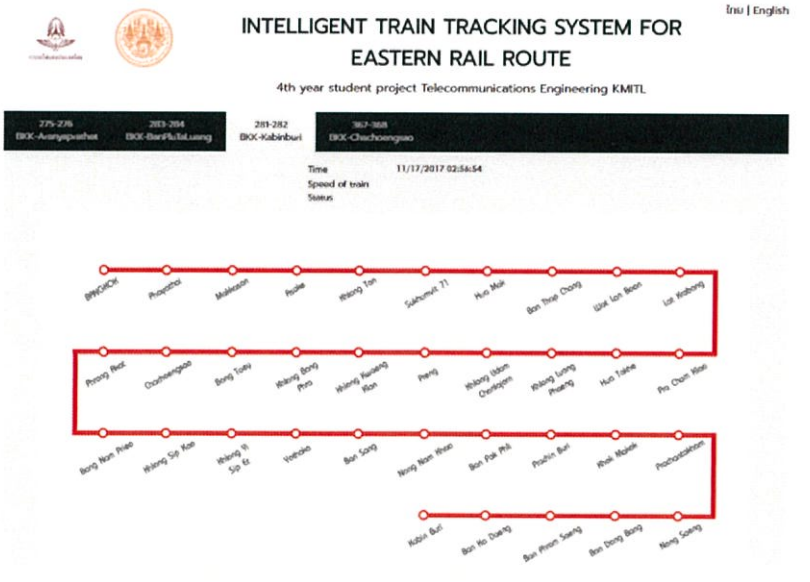
เมื่อผู้ใช้งานเลือกขบวนรถไฟจากรูปที่ 3.25 เว็บไซต์จะลิงค์ผู้ใช้งานเข้าสู่หน้าแสดงตำแหน่งของขบวนรถไฟ ซึ่งจะประกอบไปด้วยส่วนของการแสดงผลตำแหน่งรถไฟเป็นภาษาอังกฤษของรถไฟขบวน 275-276 กรุงเทพฯ-อรัญประเทศ หน้าแสดงตำแหน่งขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง หน้าแสดงตำแหน่งขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และหน้าแสดงตำแหน่งขบวน 367-368 กรุงเทพฯ-ฉะเชิงเทรา ดังรูปที่ 3.26 3.27 3.28 และ 3.29 ตามลำดับ



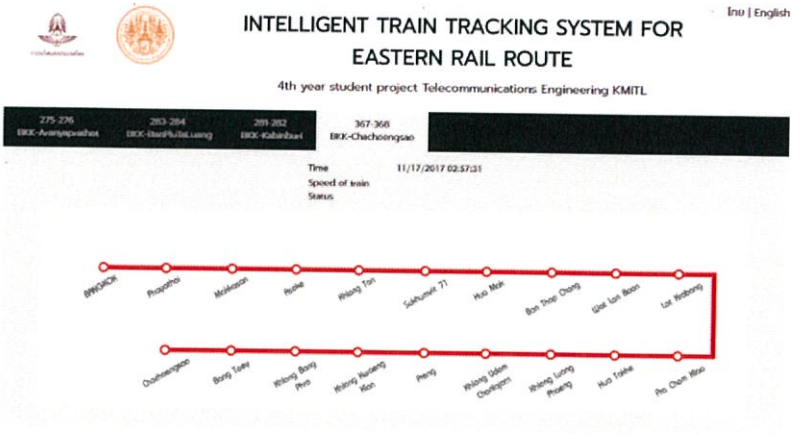
รูปที่ 3.26 หน้าภาษาอังกฤษแสดงตำแหน่งขบวน 275-276 กรุงเทพฯ-อรัญประเทศ



รูปที่ 3.27 หน้าภาษาอังกฤษแสดงตำแหน่งขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง



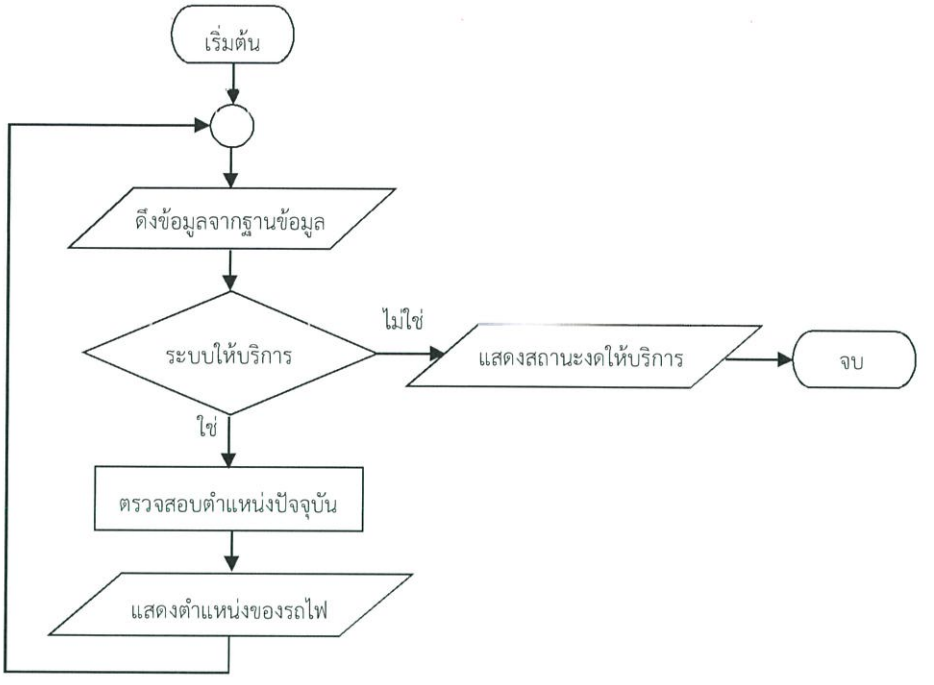
รูปที่ 3.28 หน้าภาษาอังกฤษแสดงตำแหน่งขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี



รูปที่ 3.29 หน้าภาษาอังกฤษแสดงตำแหน่งขบวน ขบวน 367-368 กรุงเทพ-ฉะเชิงเทรา

ซึ่งในแต่ละหน้านั้น สามารถกดเลือกรถไฟสายอื่น ๆ ได้โดยตรง โดยไม่จำเป็นต้องกดกลับสู่หน้าหลัก ซึ่งในขณะนี้ ได้เปิดให้ผู้ใช้สามารถกดเข้ามาในหน้าแสดงตำแหน่งรถไฟได้ 2 ขบวน คือขบวนที่ 275-276 และขบวนที่ 283-284

โดยการระบุตำแหน่งของรถไฟบนเว็บไซต์นั้น ผู้วิจัยใช้ภาษา JavaScript ในการเชื่อมต่อกับ PHP เพื่อนำค่าพิกัดของรถไฟมาประมวลผลตามเงื่อนไข โดยจะทำการดึงค่าพิกัดตำแหน่งปัจจุบันของรถไฟ สถานะการทำงาน ความเร็ว วันที่ และเวลา มาจากฐานข้อมูล และนำมาเก็บไว้ในตัวแปรที่สร้างขึ้นเพื่อใช้ในการแสดงตำแหน่งของรถไฟ จากนั้นถ้าระบบติดตามรถไฟให้บริการจะเข้าเงื่อนไขการตรวจสอบตำแหน่งปัจจุบันของรถไฟ แต่ถ้าหากระบบติดตามรถไฟไม่ให้บริการจะแสดงข้อความสถานะงดให้บริการตามลำดับ เมื่อตรวจสอบตำแหน่งปัจจุบันแล้ว จึงแสดงผลเป็นหมุดบนหน้าเว็บ ซึ่งมีแผนผังการทำงานดังรูปที่ 3.30

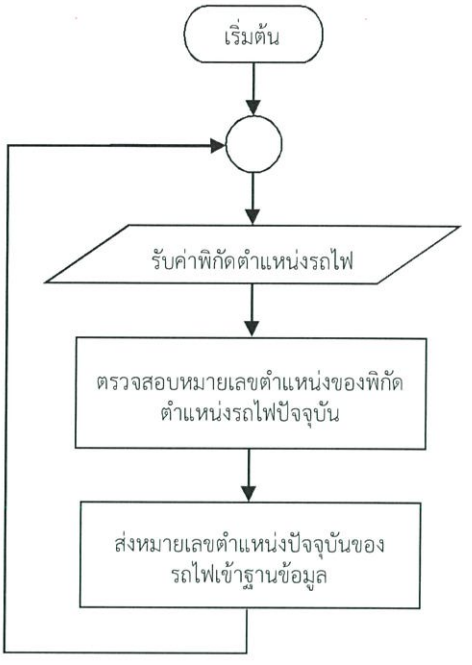


รูปที่ 3.30 แผนผังการทำงานของการระบุตำแหน่งของรถไฟบนเว็บไซต์

3.1.5 ออกแบบระบบการคำนวณตำแหน่งของรถไฟเพื่อลดภาระการทำงานของแอปพลิเคชัน

เนื่องจากการคำนวณตำแหน่งของรถไฟที่ใช้งานอยู่นั้น ได้มีการคำนวณตำแหน่งของรถไฟอยู่บนแอปพลิเคชัน ซึ่งส่งผลให้แอปพลิเคชันที่ถูกติดตั้งอยู่บนสมาร์ทโฟนทุก ๆ เครื่องนั้น ต้องมีการคำนวณตำแหน่งรถไฟเดียวกันแยกจากกัน โดยภาระการคำนวณนั้นจะไปตกอยู่ที่อุปกรณ์ของผู้ใช้งานเป็นหลัก ดังนั้นจึงได้มีการพัฒนาระบบการคำนวณตำแหน่งของรถไฟบนฐานข้อมูลเป็นหลัก เพื่อลดภาระการทำงานของแอปพลิเคชันของผู้ใช้งาน ในที่นี้ระบบฐานข้อมูลทำการคำนวณตำแหน่งของรถไฟเพียงครั้งเดียว โดยแอปพลิเคชันสามารถนำค่าดังกล่าวไปแสดงได้โดยตรง

ภาษาที่ใช้ในการเขียนโปรแกรมคำนวณตำแหน่งของรถไฟคือ ภาษา PHP โดยมีขั้นตอนการทำงานเริ่มจากรับค่าพิกัดตำแหน่งของรถไฟมาจาก Arduino และนำมาเข้าเงื่อนไขการตรวจสอบหมายเลขตำแหน่งของพิกัดตำแหน่งรถไฟปัจจุบัน ซึ่งเป็นหมายเลขที่แทนตำแหน่งสถานีรถไฟหรือตำแหน่งระหว่างสถานี เมื่อได้หมายเลขตำแหน่งปัจจุบันแล้ว โปรแกรมจะส่งหมายเลขตำแหน่งปัจจุบันของรถไฟเข้าฐานข้อมูล ซึ่งมีผังการทำงานดังรูปที่ 3.31



รูปที่ 3.31 แผนผังการคำนวณตำแหน่งของรถไฟ

3.1.6 การออกแบบระบบการประมาณเวลาถึงสถานีของรถไฟโดยใช้หลักการ K-Nearest Neighbors (K-NN)

เนื่องจากในระบบติดตามรถไฟที่ใช้งานอยู่ในปัจจุบันนั้น ผู้ใช้งานส่วนใหญ่ไม่ทราบเวลาที่รถไฟจะเข้าถึงสถานีต่าง ๆ ที่แน่นอน เนื่องจากในแอปพลิเคชันหรือเว็บไซต์จะแสดงเฉพาะตำแหน่งของรถไฟโดยคร่าวเท่านั้น ฉะนั้นในโครงการนี้จึงได้มีการออกแบบระบบการประมาณเวลาถึงสถานีของรถไฟ เพื่อให้ผู้ใช้งานได้ทราบเวลาที่รถไฟจะถึงสถานีต่าง ๆ และเป็นประโยชน์สำหรับการวางแผนเดินทาง การออกแบบในครั้งนี้จะใช้หลักการ K-Nearest Neighbors (K-NN) ในการจัดแบ่งคลาส และคำนวณหาคลาสที่เหมาะสมและใกล้เคียงกับข้อมูลที่ต้องการพิจารณามากที่สุด

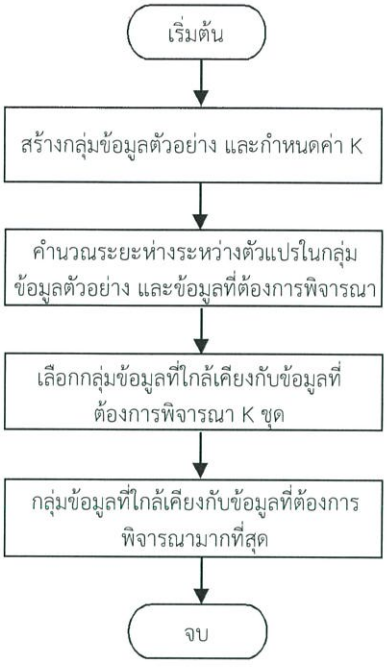
วิธีการที่ผู้วิจัยดำเนินการแสดงดังแผนผังการทำงานในรูปที่ 3.32 โดยวิธีการหาคลาสที่ใกล้เคียงกับข้อมูลที่ต้องการพิจารณามากที่สุดนั้น ขั้นแรกทำการสร้างกลุ่มข้อมูลตัวอย่างที่จะใช้ในการประมาณเวลาถึงสถานีของรถไฟ ซึ่งสำหรับรถไฟขบวน 275-276 กรุงเทพฯ-อรัญประเทศ ในขาไปจะประกอบไปด้วยข้อมูลละติจูด ลองจิจูด และเวลาของรถไฟที่พิกัดตำแหน่งต่าง ๆ ที่ใช้เข้าสู่ปลายทางหรือสถานีอรัญประเทศ ส่วนในขากลับนั้นจะประกอบไปด้วยข้อมูลละติจูด ลองจิจูด และเวลาของรถไฟที่พิกัดตำแหน่งต่าง ๆ ที่จะเข้าสู่สถานีปลายทางหรือสถานีกรุงเทพ เช่นเดียวกันกับ

ขบวน 283-284 ที่เดินทางไปกลับจากสถานีกรุงเทพฯ-บ้านพลูตาหลวง ดังนั้นในการอ้างอิง จะใช้สถานีปลายทางขาไปคือสถานีบ้านพลูตาหลวง และสถานีปลายทางขากลับคือสถานีกรุงเทพ ในโครงการนี้ผู้วิจัยเลือกใช้ข้อมูลย้อนหลัง 1 ปี จากนั้นกำหนดค่า K ซึ่งได้กำหนดให้ K เท่ากับ 7 โดย K คือจำนวนกลุ่มข้อมูลที่ใกล้เคียงกับข้อมูลที่ต้องการพิจารณามากที่สุด ซึ่งค่า K ควรกำหนดให้เหมาะสมกับกลุ่มข้อมูล หากเลือกค่าน้อยหรือมากเกินไปก็จะส่งผลต่อการพิจารณาคลาส จากนั้นทำการหาระยะห่างระหว่างแต่ละตัวแปรในกลุ่มข้อมูล สามารถคำนวณระยะห่างระหว่างแต่ละตัวแปรในกลุ่มข้อมูลได้โดยใช้สมการ Euclidean Distance ดังสมการที่ 3.3 [20] โดย $d(x_i, y_i)$ คือ ระยะห่างระหว่างตำแหน่ง x และ y แต่ละตัวแปรในกลุ่มข้อมูล x คือจุดที่ต้องการพิจารณา และ y คือจุดของข้อมูลในแต่ละกลุ่มข้อมูล และ n คือจำนวนตัวแปรในกลุ่มข้อมูล โดยในกลุ่มข้อมูลตัวอย่างมีข้อมูลละติจูด ลองจิจูด ต่าง ๆ ที่ใช้เข้าสู่ปลายทาง ซึ่งจะนำละติจูดและลองจิจูดของรถไฟปัจจุบันมาคำนวณหาระยะห่างกับละติจูดและลองจิจูดในกลุ่มข้อมูลตัวอย่าง ดังสมการที่ 3.3 เพื่อหาชุดข้อมูลที่มีพิกัดละติจูดและลองจิจูดใกล้เคียงกับจุดที่ต้องการพิจารณามากที่สุด

$$d(x_i, y_i) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.3}$$

เมื่อคำนวณระยะห่างระหว่างเงื่อนไขหรือกรณีต่าง ๆ ได้แล้ว จากนั้นก็เลือกชุดข้อมูลที่ใกล้เคียงกับข้อมูลที่ต้องการพิจารณาตามจำนวน K ที่กำหนดไว้ เมื่อพิจารณาข้อมูลจำนวน K ชุด และสังเกตว่าคลาสใดใกล้เคียงกับข้อมูลที่ต้องการพิจารณามากที่สุด ซึ่งสามารถทราบได้จากระยะห่างระหว่างแต่ละตัวแปรในข้อมูล จากนั้นทำการกำหนดคลาสให้กับข้อมูลที่ต้องการพิจารณาเพื่อจัดกลุ่มว่าข้อมูลที่ต้องการพิจารณานั้นจะมีแนวโน้มเป็นอย่างไร โดยหากรถไฟเป็นขบวน275-276 กรุงเทพฯ-อรัญประเทศ และเป็นขาไป ระบบจะนำพิกัดตำแหน่งละติจูด และลองจิจูดของรถไฟตำแหน่งปัจจุบัน และนำพิกัดตำแหน่งของสถานีรถไฟต่าง ๆ มาจัดกลุ่มเพื่อหาระยะเวลาที่จะเข้าสู่สถานีอรัญประเทศ โดยพิจารณาจากกลุ่มข้อมูลตัวอย่าง อย่างไรก็ตามหากเป็นขากลับ จะใช้วิธีการเช่นเดียวกันโดยทำย้อนกลับ ซึ่งจะกำหนดให้สถานีปลายทางคือสถานีกรุงเทพเป็นตัวอ้างอิง

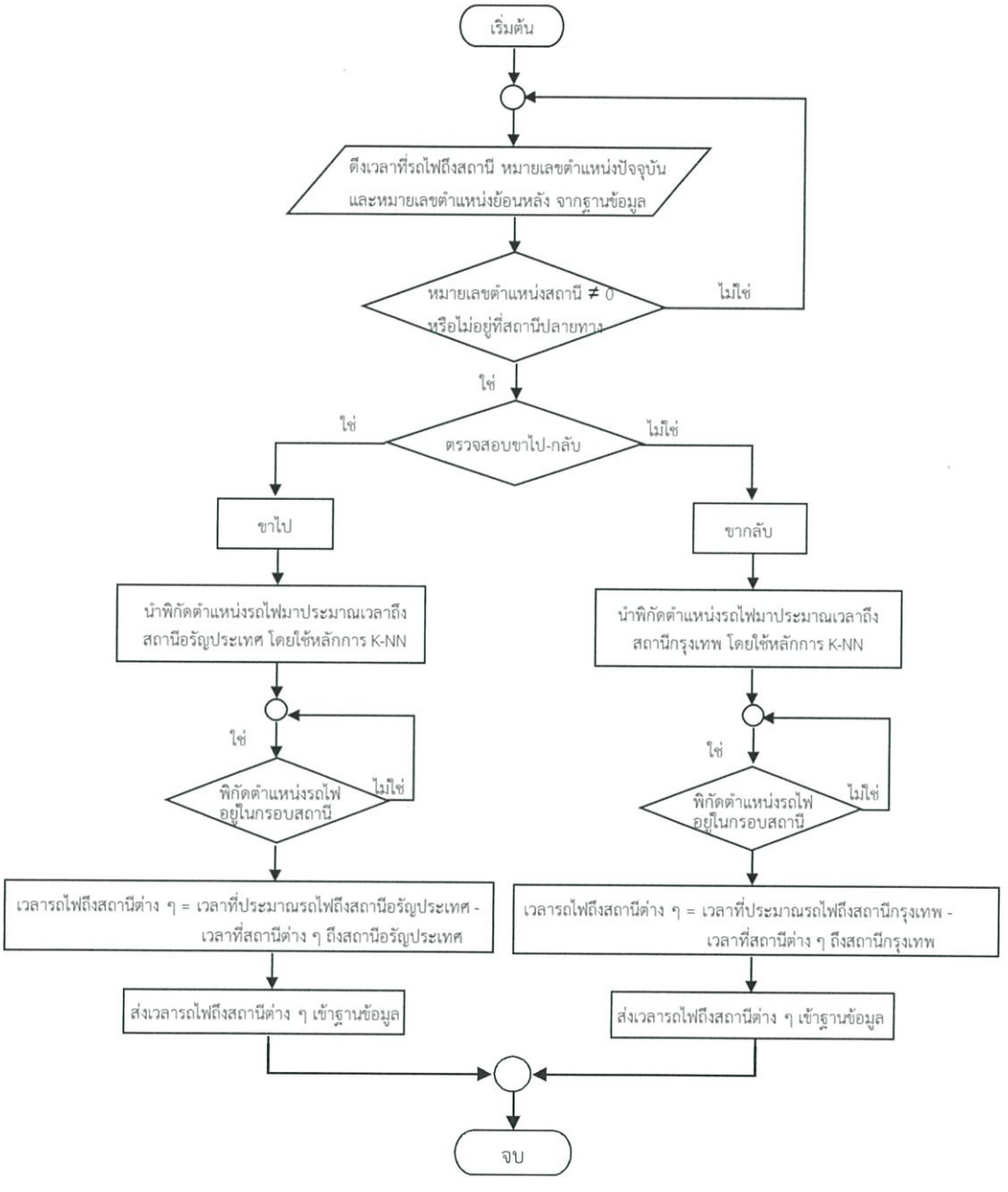
ในทางตรงกันข้าม หากรถไฟเป็นขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง และเป็นขาไป ระบบจะนำพิกัดตำแหน่งละติจูด และลองจิจูดของรถไฟตำแหน่งปัจจุบัน และนำพิกัดตำแหน่งของสถานีรถไฟต่าง ๆ มาจัดกลุ่มเพื่อหาระยะเวลาที่จะเข้าสู่สถานีบ้านพลูตาหลวง โดยพิจารณาจากกลุ่มข้อมูลตัวอย่าง อย่างไรก็ตามหากเป็นขากลับ จะใช้วิธีการเช่นเดียวกันโดยทำย้อนกลับ ซึ่งจะกำหนดให้สถานีปลายทางคือสถานีกรุงเทพเป็นตัวอ้างอิง



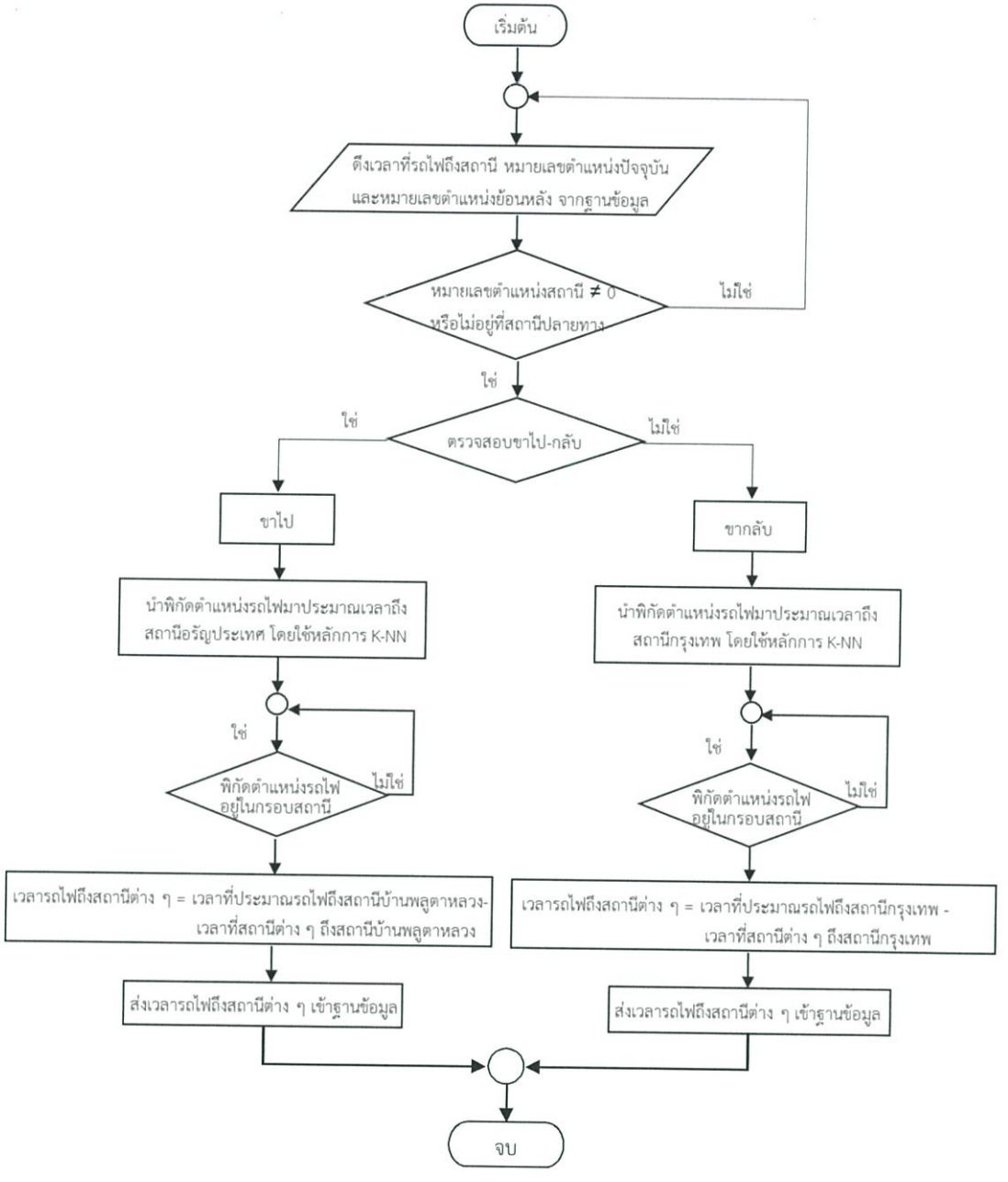
รูปที่ 3.32 แผนผังการทำงานของหลักการ K-Nearest Neighbors (K-NN)

แผนผังการทำงานเพื่อประมาณเวลาถึงสถานีของรถไฟขบวน 275-276 กรุงเทพฯ-อรัญประเทศ แสดงดังรูปที่ 3.33 และขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง แสดงดังรูปที่ 3.34 ตามลำดับ โดยจะทำการรับค่าพิกัดตำแหน่งของรถไฟ เวลาที่รถไฟถึงสถานี หมายเลขตำแหน่งปัจจุบันของรถไฟ และหมายเลขตำแหน่งย้อนหลังของรถไฟครึ่งชั่วโมงจากฐานข้อมูล มาทำการเขียนโปรแกรม เพื่อคำนวณเวลาถึงสถานีของรถไฟ โดยทำการตรวจสอบขาไป-กลับของสถานีรถไฟ ซึ่งหากเวลาที่รถไฟกำลังเคลื่อนที่เป็นขาไปของขบวนรถไฟ 275-276 ที่เดินทางไปกลับระหว่างสถานีกรุงเทพและสถานีอรัญประเทศ ระบบจะนำพิกัดตำแหน่งรถไฟมาประมาณเวลาเข้าสู่สถานีอรัญประเทศ เช่นเดียวกันกับขบวนรถไฟ 283-284 ที่เดินทางไปกลับระหว่างสถานีกรุงเทพและสถานีบ้านพลูตาหลวง โดยสถานีบ้านพลูตาหลวงจะถูกนำมาใช้เป็นตัวอ้างอิงสำหรับขาไป โดยใช้หลักการ K-NN ที่ได้กล่าวมาแล้วตอนต้น ซึ่งเมื่อได้เวลาที่พิกัดตำแหน่งรถไฟจะเข้าสู่สถานีปลายทางที่อ้างอิงแล้ว ระบบจะทำการนำพิกัดตำแหน่งรถไฟที่รับได้จากอุปกรณ์ที่ใช้ในการติดตามรถไฟ มาตรวจสอบว่าอยู่ในกรอบสถานีต่าง ๆ หรือไม่ ซึ่งถ้าหากใช่ ระบบจะนำเวลาที่พิกัดตำแหน่งรถไฟจะเข้าสู่สถานีปลายทางที่อ้างอิง มาลบกับระยะเวลาที่สถานีรถไฟต่าง ๆ ใช้เข้าสู่สถานีปลายทางที่อ้างอิง ซึ่งผลลัพธ์ที่ได้นั้นจะส่งผลให้ได้ทราบถึงเวลาที่พิกัดตำแหน่งปัจจุบันใช้เข้าสู่สถานีรถไฟอื่น ๆ ถัดไป อย่างไรก็ตามหากเป็นขากลับ จะใช้วิธีการเช่นเดียวกันโดยทำย้อนกลับ ซึ่งจะกำหนดให้สถานี

ปลายทางคือสถานีกรุงเทพเป็นตัวอ้างอิง เมื่อได้เวลาถึงสถานีของรถไฟแล้ว โปรแกรมจะส่งเวลาที่ประมาณได้เข้าสู่ฐานข้อมูล



รูปที่ 3.33 แผนผังการทำงานของระบบการประมาณเวลาถึงสถานีของรถไฟขบวน 275-276

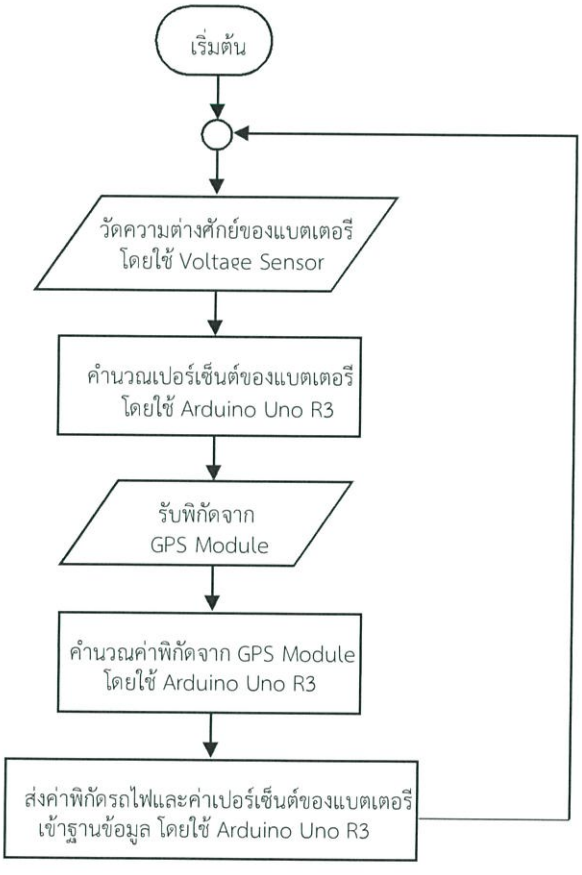


รูปที่ 3.34 แผนผังการทำงานของระบบการประมาณเวลาถึงสถานีของรถไฟขบวน 283-284

3.1.7 การออกแบบอุปกรณ์ติดตามรถไฟ

สำหรับการออกแบบอุปกรณ์ติดตามรถไฟนั้น จะมีอุปกรณ์ในการทำงานอยู่ด้วยกัน 4 อย่าง ได้แก่ GPS Module 3G Module บอร์ด Arduino และ Voltage Sensor ซึ่งอุปกรณ์ทุกตัว

จะมีแบตเตอรีสำรองเป็นแหล่งจ่ายไฟ โดยหลักการทำงาน คือ ผู้พัฒนาได้มีการเขียนโปรแกรมควบคุมอุปกรณ์ติดตามรถไฟ โดยให้รับค่าพิกัดรถไฟจาก GPS Module จากนั้นบอร์ด Arduino จะประมวลผลพิกัดจาก GPS Module และส่งค่าพิกัดรถไฟเข้าฐานข้อมูลด้วย 3G Module พร้อมทั้งวัดปริมาณแบตเตอรีคงเหลือของแบตเตอรีสำรองด้วย Voltage Sensor และส่งเข้าฐานข้อมูลด้วย 3G Module และแสดงสถานะของแบตเตอรีด้วย LED ผังการทำงานแสดงดังรูปที่ 3.35



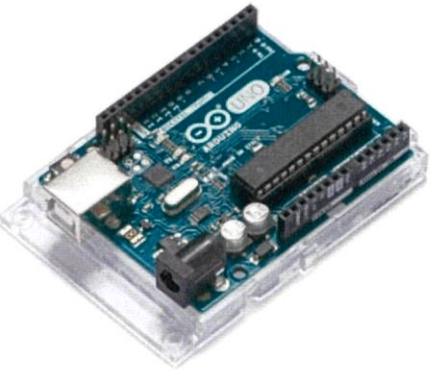
รูปที่ 3.35 แผนผังการทำงานของระบบการติดตามรถไฟ

3.2 เครื่องมือที่ใช้ในการทดลอง

ในส่วนของอุปกรณ์ติดตามรถไฟนั้น จะมีอุปกรณ์ในการทำงานอยู่ด้วยกัน 4 อย่าง ได้แก่ GPS Module Ublox NEO 6M สำหรับรับค่าพิกัดตำแหน่งของรถไฟ บอร์ด Arduino Uno R3 ที่ประมวลผลพิกัดจาก GPS Module และใช้ Voltage Sensor วัดปริมาณแบตเตอรีคงเหลือของแบตเตอรีสำรอง แล้วส่งค่าพิกัดรถไฟเข้าฐานข้อมูลด้วย 3G Module

3.2.1 บอร์ด Arduino

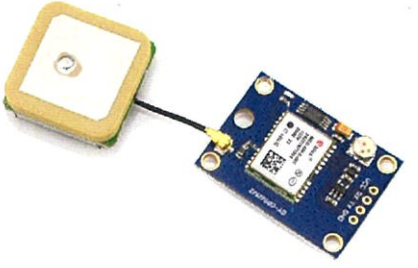
Arduino เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ซึ่งตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานง่าย สามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือสามารถต่อกับบอร์ดเสริมเพื่อความสะดวกในการใช้งาน ผู้ใช้งานสามารถโปรแกรมเพิ่มเติมในตัวบอร์ดได้โดยการเสียบสาย USB เชื่อมคอมพิวเตอร์กับบอร์ด Arduino ดังรูปที่ 3.36 แล้วทำการเขียนโปรแกรม ซึ่งบอร์ด Arduino ที่ใช้คือ Arduino Uno R3



รูปที่ 3.36 บอร์ด Arduino Uno R3 [14]

3.2.2 GPS Module Ublox NEO 6M

GPS โมดูลขนาดกระทัดรัดที่มาพร้อมกับเสาอากาศแบบเซรามิก โดยการเชื่อมต่อเข้ากับบอร์ด Arduino Uno R3 จะใช้เพียง 4 ขา ได้แก่ Rx Tx Vcc และ GND โดยแสดงดังรูปที่ 3.37 และแสดงการเชื่อมต่ออุปกรณ์ดังตารางที่ 3.1



รูปที่ 3.37 GPS Module Ublox NEO 6M [15]

ตารางที่ 3.1 รายละเอียดการต่ออุปกรณ์ GPS Module กับบอร์ด Arduino

Arduino	GPS Module
3.3V	Vcc
Gnd	Gnd
10	Tx
11	Rx

3.2.3 3G Module

3G Module เป็นบอร์ดสื่อสารเครือข่ายโทรศัพท์เคลื่อนที่ระบบ 3G ใช้ชิพสื่อสารจาก Quectel รุ่น UC15-T รองรับคลื่นความถี่ 850/2100 MHz ในระบบ 3G UMTS และรองรับคลื่นความถี่ 850/900/1800/1900 MHz ในระบบ 2G GSM มีความเร็วในการสื่อสารข้อมูลสูงสุดระหว่างโมดูลกับเครือข่าย 3.6 Mbps (downlink) และ 384 Kbps (uplink) ซึ่งในการเชื่อมต่อเข้ากับบอร์ด Arduino จะใช้ขา VIN GND Rx และ Tx โดยแสดงดังรูปที่ 3.38 และแสดงการเชื่อมต่ออุปกรณ์ดังตารางที่ 3.2

รูปที่ 3.38 3G Module (UC15-T) [16]

ตารางที่ 3.2 รายละเอียดการต่ออุปกรณ์ 3G Module กับบอร์ด Arduino

Arduino	3G Module
5V	Vcc
Gnd	Gnd
8	Tx
9	Rx

3.2.4 Voltage Sensor Module

โมดูลสำหรับวัดแรงดันไฟฟ้าช่วง 0-24 โวลต์ โดยใช้หลักการวัดแรงดันไฟฟ้าทางขาแอนะล็อกของบอร์ด Arduino ที่สามารถอ่านค่าแรงดันไฟฟ้าที่ 0-5 โวลต์ออกมาเป็นค่าดิจิทัล 0-1023 โมดูลนี้ใช้วงจรแบ่งแรงดันลดไฟจาก 0-24 โวลต์ให้อยู่ในช่วง 5 โวลต์ สามารถวัดแรงดันไฟฟ้าได้สูงสุด 24.9 โวลต์ โดยใช้ไฟเลี้ยง 5 โวลต์ หรือวัดแรงดันไฟฟ้าได้สูงสุด 16.5 โวลต์ โดยใช้ไฟเลี้ยง 3.3 โวลต์ โดยแสดงดังรูปที่ 3.39 และแสดงการเชื่อมต่ออุปกรณ์ดังตารางที่ 3.3

รูปที่ 3.39 Voltage Sensor Module [17]

ตารางที่ 3.3 รายละเอียดการต่ออุปกรณ์ Voltage Sensor Module กับบอร์ด Arduino

Arduino	Voltage Sensor Module
5V	+
Gnd	-
A0	S

3.3 การจัดเก็บผลการทดลอง

การจัดเก็บผลการทดลองแบ่งออกเป็น 9 ส่วน ได้แก่

1) การจัดเก็บผลการทดลองการวัดความต่างศักย์ของแบตเตอรี เพื่อนำค่าที่ได้ไปคำนวณหาสมการแนวโน้มการลดลงของแบตเตอรีโดยใช้โปรแกรม MATLAB

2) การจัดเก็บผลการทดลองการแสดงสถานะของแบตเตอรีด้วย LED

3) การจัดเก็บผลการทดลองระบบแจ้งเตือนบนแอปพลิเคชัน LINE

4) การจัดเก็บผลการทดลองในส่วนของการแสดงผลผ่านแอปพลิเคชัน เพื่อนำไปแสดงผลต่อผู้ใช้งานได้รับทราบตำแหน่งของรถไฟ

5) การจัดเก็บผลการทดลองในส่วนของการติดตั้งแอปพลิเคชันบน App Store

6) การจัดเก็บผลการทดลองในส่วนของการแสดงผลผ่านเว็บไซต์ เพื่อนำไปแสดงผลต่อผู้ใช้งานได้รับทราบตำแหน่งของรถไฟ

7) การจัดเก็บผลการทดลองในส่วนของระบบการคำนวณตำแหน่งของรถไฟ เพื่อลดภาระการทำงานของแอปพลิเคชัน

8) การจัดเก็บผลการทดลองระบบการประมาณเวลาถึงสถานีของรถไฟ

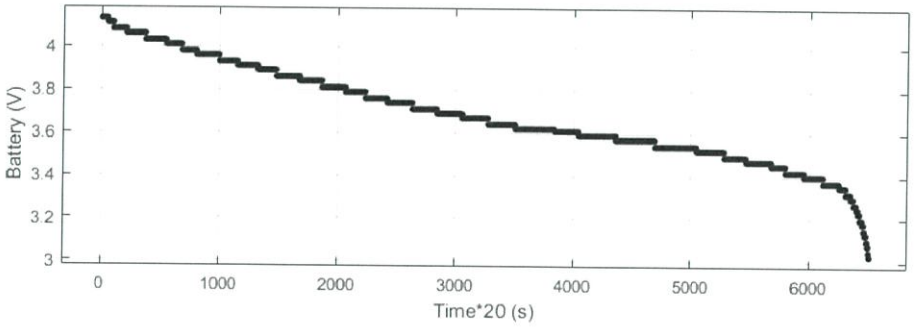
9) การจัดเก็บผลการทดลองอุปกรณ์ติดตามรถไฟ และการส่งค่าพิกัดตำแหน่งรถไฟและค่าเปอร์เซ็นต์ของแบตเตอรี

บทที่ 4

ผลการทดลอง

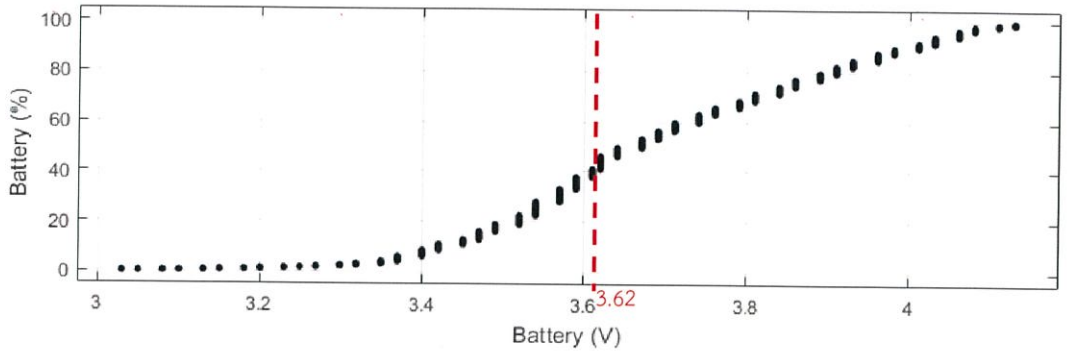
4.1 การวัดความต่างศักย์ของแบตเตอรี

ในส่วนนี้เป็นการทดลอง และวัดค่าความต่างศักย์ของแบตเตอรี เพื่อวิเคราะห์การลดลงของแบตเตอรีของแบตเตอรีสำรองที่จ่ายไฟให้กับอุปกรณ์ที่ติดตั้งบนรถไฟ โดยการทดลองในครั้งนี้จะวัดตั้งแต่แบตเตอรีเต็มจนแบตเตอรีหมด โดยใช้โปรแกรม MATLAB ในการพล็อตกราฟเพื่อแสดงผลการทดลอง แสดงดังรูปที่ 4.1



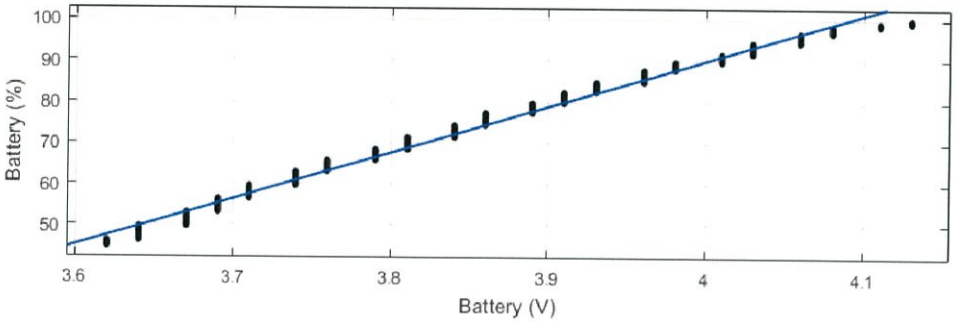
รูปที่ 4.1 การทดลองวัดความต่างศักย์ของแบตเตอรี

จากกราฟในรูปที่ 4.1 แสดงการลดลงของแบตเตอรี (แกน y) เทียบกับเวลาที่ใช้ไปในหน่วยวินาทีคูณ 20 (แกน x) ซึ่งจะเห็นได้ว่าแบตเตอรีจะลดลงตามระยะเวลาที่ใช้งาน ซึ่งระยะเวลาที่แบตเตอรีสามารถใช้งานได้คือ การนำ 6510 วินาที มาคูณกับสเกลเวลา 20 วินาที โดยจะได้ผลลัพธ์ออกมาเป็นระยะเวลาที่อุปกรณ์บนรถไฟสามารถใช้งานได้โดยปราศจากการชาร์ตแบตเตอรีหรือมีค่าประมาณเท่ากับ 36 ชั่วโมง จากนั้นนำข้อมูลที่ได้ในรูปที่ 4.1 ไปทำ curve fitting โดยใช้โปรแกรม MATLAB โดยกำหนดแกนนอนให้เป็นค่าความต่างศักย์ของแบตเตอรี และกำหนดแกนตั้งให้เป็นเปอร์เซ็นต์ของแบตเตอรีคงเหลือที่ทำการแปลงมาจากเวลา เพื่อหาสมการอัตราการลดลงของแบตเตอรี แสดงได้ดังรูปที่ 4.2

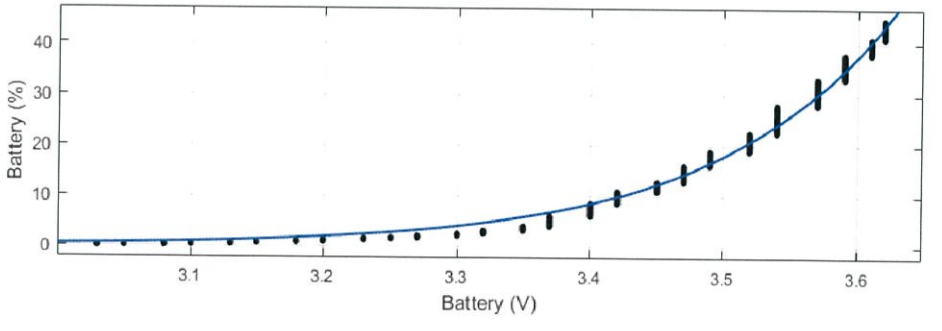


รูปที่ 4.2 การทดลองวัดเปอร์เซ็นต์ของแบตเตอรี

จากรูปที่ 4.2 พบว่าแนวโน้มของเส้นกราฟความต่างศักย์ในแกนนอน ที่มีค่ามากกว่า 3.62 โวลต์ มีลักษณะเป็นเชิงเส้น (Linear) ในทางตรงกันข้ามค่าความต่างศักย์ที่น้อยกว่า 3.62 โวลต์ มีแนวโน้มเส้นกราฟแบบเอกซ์โพเนนเชียล (Exponential) ดังนั้น ผู้วิจัยสามารถแบ่งกราฟใน รูปที่ 4.2 ออกเป็น 2 ส่วน คือ กราฟที่พิจารณาเฉพาะช่วงที่แบตเตอรีลดลงแบบเชิงเส้น ดังรูปที่ 4.3 และกราฟที่พิจารณาเฉพาะช่วงที่แบตเตอรีลดลงแบบเอกซ์โพเนนเชียล ดังรูปที่ 4.4



รูปที่ 4.3 การทำ curve fitting ในช่วงที่แบตเตอรีลดลงแบบเชิงเส้น (Linear)



รูปที่ 4.4 การทำ curve fitting ของช่วงที่แบตเตอรีลดลงแบบเอกซ์โพเนนเชียล (Exponential)

จากรูปที่ 4.3 และ 4.4 ภายหลังจากการทำ curve fitting แล้วจะพบว่าหากค่าความต่างศักย์ของแบตเตอรีมีค่ามากกว่า 3.62 โวลต์ สามารถประมาณสมการเส้นตรงของเปอร์เซ็นต์แบตเตอรีได้ดังสมการที่ 4.1

$$Y = 112X - 358 \tag{4.1}$$

โดย Y คือ ค่าเปอร์เซ็นต์แบตเตอรีคงเหลือ

X คือ ค่าความต่างศักย์ที่วัดได้จากแบตเตอรี

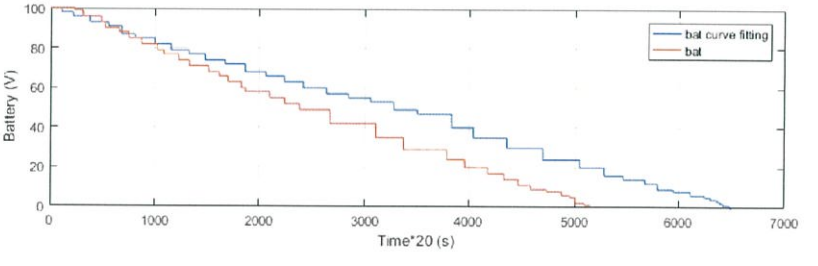
และหากค่าความต่างศักย์ของแบตเตอรีมีค่าน้อยกว่า 3.62 โวลต์ สามารถประมาณสมการเอกซ์โพเนนเชียลของเปอร์เซ็นต์แบตเตอรีได้ดังสมการที่ 4.2

$$Y = 8.832 \times 10^{-11} e^{7.439X} \tag{4.2}$$

โดย Y คือ ค่าเปอร์เซ็นต์แบตเตอรีคงเหลือ

X คือ ค่าความต่างศักย์ที่วัดได้จากแบตเตอรี

เมื่อทำการหาสมการอัตราการลดลงของแบตเตอรีได้แล้วนั้น ผู้วิจัยจะนำสมการที่ได้ไปเขียนโปรแกรมสำหรับคำนวณค่าเปอร์เซ็นต์แบตเตอรี และส่งค่าเปอร์เซ็นต์แบตเตอรีเข้าฐานข้อมูล



รูปที่ 4.5 กราฟอัตราการลดลงของแบตเตอรี

จากรูปที่ 4.5 เป็นผลการทดลองเมื่อวัดความต่างศักย์ของแบตเตอรี แล้วนำไปแปลงเป็นเปอร์เซ็นต์ของแบตเตอรีคงเหลือ เส้นสีฟ้าแสดงเป็นกราฟที่ได้จากการวัดความต่างศักย์ของแบตเตอรีครั้งแรก กราฟเส้นสีแดงแสดงเป็นผลจากการนำสมการประมาณเปอร์เซ็นต์แบตเตอรีคงเหลือจากการทำ curve fitting (สมการที่ 4.1 และ 4.2) มาใช้งาน

หมายเหตุ เนื่องจากแบตเตอรีที่ทำการทดลองได้ถูกนำมาใช้งานจากแบตเตอรีเต็มจนแบตเตอรีหมดหลายครั้ง จึงทำให้เวลาในการใช้งานของแบตเตอรีลดลงกราฟที่ได้จึงมีความคลาดเคลื่อนไปเล็กน้อย ประมาณ 10 %

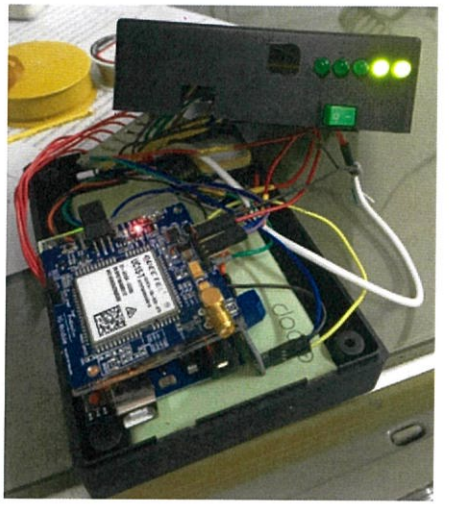
4.2 การแสดงสถานะของแบตเตอรีด้วย LED

ในส่วนของการแสดงสถานะของแบตเตอรี จะใช้หลอด l ED 5 ดวง เพื่อแสดงสถานะของแบตเตอรีคงเหลือในแต่ละช่วง (สีเขียวอย่างเดียว 4 ดวง สีเขียวและสีแดง 1 ดวง) สถานะ LED ที่กำหนดโดยแบตเตอรีคงเหลือในแต่ละช่วง แสดงดังตารางที่ 4.1

ตารางที่ 4.1 รายละเอียดการแสดงสถานะของแบตเตอรี

สถานะ LED	แบตเตอรีคงเหลือ (%)	เวลาที่สามารถใช้งานได้ (ชม.)
สีเขียว 5 ดวง	81 - 100	6
สีเขียว 4 ดวง	61 - 80	6
สีเขียว 3 ดวง	41 - 60	5
สีเขียว 2 ดวง	21 - 40	5
สีเขียว 1 ดวง	11 - 20	4
สีแดง 1 ดวง	1 - 10	3
0 ดวง	0	0

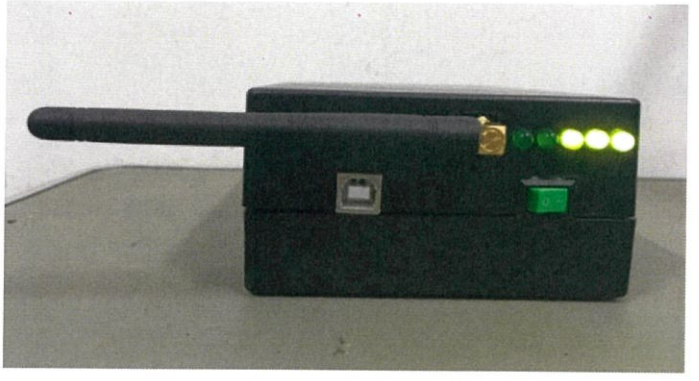
จากการออกแบบ LED เพื่อแสดงสถานะแบตเตอรีนั้น เมื่อนำ LED ติดบนกล่องอุปกรณ์ติดตามรถไฟ และทำการประกอบอุปกรณ์ในระบบติดตามรถไฟไม่ว่าจะเป็น GPS Module 3G Module บอร์ด Arduino Voltage Sensor แบตเตอรีสำรอง รวมทั้ง LED แสดงสถานะแบตเตอรีเข้าด้วยกัน และทดสอบสถานะแบตเตอรีในแต่ละช่วง จะแสดงได้ดังรูปที่ 4.6 ถึง รูปที่ 4.13



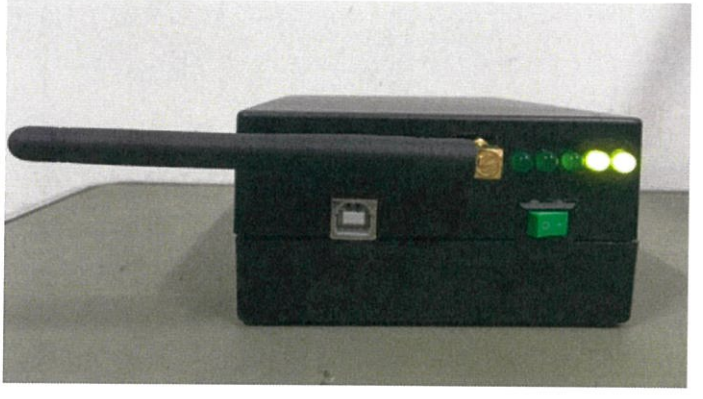
รูปที่ 4.6 อุปกรณ์ภายในกล่องอุปกรณ์ติดตามรถไฟ

รูปที่ 4.7 กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 81 - 100 เปอร์เซ็นต์

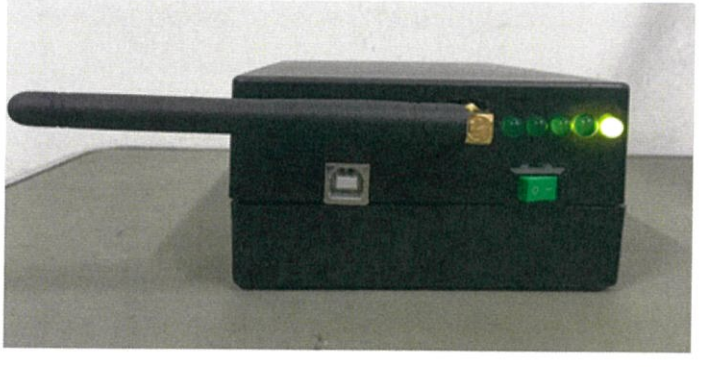
รูปที่ 4.8 กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 61 - 80 เปอร์เซ็นต์



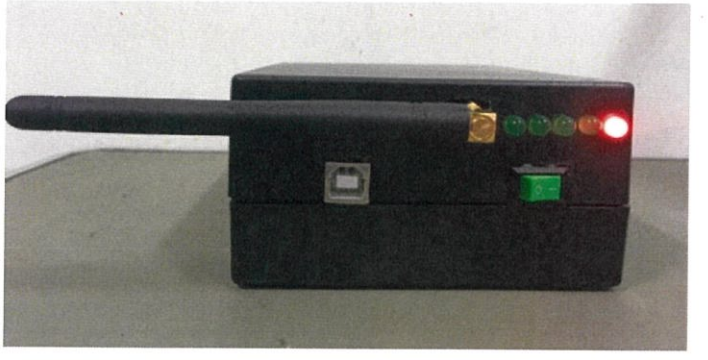
รูปที่ 4.9 กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 41 – 60 เปอร์เซ็นต์



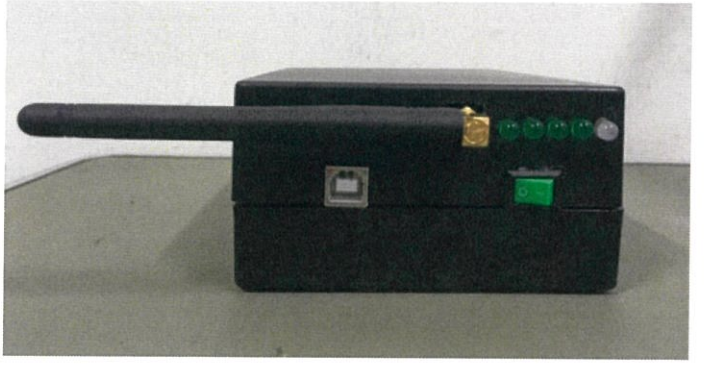
รูปที่ 4.10 กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 21 – 40 เปอร์เซ็นต์



รูปที่ 4.11 กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 11 – 20 เปอร์เซ็นต์



รูปที่ 4.12 กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 1 – 10 เปอร์เซ็นต์



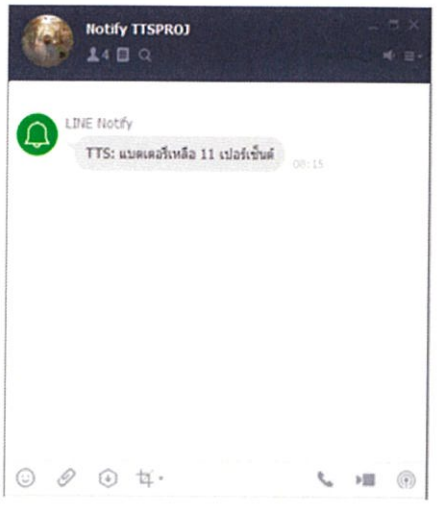
รูปที่ 4.13 กล่องอุปกรณ์ติดตามรถไฟ แสดงสถานะเมื่อแบตเตอรีคงเหลือ 0 เปอร์เซ็นต์

4.3 การแจ้งเตือนบนแอปพลิเคชัน LINE

ในส่วนของระบบแจ้งเตือนข้อความผ่านแอปพลิเคชันไลน์ โดยใช้บริการ Line Notify ซึ่งสามารถส่งข้อความการแจ้งเตือนต่าง ๆ ไปยังแอคเคาท์ผู้ขอใช้งาน

4.3.1 ระบบแจ้งเตือนปริมาณแบตเตอรี

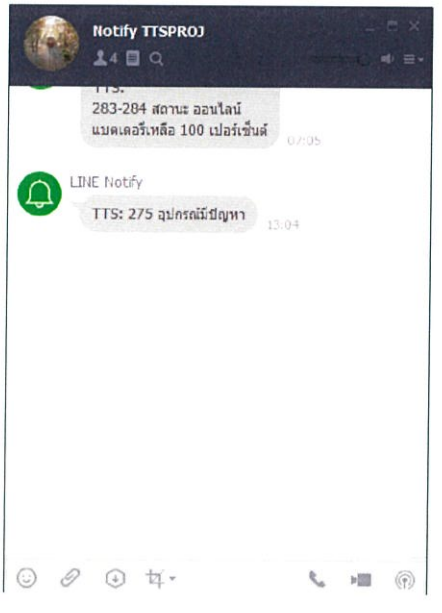
ผู้วิจัยได้ทำการกำหนดให้แจ้งเตือนเมื่อปริมาณแบตเตอรีคงเหลือ 11 เปอร์เซ็นต์ โดยแสดงเป็นข้อความ “แบตเตอรีเหลือ 11 เปอร์เซ็นต์” ผ่านแอปพลิเคชัน LINE เพื่อให้ผู้ดูแลระบบได้ทราบ แสดงดังรูปที่ 4.14



รูปที่ 4.14 การแจ้งเตือนเมื่อปริมาณแบตเตอรีคงเหลือ 11 เปอร์เซ็นต์

4.3.2 ระบบแจ้งเตือนเมื่ออุปกรณ์ที่ใช้ในการติดตามรถไฟเกิดปัญหา

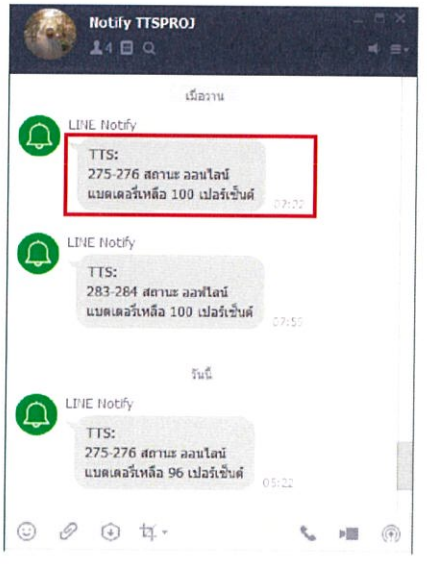
ผู้วิจัยได้ทำการกำหนดให้แจ้งเตือนเมื่อข้อมูลพิกัดตำแหน่งของรถไฟได้หยุดส่งมายังฐานข้อมูลเป็นเวลา 1 ชั่วโมง โดยแสดงเป็นข้อความ “ 275 อุปกรณ์มีปัญหา ” เมื่ออุปกรณ์ติดตามรถไฟขบวน 275 มีปัญหา และแจ้งเตือน “ 283 อุปกรณ์มีปัญหา ” เมื่ออุปกรณ์ติดตามรถไฟขบวน 283 มีปัญหา ผ่านแอปพลิเคชัน LINE เพื่อให้ผู้ดูแลระบบได้ทราบ แสดงดังรูปที่ 4.15



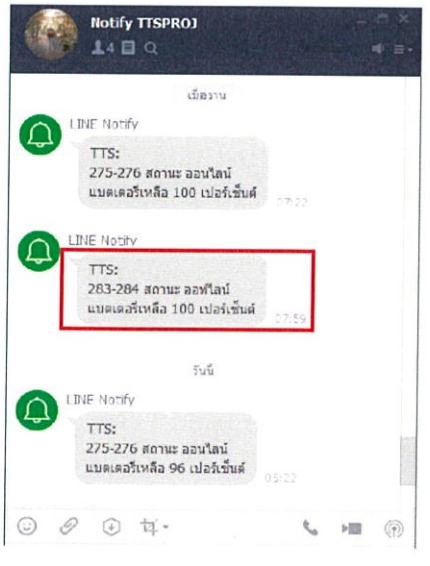
รูปที่ 4.15 การแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 เกิดปัญหา

4.3.3 ระบบแจ้งเตือนสถานะการให้บริการของอุปกรณ์ที่ใช้ในการติดตามรถไฟ

ผู้วิจัยได้ทำการกำหนดให้ระบบแจ้งเตือนเมื่ออุปกรณ์ติดตามรถไฟขบวน 275 เปิดให้บริการ ด้วยข้อความว่า " 275 สถานะ ออนไลน์ " ในทางตรงกันข้าม หากขบวน 275 งดให้บริการ ระบบจะแจ้งเตือนด้วยข้อความว่า " 275 สถานะ ออฟไลน์ " เช่นเดียวกันกับการแจ้งเตือนอุปกรณ์ติดตามรถไฟขบวน 283 ซึ่งแสดงดังรูปที่ 4.16 และรูปที่ 4.17



รูปที่ 4.16 การแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 275 ให้บริการ

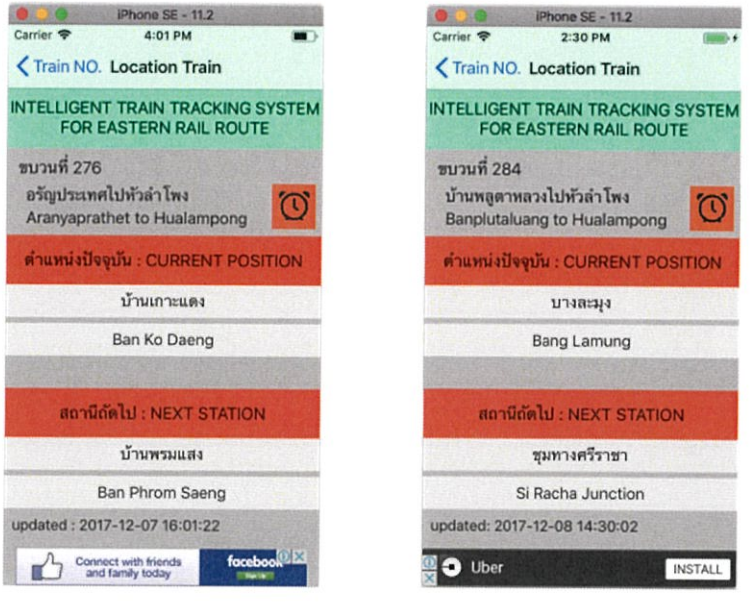


รูปที่ 4.17 การแจ้งเตือนเมื่ออุปกรณ์บนรถไฟขบวน 283 งดให้บริการ

4.4 การทดสอบแอปพลิเคชัน

4.4.1 การทดสอบแอปพลิเคชัน บนโปรแกรมจำลอง Xcode

ภายหลังจากที่นำข้อมูลจากฐานข้อมูลมาผ่านเงื่อนไขการตัดสินใจเพื่อแสดงสถานีแล้วนั้น ในหัวข้อนี้จะเป็นการทดสอบแอปพลิเคชันของสมาร์ตโฟนบนโปรแกรม Xcode แสดงดังรูปที่ 4.18(ก) ที่เป็นการแสดงผลตำแหน่งปัจจุบันของรถไฟขบวนที่ 275-276 โดยมีการตัดสินใจว่าเป็นขบวนขากลับ ซึ่งตำแหน่งปัจจุบันของรถไฟขบวนนี้คือสถานีบ้านเกาะแดงและสถานีถัดไปคือสถานีบ้านพรหมแสง สำหรับรูปที่ 4.18(ข) นั้นเป็นการแสดงผลตำแหน่งปัจจุบันของรถไฟขบวนที่ 283-284 โดยมีการตัดสินใจว่าเป็นขบวนขากลับ ซึ่งมีตำแหน่งปัจจุบันของรถไฟขบวนนี้คือสถานีบางละมุง และสถานีถัดไปคือสถานีชุมทางศรีราชา

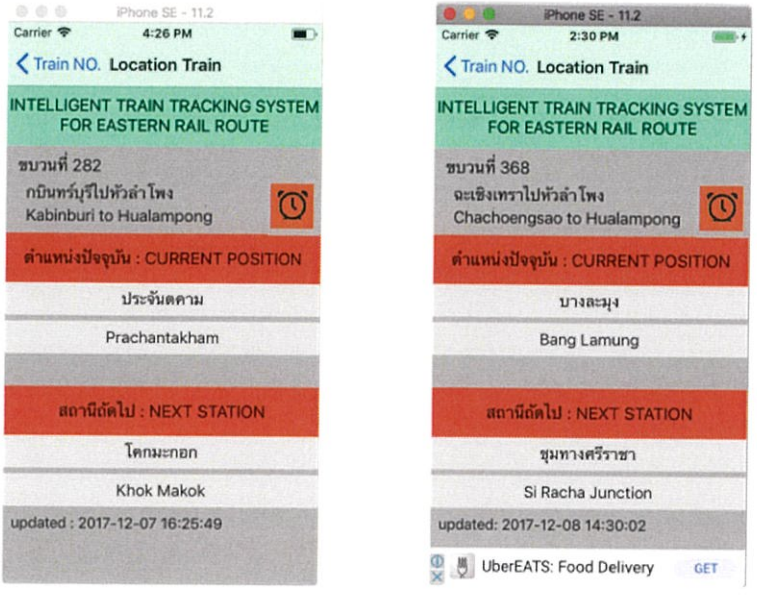


(ก) (ข)

รูปที่ 4.18 การแสดงผลตำแหน่งปัจจุบันของรถไฟขบวนที่ 275-276 (ก) และ 283-284 (ข) บนโปรแกรมจำลอง Xcode

รูปที่ 4.19(ก) เป็นการแสดงผลตำแหน่งปัจจุบันของรถไฟขบวนที่ 281-282 โดยมีการตัดสินใจว่าเป็นขบวนขากลับ ซึ่งมีตำแหน่งปัจจุบันของรถไฟคือสถานีประจันตคามและสถานีถัดไปคือสถานีโคกมะกอก สำหรับรูปที่ 4.19(ข) นั้นเป็นการแสดงผลตำแหน่งปัจจุบันของรถไฟขบวนที่ 367-368 โดยมีการตัดสินใจว่าเป็นขบวนขากลับ ซึ่งมีตำแหน่งปัจจุบันของรถไฟคือสถานีบางละมุงและสถานีถัดไปคือสถานีชุมทางศรีราชา ซึ่งตำแหน่งของรถไฟที่แสดงผลในขบวน 281-282 และ

ขบวน 367-368 ขณะนี้เป็นข้อมูลตำแหน่งของรถไฟขบวน 275-276 และขบวน 283-284 ตามลำดับ

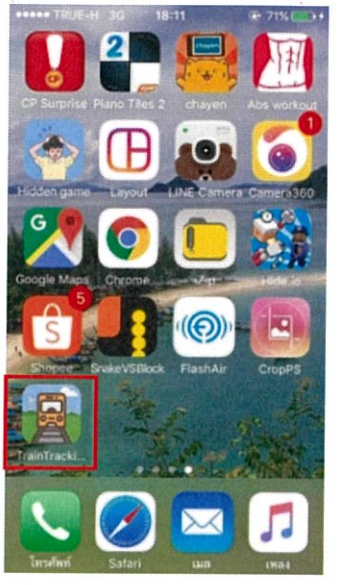


(ก) (ข)

รูปที่ 4.19 การแสดงผลตำแหน่งปัจจุบันของรถไฟขบวนที่ 281-282 (ก) และ 367-368 (ข) บนโปรแกรมจำลอง Xcode

4.4.2 การทดสอบแอปพลิเคชัน บนสมาร์ตโฟนในระบบปฏิบัติการไอโอเอส (iOS)

สำหรับการทดสอบการทำงานแอปพลิเคชันบนสมาร์ตโฟนในระบบปฏิบัติการไอโอ-เอสนั้น ขั้นตอนแรก ผู้วิจัยจะติดตั้งแอปพลิเคชัน ลงบนสมาร์ตโฟนในระบบปฏิบัติการไอโอเอส ซึ่งมีไอคอนของแอปพลิเคชันดังรูปที่ 4.20



รูปที่ 4.20 ไอคอนของแอปพลิเคชัน

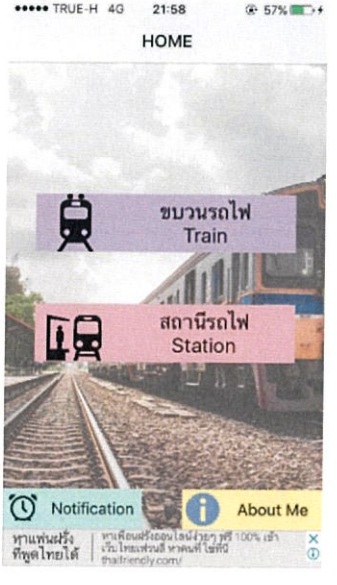
เมื่อกดเข้าสู่แอปพลิเคชันจะแสดงหน้าจอเริ่มต้น โดยหน้าจอเริ่มต้นจะมีข้อมูลชื่อโครงงานและขบวนรถไฟที่ให้บริการตามลำดับ ดังรูปที่ 4.21

รูปที่ 4.21 หน้าจอเริ่มต้นของแอปพลิเคชัน

จากนั้น แอปพลิเคชันจะแสดงหน้าต่างเพื่อขออนุญาตส่งการแจ้งเตือน ดังภาพที่ 4.22 ซึ่งถ้าหากกดไม่อนุญาต แอปพลิเคชันจะไม่ส่งการแจ้งเตือนมาให้ เมื่อต้องการให้แอปพลิเคชันแจ้งเตือน ต้องไปแก้ไขที่ส่วนของการแจ้งเตือน โดยสามารถเข้าถึงได้โดย การตั้งค่า >> การแจ้งเตือน

รูปที่ 4.22 หน้าต่างเพื่อขออนุญาตส่งการแจ้งเตือน

เมื่อกดอนุญาตจะเข้าสู่หน้าหลักของแอปพลิเคชันดังรูปที่ 4.23 โดยหน้าหลักจะประกอบไปด้วย 4 เมนู คือเมนูเลือกขบวนรถไฟ เลือกสถานีรถไฟ เมนูการแจ้งเตือน เมนูข้อมูลผู้จัดทำ และส่วนของแถบโฆษณา ตามลำดับ

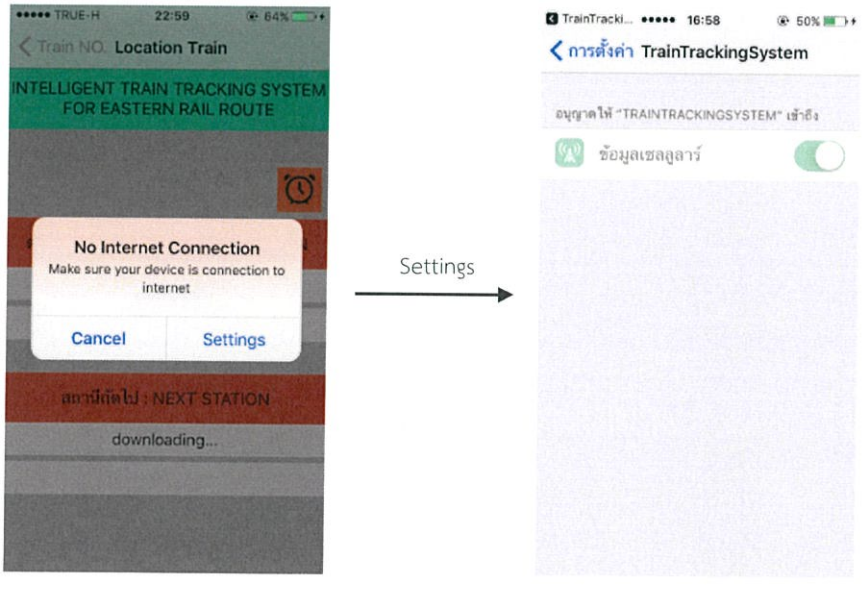


รูปที่ 4.23 หน้าหลักของแอปพลิเคชัน

ในเมนูแรกหรือเมนูเลือกขบวนรถไฟนั้นเมื่อผู้ใช้งานกดเลือก หน้าจอจะแสดงขบวนรถไฟต่าง ๆ ที่ให้บริการระบบติดตามรถไฟ ดังรูปที่ 4.24

รูปที่ 4.24 หน้าเมนูขบวนรถไฟ

เมื่อใดก็ตามที่ผู้ใช้งานกดเลือกขบวนรถไฟโดยไม่ได้ทำการเปิดใช้งานอินเทอร์เน็ต แอปพลิเคชันจะทำการแจ้งเตือนด้วยข้อความ “No Internet Connection Make sure your device is connected to internet” โดยได้ทำการเพิ่มปุ่มตั้งค่า (Settings) เพื่อให้ผู้ใช้งานสามารถกดไปหน้าการตั้งค่าได้โดยตรง เพื่อเปิดใช้งานอินเทอร์เน็ต ดังรูปที่ 4.25



รูปที่ 4.25 ปุ่ม setting เมื่อกดแล้วจะนำไปสู่หน้าการตั้งค่า

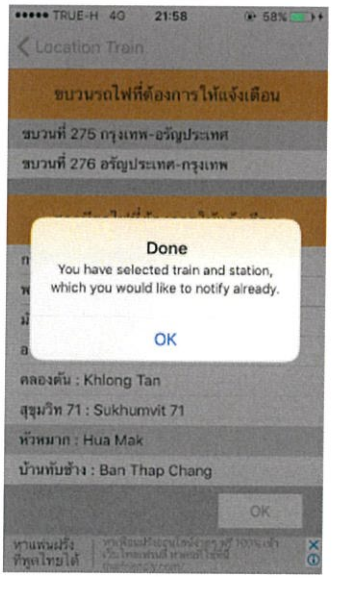
เมื่อผู้ใช้งานทำการเชื่อมต่ออินเทอร์เน็ตแล้ว แอปพลิเคชันจะแสดงหมายเลขขบวน เส้นทางรถไฟ ตำแหน่งปัจจุบัน และสถานีถัดไปของรถไฟขบวนนั้น (ทั้งในภาษาไทยและภาษาอังกฤษ) พร้อมทั้งบอกเวลาที่แอปพลิเคชันอัพเดตข้อมูลล่าสุด หากผู้ใช้ต้องการตั้งค่าการแจ้งเตือน สามารถกดที่ไอคอนรูปนาฬิกาในหน้าแสดงตำแหน่งปัจจุบันของรถไฟ ดังแสดงในรูปที่ 4.26

รูปที่ 4.26 หน้าแสดงตำแหน่งของรถไฟ

เมื่อผู้ใช้งานกดเลือกที่ไอคอนรูปนาฬิกา แอปพลิเคชันจะแสดงหน้าต่างให้เลือกขบวนรถไฟและสถานีรถไฟที่ต้องการให้แจ้งเตือน ดังรูปที่ 4.27

รูปที่ 4.27 หน้าต่างแสดงขบวนรถไฟและสถานีรถไฟที่ต้องการให้แจ้งเตือน

เมื่อผู้ใช้เลือกขบวนรถไฟและสถานีรถไฟที่ต้องการให้แจ้งเตือนและกด “OK” เรียบร้อยแล้ว แอปพลิเคชันจะแสดงหน้าต่างแจ้งเตือนว่าผู้ใช้ได้เลือกเรียบร้อยแล้ว โดยมีข้อความว่า “You have selected the train and station, which you would like to notify already .” และปุ่ม “OK” ดังรูปที่ 4.28



รูปที่ 4.28 หน้าต่างแสดงการทำรายการสำเร็จ

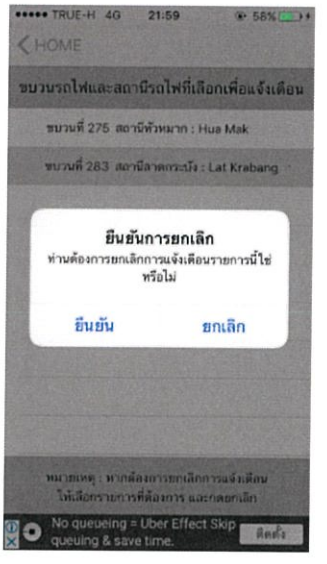
เมื่อผู้ใช้งานกด “OK” แอปพลิเคชันจะเปลี่ยนไปยังหน้าต่างแสดงรายการการแจ้งเตือน ดังรูปที่ 4.29

รูปที่ 4.29 หน้าต่างแสดงรายการการแจ้งเตือน เมื่อตั้งค่าการแจ้งเตือนแล้ว

เมื่อรถไฟกำลังเข้าสู่สถานีที่ผู้ใช้งานเลือก (ผู้พัฒนาได้ทำการออกแบบแอปพลิชันให้แจ้งเตือน 2 สถานีก่อนเข้าสถานีที่ผู้ใช้งานเลือก) แอปพลิเคชันจะแสดงการแจ้งเตือนเป็นข้อความและเสียงเตือน ดังรูปที่ 4.30

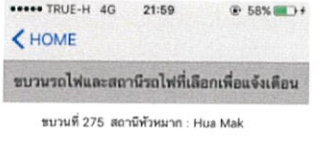
รูปที่ 4.30 หน้าแสดงการแจ้งเตือนของแอปพลิเคชัน

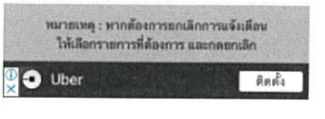
หากผู้ใช้ต้องการยกเลิกรายการแจ้งเตือน ผู้ใช้สามารถกดรายการที่ต้องการในหน้าแสดงรายการแจ้งเตือน แอปพลิเคชันจะแจ้งเตือนเพื่อยืนยันการยกเลิก และมีข้อความว่า “ท่านต้องการยกเลิกการแจ้งเตือนรายการนี้ใช่หรือไม่ ดังรูปที่ 4.31



รูปที่ 4.31 หน้าต่างยืนยันการยกเลิกการแจ้งเตือน

หากกดยกเลิก แอปพลิเคชันจะไม่ยกเลิกการแจ้งเตือนรายการนั้น แต่ถ้าหากกดยืนยัน แอปพลิเคชันจะยกเลิกการแจ้งเตือนรายการนั้น ดังรูปที่ 4.32





รูปที่ 4.32 หน้าต่างแสดงรายการแจ้งเตือน เมื่อยกเลิกการแจ้งเตือน 1 รายการ

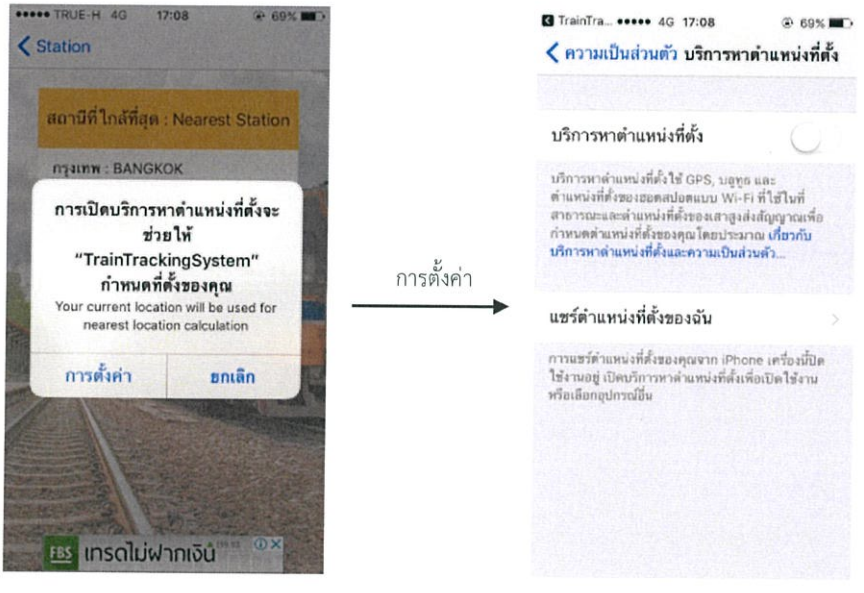
ในเมนูที่สองหรือเมนูสถานีรถไฟซึ่งถูกออกแบบมาเพื่อผู้ใช้งานที่ไม่ทราบถึงหมายเลขขบวนรถไฟ จึงทำให้ไม่สามารถใช้เมนูที่หนึ่งได้ โดยเมื่อผู้ใช้งานกดเลือกเมนูนี้ในหน้าหลัก หน้าจอจะแสดงเมนูสถานีรถไฟที่ใกล้ที่สุดและสถานีรถไฟต่าง ๆ ที่รถไฟขบวน 275-276 กรุงเทพฯ-อรัญประเทศ ขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง วิ่งผ่าน และยังมีแถบค้นหา เพื่อให้ผู้ใช้งานสามารถค้นหาชื่อสถานีรถไฟ ดังรูปที่ 4.33

รูปที่ 4.33 หน้าแสดงสถานีรถไฟ

โดยเมื่อผู้ใช้งานกดเข้าสู่เมนูสถานีที่ใกล้ที่สุด แอปพลิเคชันจะแสดงหน้าต่างแจ้งเตือนเพื่อเรียกร้องพิกัดของผู้ใช้งานจาก GPS บนมือถือ ด้วยข้อความ “อนุญาตให้ “TrainTrackingSystem”เข้าถึงตำแหน่งที่ตั้งของคุณขณะที่คุณใช้แอพหรือไม่” และมีปุ่ม “ไม่อนุญาต” และ “อนุญาต” ให้ผู้ใช้เลือกที่จะไม่อนุญาตและอนุญาตตามลำดับ ดังรูปที่ 4.34 ซึ่งถ้าหากผู้ใช้กดปุ่ม “อนุญาต” แอปพลิเคชันจะดึงพิกัดปัจจุบันของผู้ใช้งาน

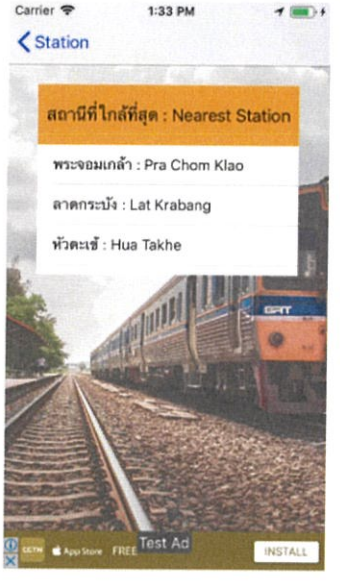
รูปที่ 4.34 หน้าต่างแจ้งเตือนเพื่อเรียกร้องพิกัดของผู้ใช้งาน

แต่ถ้าหากผู้ใช้งานกดปุ่ม “ไม่อนุญาต” แอปพลิเคชันจะแสดงหน้าต่างแจ้งเตือนด้วยข้อความ “การเปิดบริการหาตำแหน่งที่ตั้งจะช่วยให้ “TrainTrackingSystem” กำหนดตำแหน่งที่ตั้งของคุณ” และมีปุ่ม “ยกเลิก” และ “การตั้งค่า” ซึ่งถ้าหากผู้ใช้งานกดปุ่มตั้งค่า (Settings) ผู้ใช้งานสามารถจะไปหน้าการตั้งค่าเพื่อเปิดใช้งานตำแหน่งที่ตั้ง ดังรูปที่ 4.35



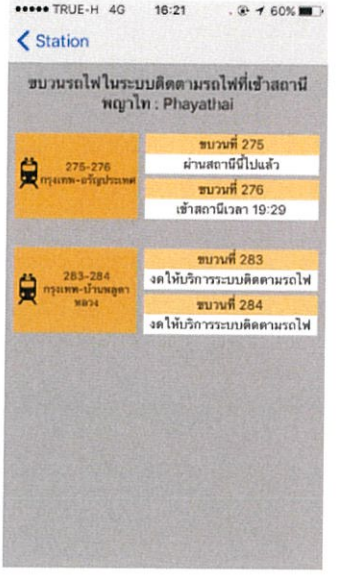
รูปที่ 4.35 ปุ่มการตั้งค่าเมื่อกดแล้วจะนำไปสู่หน้าการตั้งค่าตำแหน่งที่ตั้ง

เมื่อผู้ใช้งานอนุญาตให้แอปพลิเคชันรับพิกัดของผู้ใช้งานจาก GPS บนมือถือแล้ว แอพพลิเคชันจะแสดงสถานีใกล้ตัวที่สุดของผู้ใช้งาน 3 สถานี โดยสถานีแรกจะเป็นสถานีที่ใกล้ตัวผู้ใช้งานที่สุด และอีกสองสถานีเป็นสถานีที่ใกล้เคียงกับสถานนีที่ใกล้ตัวผู้ใช้งานมากที่สุด ดังรูปที่ 4.36 ซึ่งจากรูปที่ 4.36 ผู้ใช้งานอยู่บริเวณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งมีสถานีที่ใกล้ที่สุดคือสถานีพระจอมเกล้า และสถานีที่ใกล้เคียงกับสถานีพระจอมเกล้าคือสถานีลาดกระบังและสถานีหัวตะเข้ตามลำดับ



รูปที่ 4.36 หน้าแสดงสถานีที่ใกล้ตัวผู้ใช้งานที่สุด

ซึ่งเมื่อผู้ใช้งานกดเลือกสถานีรถไฟในรายการสถานีที่ใกล้ที่สุด หรือเลือกสถานีรถไฟในรายการสถานีอื่น ๆ แอปพลิเคชันจะแสดงขบวนรถไฟที่เข้าสถานีรถไฟที่ผู้ใช้เลือก โดยจะแสดงหมายเลขขบวนรถไฟ และเวลาที่ขบวนรถไฟนั้นจะเข้าสู่สถานี ซึ่งถ้าหากรถไฟยังไม่เข้าสู่สถานี แอปพลิเคชันจะแสดงข้อความ “เข้าสถานีเวลา...” ถ้าหากรถไฟผ่านสถานีที่เลือกไปแล้ว แอปพลิเคชันจะแสดงข้อความ “ผ่านสถานีนี้ไปแล้ว” และถ้าหากงดให้บริการระบบติดตามรถไฟ แอปพลิเคชันจะขึ้นข้อความ “งดให้บริการระบบติดตามรถไฟ” ดังรูปที่ 4.37



รูปที่ 4.37 หน้าแสดงขบวนรถไฟที่กำลังเข้าสถานีรถไฟ

4.4.3 การทดสอบแอปพลิเคชันของสมาร์ตโฟนในระบบปฏิบัติการไอโอเอสให้แสดงตำแหน่งของรถไฟให้เร็วขึ้น

เนื่องจากในก่อนหน้านี้ แอปพลิเคชันแสดงผลตำแหน่งรถไฟได้ช้าเนื่องจากคำสั่งการดึงข้อความขึ้นไปแสดงผลบนหน้าแอปพลิเคชันนั้นทำงานได้ช้า จึงเพิ่มคำสั่งในการดึงข้อความให้เร็วขึ้นเข้าไป ดังรูปที่ 4.38

```
DispatchQueue.main.async(execute: {
    self.NumberOfTrain.text =  "\(NumberTrain)"
    self.TrainNameTH.text = "\(TrainNameTH)"
    self.TrainNameEN.text = "\(TrainNameEN)"
    self.updateLB.text = "updated : \(d)"
});
```

รูปที่ 4.38 คำสั่งในการดึงข้อความให้เร็วขึ้น

จากโปรแกรมในรูปที่ 4.38 คำสั่ง DispatchQueues.main.async(excecute: {...}) เป็นคำสั่งที่ใช้ในการจัดการลำดับของการประมวลผล โดยจะเป็นการดึงคำสั่งด้านในมาประมวลผลทันที จะไม่รอให้ประมวลผลทั้งหมดก่อนแล้วจึงแสดงผล โดยคำสั่ง self.NumberOfTrain.text = “\(NumberTrain)” ที่อยู่ใน DispatchQueues จะเป็นการนำค่าที่มีอยู่ในตัวแปรที่ชื่อว่า

NumberTrain มาแสดงผลในรูปแบบของตัวอักษรใน label ที่ชื่อว่า NumberOfTrain เช่นเดียวกับคำสั่งอื่น ๆ ที่นำตัวแปรนั้น ๆ มาแสดงผลใน label

4.5 ทดสอบการติดตั้งแอปพลิเคชันบน App Store

สำหรับการติดตั้งแอปพลิเคชันบน App Store นั้น เมื่อผู้ใช้งานค้นหาแอปพลิเคชัน "ระบบติดตามรถไฟสายตะวันออกอัจฉริยะ" ผ่าน App Store โดยค้นหาคำว่า "TTS KMITL" ซึ่งจะแสดงแอปพลิเคชัน TTS KMITL จะปรากฏขึ้นมา ดังรูปที่ 4.39

รูปที่ 4.39 แอปพลิเคชันบน App Store

โดยที่ผู้พัฒนาสามารถติดตามความเคลื่อนไหวของแอปพลิเคชันได้จากในหน้า Sales and Trend โดยจะมีเมนูหน้าแดชบอร์ด แสดงจำนวนการติดตั้ง นอกจากนั้น ยังมีสถิติ คะแนน และความคิดเห็น เป็นต้น ดังรูปที่ 4.40

รูปที่ 4.40 หน้า Sales and Trend

4.6 การทดสอบการทำงานของเว็บไซต์

ในหน้าเว็บไซต์ที่เป็นภาษาอังกฤษจะแบ่งเป็น 4 ส่วน เช่นเดียวกับหน้าเว็บไซต์ที่เป็นภาษาไทย ดังรูปที่ 4.41

รูปที่ 4.41 หน้าเว็บไซต์หลัก

1) เมนูเลือกขบวนรถไฟ สำหรับแสดงตำแหน่งของรถไฟแต่ละขบวน ซึ่งประกอบไปด้วย 4 ขบวนหลัก ๆ ได้แก่

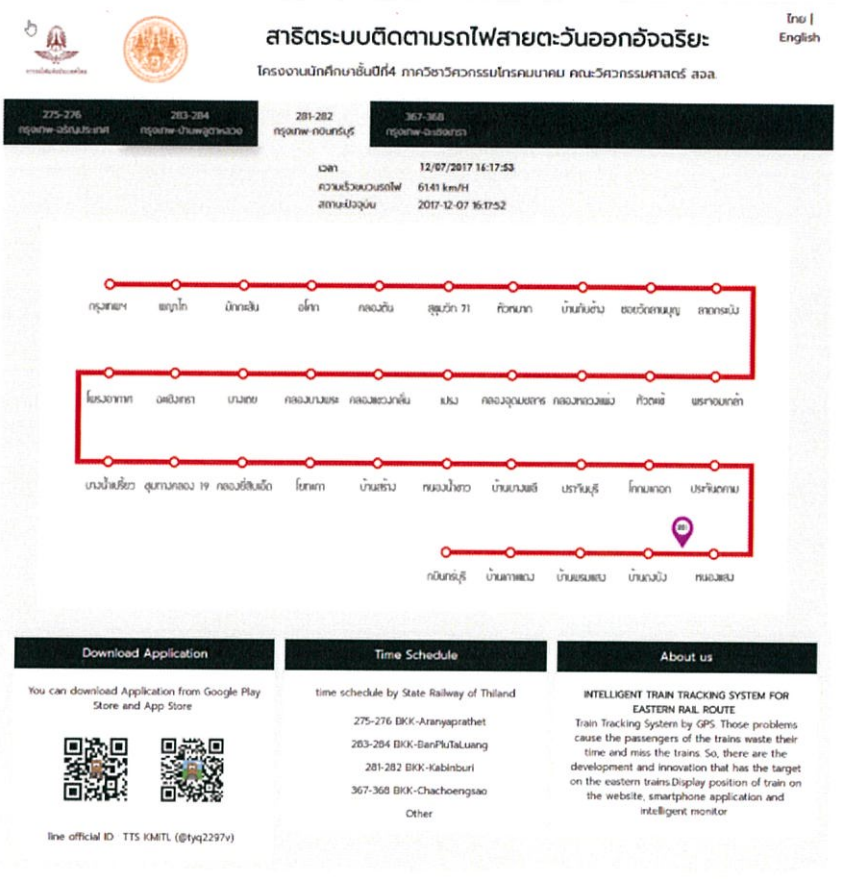
- Train 275-276 BKK-Aranyaprathet
- Train 283-284 BKK-Banplutaluang
- Train 281-282 BKK-Kabinburi
- Train 367-368 BKK-Chachoengsao

2) สำหรับดาวน์โหลดแอปพลิเคชันทั้งในระบบปฏิบัติการแอนดรอยด์ และระบบปฏิบัติการไอโอเอส

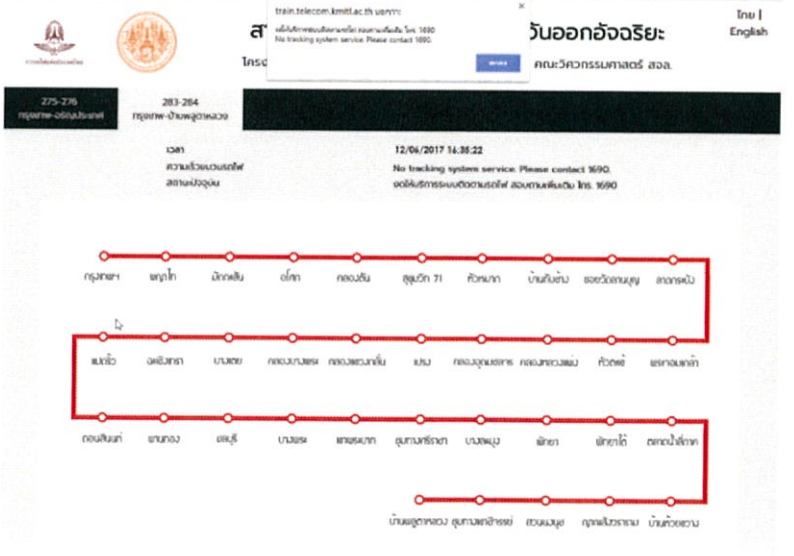
3) ส่วนตารางเวลาของรถไฟขบวนต่าง ๆ ของการรถไฟแห่งประเทศไทย

4) ส่วนแสดงข้อมูลที่มาและความสำคัญของโครงงาน

เมื่อผู้ใช้งานต้องการทราบตำแหน่งรถไฟขบวนต่าง ๆ ผ่านเว็บไซต์ ผู้ใช้งานสามารถใช้งานในส่วนของเมนูเลือกขบวนรถไฟ (ส่วนที่ 1) ผู้พัฒนาได้ทำการกำหนดเงื่อนไขในการทำงานของเว็บไซต์คือ เมื่อเว็บไซต์รับค่า s ที่มีค่าเป็น 1 เว็บไซต์จะแสดงตำแหน่งรถไฟปกติ ซึ่งภายหลังจากเลือกขบวนรถไฟแล้วนั้น หากสถานะ s มีค่าเป็น 1 เว็บไซต์จะแสดงตำแหน่งขบวนรถไฟ (หมุดสีม่วง) เวลาปัจจุบัน ความเร็วขบวนรถไฟ และสถานะปัจจุบันของรถไฟ ดังรูปที่ 4.42 ในทางตรงกันข้าม หากเว็บไซต์รับค่า s ที่มีค่าเป็น 0 แสดงการงดให้บริการของระบบติดตามรถไฟ ดังรูปที่ 4.43

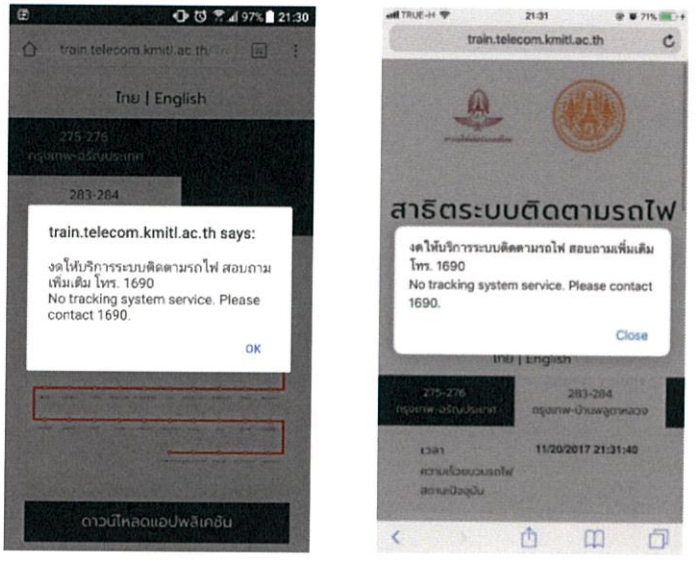


รูปที่ 4.42 หน้าเว็บไซต์แสดงตำแหน่งรถไฟ



รูปที่ 4.43 หน้าเว็บไซต์เมื่อค่า s มีค่าเป็น 0

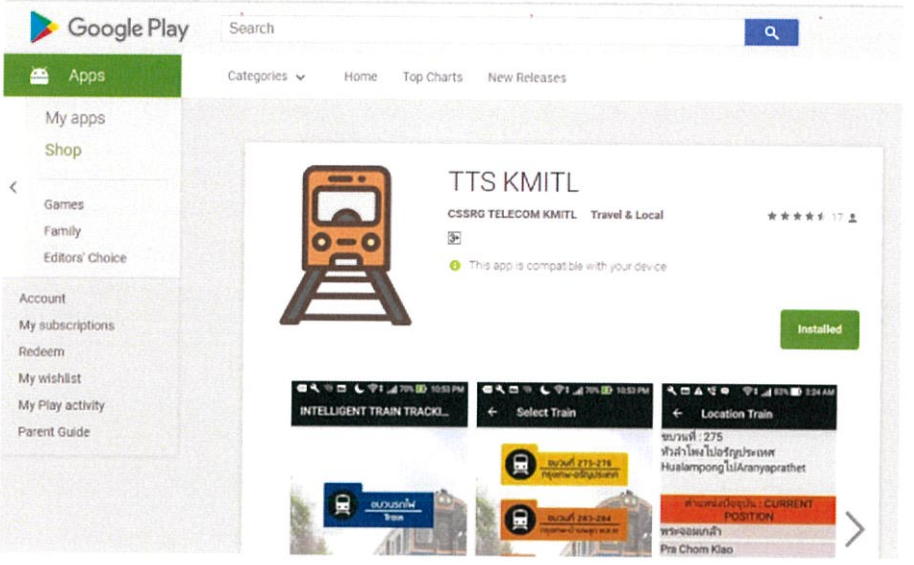
เช่นเดียวกันเมื่อมีการเปิดเว็บไซต์ผ่านสมาร์ตโฟนในระบบปฏิบัติการ android รูปที่ 4.44(ก) และ ระบบปฏิบัติการ iOS รูปที่ 4.44(ข) จะมีกล่องข้อความแจ้งเตือนเช่นเดียวกับการเปิดใช้งานผ่าน PC



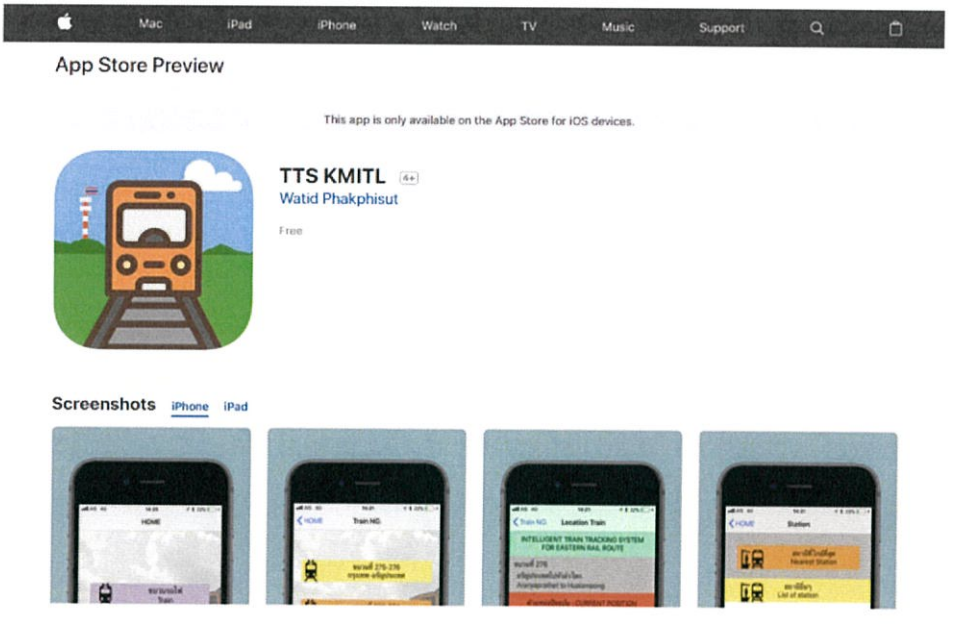
(ก) (ข)

รูปที่ 4.44 หน้าเว็บไซต์ในระบบปฏิบัติการ android (ก) และระบบปฏิบัติการ iOS (ข) ที่ใช้งานผ่านสมาร์ตโฟนเมื่อค่า s มีค่าเป็น 0

ในส่วนดาวน์โหลดแอปพลิเคชัน (ส่วนที่2) ผู้ใช้งานสามารถสแกน QR code หรือเลือกเพื่อดาวน์โหลดแอปพลิเคชันระบบติดตามรถไฟสายตะวันออกอัจฉริยะจาก Google Play Store และ App Store ดังรูปที่ 4.45 และรูปที่ 4.46 ตามลำดับ

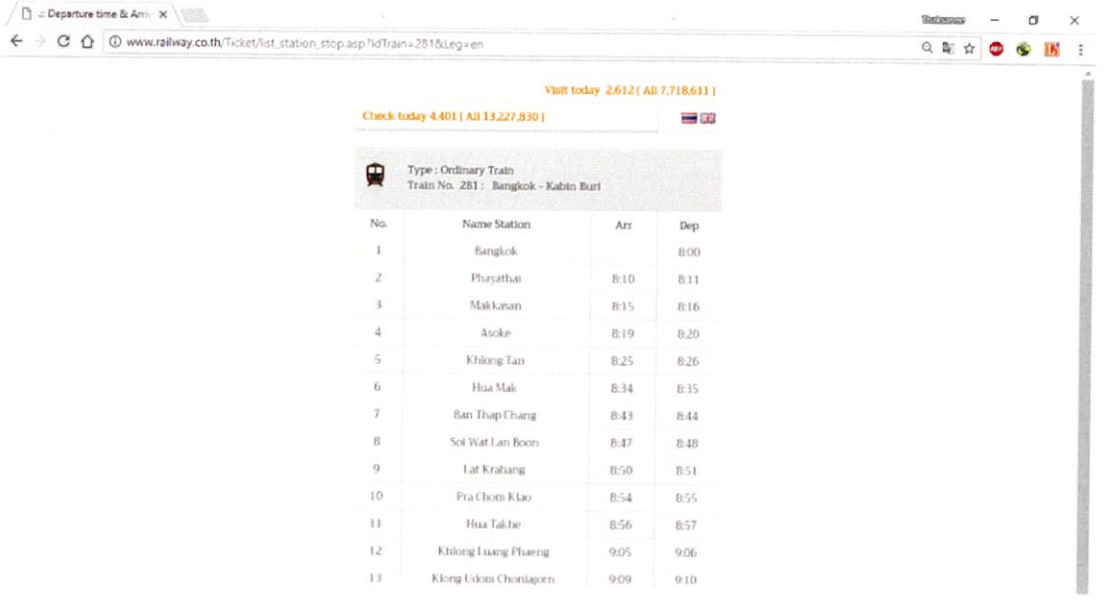


รูปที่ 4.45 หน้าดาวน์โหลดแอปพลิเคชันจาก Google Play Store



รูปที่ 4.46 หน้าดาวน์โหลดแอปพลิเคชันจากจาก App Store

สำหรับส่วนตารางเวลารถไฟ (ส่วนที่3) ผู้ใช้งานสามารถเลือกขบวนรถไฟที่ต้องการทราบตารางเวลา เป็นภาษาอังกฤษตามกำหนดการของการรถไฟแห่งประเทศไทยจากเว็บไซต์การรถไฟแห่งประเทศไทย ดังรูปที่ 4.47



รูปที่ 4.47 ตารางเวลาตามกำหนดการของการรถไฟแห่งประเทศไทย

4.7 การทดสอบของระบบการคำนวณตำแหน่งของรถไฟเพื่อลดภาระการทำงานของแอปพลิเคชัน

ในส่วนของระบบการคำนวณตำแหน่งของรถไฟเพื่อลดการทำงานของแอปพลิเคชันนั้น ผู้วิจัยได้เขียนโปรแกรมที่ใช้ในการคำนวณ และส่งค่าที่ได้จากการคำนวณนั้นเข้าสู่ฐานข้อมูล

←T→	day	id	L	n	bat	spd	No	s	sta
	2018-01-25 10:23:09	1	13.946987	101.932708	100	60.45	957311	1	76
	2018-01-25 10:26:10	1	13.927724	101.956344	100	65.80	957312	1	76
	2018-01-25 10:26:28	1	13.925725	101.958763	100	70.21	957313	1	76
	2018-01-25 10:26:47	1	13.923583	101.961387	100	64.34	957314	1	77
	2018-01-25 10:27:04	1	13.921610	101.963821	100	66.58	957315	1	77
	2018-01-25 10:27:23	1	13.919672	101.966171	100	64.26	957316	1	77

รูปที่ 4.48 ตำแหน่งของรถไฟที่คำนวณโดยใช้ระบบฐานข้อมูล

จากรูปที่ 4.48 การคำนวณหมายเลขตำแหน่งของรถไฟจะถูกส่งมายังคอลัมน์ sta โดยตัวเลขในคอลัมน์ sta จะหมายถึงหมายเลขตำแหน่งของพิกัดตำแหน่งรถไฟปัจจุบัน ซึ่งหมายเลขตำแหน่งที่ 76 แทนตำแหน่งระหว่างสถานีพระปรงและบ้านแก้ง และหมายเลขตำแหน่งที่ 77 แทนตำแหน่งสถานีบ้านแก้ง

4.8 ระบบการประมาณเวลาถึงสถานีของรถไฟ

ในส่วนนี้ผู้วิจัยได้ทำการรวมข้อมูลพิกัดตำแหน่งของรถไฟย้อนหลัง 1 ปี และได้ทำการจำแนกวันในแต่ละอาทิตย์ เพื่อนำไปใช้ประมาณเวลาถึงสถานีของรถไฟในแต่ละวัน ซึ่งผลการทดสอบที่ได้สรุปว่า การจำแนกวันในแต่ละอาทิตย์สามารถประมาณเวลาในการถึงสถานีของรถไฟได้มีความคลาดเคลื่อนเฉลี่ยรวมของแต่ละสถานีน้อยกว่าการรวมข้อมูลแบบทั้งหมด ซึ่งผลการทดสอบแสดงดังตารางที่ 4.2

ตารางที่ 4.2 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟขบวน 275-276 กรุงเทพฯ-อรัญประเทศ เมื่อใช้ชุดข้อมูลแบบรวม และชุดข้อมูลแบบจำแนกวัน

วันที่ทำนาย	ชุดข้อมูลที่ใช้	ความคลาดเคลื่อนเฉลี่ย (นาที)
จันทร์	ชุดข้อมูลแบบรวม	6
	ชุดข้อมูลวันจันทร์	5
อังคาร	ชุดข้อมูลแบบรวม	6
	ชุดข้อมูลวันอังคาร	5
พุธ	ชุดข้อมูลแบบรวม	5
	ชุดข้อมูลวันพุธ	4
พฤหัสบดี	ชุดข้อมูลแบบรวม	7
	ชุดข้อมูลวันพฤหัสบดี	6
ศุกร์	ชุดข้อมูลแบบรวม	6
	ชุดข้อมูลวันศุกร์	4
เสาร์	ชุดข้อมูลแบบรวม	7
	ชุดข้อมูลวันเสาร์	6
อาทิตย์	ชุดข้อมูลแบบรวม	7
	ชุดข้อมูลวันอาทิตย์	5

เมื่อทำการทดสอบระบบการประมาณเวลาถึงสถานีของรถไฟ โดยใช้ชุดข้อมูลแบบจำแนกวันแล้ว ผู้วิจัยได้ทำการประมาณเวลาถึงสถานีของรถไฟโดยใช้ชุดข้อมูลแบบจำแนกวัน พบว่ามีความคลาดเคลื่อนน้อยกว่าการใช้ชุดข้อมูลแบบรวม นอกจากนั้น ผู้วิจัยได้ทดลองเพิ่มตัวแปรในข้อมูลคือ ความเร็วเฉลี่ยของรถไฟ โดยทำการเฉลี่ยความเร็ว 10 ค่าย้อนหลัง ซึ่งพบว่าการประมาณเวลาถึงสถานีของรถไฟโดยใช้ชุดข้อมูลแบบที่มีความเร็วเฉลี่ยมีความคลาดเคลื่อนเฉลี่ยรวมของแต่ละสถานีมากกว่าการใช้ชุดข้อมูลแบบไม่มีความเร็วเฉลี่ย ซึ่งผลการทดสอบแสดงดังตารางที่ 4.3

ตารางที่ 4.3 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟขบวน 275 276 กรุงเทพฯ-อรัญประเทศ เมื่อใช้ชุดข้อมูลแบบมีความเร็วเฉลี่ย และชุดข้อมูลแบบไม่มีความเร็วเฉลี่ย

สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลที่มีความเร็วเฉลี่ย (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลที่ไม่มีความเร็วเฉลี่ย (นาที)
กรุงเทพ	4	7
พญาไท	5	4
มักกะสัน	5	5
อโศก	6	4
คลองตัน	6	4
สุขุมวิท 71	7	3
หัวหมาก	6	3
บ้านทับช้าง	6	5
ซอยวัดลานบุญ	5	4
ลาดกระบัง	5	4
พระจอมเกล้า	4	3
หัวตะเข้	6	4
คลองหลวงแพ่ง	9	5
คลองอุดมชลจร	4	5
เปรง	5	6
คลองแขวงกลั่น	6	4
คลองบางพระ	5	5
บางเตย	7	4
ชุมทางฉะเชิงเทรา	6	4
โพรงอากาศ	5	5
บางน้ำเปรี้ยว	11	3
ชุมทางคลองสิบเก้า	6	4
คลองยี่สิบเอ็ด	5	5
โยทะกา	6	5
บ้านสร้าง	6	5
หนองน้ำขาว	6	5
บ้านปากพลี	5	3
ปราจีนบุรี	4	3
โคกมะกอก	7	4
ประจันตคาม	5	3
หนองแสง	4	4
บ้านดงบัง	4	4
บ้านพรมแสง	4	3
บ้านเกาะแดง	3	4
กบินทร์บุรี	5	3
กบินทร์เก่า	4	3
หนองสัง	4	4
พระปรง	3	3
บ้านแก้ง	4	4
ศาลาลำดวน	4	2

สระแก้ว	6	3
ท่าเกษม	4	2
ห้วยโจด	3	2
วัฒนานคร	6	2
บ้านโป่งคอม	4	2
ห้วยเดื่อ	3	4
อรัญประเทศ	10	5
เฉลี่ยรวม	5	4

จากการทดสอบระบบการประมาณเวลาถึงสถานีของรถไฟ ซึ่งผลการทดสอบในตารางที่ 4.2 และตารางที่ 4.3 สามารถสรุปได้ว่า การใช้กลุ่มข้อมูลตัวอย่างในการประมาณเวลาถึงสถานีของรถไฟแบบจำแนกวัน และไม่ใช้ข้อมูลความเร็วเฉลี่ยของรถไฟ มีความคลาดเคลื่อนเฉลี่ยรวมของแต่ละสถานีน้อยที่สุด ฉะนั้น ผู้วิจัยจึงเลือกใช้การประมาณเวลาถึงสถานีของรถไฟด้วยเกณฑ์ดังกล่าวสำหรับโครงการนี้ โดยผลการทดสอบระบบการประมาณเวลาถึงสถานีรถไฟจำนวน 1 เดือนของรถไฟขบวน 275-276 กรุงเทพฯ-อรัญประเทศ โดยทำการจำแนกวัน แสดงดังตารางที่ 4.4 ถึงตารางที่ 4.10 และผลการทดสอบระบบการประมาณเวลาถึงสถานีรถไฟของรถไฟขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง แสดงดังตารางที่ 4.11

ตารางที่ 4.4 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีรถไฟในวันจันทร์ เมื่อใช้ชุดข้อมูลวันจันทร์

สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันจันทร์ ขาไป (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันจันทร์ ขากลับ (นาที)
กรุงเทพ	9	-
พญาไท	9	3
มักกะสัน	5	6
อโศก	5	4
คลองตัน	9	3
สุขุมวิท 71	5	3
หัวหมาก	5	5
บ้านทับช้าง	7	3
ซอยวัดลานบุญ	7	3
ลาดกระบัง	6	3
พระจอมเกล้า	7	3
หัวตะเข้	6	8
คลองหลวงแพ่ง	6	3
คลองอุดมชลจร	7	3
เปรง	7	3
คลองแขวงกลั่น	5	4
คลองบางพระ	8	4
บางเตย	8	3

ชุมทางฉะเชิงเทรา	7	6
โพรงอากาศ	6	4
บางน้ำเปรี้ยว	6	6
ชุมทางคลองสิบเก้า	6	3
คลองยี่สิบเอ็ด	6	6
โยทะกา	6	7
บ้านสร้าง	7	6
หนองน้ำขาว	8	6
บ้านปากพลี	7	10
ปราจีนบุรี	6	7
โคกมะกอก	2	5
ประจันตคาม	7	9
หนองแสง	2	8
บ้านดงบัง	3	12
บ้านพรมแสง	4	5
บ้านเกาะแดง	3	6
กบินทร์บุรี	3	12
กบินทร์เก่า	2	4
หนองสัง	2	5
พระปรง	5	5
บ้านแก้ง	3	5
ศาลาลำดวน	2	10
สระแก้ว	2	7
ท่าเกษม	2	8
ห้วยโจด	1	9
วัฒนานคร	2	7
บ้านโป่งคอม	1	7
ห้วยเดื่อ	1	5
อรัญประเทศ	-	5
เฉลี่ยรวม	5	6

ตารางที่ 4.5 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันอังคาร เมื่อใช้ชุดข้อมูลวันอังคาร

สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันอังคาร ขาไป (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันอังคาร ขากลับ (นาที)
กรุงเทพ	6	-
พญาไท	6	2
มักกะสัน	5	3
อโศก	6	5
คลองตัน	7	2
สุขุมวิท 71	8	3
หัวหมาก	6	2
บ้านทับช้าง	5	3
ซอยวัดลานบุญ	7	4
ลาดกระบัง	6	5
พระจอมเกล้า	7	4
หัวตะเข้	8	3
คลองหลวงแพ่ง	7	5
คลองอุดมชลจร	9	4
เปรง	6	4
คลองแขวงกลัน	7	4
คลองบางพระ	6	4
บางเตย	5	4
ชุมทางฉะเชิงเทรา	6	5
โพรงอากาศ	5	5
บางน้ำเปรี้ยว	5	7
ชุมทางคลองสิบเก้า	5	5
คลองยี่สิบเอ็ด	5	4
โยทะกา	4	9
บ้านสร้าง	4	5
หนองน้ำขาว	4	7
บ้านปากพลี	4	7
ปราจีนบุรี	6	4
โคกมะกอก	3	3
ประจันตคาม	2	4
หนองแสง	3	4
บ้านดงบัง	3	4
บ้านพรมแสง	3	3
บ้านเกาะแดง	2	3
กบินทร์บุรี	2	4
กบินทร์เก่า	2	4
หนองสัง	2	4
พระปรง	3	3
บ้านแก้ง	2	4
ศาลาลำดวน	2	3
สระแก้ว	2	5

ท่าเกษม	2	3
ห้วยโจด	2	7
วัฒนานคร	1	5
บ้านโป่งคอม	1	3
ห้วยเดื่อ	1	14
อรัญประเทศ	-	4
เฉลี่ยรวม	4	4

ตารางที่ 4.6 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันพุธ เมื่อใช้ชุดข้อมูลวันพุธ

สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันพุธ ขาไป (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันพุธ ขากลับ (นาที)
กรุงเทพ	6	-
พญาไท	7	2
มักกะสัน	8	2
อโศก	7	2
คลองตัน	5	4
สุขุมวิท 71	5	3
หัวหมาก	5	3
บ้านทับช้าง	5	3
ซอยวัดลานบุญ	7	4
ลาดกระบัง	7	2
พระจอมเกล้า	5	3
หัวตะเข้	4	3
คลองหลวงแพ่ง	7	4
คลองอุดมชลจร	5	3
เปรง	5	4
คลองแขวงกลั่น	5	5
คลองบางพระ	6	4
บางเตย	5	3
ชุมทางฉะเชิงเทรา	4	2
โพรงอากาศ	4	4
บางน้ำเปรี้ยว	6	3
ชุมทางคลองสิบเก้า	4	4
คลองยี่สิบเอ็ด	3	4
โยทะกา	5	3
บ้านสร้าง	5	2
หนองน้ำขาว	4	7
บ้านปากพลี	2	4
ปราจีนบุรี	8	3
โคกมะกอก	6	6
ประจันตคาม	4	8
หนองแสง	2	8
บ้านดงบัง	5	9

บ้านพรมแสง	4	4
บ้านเกาะแดง	2	3
กบินทร์บุรี	5	4
กบินทร์เก่า	3	3
หนองสัง	3	5
พระปรง	4	4
บ้านแก้ง	3	5
ศาลาลำดวน	2	3
สระแก้ว	2	8
ท่าเกษม	2	3
ห้วยโจด	2	9
วัฒนานคร	2	6
บ้านโป่งคอม	2	4
ห้วยเดื่อ	1	10
อรัญประเทศ	-	9
เฉลี่ยรวม	4	4

ตารางที่ 4.7 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันพฤหัสบดี เมื่อใช้ชุดข้อมูลวันพฤหัสบดี

สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันพฤหัสบดี ขาไป (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันพฤหัสบดี ขากลับ (นาที)
กรุงเทพ	6	-
พญาไท	7	3
มักกะสัน	4	6
อโศก	4	2
คลองตัน	8	2
สุขุมวิท 71	5	2
หัวหมาก	4	3
บ้านทับช้าง	9	6
ซอยวัดลานบุญ	5	2
ลาดกระบัง	9	3
พระจอมเกล้า	4	3
หัวตะเข้	7	6
คลองหลวงแพ่ง	6	4
คลองอุดมชลจร	5	3
เปรง	4	3
คลองแขวงกลั่น	4	5
คลองบางพระ	5	6
บางเตย	4	6
ชุมทางฉะเชิงเทรา	7	9
โพรงอากาศ	3	9
บางน้ำเปรี้ยว	9	3
ชุมทางคลองสิบเก้า	2	5
คลองยี่สิบเอ็ด	2	7

โยทะกา	6	19
บ้านสร้าง	4	9
หนองน้ำขาว	6	6
บ้านปากพลี	5	17
ปราจีนบุรี	8	6
โคกมะกอก	5	7
ประจันตคาม	1	5
หนองแสง	5	10
บ้านดงบัง	3	5
บ้านพรมแสง	4	16
บ้านเกาะแดง	2	6
กบินทร์บุรี	4	4
กบินทร์เก่า	2	5
หนองสัง	2	9
พระปรง	4	5
บ้านแก้ง	2	6
ศาลาลำดวน	2	5
สระแก้ว	2	5
ท่าเกษม	2	7
ห้วยโจด	1	14
วัฒนานคร	1	7
บ้านโป่งคอม	1	15
ห้วยเดื่อ	1	7
อรัญประเทศ	-	6
เฉลี่ยรวม	4	6

ตารางที่ 4.8 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันศุกร์ เมื่อใช้ชุดข้อมูลวันศุกร์

สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันศุกร์ ขาไป (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันศุกร์ ขากลับ (นาที)
กรุงเทพ	6	-
พญาไท	6	1
มักกะสัน	12	2
อโศก	7	2
คลองตัน	5	5
สุขุมวิท 71	5	2
หัวหมาก	4	2
บ้านทับช้าง	4	3
ซอยวัดลานบุญ	3	3
ลาดกระบัง	3	3
พระจอมเกล้า	6	3
หัวตะเข้	5	2
คลองหลวงแพ่ง	6	4
คลองอุดมชลจร	3	3

เปรง	4	3
คลองแขวงกลั่น	4	4
คลองบางพระ	5	3
บางเตย	5	4
ชุมทางฉะเชิงเทรา	4	5
โพรงอากาศ	8	2
บางน้ำเปรี้ยว	3	2
ชุมทางคลองสิบเก้า	4	3
คลองยี่สิบเอ็ด	3	4
โยทะกา	4	4
บ้านสร้าง	5	5
หนองน้ำขาว	5	5
บ้านปากพลี	5	4
ปราจีนบุรี	5	7
โคกมะกอก	2	8
ประจันตคาม	2	7
หนองแสง	2	6
บ้านดงบัง	2	6
บ้านพรมแสง	2	5
บ้านเกาะแดง	2	5
กบินทร์บุรี	1	5
กบินทร์เก่า	2	4
หนองสัง	2	4
พระปรง	4	4
บ้านแก้ง	4	4
ศาลาลำดวน	2	3
สระแก้ว	1	4
ท่าเกษม	1	4
ห้วยโจด	1	3
วัฒนานคร	1	4
บ้านโป่งคอม	1	3
ห้วยเดื่อ	1	3
อรัญประเทศ	-	4
เฉลี่ยรวม	4	4

ตารางที่ 4.9 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันเสาร์ เมื่อใช้ชุดข้อมูลวันเสาร์

สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันเสาร์ ขาไป (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันเสาร์ ขากลับ (นาที)
กรุงเทพ	7	-
พญาไท	6	6
มักกะสัน	7	5
อโศก	7	2
คลองตัน	8	3
สุขุมวิท 71	6	4
หัวหมาก	8	3
บ้านทับช้าง	5	3
ซอยวัดลานบุญ	6	3
ลาดกระบัง	6	3
พระจอมเกล้า	8	3
หัวตะเข้	7	2
คลองหลวงแพ่ง	6	3
คลองอุดมชลจร	6	4
เปรง	10	4
คลองแขวงกลั่น	6	3
คลองบางพระ	6	5
บางเตย	7	4
ชุมทางฉะเชิงเทรา	11	3
โพรงอากาศ	8	3
บางน้ำเปรี้ยว	9	3
ชุมทางคลองสิบเก้า	6	7
คลองยี่สิบเอ็ด	6	11
โยทะกา	4	4
บ้านสร้าง	5	3
หนองน้ำขาว	5	5
บ้านปากพลี	5	3
ปราจีนบุรี	6	5
โคกมะกอก	2	5
ประจันตคาม	4	5
หนองแสง	4	4
บ้านดงบัง	2	7
บ้านพรมแสง	4	3
บ้านเกาะแดง	5	6
กบินทร์บุรี	6	8
กบินทร์เก่า	2	7
หนองสัง	6	3
พระปรง	3	3
บ้านแก้ง	3	8
ศาลาลำดวน	4	3
สระแก้ว	2	4

ท่าเกษม	3	10
ห้วยโจด	2	3
วัฒนานคร	2	1
บ้านโป่งคอม	1	6
ห้วยเดื่อ	1	8
อรัญประเทศ	-	5
เฉลี่ยรวม	5	5

ตารางที่ 4.10 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟในวันอาทิตย์ เมื่อใช้ชุดข้อมูลวันอาทิตย์

สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันอาทิตย์ ขาไป (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลวันอาทิตย์ ขากลับ (นาที)
กรุงเทพ	4	-
พญาไท	8	6
มักกะสัน	5	6
อโศก	6	2
คลองตัน	6	3
สุขุมวิท 71	12	2
หัวหมาก	4	6
บ้านทับช้าง	8	3
ซอยวัดลานบุญ	3	2
ลาดกระบัง	3	2
พระจอมเกล้า	6	3
หัวตะเข้	3	2
คลองหลวงแพ่ง	5	4
คลองอุดมชลจร	11	7
เปรง	5	3
คลองแขวงกลั่น	5	2
คลองบางพระ	5	5
บางเตย	5	3
ชุมทางฉะเชิงเทรา	8	3
โพรงอากาศ	8	4
บางน้ำเปรี้ยว	4	9
ชุมทางคลองสิบเก้า	5	14
คลองยี่สิบเอ็ด	4	5
โยทะกา	3	4
บ้านสร้าง	7	4
หนองน้ำขาว	4	3
บ้านปากพลี	3	6
ปราจีนบุรี	4	6
โคกมะกอก	2	4
ประจันตคาม	4	4
หนองแสง	3	3
บ้านดงบัง	3	3

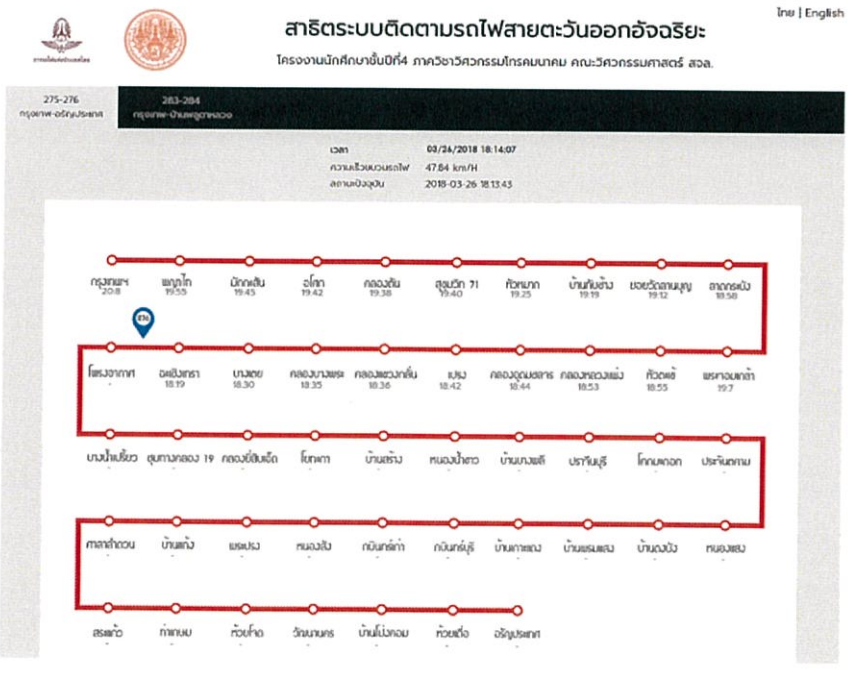
บ้านพรมแสง	2	3
บ้านเกาะแดง	2	3
กบินทร์บุรี	4	9
กบินทร์เก่า	3	3
หนองสัง	2	4
พระปรง	3	5
บ้านแก้ง	3	5
ศาลาลำดวน	2	5
สระแก้ว	1	3
ท่าเกษม	2	4
ห้วยโจด	2	9
วัฒนานคร	1	4
บ้านโป่งคอม	1	5
ห้วยเดื่อ	1	9
อรัญประเทศ	-	6
เฉลี่ยรวม	4	4

ตารางที่ 4.11 ความคลาดเคลื่อนเฉลี่ยรวมจากการประมาณเวลาถึงสถานีของรถไฟขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง เมื่อใช้ชุดข้อมูลแบบจำแนกวัน

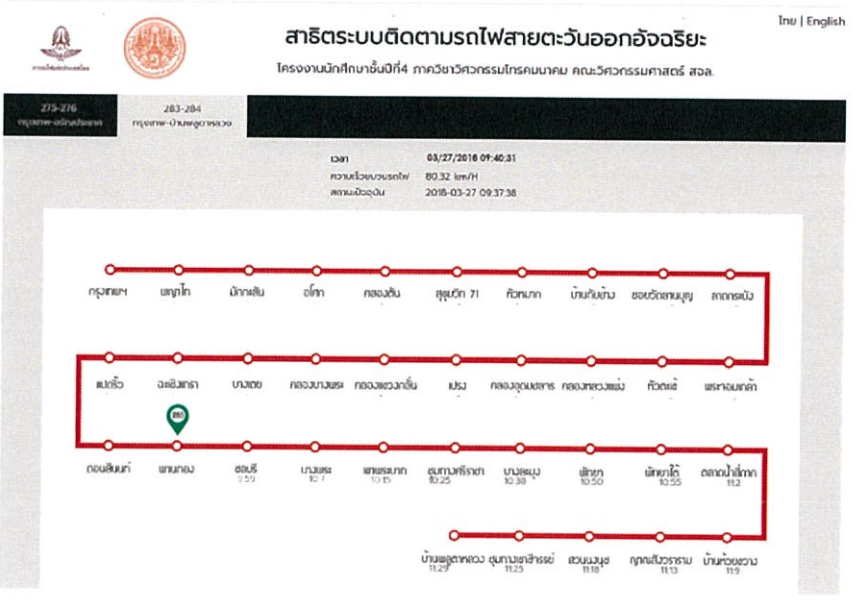
สถานีรถไฟ	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลจำแนกวัน ขาไป (นาที)	ความคลาดเคลื่อนเฉลี่ยของชุดข้อมูลจำแนกวัน ขากลับ (นาที)
กรุงเทพ	5	-
พญาไท	4	1
มักกะสัน	3	1
อโศก	2	2
คลองตัน	4	2
สุขุมวิท 71	5	2
หัวหมาก	5	4
บ้านทับช้าง	4	4
ชอยวัดลานบุญ	5	4
ลาดกระบัง	3	4
พระจอมเกล้า	6	6
หัวตะเข้	3	5
คลองหลวงแพ่ง	5	5
คลองอุดมชลจร	5	7
เปรง	3	7
คลองแขวงกลั่น	3	6
คลองบางพระ	4	6
บางเตย	11	7
ชุมทางฉะเชิงเทรา	4	7
แปดริ้ว	5	5
ดอนสีนนท์	4	6

พานทอง	6	7
ชลบุรี	3	5
บางพระ	3	5
เขาพระบาท	5	6
ชุมทางศรีราชา	4	7
บางละมุง	5	5
พัทยา	4	7
พัทยาใต้	4	7
ตลาดน้ำสี่ภาค	6	6
บ้านห้วยขวาง	6	6
ญาณสังวราราม	3	7
สวนนงนุช	5	10
ชุมทางเขาชีจรรย์	4	7
บ้านพลูตาหลวง	-	10
เฉลี่ยรวม	4	5

เมื่อทำการประมาณเวลาถึงสถานีของรถไฟและส่งข้อมูลไปยังฐานข้อมูลแล้ว ผู้ใช้งานสามารถทราบเวลาถึงสถานีของรถไฟได้จากหน้าเว็บไซต์ โดยแสดงเวลาถึงสถานีของรถไฟในขบวน 275-276 กรุงเทพฯ-อรัญประเทศ และขบวน 283-284 กรุงเทพฯ-บ้านพลูตาหลวง ดังรูปที่ 4.49 และรูปที่ 4.50 ตามลำดับ



รูปที่ 4.49 หน้าเว็บไซต์การประมาณเวลาถึงสถานีของรถไฟขบวน 275-276

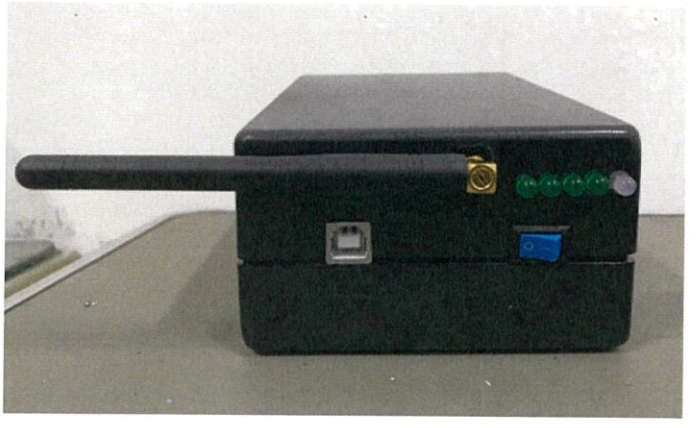


รูปที่ 4.50 หน้าเว็บไซต์การประมาณเวลาถึงสถานีของรถไฟขบวน 283-284

4.9 การส่งค่าพิกัดตำแหน่งรถไฟ และค่าเปอร์เซ็นต์ของแบตเตอรี

จากการออกแบบกล่องอุปกรณ์ติดตามรถไฟ และทำการประกอบอุปกรณ์ในระบบติดตามรถไฟเข้าด้วยกัน ได้แก่ GPS Module 3G Module บอร์ด Arduino Voltage Sensor แบตเตอรีสำรอง รวมทั้ง LED แสดงสถานะแบตเตอรี ซึ่งกล่องอุปกรณ์ติดตามรถไฟขบวนที่ 281-282 กรุงเทพฯ-กบินทร์บุรี และขบวนที่ 367-368 กรุงเทพ-ฉะเชิงเทรา แสดงดังรูปที่ 4.51 และรูปที่ 4.52 ตามลำดับ

รูปที่ 4.51 กล่องอุปกรณ์ติดตามรถไฟ ขบวนที่ 281-282 กรุงเทพฯ-กบินทร์บุรี



รูปที่ 4.52 กล่องอุปกรณ์ติดตามรถไฟ ขบวนที่ 367-368 กรุงเทพ-ฉะเชิงเทรา

ในส่วนของการส่งค่าพิกัดตำแหน่งรถไฟ และค่าเปอร์เซ็นต์ของแบตเตอรีเข้าฐานข้อมูล ได้ทำการเขียนโปรแกรมให้ระบบส่งค่าพิกัดตำแหน่งรถไฟ และค่าเปอร์เซ็นต์ของแบตเตอรีเข้าฐานข้อมูล โดยค่าพิกัดตำแหน่งรถไฟจะได้จาก GPS Module ส่วนค่าเปอร์เซ็นต์ของแบตเตอรีจะได้จากการคำนวณค่าความต่างศักย์ที่อ่านได้จาก Voltage Sensor ซึ่งจะส่งค่าข้อมูลทั้งหมดเข้าฐานข้อมูลโดยใช้ 3G Module แสดงดังรูปที่ 4.53

←T→	day	id	L	n	bat	spd	No	s	w
	2017-11-04 16:56:30	1	14.020529	101.213134	63	51.34	841061	1	0
	2017-11-04 16:56:51	1	14.019393	101.210975	63	49.60	841062	1	0
	2017-11-04 16:58:45	1	14.015501	101.203643	63	0.74	841063	1	0
	2017-11-04 17:01:35	1	14.009209	101.191581	63	46.74	841064	1	0
	2017-11-04 17:04:47	1	14.000437	101.174736	63	51.97	841065	1	0
	2017-11-04 17:05:06	1	13.996809	101.168258	63	48.13	841066	1	0
	2017-11-04 17:05:25	1	13.995144	101.166595	63	46.24	841067	1	0
	2017-11-04 17:05:44	1	13.993308	101.165313	63	45.21	841068	1	0
	2017-11-04 17:06:03	1	13.991547	101.164062	63	44.80	841069	1	0
	2017-11-04 17:07:36	1	13.983475	101.158393	63	36.87	841070	1	0
	2017-11-04 17:08:16	1	13.980616	101.156425	63	35.82	841071	1	0
	2017-11-04 17:09:50	1	13.973967	101.151809	63	30.61	841072	1	0
	2017-11-04 17:11:07	1	13.971530	101.150100	63	1.19	841073	1	0
	2017-11-04 17:11:24	1	13.971494	101.150093	63	0.83	841074	1	0
	2017-11-04 17:16:11	1	13.964520	101.145294	60	58.13	841075	1	0
	2017-11-04 17:16:31	1	13.964520	101.145294	60	58.13	841076	1	0

รูปที่ 4.53 ค่าพิกัดตำแหน่งรถไฟ และค่าเปอร์เซ็นต์ของแบตเตอรีในฐานข้อมูล

จากรูปที่ 4.53 ข้อมูลในระบบติดตามรถไฟได้ส่งไปยังฐานข้อมูล ซึ่งคอลัมน์ day คือข้อมูลวันที่และเวลาที่ส่งข้อมูล คอลัมน์ id คือหมายเลขขบวนรถไฟ คอลัมน์ L คือค่าละติจูดที่รับได้จาก GPS Module คอลัมน์ n คือค่าลองจิจูดที่รับได้จาก GPS Module คอลัมน์ bat คือค่า

เปอร์เซ็นต์แบตเตอรีคงเหลือ ซึ่งได้ทำการเขียนโปรแกรมและส่งค่าเปอร์เซ็นต์แบตเตอรีเข้าฐานข้อมูล คอลัมน์ spd คือค่าความเร็วของรถไฟ คอลัมน์ No คือค่าลำดับข้อมูลที่เข้ามา และคอลัมน์ s คือสถานะการให้บริการของขบวนรถไฟ หากสถานะ s มีค่าเป็น 1 ระบบติดตามรถไฟจะให้บริการตามปกติ และหากสถานะ s มีค่าเป็น 0 ระบบติดตามรถไฟจะงดให้บริการ

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ในปริญญานิพนธ์นี้ ผู้วิจัยได้พัฒนาระบบติดตามรถไฟในสายตะวันออก โดยแบ่งเป็นสามส่วน ส่วนแรก ผู้วิจัยได้พัฒนาอุปกรณ์ในระบบติดตามรถไฟ โดยทำการติดตั้งอุปกรณ์ในการวัดแบตเตอรีคงเหลือ โดยแสดงผลสถานะแบตเตอรีคงเหลือในรูปแบบ LED บนกล่องอุปกรณ์ติดตามรถไฟ เพื่อให้ผู้ที่ถืออุปกรณ์ติดตามรถไฟทราบปริมาณแบตเตอรีคงเหลือขณะนั้น ๆ และเมื่อติดตั้งอุปกรณ์ในการวัดแบตเตอรีคงเหลือแล้ว จะทำการส่งค่าปริมาณแบตเตอรีคงเหลือเข้าฐานข้อมูล เพื่อนำค่าปริมาณแบตเตอรีไปสร้างระบบแจ้งเตือนผ่านทางแอปพลิเคชันไลน์ให้ผู้ดูแลระบบทราบเมื่อปริมาณแบตเตอรีเหลือ 11 เปอร์เซ็นต์ แจ้งเตือนเมื่ออุปกรณ์ติดตามรถไฟมีปัญหา และแจ้งเตือนสถานะการให้บริการของอุปกรณ์ติดตามรถไฟ ส่วนที่สอง ผู้วิจัยได้ทำการพัฒนาส่วนการแสดงผลตำแหน่งรถไฟเพิ่มเติมจากระบบเดิม โดยได้พัฒนาแอปพลิเคชันบนระบบปฏิบัติการไอโอเอส (iOS) และเว็บไซต์ ซึ่งในส่วนของแอปพลิเคชันบนระบบปฏิบัติการไอโอเอสนั้น ผู้วิจัยได้พัฒนาระบบการแสดงผลตำแหน่งรถไฟ โดยแสดงผลตำแหน่งรถไฟเพิ่มอีก 2 ขบวน ได้แก่ ขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และขบวน 367-368 กรุงเทพฯ-ฉะเชิงเทรา และยังเพิ่มฟังก์ชันแสดงขบวนรถไฟที่กำลังเข้าสถานีรถไฟ ทั้งสถานีรถไฟที่อยู่ใกล้ผู้ใช้งานที่สุด และสถานีอื่น ๆ พร้อมทั้งปรับปรุงการแสดงผลของแอปพลิเคชันให้เร็วขึ้น นอกจากนั้นยังได้เพิ่มระบบแจ้งเตือนเมื่อรถไฟกำลังจะเข้าสู่สถานี สำหรับส่วนของการแสดงผลผ่านทางเว็บไซต์นั้น ผู้วิจัยได้พัฒนาระบบการแสดงผลตำแหน่งรถไฟในเวอร์ชันภาษาอังกฤษเพิ่มอีก 2 ขบวน ได้แก่ ขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และขบวน 367-368 กรุงเทพฯ-ฉะเชิงเทรา พร้อมทั้งเพิ่มกล่องข้อความแจ้งเตือนเมื่อระบบติดตามรถไฟงดให้บริการ ส่วนสุดท้าย ผู้วิจัยได้พัฒนาระบบติดตามรถไฟให้มีประสิทธิภาพมากขึ้น โดยปรับปรุงการคำนวณตำแหน่งของรถไฟ เพื่อลดภาระการทำงานของแอปพลิเคชัน และยังสร้างระบบการประมาณเวลาถึงสถานีของรถไฟ เพื่อให้ผู้ใช้งานได้ทราบเวลาที่รถไฟจะถึงสถานีต่าง ๆ และเพิ่มระบบติดตามรถไฟสายตะวันออกขบวน 281-282 กรุงเทพฯ-กบินทร์บุรี และขบวน 367-368 กรุงเทพฯ-ฉะเชิงเทรา

5.2 ข้อเสนอแนะ

เนื่องจากในบางวันผู้ที่ถืออุปกรณ์ติดตามรถไฟไม่ได้นำอุปกรณ์ติดตามรถไฟไปด้วย หรือไม่ได้ชาร์ตอุปกรณ์ติดตามรถไฟ ทำให้แบตเตอรีหมดกลางทางขณะใช้งาน และบางเวลาอุปกรณ์ไม่สามารถรับสัญญาณจีพีเอสได้ ส่งผลให้ผู้ใช้บริการไม่สามารถทราบตำแหน่งของรถไฟ อีกทั้งหากแบตเตอรีสำรองเสื่อม จะไม่สามารถใช้สมการอัตราการลดลงของแบตเตอรีแบบเดิมได้ จึงต้องมีการปรับสมการอัตราการลดลงใหม่อยู่เสมอ

ซึ่งถ้าหากไม่สามารถติดตามรถไฟได้ สามารถตรวจสอบตารางเวลารถไฟได้ที่เว็บไซต์การรถไฟแห่งประเทศไทย http://www.railway.co.th/checktime/checktime.asp หรือเบอร์โทรศัพท์ 1690

บรรณานุกรม

[1] มปป. "GPS." http://www.xn--12cg1cxchd0a2gzc1c5d5a.net/gps/.

[2] Global5 Co., Ltd. "หลักการทำงานของGPS." http://www.global5thailand.com/thai/gps.htm.

[3] มปป. "Arduino UNO R3." http://help.blynk.cc/how-to-connect-different-hardware-with-blynk/arduino/usb-serial.

[4] Thaieasyelec. "Arduino UNO R3." https://goo.gl/f6hXKW.

[5] บริษัท Arduitronics. "GPS Module U Blox NEO 6M." https://www.arduitronics.com/product/348/gps-module-ublox-neo-6mv2-free-gps-antenna.

[6] บริษัท Thaieasyelec. "3G Module." http://www.thaieasyelec.com/products/wireless-modules/gsm-3g/3g-module-uc15-t-1892-detail.html.

[7] บริษัท My Arduino. "Voltage Sensor Module." https://goo.gl/ecDFM7.

[8] Commandrone. "LED." http://commandronestore.com/learning/resistor001.php.

[9] LINE Corporation."LINE Notify." https://help2.line.me/line_notify/web/pc?lang=th.

[10] วิกิพีเดีย สารานุกรมเสรี. "iOS." https://goo.gl/cwbXmQ.

[11] นายทศพล ต้นสมบัติ. "iOS." https://goo.gl/X99bkd.

[12] MacThai. "ภาษา swift." https://www.macthai.com/2014/06/07/intruduction-to-swift-programming-language-from-apple/.

[13] ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร. "HTML." https://www.m-culture.go.th/it/ewt_news.php?nid=69&filename=index

[14] ดร.ปรัชญนันท์ นิลสุข. "JavaScript." http://www.oocities.org/wimonrat555/page1j.html.

[15] มปป. "CSS." http://www.enjoyday.net/webtutorial/css/css_chapter01.html.

[16] Banyapon Poolsawasd. "Xcode." http://www.daydev.com/developer/s6-programming-language/ios-developer-chapter-1-xcode.html.

[17] Rightsoft Corporation Co.,Ltd.. "PHP."
http://rightsoftcorp.com/index.php?name news&file=readnews&id=11.
[18] Thitiwan Sakeepan. "ระบบฐานข้อมูล." http://sakepan023.blogspot.com/p/1.html.
[19] มปป. "Firebase." https://goo.gl/Zbfm6Y.
[20] วิกิพีเดีย สารานุกรมเสรี. "K-NN." https://goo.gl/VgWG3g
[21] มปป. "Haversine formula." https://en.wikipedia.org/wiki/Haversine_formula.

ภาคผนวก

โค้ดคำสั่ง

คำสั่งสำหรับการออกแบบอุปกรณ์ติดตามรถไฟ

คำสั่งในออกแบบอุปกรณ์ติดตามรถไฟ แบ่งเป็น 4 ส่วน คือ ส่วนการรับสัญญาณ GPS ส่วนวัดปริมาณแบตเตอรีคงเหลือ ส่วนการแสดงสถานะแบตเตอรี และส่วนการส่งข้อมูลเข้าฐานข้อมูล โดยใช้ 3G Module

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <AltSoftSerial.h>
#include <Average.h>

AltSoftSerial mySerial;
TinyGPS gps;
SoftwareSerial ss(10,11);

//GPS
float flat, flon , s;
float L = 0.000000;
float n = 0.000000;

//BATTERY
int inputPin = A0;
int val1;
float val2;
float y;
int z;
int a = 100;
Average <int>ave(20);

//LED
int ledgreen1 = 2;
int ledgreen2 = 3;
int ledgreen3 = 4;
int ledgreen4 = 5;
int ledgreen5 = 6;
int ledgreen6 = 7;

void setup()
{
  Serial.begin(9600);
  ss.begin(9600);
  mySerial.begin(9600);
  pinMode(ledgreen1,OUTPUT);
  pinMode(ledgreen2,OUTPUT);
  pinMode(ledgreen3,OUTPUT);
  pinMode(ledgreen4,OUTPUT);
  pinMode(ledgreen5,OUTPUT);
  pinMode(ledgreen6,OUTPUT);
  delay(5000);
}

void loop()
{
  for (int i = 0; i < 20; i++) {
        ave.push(analogRead(inputPin));
    }
  val1 = ave.mean();
  val2 = val1/40.92;
       if (val2 >= 3.62)
       {
        y = (112*val2)-358;
```

```
        z = (int) y;
           if (z > 100)
           {
             z = 100;
           }
       }
       else if (val2 < 3.62)
       {
          y = (8.832e-11)*exp(7.439*val2);
          z = (int) y;
           if (z > 47)
           {
              z = 47;
           }
       }
if (z <= a)
 {
   a = z;
 }
 else
 {
   a = a;
 }

 if (a > 80 && a <= 100)
 {
   digitalWrite(ledgreen1,HIGH);
   digitalWrite(ledgreen2,HIGH);
   digitalWrite(ledgreen3,HIGH);
   digitalWrite(ledgreen4,HIGH);
   digitalWrite(ledgreen5,HIGH);
   digitalWrite(ledgreen6,LOW);
 }
 else if (a > 60 && a <= 80)
 {
   digitalWrite(ledgreen1,LOW);
   digitalWrite(ledgreen2,HIGH);
   digitalWrite(ledgreen3,HIGH);
   digitalWrite(ledgreen4,HIGH);
   digitalWrite(ledgreen5,HIGH);
   digitalWrite(ledgreen6,LOW);
 }
 else if (a > 40 && a <= 60)
 {
   digitalWrite(ledgreen1,LOW);
   digitalWrite(ledgreen2,LOW);
   digitalWrite(ledgreen3,HIGH);
   digitalWrite(ledgreen4,HIGH);
   digitalWrite(ledgreen5,HIGH);
   digitalWrite(ledgreen6,LOW);
 }
 else if (a > 20 && a <= 40)
 {
   digitalWrite(ledgreen1,LOW);
   digitalWrite(ledgreen2,LOW);
   digitalWrite(ledgreen3,LOW);
   digitalWrite(ledgreen4,HIGH);
   digitalWrite(ledgreen5,HIGH);
   digitalWrite(ledgreen6,LOW);
 }
 else if (a > 10 && a <= 20)
 {
   digitalWrite(ledgreen1,LOW);
   digitalWrite(ledgreen2,LOW);
   digitalWrite(ledgreen3,LOW);
```

```
    digitalWrite(ledgreen4,LOW);
    digitalWrite(ledgreen5,HIGH);
    digitalWrite(ledgreen6,LOW);
  }
  else if (a > 0 && a <= 10)
  {
    digitalWrite(ledgreen1,LOW);
    digitalWrite(ledgreen2,LOW);
    digitalWrite(ledgreen3,LOW);
    digitalWrite(ledgreen4,LOW);
    digitalWrite(ledgreen5,LOW);
    digitalWrite(ledgreen6,HIGH);
  }
  else
  {
    digitalWrite(ledgreen1,LOW);
    digitalWrite(ledgreen2,LOW);
    digitalWrite(ledgreen3,LOW);
    digitalWrite(ledgreen4,LOW);
    digitalWrite(ledgreen5,LOW);
    digitalWrite(ledgreen6,LOW);
  }

  //GPS
  bool newData = false;
  for ( unsigned long start = millis(); millis() - start < 1000;)
  {
    while (ss.available())
    {
      char c = ss.read();
      if (gps.encode(c))
      newData = true;
      }
    }
     if (newData)
    {
      unsigned long age;
      gps.f_get_position(&flat, &flon, &age);
      s = gps.f_speed_kmph();
      if ((flat != L) && (flon != n))
      {
        L = flat;
        n = flon;
        SendSQL();
      }
    }
}

void SendSQL()
{
  mySerial.println("AT+QHTTPCFG=\"contextid\",1");
  delay(2000);
  mySerial.println("AT+QHTTPCFG=\"responseheader\",1");
  delay(2000);
  mySerial.println("AT+QICSGP=1,1,\"www.dtac.co.th\",\"\",\"\",1");
  delay(2000);
  mySerial.println("AT+QIACT=1");
  delay(3000);
  mySerial.println("AT+QHTTPURL=93,80");
  delay(2000);
  mySerial.print("http://train.telecom.kmitl.ac.th/ttrack275?");
  mySerial.print("&Lat=");//5
  mySerial.print(L, 6);//9
  mySerial.print("&long=");//6
  mySerial.print(n, 6);//10
```

```
    mySerial.print("&bat=");//5
    mySerial.print(a);//3
    mySerial.print("&speed=");//7
    mySerial.println(s, 2);//5
    delay(2000);
    mySerial.println("AT+QHTTPGET=80");
    delay(4000);
    mySerial.println("AT+QIDEACT=1");
    delay(2000);
}
```

คำสั่งสำหรับระบบแจ้งเตือนปริมาณแบตเตอรี

```
<?ob_start();?>
<?php
$now = time();
$today = strtotime('8:01');
$tomorrow = strtotime('tomorrow 8:01');
if (($today - $now) > 0)
{
  $refreshTime = $today - $now;
}
else
{
  $refreshTime = $tomorrow - $now;
}
header("Refresh: $refreshTime;
url='http://train.telecom.kmitl.ac.th/testlinebat'");
mysql_connect("192.168.1.240","root","ttsproj");
mysql_select_db("tts");
mysql_query("SET NAMES UTF8");
$q = mysql_query("SELECT bat FROM track2 WHERE No=(SELECT max(No) FROM
track2)");
while($e=mysql_fetch_assoc($q))
{
  $a = $e["bat"];
}
if ($a == 11)
{
  line_notify();
}
function line_notify()
{
  $chOne = curl_init();
  curl_setopt( $chOne, CURLOPT_URL, "https://notify-
api.line.me/api/notify");
  curl_setopt( $chOne, CURLOPT_SSL_VERIFYHOST, 0);
  curl_setopt( $chOne, CURLOPT_SSL_VERIFYPEER, 0);
  curl_setopt( $chOne, CURLOPT_POST, 1);
  curl_setopt( $chOne, CURLOPT_POSTFIELDS, "message=แบตเตอรีเหลือ 11 เปอร์เซ็นต์");
  curl_setopt( $chOne, CURLOPT_FOLLOWLOCATION, 1);
  $headers = array( 'Content-type: application/x-www-form-
urlencoded', 'Authorization: Bearer
g7Nf1aZ2eN7nhYHHIHm8lTMbq2s1uqDUO78JHDTA6eG', );
  curl_setopt($chOne, CURLOPT_HTTPHEADER, $headers);
  curl_setopt( $chOne, CURLOPT_RETURNTRANSFER, 1);
  $result = curl_exec( $chOne );
  if(curl_error($chOne)) { echo 'error:' . curl_error($chOne); }
  else { $result_ = json_decode($result, true);
echo "status : ".$result_['status']; echo "message : ". $result_['message'];
}
  curl_close( $chOne );
```

```
  exit();
}
?>
<? ob_end_flush() ?>
```

คำสั่งสำหรับระบบแจ้งเตือนเมื่ออุปกรณ์ที่ใช้ในการติดตามรถไฟมีปัญหา

```
<?php
$now = time();
$today = strtotime('7:01');
$tomorrow = strtotime('tomorrow 7:01');
if (($today - $now) > 0)
{
  $refreshTime = $today - $now;
}
else
{
  $refreshTime = $tomorrow - $now;
}
header("Refresh: $refreshTime;
url='http://train.telecom.kmitl.ac.th/testlinetrack283errornew.php'");
mysql_connect("192.168.1.240","root","ttsproj");
mysql_select_db("tts");
mysql_query("SET NAMES UTF8");
date_default_timezone_set("Asia/Bangkok");
$timeH = date("H");
$timei = date("i");
$t283 = mysql_query("SELECT date_format(day, '%H:%i:%s') as time FROM track
WHERE No=(SELECT max(No) FROM track)");
while($a = mysql_fetch_assoc($t283))
{
  $b = $a["time"];
}
$time283 = explode(":", $b);
$s283 = mysql_query("SELECT s FROM track WHERE No=(SELECT max(No) FROM
track)");
while($c = mysql_fetch_assoc($s283))
{
  $st283 = $c["s"];
}
$timeNow = (($timeH*60)+$timei);
$timeServer = (($time283[0]*60) + $time283[1]);
if (($timeH >= 7 && $timeH <= 18) && ($time283[0] >= 7 && $time283[0] <= 18)
&& ($timeNow - $timeServer >= 60) && ($st283 == 1))
{
  line_notify();
}
function line_notify()
{
  $chOne = curl_init();
  curl_setopt( $chOne, CURLOPT_URL, "https://notify-
api.line.me/api/notify");
  curl_setopt( $chOne, CURLOPT_SSL_VERIFYHOST, 0);
  curl_setopt( $chOne, CURLOPT_SSL_VERIFYPEER, 0);
  curl_setopt( $chOne, CURLOPT_POST, 1);
  curl_setopt( $chOne, CURLOPT_POSTFIELDS, "message=283 อุปกรณ์มีปัญหา");
  curl_setopt( $chOne, CURLOPT_FOLLOWLOCATION, 1);
  $headers = array( 'Content-type: application/x-www-form-urlencoded',
'Authorization: Bearer g7Nf1aZ2eN7nhYHHIHm8lTMbq2s1uqDUO78JHDTA6eG', );
  curl_setopt($chOne, CURLOPT_HTTPHEADER, $headers);
  curl_setopt( $chOne, CURLOPT_RETURNTRANSFER, 1);
  $result = curl_exec( $chOne );
```

```
  if(curl_error($chOne)) { echo 'error:' . curl_error($chOne); }
  else { $result_ = json_decode($result, true);
  echo "status : ".$result_['status']; echo "message : ".
$result ['message']; }
  curl_close( $chOne );
  exit();
}
?>
```

คำสั่งสำหรับระบบแจ้งเตือนสถานะการให้บริการของอุปกรณ์ที่ใช้ในการติดตามรถไฟ

1. คำสั่งเมื่ออุปกรณ์ที่ใช้ในการติดตามรถไฟให้บริการ

```
<?php
$now = time();
$today = strtotime('5:01');
$tomorrow = strtotime('tomorrow 5:01');
if (($today - $now) > 0)
{
  $refreshTime = $today - $now;
}
else
{
  $refreshTime = $tomorrow - $now;
}
header("Refresh: $refreshTime;
url='http://train.telecom.kmitl.ac.th/testlinetrack275online.php'");
mysql_connect("192.168.1.240","root","ttsproj");
mysql_select_db("tts");
mysql_query("SET NAMES UTF8");
date_default_timezone_set("Asia/Bangkok");
$date = date("d");
$d275 = mysql_query("SELECT date_format(day, '%Y-%m-%d') as date FROM track2
WHERE No=(SELECT max(No) FROM track2)");
while($a = mysql_fetch_assoc($d275))
{
  $b = $a["date"];
}
$date275 = explode("-", $b);
$s275 = mysql_query("SELECT s FROM track2 WHERE No=(SELECT max(No) FROM
track2)");
while($c = mysql_fetch_assoc($s275))
{
  $st275 = $c["s"];
}
$b275 = mysql_query("SELECT bat FROM track2 WHERE No=(SELECT max(No) FROM
track2)");
while($d=mysql_fetch_assoc($b275))
{
  $bat275 = $d["bat"];
}
if (($date - $date275[2] == 0) && ($st275 == 1))
{
  line_notify();
}
function line_notify()
{
  $chOne = curl_init();
  curl_setopt( $chOne, CURLOPT_URL, "https://notify-api.line.me/api/notify");
  curl_setopt( $chOne, CURLOPT_SSL_VERIFYHOST, 0);
  curl_setopt( $chOne, CURLOPT_SSL_VERIFYPEER, 0);
  curl_setopt( $chOne, CURLOPT_POST, 1);
```

```
  curl_setopt( $chOne, CURLOPT_POSTFIELDS, "message=\n275-276 สถานะ ออนไลน์\nแบตเตอรี่
เหลือ $bat275 เปอร์เซ็นต์");
  curl_setopt( $chOne, CURLOPT_FOLLOWLOCATION, 1);
  $headers = array( 'Content type: application/x www-form-urlencoded',
'Authorization: Bearer g7N11aZ2eN7nhYHHlHm8lTMbq2sluqDUO78JHDTA6eG', );
  curl_setopt($chOne, CURLOPT_HTTPHEADER, $headers);
  curl_setopt( $chOne, CURLOPT_RETURNTRANSFER, 1);
  $result = curl_exec( $chOne );
  if(curl_error($chOne)) { echo 'error:' . curl_error($chOne); }
  else { $result_ = json_decode($result, true);
  echo "status : ".$result_['status']; echo "message : ".
$result_['message']; }
  curl_close( $chOne );
  exit();
}
?>
```

2. คำสั่งเมื่ออุปกรณ์ที่ใช้ในการติดตามรถไฟงดให้บริการ

```
<?php
$now = time();
$today = strtotime('8:01');
$tomorrow = strtotime('tomorrow 8:01');
if (($today - $now) > 0)
{
  $refreshTime = $today - $now;
}
else
{
  $refreshTime = $tomorrow - $now;
}
header("Refresh: $refreshTime;
url='http://train.telecom.kmitl.ac.th/testlinetrack275offline.php'");
mysql_connect("192.168.1.240","root","ttsproj");
mysql_select_db("tts");
mysql_query("SET NAMES UTF8");
date_default_timezone_set("Asia/Bangkok");
$date = date("d");
$d275 = mysql_query("SELECT date_format(day, '%Y-%m-%d') as date FROM track2
WHERE No=(SELECT max(No) FROM track2)");
while($a = mysql_fetch_assoc($d275))
{
  $b = $a["date"];
}
$date275 = explode("-", $b);
$s275 = mysql_query("SELECT s FROM track2 WHERE No=(SELECT max(No) FROM
track2)");
while($c = mysql_fetch_assoc($s275))
{
  $st275 = $c["s"];
}
$b275 = mysql_query("SELECT bat FROM track2 WHERE No=(SELECT max(No) FROM
track2)");
while($d=mysql_fetch_assoc($b275))
{
  $bat275 = $d["bat"];
}
if (($date - $date275[2] != 0))
{
  line_notify();
}
function line_notify()
{
```

```
  $chOne = curl_init();
  curl_setopt( $chOne, CURLOPT_URL, "https://notify-api.line.me/api/notify");
  curl_setopt( $chOne, CURLOPT_SSL_VERIFYHOST, 0);
  curl_setopt( $chOne, CURLOPT_SSL_VERIFYPEER, 0);
  curl_setopt( $chOne, CURLOPT_POST, 1);
  curl_setopt( $chOne, CURLOPT_POSTFIELDS, "message=\n275-276 สถานะ ออฟไลน์\nแบตเตอรี
เหลือ $bat275 เปอร์เซ็นต์");
  curl_setopt( $chOne, CURLOPT_FOLLOWLOCATION, 1);
  $headers = array( 'Content-type: application/x-www-form-urlencoded',
'Authorization: Bearer g7Nf1aZ2eN7nhYHHIHm8lTMbq2s1uqDUO78JHDTA6eG', );
  curl_setopt($chOne, CURLOPT_HTTPHEADER, $headers);
  curl_setopt( $chOne, CURLOPT_RETURNTRANSFER, 1);
  $result = curl_exec( $chOne );
  if(curl_error($chOne)) { echo 'error:' . curl_error($chOne); }
  else { $result_ = json_decode($result, true);
  echo "status : ".$result_['status']; echo "message : ".
$result_['message']; }
  curl_close( $chOne );
  exit();
}
?>
```

คำสั่งสำหรับเว็บไซต์

คำสั่งในการแสดงผลหน้าหลักของเว็บไซต์ภาษาอังกฤษจะใช้ไฟล์ indexEN.html ในการแสดงผล โดยแบ่งเป็น 5 ส่วน คือ ส่วนหัวของเว็บไซต์ ส่วนเลือกขบวนรถไฟ ส่วนดาวน์โหลด ส่วนตารางเวลารถไฟและส่วนเกี่ยวกับเรา

```
<!DOCTYPE html>
<html>

<title>INTELLIGENT TRAIN TRACKING SYSTEM FOR EASTERN RAIL ROUTE </title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css">
<link href="https://fonts.googleapis.com/css?family=Prompt" rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Overpass"
rel="stylesheet">
<style>
h1,h2,h3,h4,h5,h6 {font-family: "Prompt"}
body {font-family: "Prompt"}

       a {
      text-decoration:none;
       }
</style>

<body class="w3-light-gray" >

       <!-- Header -->
       <header class="w3-row w3-white" >
       <a style="display:block"
href="http://train.telecom.kmitl.ac.th/Traintrack/new/indexEN">
              <div class="w3-col w3-padding-24 w3-center m12 l3"  ><img
src="img2559/logo.gif" alt="logo" height="100px"></div>
              <div class="w3-col w3-padding-16 w3-center m12
18"><h1><b>INTELLIGENT TRAIN TRACKING SYSTEM FOR EASTERN RAIL ROUTE </b></h1>
              <h4>4th year student project Telecommunications Engineering
KMITL</h4>
              </div>
       </a>
```

```
            <div class="w3-col w3-center m12 l1"><h5><a
href="http://train.telecom.kmitl.ac.th">ไทย</a> | <a
href="http://train.telecom.kmitl.ac.th/Traintrack/new/indexEN.html">English</
a></h5></div>
      </header>

      <!-- Image header -->
      <header class="w3-display-container w3-wide w3-center" style=""
id="home">
            <img src="img2559\รถไฟ3.jpg" alt="train" height="600px">
            <div class="w3-display-topleft w3-padding-48 w3-center"
style="left:10% ;"  >
            <h2 class="w3-xxlarge w3-text-light-black w3-light-gray w3-
opacity">Choose train</h2>
            <h6><a class="w3-btn w3-white w3-padding-large w3-large w3-
opacity w3-hover-opacity-off" style="width:100%"
href="http://train.telecom.kmitl.ac.th/Traintrack/new/select275EN.html"
><span class="w3-hide-small">275-276</span> BKK-Aranyaprathet</a></h6>
            <h6><a class="w3-btn w3-white w3-padding-large w3-large w3-
opacity w3-hover-opacity-off" style="width:100%"
href="http://train.telecom.kmitl.ac.th/Traintrack/new/select283EN.html"
><span class="w3-hide-small">283-284</span> BKK-BanPluTaLuang</a></h6>
            <h6><a class="w3-btn w3-white w3-padding-large w3-large w3-
opacity w3-hover-opacity-off" style="width:100%" href="" ><span class="w3-
hide-small">281-282</span> BKK-Kabinburi<br>(future)</a></h6>
            <h6><a class="w3-btn w3-white w3-padding-large w3-large w3-
opacity w3-hover-opacity-off" style="width:100%" href="" ><span class="w3-
hide-small">367-368</span> BKK-Chachoengsao<br>(future)</a></h6>
            </div>
       </header>

       <!-- grid-->
      <div class="w3-row-padding w3-container w3-center">
            <!-- download-->
            <div class="w3-col w3-padding-32  m6 l4 w3-center " >
            <div class="w3-container  w3-black" ><h4 class="w3-
large">Download Application</h4></div>
            <div class="w3-container w3-padding w3-white">
            <p>You can download Application from Google Play Store and App
Store <br></p>
            <a href="https://goo.gl/V4rVmc"><img class="w3-margin"
src="img2559/qrcode.jpg" alt="qrcode" width="30%"></a>
            <a href="https://goo.gl/zWbYQj"><img class="w3-margin"
src="img2559/qrcode_ios.png" alt="qrcode" width="30%"></a>
            <p>line official ID : TTS KMITL (@tyq2297v) <br></p>
            </div>
            </div>

            <!-- table-->
            <div class="w3-col w3-padding-32  m6 l4 w3-center " >
            <div class="w3-container  w3-black" ><h4 class="w3-large">Time
Schedule</h4></div>
            <div class="w3-container w3-padding w3-white">
            <p>time schedule by State Railway of Thiland</p>
            <ul class="w3-ul w3-hoverable">
            <li><a
href="http://www.railway.co.th/Ticket/list_station_stop.asp?IdTrain=275&Leg=e
n">275-276 BKK-Aranyaprathet</a></li>
            <li><a
href="http://www.railway.co.th/Ticket/list_station_stop.asp?IdTrain=283&Leg=e
n">283-284 BKK-BanPluTaLuang</a></li>
            <li><a
href="http://www.railway.co.th/Ticket/list_station_stop.asp?IdTrain=281&Leg=e
n">281-282 BKK-Kabinburi</a></li>
```

```
            <li><a
href="http://www.railway.co.th/Ticket/list_station_stop.asp?IdTrain=367&Leg=e
n">367-368 BKK-Chachoengsao</a></li>
            <li><a
href="http://www.railway.co.th/checktime/checktime.asp?lenguage=Eng">Other</a
></li>
            </ul></div>
            </div>

            <!-- about-->
            <div class="w3-col w3-padding-32  m12 l4 w3-center " >
            <div class="w3-container  w3-black" ><h4 class="w3-large">About
us</h4></div>
            <div class="w3-container w3-padding w3-white">
            <p><b>INTELLIGENT TRAIN TRACKING SYSTEM FOR EASTERN RAIL
ROUTE</b> <br>Train Tracking System by GPS. Those problems cause the
passengers of the trains waste their time and miss the trains. So, there are
the development and innovation that has the target on the eastern
trains.Display position of train on the website, smartphone application and
intelligent monitor. </p>
            </div>
            </div>

      </div>

      <footer class="w3-container w3-dark-grey w3-padding-32 w3-padding-
xlarge">
            <p style="text-align:right;">submitted by : Apitchaya
Siriwanitpong , Apichet Promgate and Arif Dataesatu
            <br>Thaksanee Jianwibulsuk, Thanyaporn Laojirachetkul
            <br>Advisors : Prof.Dr.Pornchai Supnithi , Asst.Prof.Dr.Nuttakan
Puttarak and Dr.Watid Phakphisut</p>
            <p style="text-align:right;"><script
src="https://www.w3counter.com/tracker.js?id=109422"></script></p>
            <a href="#" class="w3-btn w3-padding-large w3-margin-bottom"><i
class="fa fa-arrow-up w3-margin-right"></i>To the top</a>

      </footer>

</body>

</html>
```

คำสั่งในการแสดงตำแหน่งปัจจุบันของรถไฟบนเว็ปไซต์ จะใช้ไฟล์ selected281.html

```
<!DOCTYPE html>
<html>

<title>ระบบติดตามรถไฟสายตะวันออกอัจฉริยะ </title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css">
<link href="https://fonts.googleapis.com/css?family=Prompt" rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Overpass"
rel="stylesheet">
<style>
h1,h2,h3,h4,h5,h6 {font-family: "Prompt"}
body {font-family: "Prompt"}
</style>
<!DOCTYPE html>
<html>
<head>

<title>สาธิตระบบติดตามรถไฟสายตะวันออกอัจฉริยะ </title>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css">
<link href="https://fonts.googleapis.com/css?family=Prompt" rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Overpass"
rel="stylesheet">
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></scri
pt>
<script type="text/javascript" src='map281.js'></script>

</head>

<body class="w3-light-gray" >

<style>
h1,h2,h3,h4,h5,h6 {font-family: "Prompt"}
body {font-family: "Prompt"}

       a {
      text-decoration:none;
       }
       .c {width:100%;}
       .d {width:100%;}
</style>

<script>
            $(document).ready(function(){
                  $('.d').css({
                        position: 'absolute',
                        top: $('.c').offset().top,
                        left: $('.c').offset().left,
                        width: $('.c').offset().width
                  });
            });

</script>

       <!-- Header -->
       <header class="a w3-row w3-white" >
       <a style="display:block" href="http://train.telecom.kmitl.ac.th">
              <div class="w3-col w3-padding-24 w3-center m12 l3"  ><img
src="img2559\logo.gif" alt="logo" height="100px"></div>
              <div class="w3-col w3-padding-16 w3-center m12 l8"><h1><b>สาธิตระบบ
ติดตามรถไฟสายตะวันออกอัจฉริยะ </b></h1>
              <h4>โครงงานนักศึกษาชั้นปีที่4 ภาควิชาวิศวกรรมโทรคมนาคม  คณะวิศวกรรมศาสตร์ สจล.</h4>
              </div>
       </a>
              <div class="w3-col w3-center m12 l1"><h5><a
href="http://train.telecom.kmitl.ac.th/Traintrack/new/select281.html">ไทย</a>
| <a
href="http://train.telecom.kmitl.ac.th/Traintrack/new/select281EN.html">Engli
sh</a></h5></div>
       </header>

       <div class=" w3-black w3-btn-group">
              <a class="w3-btn w3-black w3-padding-large w3-opacity w3-hover-
gray" href="http://train.telecom.kmitl.ac.th/Traintrack/new/select275.html"
>275-276 <br>กรุงเทพ-อรัญประเทศ</a>
              <a class="w3-btn w3-black w3-padding-large w3-opacity w3-hover-
gray" href="http://train.telecom.kmitl.ac.th/Traintrack/new/select283.html"
>283-284 <br>กรุงเทพ-บ้านพลูตาหลวง</a>
              <a class="w3-btn w3-light-gray w3-padding-large">281-282<br>
กรุงเทพ-กบินทร์บุรี</a>
```

```
            <a class="w3-btn w3-black w3-padding-large w3-opacity w3-hover-
gray" href="http://train.telecom.kmitl.ac.th/Traintrack/new/select367.html"
>367-368 <br>กรุงเทพ-ฉะเชิงเทรา</a>

        </div>

        <div class="b w3-light-gray w3-margin">
        <table class="" align="center" >
        <tr>
       <td style="width:50%;"><b>เวลา</b> </td>
          <td><iframe
src="http://free.timeanddate.com/clock/i5n6uxfc/n28/fn6/fs15/tct/pct/ftb/tt0/
tw0/tm3/td2/th1/tb2" frameborder="0" width="147" height="20"
allowTransparency="true"></iframe>
        </td>
    </tr>

    <tr>
       <td>ความเร็วขบวนรถไฟ</td>
       <td><a id="myspeed"></a>
    </tr>

       <tr>
      <td>สถานะปัจจุบัน </td>
        <td><a id="update"></a>
    </tr>
    </tr>

       </table>
       </div>

       <div class="c" align="center">
       <img src="img2559\select281.jpg" alt="qrcode" width="90%">
       </div>
       <div class="d" align="center">
       <img id="position" width="90%" />
       </div>


       <div>
        <!-- grid-->
       <div class="w3-row-padding w3-container w3-center">
             <!-- download-->
             <div class="w3-col w3-padding-32  m6 l4 w3-center " >
             <div class="w3-container  w3-black" ><h4 class="w3-large">ดาวน์โหลด
แอปพลิเคชัน</h4></div>
             <div class="w3-container w3-padding w3-white">
             <p>ท่านสามารถดาวน์โหลดแอปพลิเคชันระบบติดตามรถไฟสายตะวันออกอัจฉริยะผ่านทาง  Google Play Store
<br></p>
             <a href="https://goo.gl/V4rVmc"><img class="w3-margin"
src="img2559/qrcode.jpg" alt="qrcode" width="30%"></a>
             </div>
             </div>

             <!-- table-->
             <div class="w3-col w3-padding-32  m6 l4 w3-center " >
             <div class="w3-container  w3-black" ><h4 class="w3-large">ตารางเวลา
รถไฟ</h4></div>
             <div class="w3-container w3-padding w3-white">
             <p>ตารางเวลารถไฟตามกำหนดการของการรถไฟแห่งประเทศไทย</p>
             <ul class="w3-ul w3-hoverable">
             <li><a
href="http://www.railway.co.th/Ticket/list_station_stop.asp?IdTrain=275&Leg=t
h">275-276 กรุงเทพ-อรัญประเทศ</a></li>
```

```
                <li><a
href="http://www.railway.co.th/Ticket/list_station_stop.asp?IdTrain=283&Leg=t
h">283-284 กรุงเทพ-บ้านพลูตาหลวง</a></li>
                <li><a
href="http://www.railway.co.th/Ticket/list_station_stop.asp?IdTrain 281&Leg=t
h">281-282 กรุงเทพ-กบินทร์บุรี</a></li>
                <li><a
href="http://www.railway.co.th/Ticket/list_station_stop.asp?IdTrain=367&Leg=t
h">367-368 กรุงเทพ-ฉะเชิงเทรา</a></li>
                <li><a href="http://www.railway.co.th/checktime/checktime.asp">
ขบวนอื่นๆ</a></li>
                </ul></div>
                </div>

                <!-- about-->
                <div class="w3-col w3-padding-32  m12 l4 w3-center " >
                <div class="w3-container  w3-black" ><h4 class="w3-large">เกี่ยวกับเรา
</h4></div>
                <div class="w3-container w3-padding w3-white">
                <p><b>โครงงานระบบติดตามรถไฟสายตะวันออกอัจฉริยะ</b> <br>เป็นระบบติดตามรถไฟด้วยระบบจีพีเอส จัดทำขึ้นเพื่อ
แก้ไขปัญหาการรอรถไฟเป็นเวลานาน ระบบจะแสดงตำแหน่งของรถไฟผ่านทางเว็บไซต์ หน้าจอแสดงผลบริเวณหน้าตึกภาควิชาโทรคมนาคม และแอปพลิเคชัน ให้
ผู้ใช้งานสามารถเลือกใช้ได้</p>
                </div>
                </div>

        </div>

        <footer class="w3-container w3-dark-grey w3-padding-32 w3-padding-
xlarge">
                <p style="text-align:right;">จัดทำโดย : นางสาวอภิชญา สิริวณิชพงศ์ , นายอภิเชษฐ์
พรมเกตุ , นายอารีฟ ดาแดสาดู <br>อาจารย์ที่ปรึกษา : ศ.ดร.พรชัย ทรัพย์นิธิ ผศ.ดร.ณัฐกานต์ พุทธรักษ์ และดร.เวชิต ภาคย์พิสุทธิ์
</p>
                <p style="text-align:right;"><script
src="https://www.w3counter.com/tracker.js?id=109423"></script></p>
                <a href="#" class="w3-btn w3-padding-large w3-margin-bottom"><i
class="fa fa-arrow-up w3-margin-right"></i>To the top</a>

        </footer>

        </div>

</body>

</html>
```

คำสั่งในการดึงข้อมูลจากฐานข้อมูลมาแสดงผลบนเว็บไซต์ในหน้าแสดงตำแหน่งของรถไฟขบวน 281 จะใช้ไฟล์ map281.js โดยแบ่งเป็น 3 ส่วน คือ ส่วนกำหนดพิกัดสถานีรถไฟ ส่วน refresh และทำซ้ำ และส่วนเงื่อนไขการแสดงตำแหน่งรถไฟ

```
var train=275;
var station = [
[       13.734988       ,       13.743988       ,       100.512247      ,       100.521247
        ],      //      กรุงเทพ
[       13.743998       ,       13.767855       ,       100.512247      ,       100.528571
        ],      //      กรุงเทพ-พญาไท
[       13.752534       ,       13.761534       ,       100.528581      ,       100.537581
        ],      //      พญาไท
[       13.749211       ,       13.761534       ,       100.537591      ,       100.542881
        ],      //      พญาไท-มักกะสัน
[       13.749211       ,       13.758211       ,       100.542891      ,       100.551891
        ],      //      มักกะสัน
```

```
[ 13.745856 , 13.758211 , 100.551901 , 100.558616 ], // มักกะสัน-อโศก
[ 13.745856 , 13.754856 , 100.558626 , 100.567626 ], // อโศก
[ 13.741226 , 13.754856 , 100.567636 , 100.586695 ], // อโศก-คลองตัน
[ 13.741226 , 13.747226 , 100.586705 , 100.592705 ], // คลองตัน
[ 13.739971 , 13.747226 , 100.592715 , 100.597254 ], // คลองตัน-สุขุมวิท 71
[ 13.739971 , 13.745971 , 100.597264 , 100.603264 ], // สุขุมวิท 71
[ 13.733767 , 13.745971 , 100.603274 , 100.635342 ], // สุขุมวิท 71-หัวหมาก
[ 13.733767 , 13.742767 , 100.635352 , 100.644352 ], // หัวหมาก
[ 13.728604 , 13.742767 , 100.644362 , 100.684222 ], // หัวหมาก-บ้านทับช้าง
[ 13.728604 , 13.737604 , 100.684232 , 100.693232 ], // บ้านทับช้าง
[ 13.725295 , 13.737604 , 100.693242 , 100.713642 ], // บ้านทับช้าง-ซอยวัดลานบุญ
[ 13.725295 , 13.734295 , 100.713652 , 100.722652 ], // ซอยวัดลานบุญ
[ 13.723117 , 13.734295 , 100.722662 , 100.744137 ], // ซอยวัดลานบุญ-ลาดกระบัง
[ 13.723117 , 13.732117 , 100.744147 , 100.753147 ], // ลาดกระบัง
[ 13.72524 , 13.732117 , 100.753157 , 100.77239 ], // ลาดกระบัง-พระจอมเกล้า
[ 13.72524 , 13.73124 , 100.7724 , 100.7784 ], // พระจอมเกล้า
[ 13.72524 , 13.731468 , 100.77841 , 100.779861 ], // พระจอมเกล้า-หัวตะเข้
[ 13.725468 , 13.731468 , 100.779871 , 100.785871 ], // หัวตะเข้
[ 13.715631 , 13.731468 , 100.785881 , 100.858345 ], // หัวตะเข้-คลองหลวงแพ่ง
[ 13.715631 , 13.724631 , 100.858355 , 100.867355 ], // คลองหลวงแพ่ง
[ 13.712106 , 13.724631 , 100.867365 , 100.89476 ], // คลองหลวงแพ่ง-คลองอุดมชลจร
[ 13.712106 , 13.721106 , 100.89477 , 100.90377 ], // คลองอุดมชลจร
[ 13.706353 , 13.721106 , 100.90378 , 100.923881 ], // คลองอุดมชลจร-เปรง
[ 13.706353 , 13.715353 , 100.923891 , 100.932891 ], // เปรง
[ 13.702468 , 13.715353 , 100.932901 , 100.958396 ], // เปรง-คลองแขวงกลั่น
[ 13.702468 , 13.711468 , 100.958406 , 100.967406 ], // คลองแขวงกลั่น
[ 13.699568 , 13.711468 , 100.967416 , 100.986945 ], // คลองแขวงกลั่น-คลองบางพระ
[ 13.699568 , 13.708568 , 100.986955 , 100.995955 ], // คลองบางพระ
[ 13.695743 , 13.708568 , 100.995965 , 101.013495 ], // คลองบางพระ-บางเตย
[ 13.695743 , 13.704743 , 101.013505 , 101.022505 ], // บางเตย
[ 13.686145 , 13.704743 , 101.022515 , 101.050713 ], // บางเตย-ชุมทางฉะเชิงเทรา
```

```
[      13.686145      ,      13.704145      ,      101.050723      ,      101.070902
],     //
ชุมทางฉะเชิงเทรา
[      13.704155      ,      13.799977      ,      101.050723      ,      101.103407
],     //
ชุมทางฉะเชิงเทรา-โพรงอากาศ
[      13.799987      ,      13.817987      ,      101.085407      ,      101.103407
],     //
โพรงอากาศ
[      13.817997      ,      13.841611      ,      101.085407      ,      101.110765
],     //
โพรงอากาศ-บางน้ำเปรี้ยว
[      13.841621      ,      13.859621      ,      101.092765      ,      101.110765
],     //
บางน้ำเปรี้ยว
[      13.859631      ,      13.898502      ,      101.092765      ,      101.123012
],     //
บางน้ำเปรี้ยว-ชุมทางคลองสิบเก้า
[      13.898512      ,      13.916512      ,      101.105012      ,      101.123012
],     //
ชุมทางคลองสิบเก้า
[      13.916522      ,      13.934507      ,      101.105012      ,      101.136279
],     //
ชุมทางคลองสิบเก้า-คลองยี่สิบเอ็ด
[      13.934517      ,      13.943517      ,      101.118279      ,      101.136279
],     //
คลองยี่สิบเอ็ด
[      13.943527      ,      13.961833      ,      101.118279      ,      101.158473
],     //
คลองยี่สิบเอ็ด-โยทะกา
[      13.961843      ,      13.979843      ,      101.140473      ,      101.158473
],     //
โยทะกา
[      13.979853      ,      14.004881      ,      101.140473      ,      101.210905
],     //
โยทะกา-บ้านสร้าง
[      14.004891      ,      14.022891      ,      101.192905      ,      101.210905
],     //
บ้านสร้าง
[      14.004891      ,      14.053974      ,      101.210915      ,      101.263781
],     //
บ้านสร้าง-หนองน้ำขาว
[      14.044974      ,      14.053974      ,      101.263791      ,      101.272791
],     //
หนองน้ำขาว
[      14.044974      ,      14.08335       ,      101.272801      ,      101.306228
],     //
หนองน้ำขาว-บ้านปากพลี
[      14.06535       ,      14.08335       ,      101.306238      ,      101.324238
],     //
บ้านปากพลี
[      14.061812      ,      14.08335       ,      101.324248      ,      101.364586
],     //
บ้านปากพลี-ปราจีนบุรี
[      14.061812      ,      14.079812      ,      101.364596      ,      101.382596
],     //
ปราจีนบุรี
[      14.055777      ,      14.079812      ,      101.382606      ,      101.44845
],     //
ปราจีนบุรี-โคกมะกอก
[      14.055777      ,      14.073777      ,      101.44855       ,      101.46655
],     //
โคกมะกอก
[      14.054668      ,      14.073777      ,      101.46665       ,      101.510048
],     //
โคกมะกอก-ประจันตคาม
[      14.054668      ,      14.072668      ,      101.510058      ,      101.528058
],     //
ประจันตคาม
[      14.048027      ,      14.072668      ,      101.528068      ,      101.565207
],     //
ประจันตคาม-หนองแสง
[      14.048027      ,      14.057027      ,      101.565217      ,      101.574217
],     //
หนองแสง
[      14.031123      ,      14.057027      ,      101.574227      ,      101.589905
],     //
หนองแสง-บ้านดงบัง
[      14.031123      ,      14.049123      ,      101.589915      ,      101.607915
],     //
บ้านดงบัง
[      14.013339      ,      14.049123      ,      101.607925      ,      101.63332
],     //
บ้านดงบัง-บ้านพรมแสง
[      14.013339      ,      14.031339      ,      101.63338       ,      101.65138
],     //
บ้านพรมแสง
[      14.002926      ,      14.031339      ,      101.65139       ,      101.674315
],     //
บ้านพรมแสง-บ้านเกาะแดง
[      14.002926      ,      14.011926      ,      101.674325      ,      101.683325
],     //
บ้านเกาะแดง
```

```
[       13.98082        ,       14.011926       ,       101.683335      ,       101.713852
        ],      //      บ้านเกาะแดง-กบินทร์บุรี
[       13.98082        ,       13.99882        ,       101.713862      ,       101.731862
        ],      //      กบินทร์บุรี
];

$(document).ready(function(){
        //on load finish
        console.log('start');
        var t = setInterval(function(){
                refresh();
        }, 3000);

});

function refresh(){
        $.getJSON('http://train.telecom.kmitl.ac.th/monitor/json.php',
function(json){
                //เมื่อได้ข้อมูล
                var hours = new Date().getHours();
                console.log('length = '+json.marker.length);
                for(var i=0; i< json.marker.length ;i++ ){
                        var serve = parseInt( json.marker[i].attr.s );
                        console.log(json.marker[i].attr.s);
                        var name = parseInt( json.marker[i].attr.name );

                        console.log("t"+name);

                        if(name==train){

                        if(serve == 1 ){

                                var update = json.marker[i].attr.update;
                                console.log(update);
                                document.getElementById("update").innerHTML
=update;


                                var canx  = json.marker[i].attr.lat;
                                var cany = json.marker[i].attr.lng;

                                var speed = json.marker[i].attr.spd;
                                var textspeed = speed+"  km/H";
                                console.log(textspeed);
                                document.getElementById("myspeed").innerHTML
=textspeed;


                                var current

                                for (var m=0;m<station.length && m%1==0;m++)
                                {
                                        console.log(m);
                                        if(canx  >=  station[m][0] && canx<=
      station[m][1] && cany>= station[m][2]    && cany<= station[m][3]    )

                                        {
                                        // console.log(station[m]);
                                        console.log(m);
                                        current = m+1;
                                        break;
                                        }

                                }
                                console.log(current);
                                console.log(speed);
```

```
                        if (hours >= 0 && hours < 12) {
        document.getElementById("position").src="img2559/281/"+current+".gif";
                        }
                        else{
        document.getElementById("position").src="img2559/281/a"+current+".gif"
;
                        }
                    }
                    else{
        document.getElementById("position").src="img2559/null_screen.png";
                        var noupdate="งดให้บริการระบบติดตามรถไฟ สอบถามเพิ่มเติม โทร. 1690";
                        document.getElementById("update").innerHTML
=noupdate;

                            // var noupdate="งดให้บริการระบบติดตามรถไฟ สอบถามเพิ่มเติม โทร.
1690";
                               // document.getElementById("demo").innerHTML
=noupdate;

                        var noupdateEN="No tracking system service. Please
contact 1690.";
                        document.getElementById("myspeed").innerHTML
=noupdateEN;
                        alert("งดให้บริการระบบติดตามรถไฟ สอบถามเพิ่มเติม โทร. 1690\nNo
tracking system service. Please contact 1690.");
                        }
                    }
                }


        });
}
```

คำสั่งการออกแบบแอปพลิเคชันบนสมาร์ตโฟนในระบบปฏิบัติการไอโอเอส (iOS)

1. คำสั่งการทำงานแอปพลิเคชันในหน้าแสดงตำแหน่งรถไฟ

```
import GoogleMobileAds
import UIKit
class LocatedTrain275: UIViewController {

    @IBOutlet weak var HeadLB: UILabel!
    @IBOutlet weak var NumberOfTrain: UILabel!
    @IBOutlet weak var TrainNameTH: UILabel!
    @IBOutlet weak var TrainNameEN: UILabel!
    @IBOutlet weak var CurrentPositionLB: UILabel!
    @IBOutlet weak var NextStationLB: UILabel!
    @IBOutlet weak var NowStationTH: UILabel!
    @IBOutlet weak var NowStationEN: UILabel!
    @IBOutlet weak var NextStationTH: UILabel!
    @IBOutlet weak var NextStationEN: UILabel!
    @IBOutlet weak var updateLB: UILabel!
    @IBOutlet weak var GoogleBannerView: GADBannerView!
    @IBOutlet weak var AlertBT: UIButton!
```

```
    override func viewDidLoad() {
        super.viewDidLoad()

        GoogleBannerView.adUnitID = "ca-app-pub-6212478803628615/7163639657"
        GoogleBannerView.rootViewController = self
        GoogleBannerView.load(GADRequest())

        // Do any additional setup after loading the view.

          _ = Timer.scheduledTimer(timeInterval: 20, target: self, selector:
#selector(self.viewDidAppear(_:)), userInfo: nil, repeats: true);
    }

    override func viewDidAppear(_ animated: Bool) {
        if Reachability.isConnectedToNetwork() == true
        {

            let url = URL(string:
"http://train.telecom.kmitl.ac.th/downloadlatlng275edit.php")
            let task = URLSession.shared.dataTask(with: url!) { (data,
response, error) in
                if error != nil
                {
                    print ("ERROR")
                }
                else
                {
                    if let content = data
                    {
                        do
                        {
                            var direction = 0
                            var stat :Array<Int> = []
                            var d = ""
                            var sv = ""
                            var station = ""

                            if let myJsonArray = try
JSONSerialization.jsonObject(with: content, options: .allowFragments) as?
Array<AnyObject>
                            {
                                for myJson in myJsonArray
                                {
                                    if let day = myJson["day"] as? String
                                    {
                                        d = day
                                    }
                                    else
                                    {
                                        d = ""
                                    }
                                    if let s = myJson["s"] as? String
                                    {
                                        sv = s
                                    }
                                    else
                                    {
                                        sv = ""
                                    }
                                    if let sta = myJson["sta"] as? String
                                    {
                                        station = sta
                                    }
```

```
                    else
                    {
                        station = ""
                    }

                    if let st = Int(station) {
                        stat.append(st)
                    }

                } //end get data from json

            }//end JSON data check

            let service = Int(sv)
            let date = Date()
            let calender = Calendar.current
            let hour = calender.component(.hour, from: date)

            let stationName:[String] = ["กรุงเทพ", "พญาไท", "มักกะสัน",
"อโศก", "คลองตัน", "สุขุมวิท71", "หัวหมาก", "บ้านทับช้าง","ซอยวัดลานบุญ", "ลาดกระบัง", "พระจอมเกล้า", "หัวตะเข้","คลอง
หลวงแพ่ง", "คลองอุดมชลจร","เปรง", "คลองแขวงกลั่น", "คลองบางพระ", "บางเตย", "ชุมทางฉะเชิงเทรา", "โพรงอากาศ","บางน้ำเปรี้ยว",
"ชุมทางคลองสิบเก้า", "คลองยี่สิบเอ็ด", "ไชยะกา", "บ้านสร้าง", "หนองน้ำขาว", "บ้านปากพลี","ปราจีนบุรี", "โคกมะกอก", "ประจันต
คาม", "หนองแสง", "บ้านดงบัง", "บ้านพรมแสง", "บ้านเกาะแดง","กบินทร์บุรี", "กบินทร์เก่า", "หนองสัง", "พระปรง", "บ้านแก้ง",
"ศาลาลำดวน", "สระแก้ว", "ท่าเกษม", "ห้วยโจด","วัฒนานคร", "บ้านโป่งคอม","ห้วยเดื่อ", "อรัญประเทศ"]

            let stationNameENG:[String] = ["BANGKOK",
"Phayathai", "Makkasan", "Asoke", "Khlong Tan", "Sukhumvit71", "Hua Mak","Ban
Thap Chang","Soi Wat Lan Boon", "Lat Krabang", "Pra Chom Klao", "Hua
Takhe","Khlong Luang Phaeng", "Khlong Udom Chonlajorn", "Preng","Khlong
Kwaeng Klan",                                                    "Khlong Bang Phra",
"Bang Toey", "Chachoengsao Junction", "Phrong Akat", "Bang Nam Prieo","Khlong
Sip Kao Junction","Khlong Yi Sip Et", "Yothaka", "Ban Sang", "Nong Nam
Khao","Ban Pak Phli", "Prachin Buri", "Khok Makok", "Prachantakham", "Nong
Saeng", "Ban Dong Bang","Ban Phrom Saeng", "Ban Ko Daeng", "Kabin Buri",
"Kabin Kao", "Nong Sang", "Phra Prong","Ban Kaeng", "Sala Lamduan", "Sa
Kaeo", "Tha Kasem", "Huai Chot","Watthana Nakhon", "Ban Pong Kom", "Huai
Dua", "Aranyaprathet"]

            //trainnumber
            var NumberTrain = " "
            var TrainNameTH = " "
            var TrainNameEN = " "

            if (stat[1] != 0 || (stat[1]-stat[0] != 0)){
                if ((hour >= 5 && hour <= 10) || (stat[1] >
stat[0])) {
                    direction = 1
                }
                else{
                    direction = -1
                }
            }

            if direction == 1 {
                NumberTrain = "ขบวนที่ 275"
                TrainNameTH = "หัวลำโพงไปอรัญประเทศ"
                TrainNameEN = "Hualampong to Aranyaprathet"
            }

            else if direction == -1 {
                NumberTrain = "ขบวนที่ 276"
                TrainNameTH = "อรัญประเทศไปหัวลำโพง"
                TrainNameEN = "Aranyaprathet to Hualampong"
```

```
                }

                else {
                    NumberTrain = " "
                    TrainNameTH = " "
                    TrainNameEN = " "

                }

                DispatchQueue.main.async(execute: {
                    self.NumberOfTrain.text =  "\(NumberTrain)"
                    self.TrainNameTH.text = "\(TrainNameTH)"
                    self.TrainNameEN.text = "\(TrainNameEN)"
                    self.updateLB.text = "updated : \(d)"
                });


                var NowStTH = " "
                var NowStEN = " "
                var NextStTH = " "
                var NextStEN = " "
                let CurrentStat = Int(station)
                let CurrentStation = CurrentStat! - 1

                if (service==1){
                    var NextStation = 0

                    if (CurrentStation<1000 && CurrentStation > -
1){
                        //บอกสถานีปัจจุบัน
                        if (CurrentStation % 2 == 0){
                            NowStTH = stationName[CurrentStation
/ 2]
                            NowStEN =
stationNameENG[CurrentStation / 2]

                            NextStation = (CurrentStation / 2) +
direction; //บอกสถานีถัดไป
                            if (NextStation < 0 || NextStation >
(stationName.count - 1)){
                                NextStTH = "-"
                                NextStEN = "-"
                            }
                            else {
                                NextStTH =
stationName[NextStation]
                                NextStEN =
stationNameENG[NextStation]
                            }
                        }
                        else{
                            NextStation = (CurrentStation +
direction) / 2
                            NowStTH = "ระหว่างทางไป
\(stationName[NextStation])"
                            NowStEN = "On the way to
\(stationNameENG[NextStation])"
                            NextStTH = stationName[NextStation]
                            NextStEN =
stationNameENG[NextStation]
                        }
                    }
                    else {
                        NowStTH = " "
```

```
                                    NowStEN = " "
                                    NextStTH = " "
                                    NextStEN = " "
                                }


                            }
                            else{
                                NowStTH = "งดให้บริการระบบติดตามรถไฟ"
                                NowStEN = " "
                                NextStTH = "No tracking system service"
                                NextStEN = " "
                            }


                            DispatchQueue.main.async(execute: {
                                self.NowStationTH.text = "\(NowStTH)"
                                self.NowStationEN.text = "\(NowStEN)"
                                self.NextStationTH.text = "\(NextStTH)"
                                self.NextStationEN.text = "\(NextStEN)"
                            });

                        } // end do loop
                        catch
                        {
                            print(error)
                        }
                    }
                }

            }
            task.resume()

        } //end if reachability is true

        else
        {
            let AlertController = UIAlertController(title: "No Internet
Connection",
                                                    message: "Make sure your
device is connection to internet",
                                                    preferredStyle: .alert)
            let settingsAction = UIAlertAction(title: "Settings", style:
.default) { (_) -> Void in
                guard let settingsUrl = URL(string:
UIApplicationOpenSettingsURLString) else{
                    return
                }

                if UIApplication.shared.canOpenURL(settingsUrl)
                {
                    UIApplication.shared.open(settingsUrl)
                }
            }

            let cancel = UIAlertAction(title: "Cancel", style: .cancel,
handler: nil)

            AlertController.addAction(settingsAction)
            AlertController.addAction(cancel)

            present(AlertController, animated: true, completion: nil)

        }
    }
```

```
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    func reNew(){
        //reload application data (renew root view )
        UIApplication.shared.keyWindow?.rootViewController =
storyboard!.instantiateViewController(withIdentifier: "LocatedTrain275")
    }

}
```

2. คำสั่งการทำงานแอปพลิเคชันในหน้าแสดงสถานีรถไฟ

```
import GoogleMobileAds
import UIKit

class StationViewController: UIViewController, UITableViewDelegate,
UITableViewDataSource, UISearchBarDelegate {


    @IBOutlet weak var NearsetStationIcon: UIImageView!
    @IBOutlet weak var NearestStationButton: UIButton!
    @IBOutlet weak var OtherStationIcon: UIImageView!
    @IBOutlet weak var ListOfStationLabel: UILabel!
    @IBOutlet var OtherStationTable: UITableView!
    @IBOutlet weak var GoogleBannerView: GADBannerView!
    @IBOutlet weak var searchBar: UISearchBar!


    var searchActive : Bool = false
    var stationArray = ["กรุงเทพ:BANGKOK","พญาไท:Phayathai","มักกะสัน:Makkasan","อโศก:
Asoke","คลองตัน:Khlong Tan","สุขุมวิท71 : Sukhumvit 71","หัวหมาก:Hua Mak","บ้านทับช้าง:Ban
Thap Chang","ซอยวัดลานบุญ:Soi Wat Lan Boon","ลาดกระบัง:Lat Krabang","พระจอมเกล้า:Pra Chom
Klao","หัวตะเข้:Hua Takhe","คลองหลวงแพ่ง:Khlong Luang Phaeng","คลองอุดมชลจร:Khlong Udom
Chonlajorn","เปรง:Preng","คลองแขวงกลั่น:Khlong Kwaeng Klan","คลองบางพระ:Khlong Bang
Phra","บางเตย:Bang Toey","ชุมทางฉะเชิงเทรา:Chachoengsao Junction","โพรงอากาศ:Phrong Akat","
บางน้ำเปรี้ยว:Bang Nam Prieo","ชุมทางคลองสิบเก้า:Khlong Sip Kao Junction","คลองยี่สิบเอ็ด:Khlong Yi
Sip Et","โยทะกา:Yothaka","บ้านสร้าง:Ban Sang","หนองน้ำขาว:Nong Nam Khao","บ้านปากพลี:Ban Pak
Phli","ปราจีนบุรี:Prachin Buri","โคกมะกอก:Khok Makok","ประจันตคาม:Prachantakham","หนองแสง:
Nong Saeng","บ้านดงบัง:Ban Dong Bang","บ้านพรมแสง:Ban Phrom Saeng","บ้านเกาะแดง:Ban Ko
Daeng","กบินทร์บุรี:Kabin Buri","กบินทร์เก่า:Kabin Kao","หนองสัง:Nong Sang","พระปรง:Phra
Prong","บ้านแก้ง:Ban Kaeng","ศาลาลำดวน:Sala Lamduan","สระแก้ว:Sa Kaeo","ท่าเกษม:Tha Kasem","
ห้วยโจด:Huai Chot","วัฒนานคร:Watthana Nakhon","บ้านโป่งคอม:Ban Pong Kom","ห้วยเดื่อ:Huai Dua","
อรัญประเทศ:Aranyaprathet","แปดริ้ว:Paet Riu","ดอนสีนนท์:Don Si Non","พานทอง:Phan Thong","ชลบุรี
:Chon Buri","บางพระ:Bang Phra","เขาพระบาท:Khao Phrabat","ชุมทางศรีราชา:Si Racha
Junction","บางละมุง:Bang Lamung","พัทยา:Pattaya","พัทยาใต้:Pattaya Tai","ตลาดน้ำสี่ภาค:Talad
Nam Si Phak","บ้านห้วยขวาง:Ban Huai Kwang","ญาณสังวรารามฯ:Yanasangwararam","สวนนงนุช:Suan
Nong Nuch","ชุมทางเขาชีจรรย์:Kaochichan Junction","บ้านพลูตาหลวง:Ban Plu Ta Luang"]
    var filtered:[String] = []

    let searchController = UISearchController(searchResultsController: nil)
    var filteredStationArray: [String]?

    override func viewDidLoad() {
        super.viewDidLoad()
        searchBar.delegate = self
```

```
        GoogleBannerView.adUnitID = "ca-app-pub-6212478803628615/7163639657"
        GoogleBannerView.rootViewController = self
        GoogleBannerView.load(GADRequest())

        _ = Timer.scheduledTimer(timeInterval: 20, target: self, selector:
#selector(self.viewDidAppear(_:)), userInfo: nil, repeats: true);

        self.OtherStationTable.dataSource = self
        self.OtherStationTable.delegate = self
        self.OtherStationTable.register(UITableViewCell.self,
forCellReuseIdentifier: "cell")
        self.OtherStationTable.allowsSelection = true


        NearestStationButton.titleLabel?.textAlignment =
NSTextAlignment.center
        ListOfStationLabel.numberOfLines = 2
        ListOfStationLabel.textAlignment = NSTextAlignment.center

    }

    func searchBarTextDidBeginEditing(_ searchBar: UISearchBar) {
        searchActive = true;
    }

    func searchBarTextDidEndEditing(_ searchBar: UISearchBar) {
        searchActive = false;
    }

    func searchBarCancelButtonClicked(_ searchBar: UISearchBar) {
        searchActive = false;
    }

    func searchBarSearchButtonClicked(_ searchBar: UISearchBar) {

        searchActive = false;
        self.searchBar.endEditing(true)

    }

    func searchBar(_ searchBar: UISearchBar, textDidChange searchText:
String) {

        filtered = stationArray.filter({ (text) -> Bool in
            let tmp: NSString = text as NSString
            let range = tmp.range(of: searchText, options:
NSString.CompareOptions.caseInsensitive)
            return range.location != NSNotFound
        })
        if(filtered.count == 0){
            searchActive = false;
        } else {
            searchActive = true;
        }
        self.OtherStationTable.reloadData()
    }


    //Table
    func tableView(_ tableView: UITableView, heightForRowAt indexPath:
IndexPath) -> CGFloat {
        return 40
    }
```

```
    func tableView(_ tableView: UITableView, numberOfRowsInSection section:
Int) -> Int {
        if(searchActive) {
            return filtered.count
        }
        return stationArray.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath:
IndexPath) -> UITableViewCell {

        let cell:UITableViewCell =
self.OtherStationTable.dequeueReusableCell(withIdentifier:
"cellReuseIdentifier") as UITableViewCell!

            if(searchActive){
                if filtered.count > indexPath.row {
                    cell.textLabel?.text = filtered[indexPath.row]
                }
            } else {
                cell.textLabel?.text = stationArray[indexPath.row];
            }
        cell.textLabel?.font = UIFont.systemFont(ofSize: 14.8)
        return cell
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath:
IndexPath) {

        let currentCell = tableView.cellForRow(at: indexPath)! as
UITableViewCell
        let currentItem = currentCell.textLabel!.text

        var selectedLabel:String?
        for i in (0..<stationArray.count) {
            if  currentItem! == stationArray[i] {
                selectedLabel = String(i)
            }
        }
        performSegue(withIdentifier: "GotoStation", sender: selectedLabel)
    }

    func updateSearchResults(for searchController: UISearchController) {
        if let searchText = searchController.searchBar.text,
!searchText.isEmpty {
            filteredStationArray = stationArray.filter { team in
                return team.lowercased().contains(searchText.lowercased())
            }
        } else {
            filteredStationArray = stationArray
        }
        OtherStationTable.reloadData()
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if let vc = segue.destination as? TrainToStation {
            if let strTemp = sender as? String
            {
                vc.mytext = strTemp
            }
        }
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
```

```
        // Dispose of any resources that can be recreated.
    }

}
```

3. คำสั่งการทำงานแอปพลิเคชันในหน้าแสดงสถานีที่ใกล้ตัวผู้ใช้งานที่สุด

```
import GoogleMobileAds
import UIKit
import CoreLocation

class NearestStation: UIViewController, CLLocationManagerDelegate ,
UITableViewDelegate, UITableViewDataSource {

    @IBOutlet weak var NearlessLabel: UILabel!
    @IBOutlet weak var GoogleBannerView: GADBannerView!
    @IBOutlet weak var NearlessStationTableView: UITableView!

    var locManager = CLLocationManager()
    var currentLocation: CLLocation!
    var UserLat: Double = 0.00
    var UserLong: Double = 0.00
    var NearlessStationArray:Array<String> = []
    var StationLocation = [
        [    13.739488    ,    100.516747    ]    ,    //    กรุงเทพ
        [    13.757034    ,    100.533081    ]    ,    //    พญาไท
        [    13.753711    ,    100.547391    ]    ,    //    มักกะสัน
        [    13.750356    ,    100.563126    ]    ,    //    อโศก
        [    13.744226    ,    100.589705    ]    ,    //    คลองตัน
        [    13.742971    ,    100.600264    ]    ,    //    สุขุมวิท 71
        [    13.738267    ,    100.639852    ]    ,    //    หัวหมาก
        [    13.733104    ,    100.688732    ]    ,    //    บ้านทับช้าง
        [    13.729795    ,    100.718152    ]    ,    //    ซอยวัดลานบุญ
        [    13.727617    ,    100.748647    ]    ,    //    ลาดกระบัง
        [    13.72824    ,    100.7754    ]    ,    //    พระจอมเกล้า
        [    13.728468    ,    100.782871    ]    ,    //    หัวตะเข้
        [    13.720131    ,    100.862855    ]    ,    //    คลองหลวงแพ่ง
        [    13.716606    ,    100.89927    ]    ,    //    คลองอุดมชลจร
        [    13.710853    ,    100.928391    ]    ,    //    เปรง
        [    13.706968    ,    100.962906    ]    ,    //    คลองแขวงกลั่น
        [    13.704068    ,    100.991455    ]    ,    //    คลองบางพระ
        [    13.700243    ,    101.018005    ]    ,    //    บางเตย
        [    13.695145    ,    101.059723    ]    ,    //    ชุมทางฉะเชิงเทรา
        [    13.808987    ,    101.094407    ]    ,    //    โพรงอากาศ
        [    13.850621    ,    101.101765    ]    ,    //    บางน้ำเปรี้ยว
        [    13.907512    ,    101.114012    ]    ,    //    ชุมทางคลองสิบเก้า
        [    13.939017    ,    101.127279    ]    ,    //    คลองขี่สิบเอ็ด
        [    13.970843    ,    101.149473    ]    ,    //    โยทะกา
        [    14.013891    ,    101.201905    ]    ,    //    บ้านสร้าง
        [    14.049474    ,    101.268291    ]    ,    //    หนองน้ำขาว
        [    14.07435    ,    101.315238    ]    ,    //    บ้านปากพลี
        [    14.070812    ,    101.373596    ]    ,    //    ปรจีนบุรี
        [    14.064777    ,    101.45755    ]    ,    //    โคกมะกอก
        [    14.063668    ,    101.519058    ]    ,    //    ประจันตคาม
        [    14.052527    ,    101.569717    ]    ,    //    หนองแสง
        [    14.040123    ,    101.598915    ]    ,    //    บ้านคงบัง
```

```
[    14.022339    ,    101.64238    ]    ,    //    บ้านพรมแสง
[    14.007426    ,    101.678825    ]    ,    //    บ้านเกาะแดง
[    13.98982    ,    101.722862    ]    ,    //    กบินทร์บุรี
[    13.996773    ,    101.760457    ]    ,    //    กบินทร์เก่า
[    13.980702    ,    101.82544    ]    ,    //    หนองสัง
[    13.952985    ,    101.92112    ]    ,    //    พระปรง
[    13.917303    ,    101.969036    ]    ,    //    บ้านแก้ง
[    13.882769    ,    102.009475    ]    ,    //    ศาลาลำดวน
[    13.826814    ,    102.075211    ]    ,    //    สระแก้ว
[    13.7902    ,    102.168762    ]    ,    //    ท่าเกษม
[    13.7706    ,    102.231258    ]    ,    //    ห้วยโจด
[    13.741161    ,    102.323792    ]    ,    //    วัฒนานคร
[    13.724712    ,    102.380528    ]    ,    //    บ้านโป่งคอม
[    13.712872    ,    102.422393    ]    ,    //    ห้วยเดื่อ
[    13.692745    ,    102.50496    ]    ,    //    อรัญประเทศ
[    13.692895    ,    101.076773    ]    ,    //    แปดริ้ว
[    13.602066    ,    101.110032    ]    ,    //    ดอนสีนนท์
[    13.462968    ,    101.081307    ]    ,    //    พานทอง
[    13.342336    ,    100.998188    ]    ,    //    ชลบุรี
[    13.235714    ,    100.953529    ]    ,    //    บางพระ
[    13.202484    ,    100.941771    ]    ,    //    เขาพระบาท
[    13.156006    ,    100.945726    ]    ,    //    ชุมทางศรีราชา
[    13.034484    ,    100.939955    ]    ,    //    บางละมุง
[    12.940268    ,    100.909307    ]    ,    //    พัทยา
[    12.908361    ,    100.901202    ]    ,    //    พัทยาใต้
[    12.869609    ,    100.90962    ]    ,    //    ตลาดน้ำสี่ภาค
[    12.824438    ,    100.921441    ]    ,    //    บ้านห้วยขวาง
[    12.801388    ,    100.922488    ]    ,    //    ญาณสังวรารามม
[    12.774687    ,    100.924301    ]    ,    //    สวนนงนุช
[    12.733719    ,    100.954497    ]    ,    //    ชุมทางเขาชีจรรย์
[    12.704553    ,    100.972444    ] ]    //    บ้านพลูตาหลวง
```

```
let stationName:[String] = ["กรุงเทพ:BANGKOK","พญาไท:Phayathai","มักกะสัน:Makkasan","อโศก:Asoke","คลองตัน:Khlong Tan","สุขุมวิท 71 : Sukhumvit 71","หัวหมาก:Hua Mak","บ้านทับช้าง:Ban Thap Chang","ซอยวัดลานบุญ:Soi Wat Lan Boon","ลาดกระบัง:Lat Krabang","พระจอมเกล้า:Pra Chom Klao","หัวตะเข้:Hua Takhe","คลองหลวงแพ่ง:Khlong Luang Phaeng","คลองอุดมชลจร:Khlong Udom Chonlajorn","เปรง:Preng","คลองแขวงกลั่น:Khlong Kwaeng Klan","คลองบางพระ:Khlong Bang Phra","บางเตย:Bang Toey","ชุมทางฉะเชิงเทรา:Chachoengsao Junction","โพรงอากาศ:Phrong Akat","บางน้ำเปรี้ยว:Bang Nam Prieo","ชุมทางคลองสิบเก้า:Khlong Sip Kao Junction","คลองยี่สิบเอ็ด:Khlong Yi Sip Et","โยทะกา:Yothaka","บ้านสร้าง:Ban Sang","หนองน้ำขาว:Nong Nam Khao","บ้านปากพลี:Ban Pak Phli","ปราจีนบุรี:Prachin Buri","โคกมะกอก:Khok Makok","ประจันตคาม:Prachantakham","หนองแสง:Nong Saeng","บ้านดงบัง:Ban Dong Bang","บ้านพรมแสง:Ban Phrom Saeng","บ้านเกาะแดง:Ban Ko Daeng","กบินทร์บุรี:Kabin Buri","กบินทร์เก่า:Kabin Kao","หนองสัง:Nong Sang","พระปรง:Phra Prong","บ้านแก้ง:Ban Kaeng","ศาลาลำดวน:Sala Lamduan","สระแก้ว:Sa Kaeo","ท่าเกษม:Tha Kasem","ห้วยโจด:Huai Chot","วัฒนานคร:Watthana Nakhon","บ้านโป่งคอม:Ban Pong Kom","ห้วยเดื่อ:Huai Dua","อรัญประเทศ:Aranyaprathet","แปดริ้ว:Paet Riu","ดอนสีนนท์:Don Si Non","พานทอง:Phan Thong","ชลบุรี:Chon Buri","บางพระ:Bang Phra","เขาพระบาท:Khao Phrabat","ชุมทางศรีราชา:Si Racha Junction","บางละมุง:Bang Lamung","พัทยา:Pattaya","พัทยาใต้:Pattaya Tai","ตลาดน้ำสี่ภาค:Talad Nam Si Phak","บ้านห้วยขวาง:Ban Huai Kwang","ญาณสังวรารามม:Yanasangwararam","สวนนงนุช:Suan Nong Nuch","ชุมทางเขาชีจรรย์:Kaochichan Junction","บ้านพลูตาหลวง:Ban Plu Ta Luang"]
```

```
override func viewDidLoad() {
    super.viewDidLoad()
    GoogleBannerView.adUnitID = "ca-app-pub-6212478803628615/7163639657"
```

```
        GoogleBannerView.rootViewController = self
        GoogleBannerView.load(GADRequest())

        _ = Timer.scheduledTimer(timeInterval: 20, target: self, selector:
#selector(self.viewDidAppear(_:)), userInfo: nil, repeats: true);

        self.NearlessStationTableView.dataSource = self
        self.NearlessStationTableView.delegate = self
        self.NearlessStationTableView.register(UITableViewCell.self,
forCellReuseIdentifier: "cell")
        self.NearlessStationTableView.allowsSelection = true

        //GPSGetLocation
        locManager.requestWhenInUseAuthorization()
        locManager.delegate = self
        locManager.desiredAccuracy = kCLLocationAccuracyBest

        if( CLLocationManager.authorizationStatus() ==
CLAuthorizationStatus.authorizedWhenInUse ||
            CLLocationManager.authorizationStatus() ==
CLAuthorizationStatus.authorizedAlways){

            currentLocation = locManager.location
            UserLat = currentLocation.coordinate.latitude
            UserLong = currentLocation.coordinate.longitude
            self.NearlessStationTableView.reloadData()
        }

        func DegreesToRadians(_ degrees: Double) -> Double {
            return degrees * Double.pi / 180
        }

        func computeDistance(_ row: Int,_ UserLat: Double,_ UserLong: Double)
-> Double {
            let radius: Double = 6371; // km
            let deltaP = DegreesToRadians(StationLocation[row][0]) -
DegreesToRadians(UserLat)
            let deltaL = DegreesToRadians(StationLocation[row][1]) -
DegreesToRadians(UserLong)
            let a = sin(deltaP/2) * sin(deltaP/2) +
                cos(DegreesToRadians(UserLat)) *
cos(DegreesToRadians(StationLocation[0][0])) * sin(deltaL/2) * sin(deltaL/2)
            let c = 2 * atan2(sqrt(a), sqrt(1-a))
            let d = radius * c

            return d
        }

        var dist :Array<Double> = []
        for i in (0..<StationLocation.count) {
            let distant = computeDistance(i,UserLat,UserLong)
            dist.append(distant)
        }

        let MinIndex = dist.index(of: dist.min()!)
        if MinIndex == 0 {
            NearlessStationArray = [stationName[MinIndex!],
stationName[MinIndex!+1]]
        }
        else if MinIndex == 46 {
            NearlessStationArray = [stationName[MinIndex!],
stationName[MinIndex!-1]]
        }
        else if MinIndex == 62 {
```

```
            NearlessStationArray = [stationName[MinIndex!],
stationName[MinIndex!-1]]
        }
        else {
            NearlessStationArray = [stationName[MinIndex!],
stationName[MinIndex!-1] , stationName[MinIndex!+1]]
        }

    }//end view did load

    //Table
    func tableView(_ tableView: UITableView, heightForRowAt indexPath:
IndexPath) -> CGFloat {
        return 40
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSection section:
Int) -> Int {
        return self.NearlessStationArray.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath:
IndexPath) -> UITableViewCell {

        let cell:UITableViewCell =
self.NearlessStationTableView.dequeueReusableCell(withIdentifier:
"NearlessIdentifier",

for:indexPath) as UITableViewCell!
        cell.textLabel?.text = self.NearlessStationArray[indexPath.row]
        cell.textLabel?.font = UIFont.systemFont(ofSize: 14.8)
        return cell
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath:
IndexPath) {

        let currentCell = tableView.cellForRow(at: indexPath)! as
UITableViewCell
        let currentItem = currentCell.textLabel!.text

        var selectedLabel:String?
        for i in (0..<stationName.count) {
            if  currentItem! == stationName[i] {
                selectedLabel = String(i)
            }
        }
        performSegue(withIdentifier: "GotoStation2", sender: selectedLabel)
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if let vc = segue.destination as? TrainToStation {
            if let strTemp = sender as? String
            {
                vc.mytext = strTemp
            }
        }
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

4. คำสั่งสำหรับหน้าแสดงขบวนรถไฟที่กำลังเข้าสถานีรถไฟ

```
import GoogleMobileAds
import UIKit

class TrainToStation: UIViewController {
    var mytext = ""

    @IBOutlet weak var GoogleBannerView: GADBannerView!
    @IBOutlet weak var HeadLabel: UILabel!
    @IBOutlet weak var HeadLabel2: UILabel!
    @IBOutlet weak var IconTrain275: UIImageView!
    @IBOutlet weak var IconTrain283: UIImageView!
    @IBOutlet weak var OBTrainLB_1: UILabel!
    @IBOutlet weak var IBTrainLB_1: UILabel!
    @IBOutlet weak var OBTimeLB_1: UILabel!
    @IBOutlet weak var IBTimeLB_1: UILabel!
    @IBOutlet weak var OBTrainLB_2: UILabel!
    @IBOutlet weak var IBTrainLB_2: UILabel!
    @IBOutlet weak var OBTimeLB_2: UILabel!
    @IBOutlet weak var IBTimeLB_2: UILabel!
    @IBOutlet weak var view275: UIView!
    @IBOutlet weak var view283: UIView!
    @IBOutlet weak var NumTrain275: UILabel!
    @IBOutlet weak var NameTrain275: UILabel!
    @IBOutlet weak var NumTrain283: UILabel!
    @IBOutlet weak var NameTrain283: UILabel!

    var StationArray = ["กรุงเทพ:BANGKOK","พญาไท:Phayathai","มักกะสัน:Makkasan","อโศก:
Asoke","คลองตัน:Khlong Tan","สุขุมวิท 71 : Sukhumvit 71","หัวหมาก:Hua Mak","บ้านทับช้าง:Ban
Thap Chang","ซอยวัดลานบุญ:Soi Wat Lan Boon","ลาดกระบัง:Lat Krabang","พระจอมเกล้า:Pra Chom
Klao","หัวตะเข้:Hua Takhe","คลองหลวงแพ่ง:Khlong Luang Phaeng","คลองอุดมชลจร:Khlong Udom
Chonlajorn","เปรง:Preng","คลองแขวงกลั่น:Khlong Kwaeng Klan","คลองบางพระ:Khlong Bang
Phra","บางเตย:Bang Toey","ชุมทางฉะเชิงเทรา:Chachoengsao Junction","โพรงอากาศ:Phrong Akat","
บางน้ำเปรี้ยว:Bang Nam Prieo","ชุมทางคลองสิบเก้า:Khlong Sip Kao Junction","คลองยี่สิบเอ็ด:Khlong Yi
Sip Et","โยทะกา:Yothaka","บ้านสร้าง:Ban Sang","หนองน้ำขาว:Nong Nam Khao","บ้านปากพลี:Ban Pak
Phli","ปราจีนบุรี:Prachin Buri","โคกมะกอก:Khok Makok","ประจันตคาม:Prachantakham","หนองแสง:
Nong Saeng","บ้านดงบัง:Ban Dong Bang","บ้านพรมแสง:Ban Phrom Saeng","บ้านเกาะแดง:Ban Ko
Daeng","กบินทร์บุรี:Kabin Buri","กบินทร์เก่า:Kabin Kao","หนองสัง:Nong Sang","พระปรง:Phra
Prong","บ้านแก้ง:Ban Kaeng","ศาลาลำดวน:Sala Lamduan","สระแก้ว:Sa Kaeo","ท่าเกษม:Tha Kasem","
ห้วยโจด:Huai Chot","วัฒนานคร:Watthana Nakhon","บ้านโป่งคอม:Ban Pong Kom","ห้วยเดื่อ:Huai Dua","
อรัญประเทศ:Aranyaprathet","แปดริ้ว:Paet Riu","ดอนสีนนท์:Don Si Non","พานทอง:Phan Thong","ชลบุรี
:Chon Buri","บางพระ:Bang Phra","เขาพระบาท:Khao Phrabat","ชุมทางศรีราชา:Si Racha
Junction","บางละมุง:Bang Lamung","พัทยา:Pattaya","พัทยาใต้:Pattaya Tai","ตลาดน้ำสี่ภาค:Talad
Nam Si Phak","บ้านห้วยขวาง:Ban Huai Kwang","ญาณสังวราราม:Yanasangwararam","สวนนงนุช:Suan
Nong Nuch","ชุมทางเขาชีจรรย์:Kaochichan Junction","บ้านพลูตาหลวง:Ban Plu Ta Luang"]


    var TimeTable275 =
["5:55","6:09","6:19","6:25","6:35","6:40","6:42","6:47","6:51","6:54","6:59"
,"7:02","7:10","7:14","7:18","7:23","7:27","7:30","7:36","7:52","8:00","8:08"
,"8:15","8:22","8:32","8:40","8:49","8:56","9:08","9:15","9:22","9:29","9:35"
,"9:40","9:47","9:52","9:58","10:09","10:17","10:24","10:33","10:44","10:54",
"11:05","11:15","11:24","11:35","","","","","","","","","","","","","","","",
""]
    var TimeTable276 =
["19:40","19:29","19:24","19:19","19:14","19:07","19:05","18:58","18:54","18:
50","18:44","18:41","18:32","18:26","18:21","18:15","18:10","18:05","17:58","
17:45","17:37","17:28","17:19","17:10","16:59","16:48","16:40","16:29","16:19
","16:10","16:02","15:56","15:50","15:45","15:38","15:32","15:24","15:13","15
```

```
:05","14:58","14:49","14:37","14:28","14:17","14:10","14:04","13:55","","",""
,"","","","","","","","","","","","",""]
    var TimeTable283 =
["6:55","7:07","7:15","7:21","7:31","7:42","7:49","7:57","8:01","8:05","8:09"
,"8:13","8:23","8:28","8:33","8:39","8:45","8:50","8:56","9:03","9:15","9:30"
,"9:48","9:59","10:06","10:12","10:24","10:34","10:39","10:44","10:50","10:56
","11:03","11:12","11:20"]
    var TimeTable284 =
["18:15","18:05","18:00","17:55","17:50","17:46","17:36","17:26","17:20","17:
14","17:08","17:05","16:54","16:48","16:43","16:37","16:32","16:26","16:12","
16:08","15:53","15:37","15:20","15:05","15:00","14:47","14:31","14:20","14:13
","14:06","14:01","13:56","13:51","13:43","13:35"]

    override func viewDidLoad() {
        super.viewDidLoad()

        GoogleBannerView.adUnitID = "ca-app-pub-6212478803628615/7163639657"
        GoogleBannerView.rootViewController = self
        GoogleBannerView.load(GADRequest())

        let SelectedStation = Int(mytext)
        HeadLabel.text = "ขบวนรถไฟในระบบติดตามรถไฟที่เข้าสถานี"
        HeadLabel2.text = "\(StationArray[SelectedStation!])"

        if Reachability.isConnectedToNetwork() == true
        {
            let SelectedStation = Int(mytext)
            let url = URL(string:
"http://train.telecom.kmitl.ac.th/downloadlatlng275_283.php")
            let task = URLSession.shared.dataTask(with: url!) { (data,
response, error) in
                if error != nil
                {
                    print ("ERROR")
                }
                else
                {
                    if let content = data
                    {
                        do
                        {
                            var direction275 = 0
                            var direction283 = 0
                            var stat:Array<Int> = []
                            var service:Array<Int> = []
                            var d = ""
                            var sv = ""
                            var station = ""

                            if let myJsonArray = try
JSONSerialization.jsonObject(with: content, options: .allowFragments) as?
Array<AnyObject>
                            {
                                for myJson in myJsonArray
                                {
                                    if let day = myJson["day"] as? String
                                    {
                                        d = day
                                    }
                                    else
                                    {
                                        d = ""
                                    }
                                    if let s = myJson["s"] as? String
                                    {
```

```
                    sv = s
                }
                else
                {
                    sv = ""
                }
                if let sta = myJson["sta"] as? String
                {
                    station = sta
                }
                else
                {
                    station = ""
                }

                if let st = Int(station) {
                    stat.append(st)
                }

                if let serv = Int(sv) {
                    service.append(serv)
                }

            } //end get data from json

        }//end JSON data check

        print ("\(stat)")

        let date = Date()
        let calender = Calendar.current
        let hour = calender.component(.hour, from: date)

        //directionTrain275
        if (stat[1] != 0 || (stat[1]-stat[0] != 0)){
            if ((hour >= 5 && hour <= 10) || (stat[1] >
stat[0])) {
                direction275 = 1
            }
            else{
                direction275 = -1
            }
        }

        //directionTrain283
        if (stat[3] != 0 || (stat[3]-stat[2] != 0)){
            if ((hour >= 5 && hour <= 10) || (stat[3] >
stat[2])) {
                direction283 = 1
            }
            else{
                direction283 = -1
            }
        }

        var OBTrainNumber = ["ขบวนที่275","ขบวนที่283"]
        var IBTrainNumber = ["ขบวนที่276","ขบวนที่284"]
        var Time275 = ""
        var Time276 = ""
        var Time283 = ""
        var Time284 = ""
        var CuStation283 = 0

        func CurrentSta283 (_ stat : Int) -> Int
```

```
                                {
                                    if SelectedStation! <= 18 {
                                        CuStation283 = (stat - 1)/2
                                    }
                                    else if (SelectedStation! >= 47 &&
SelectedStation! <= 62) {
                                        CuStation283 = (stat - 1)/2 + 28
                                    }
                                    return CuStation283
                                }

                                let CurrentStation275 = (stat[1] - 1)/2
                                let CurrentStation283 = CurrentSta283(stat[3])

                                func TimeOf283 (_ SelectedStat : Int) -> String
                                {
                                    if SelectedStation! <= 18 {
                                        Time283 =
self.TimeTable283[SelectedStation!]
                                    }
                                    else if (SelectedStation! >= 47 &&
SelectedStation! <= 62) {
                                        Time283 =
self.TimeTable283[SelectedStation!-28]
                                    }
                                    return Time283
                                }

                                func TimeOf284 (_ SelectedStat : Int) -> String
                                {
                                    if SelectedStation! <= 18 {
                                        Time284 =
self.TimeTable284[SelectedStation!]
                                    }
                                    else if (SelectedStation! >= 47 &&
SelectedStation! <= 62) {
                                        Time284 =
self.TimeTable284[SelectedStation!-28]
                                    }
                                    return Time284
                                }

                                if service[1]==1 {

                                    if (direction275 == 1 ){

                                        if CurrentStation275 < SelectedStation!{
                                            Time275 = "เข้าสถานีเวลา
\(self.TimeTable275[SelectedStation!])"
                                            Time276 = "เข้าสถานีเวลา
\(self.TimeTable276[SelectedStation!])"
                                        }
                                        else{
                                            Time275 = "ผ่านสถานีนี้ไปแล้ว"
                                            Time276 = "เข้าสถานีเวลา
\(self.TimeTable276[SelectedStation!])"
                                        }
                                    }

                                    if (direction275 == -1 ){
                                        if CurrentStation275 > SelectedStation! {
                                            Time275 = "ผ่านสถานีนี้ไปแล้ว"
                                            Time276 = "เข้าสถานีเวลา
\(self.TimeTable276[SelectedStation!])"
```

```
                    }
                    else{
                        Time275 = "ผ่านสถานีนี้ไปแล้ว"
                        Time276 = "ผ่านสถานีนี้ไปแล้ว"
                    }
                }
            }
            // ถ้าservice275 =0
            else{
                Time275 = "งดให้บริการระบบติดตามรถไฟ"
                Time276 = "งดให้บริการระบบติดตามรถไฟ"
            }

            if service[3]==1 {

                if (direction283 == 1){
                    if CurrentStation283 < SelectedStation!{
                        Time283 = "เข้าสถานีเวลา
\(TimeOf283(SelectedStation!))"
                        Time284 = "เข้าสถานีเวลา
\(TimeOf284(SelectedStation!))"
                    }
                    else{
                        Time283 = "ผ่านสถานีนี้ไปแล้ว"
                        Time284 = "เข้าสถานีเวลา
\(TimeOf284(SelectedStation!))"
                    }
                }

                if (direction283 == -1){
                    if CurrentStation283 > SelectedStation!{
                        Time283 = "ผ่านสถานีนี้ไปแล้ว"
                        Time284 = "เข้าสถานีเวลา
\(TimeOf284(SelectedStation!))"
                    }
                    else{
                        Time283 = "ผ่านสถานีนี้ไปแล้ว"
                        Time284 = "ผ่านสถานีนี้ไปแล้ว"
                    }
                }
            }
            //ถ้าservice283 = 0
            else{
                Time283 = "งดให้บริการระบบติดตามรถไฟ"
                Time284 = "งดให้บริการระบบติดตามรถไฟ"
            }

            var NumTrain = ["ขบวนที่ 275-276", "ขบวนที่ 283-284"]
            var NameTrain = ["กรุงเทพ-อรัญประเทศ", "กรุงเทพ-บ้านพลูตาหลวง"]

            if SelectedStation! <= 18 {
                DispatchQueue.main.async(execute: {
                    self.NumTrain275.text = "\(NumTrain[0])"
                    self.NameTrain275.text =
"\(NameTrain[0])"
                    self.OBTrainLB_1.text =
"\(OBTrainNumber[0])"
                    self.OBTimeLB_1.text = "\(Time275)"
                    self.IBTrainLB_1.text =
"\(IBTrainNumber[0])"
                    self.IBTimeLB_1.text = "\(Time276)"
```

```
                                    self.NumTrain283.text = "\(NumTrain[1])"
                                    self.NameTrain283.text =
"\(NameTrain[1])"
                                    self.OBTrainLB_2.text =
"\(OBTrainNumber[1])"
                                    self.OBTimeLB_2.text = "\(Time283)"
                                    self.IBTrainLB_2.text =
"\(IBTrainNumber[1])"
                                    self.IBTimeLB_2.text = "\(Time284)"
                                })
                            }
                            else if (SelectedStation! >= 18 &&
SelectedStation! <= 46) {
                                DispatchQueue.main.async(execute: {
                                    self.view283.isHidden = true
                                    self.IconTrain283.isHidden = true
                                    self.OBTrainLB_2.isHidden = true
                                    self.IBTrainLB_2.isHidden = true
                                    self.OBTimeLB_2.isHidden = true
                                    self.IBTimeLB_2.isHidden = true
                                    self.NumTrain283.isHidden = true
                                    self.NameTrain283.isHidden = true

                                    self.NumTrain275.text = "\(NumTrain[0])"
                                    self.NameTrain275.text =
"\(NameTrain[0])"
                                    self.OBTrainLB_1.text =
"\(OBTrainNumber[0])"
                                    self.OBTimeLB_1.text = "\(Time275)"
                                    self.IBTrainLB_1.text =
"\(IBTrainNumber[0])"
                                    self.IBTimeLB_1.text = "\(Time276)"
                                })
                            }
                            else if (SelectedStation! >= 47 &&
SelectedStation! <= 62) {
                                DispatchQueue.main.async(execute: {
                                    self.view283.isHidden = true
                                    self.IconTrain283.isHidden = true
                                    self.OBTrainLB_2.isHidden = true
                                    self.IBTrainLB_2.isHidden = true
                                    self.OBTimeLB_2.isHidden = true
                                    self.IBTimeLB_2.isHidden = true
                                    self.NumTrain283.isHidden = true
                                    self.NameTrain283.isHidden = true

                                    self.NumTrain275.text = "\(NumTrain[1])"
                                    self.NameTrain275.text =
"\(NameTrain[1])"
                                    self.OBTrainLB_1.text =
"\(OBTrainNumber[1])"
                                    self.OBTimeLB_1.text = "\(Time283)"
                                    self.IBTrainLB_1.text =
"\(IBTrainNumber[1])"
                                    self.IBTimeLB_1.text = "\(Time284)"
                                })
                            }

                        } // end do loop
                        catch
                        {
                            print(error)
                        }
                    }
```

```
                }

            }
            task.resume()

        } //end if reachability is true


        else
        {
            let AlertController = UIAlertController(title: "No Internet
Connection",
                                                    message: "Make sure your
device is connection to internet",
                                                    preferredStyle: .alert)
            let settingsAction = UIAlertAction(title: "Settings", style:
.default) { (_) -> Void in
                guard let settingsUrl = URL(string:
UIApplicationOpenSettingsURLString) else{
                    return
                }

                if UIApplication.shared.canOpenURL(settingsUrl)
                {
                    UIApplication.shared.open(settingsUrl)
                }
            }

            let cancel = UIAlertAction(title: "Cancel", style: .cancel,
handler: nil)

            AlertController.addAction(settingsAction)
            AlertController.addAction(cancel)

            present(AlertController, animated: true, completion: nil)

        }
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    func reNew(){
        //reload application data (renew root view )
        UIApplication.shared.keyWindow?.rootViewController =
storyboard!.instantiateViewController(withIdentifier: "TrainToSatation")
    }
}
```

คำสั่งสำหรับระบบประมาณเวลาถึงสถานีของรถไฟ

```
def refresh():
    t.sleep(20)
import time as t
import pymysql
import numpy as np
import csv
from sklearn.metrics import accuracy_score
import math
import datetime
```

```
station = [
[13.734988, 13.743988, 100.512247, 100.521247],        #     กรุงเทพ
[13.752534, 13.761534, 100.528581, 100.537581],        #     พญาไท
[13.749211, 13.758211, 100.542891, 100.551891],        #     มักกะสัน
[13.745856, 13.754856, 100.558626, 100.567626],        #     อโศก
[13.741226, 13.747226, 100.586705, 100.592705],        #     คลองตัน
[13.739971, 13.745971, 100.597264, 100.603264],        #     สุขุมวิท 71
[13.733767, 13.742767, 100.635352, 100.644352],        #     หัวหมาก
[13.728604, 13.737604, 100.684232, 100.693232],        #     บ้านทับช้าง
[13.725295, 13.734295, 100.713652, 100.722652],        #     ซอยวัดลานบุญ
[13.723117, 13.732117, 100.744147, 100.753147],        #     ลาดกระบัง
[13.72524,  13.73124,  100.7724  ,  100.7784],         #     พระจอมเกล้า
[13.725468, 13.731468, 100.779871, 100.785871],        #     หัวตะเข้
[13.715631, 13.724631, 100.858355, 100.867355],        #     คลองหลวงแพ่ง
[13.712106, 13.721106, 100.89477 , 100.90377],         #     คลองอุดมชลจร
[13.706353, 13.715353, 100.923891, 100.932891],        #     เปรง
[13.702468, 13.711468, 100.958406, 100.967406],        #     คลองแขวงกลั่น
[13.699568, 13.708568, 100.986955, 100.995955],        #     คลองบางพระ
[13.695743, 13.704743, 101.013505, 101.022505],        #     บางเคย
[13.686145, 13.704145, 101.050723, 101.070902],        #     ชุมทางฉะเชิงเทรา
[13.799987, 13.817987, 101.085407, 101.103407],        #     โพรงอากาศ
[13.841621, 13.859621, 101.092765, 101.110765],        #     บางน้ำเปรี้ยว
[13.898512, 13.916512, 101.105012, 101.123012],        #     ชุมทางคลองสิบเก้า
[13.934517, 13.943517, 101.118279, 101.136279],        #     คลองซี่สิบเอ็ด
[13.961843, 13.979843, 101.140473, 101.158473],        #     โยทะกา
[14.004891, 14.022891, 101.192905, 101.210905],        #     บ้านสร้าง
[14.044974, 14.053974, 101.263791, 101.272791],        #     หนองน้ำขาว
[14.06535,  14.08335,  101.306238, 101.324238],        #     บ้านปากพลี
[14.061812, 14.079812, 101.364596, 101.382596],        #     ปราจีนบุรี
[14.055777, 14.073777, 101.44855,  101.46655],         #     โคกมะกอก
[14.054668, 14.072668, 101.510058, 101.528058],        #     ประจันตคาม
[14.048027, 14.057027, 101.565217, 101.574217],        #     หนองแสง
[14.031123, 14.049123, 101.589915, 101.607915],        #     บ้านดงบัง
[14.013339, 14.031339, 101.63338,  101.65138],         #     บ้านพรมแสง
[14.002926, 14.011926, 101.674325, 101.683325],        #     บ้านเกาะแดง
[13.98082,  13.99882,  101.713862, 101.731862],        #     กบินทร์บุรี
[13.992273, 14.001273, 101.755957, 101.764957],        #     กบินทร์เก่า
[13.971702, 13.989702, 101.81644,  101.83444],         #     หนองสัง
[13.943985, 13.961985, 101.91212,  101.93012],         #     พระปรง
[13.908303, 13.926303, 101.960036, 101.978036],        #     บ้านแก้ง
[13.873769, 13.891769, 102.000475, 102.018475],        #     ศาลาลำดวน
[13.817814, 13.835814, 102.066211, 102.084211],        #     สระแก้ว
[13.7812,   13.7992,   102.159762, 102.177762],        #     ท่าเกษม
[13.7616,   13.7796,   102.222258, 102.240258],        #     ห้วยโจด
[13.732161, 13.750161, 102.314792, 102.332792],        #     วัฒนานคร
[13.720212, 13.729212, 102.376028, 102.385028],        #     บ้านโป่งคอม
[13.703872, 13.721872, 102.413393, 102.431393],        #     ห้วยเดื่อ
[13.683745, 13.701745, 102.49596,  102.51396],];       #     อรัญประเทศ

location =
[[13.740780,100.516868],[13.757034,100.533081],[13.753711,100.547391],[13.750
356,100.563126],[13.744226,100.589705],[13.742971,100.600264],[13.738267,100.
639852],[13.733104,100.688732],[13.729795,100.718152],[13.727617,100.748647],
[13.72824,100.7754],[13.728468,100.782871],[13.720131,100.862855],[13.716606,
100.89927],[13.710853,100.928391],[13.706968,100.962906],[13.704068,100.99145
```

```
5],[13.700243,101.018005],[13.695145,101.059723],[13.808987,101.094407],[13.8
50621,101.101765],[13.907512,101.114012],[13.939017,101.127279],[13.970843,10
1.149473],[14.013891,101.201905],[14.049474,101.268291],[14.07435,101.315238]
,[14.070812,101.373596],[14.064777,101.45755],[14.063668,101.519058],[14.0525
27,101.569717],[14.040123,101.598915],[14.022339,101.64238],[14.007426,101.67
8825],[13.98982,101.722862],[13.996773,101.760457],[13.980702,101.82544],[13.
952985,101.92112],[13.917303,101.969036],[13.882769,102.009475],[13.826814,10
2.075211],[13.7902,102.168762],[13.7706,102.231258],[13.741161,102.323792],[1
3.724712,102.380528],[13.712872,102.422393],[13.692745,102.50496]]

def estimatetoAran(latitude, longitude, station, location, timedatabaseH,
timedatabaseM, timedatabaseS, s275):
    linesc = csv.reader(csvfile)
    dataset = []
    dataset1 = []
    dataspeed = []
    targetsetc = []
    for linec in linesc:
        row1c = (' '.join(linec[0:1]))
        lac = float(row1c)
        row2c = (' '.join(linec[1:2]))
        lnc = float(row2c)
        row4c = (' '.join(linec[3:4]))
        minutec = float(row4c)
        inputsetc = [lac,lnc]
        inputsetc1 = [lac,lnc]
        targetsetc.append(minutec)
        dataset.append(inputsetc)
        dataset1.append(inputsetc1)
    testset = []
    inputset = [latitude, longitude]
    testset.append(inputset)
    from sklearn.neighbors import KNeighborsRegressor
    neigh = KNeighborsRegressor(n_neighbors=7)
    neigh.fit(dataset, targetsetc)
    KNeighborsRegressor(...)
    time_pred = math.floor(neigh.predict(testset))
    time_pred1 = []
    for l in range(0, 47):
        neigh1 = KNeighborsRegressor(n_neighbors=7)
        neigh1.fit(dataset1, targetsetc)
        time_pred1.append(math.floor(neigh1.predict([location[l]])))
    timearrive = []
    timeHour = []
    timeMinute = []
    timeMinuteM = []
    timeindatabase = []
    timeestimate = []
    for m in range(0, 47):
        if latitude >= station[m][0] and latitude <= station[m][1] and
longitude >= station[m][2] and longitude <= station[m][3] and (s275 != 0):
            a = m
            for e in range(0, 47):
                timearrive.append(math.floor(abs(time_pred - time_pred1[e])))
                timeindatabase.append(math.floor(timedatabaseH*3600 +
timedatabaseM*60 + timedatabaseS + timearrive[e]))
                timeestimate.append(math.floor((timeindatabase[e] / 60)))
            sql = ("INSERT INTO estimatetime275(Bangkok, Phayathai, Makkasan,
Asoke, khlongtan, Sukhumvit, Haumak, Banthapchang, Soiwatlanboon, Ladkrabang,
Prachomklao,Hautakhe, Klongluangphang, Klongudomchon, Preng, Klongkwaeng,
Klongbangphra, Bangtoey, Chachoengsao, Phrongakat, Bangnamprieo,
Khlongsipkao, Khlongyisipet,Yothaka, Bansang, Nongnamkhao, Banpakphli,
Prachinburi, Khokmakok, Prachantakham, Nongsaeng, Bandongbang, Banphromsaeng,
Bankodaeng, Kabinburi, Kabinkao,Nongsang, Phraprong, Bankaeng, Salalamduan,
Sakaeo, Thakasem, Huaichot, Vatthananakhon, Banpongkom, Huaidua,
```

```
Aranyaprathet) VALUES('%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s',
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s',
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s','%s', '%s', '%s',
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')" %
(timeestimate[0], timeestimate[1], timeestimate[2],
timeestimate[3],timeestimate[4], timeestimate[5], timeestimate[6],
timeestimate[7], timeestimate[8], timeestimate[9], timeestimate[10],
timeestimate[11], timeestimate[12], timeestimate[13],timeestimate[14],
timeestimate[15], timeestimate[16], timeestimate[17], timeestimate[18],
timeestimate[19], timeestimate[20], timeestimate[21], timeestimate[22],
timeestimate[23], timeestimate[24], timeestimate[25], timeestimate[26],
timeestimate[27], timeestimate[28], timeestimate[29], timeestimate[30],
timeestimate[31], timeestimate[32], timeestimate[33], timeestimate[34],
timeestimate[35], timeestimate[36], timeestimate[37], timeestimate[38],
timeestimate[39], timeestimate[40], timeestimate[41], timeestimate[42],
timeestimate[43], timeestimate[44], timeestimate[45], timeestimate[46]))
            cur.execute(sql)
            conn.commit()
            break


def estimatetoBangkok(latitude, longitude, station, location, timedatabaseH,
timedatabaseM, timedatabaseS, s275):
    linesc = csv.reader(csvfile)
    dataset = []
    dataset1 = []
    dataspeed = []
    targetsetc = []
    for linec in linesc:
        row1c = (' '.join(linec[0:1]))
        lac = float(row1c)
        row2c = (' '.join(linec[1:2]))
        lnc = float(row2c)
        row4c = (' '.join(linec[3:4]))
        minutec = float(row4c)
        inputsetc = [lac,lnc]
        inputsetc1 = [lac,lnc]
        targetsetc.append(minutec)
        dataset.append(inputsetc)
        dataset1.append(inputsetc1)
    testset = []
    inputset = [latitude, longitude]
    testset.append(inputset)
    from sklearn.neighbors import KNeighborsRegressor
    neigh = KNeighborsRegressor(n_neighbors=7)
    neigh.fit(dataset, targetsetc)
    KNeighborsRegressor(...)
    time_pred = math.floor(neigh.predict(testset))
    time_pred1 = []
    for l in range(0, 47):
        neigh1 = KNeighborsRegressor(n_neighbors=7)
        neigh1.fit(dataset1, targetsetc)
        time_pred1.append(math.floor(neigh1.predict([location[l]])))
    timearrive = []
    timeHour = []
    timeMinute = []
    timeMinuteM = []
    timeindatabase = []
    timeestimate = []
    for m in range(0, 47):
        if latitude >= station[m][0] and latitude <= station[m][1] and
longitude >= station[m][2] and longitude <= station[m][3] and (s275 != 0):
            a = m
            for e in range(0, a):
                timearrive.append(math.floor(abs(time_pred - time_pred1[e])))
```

```
                timeindatabase.append(math.floor(timedatabaseH*3600 +
timedatabaseM*60 + timedatabaseS + timearrive[e]))
                timeestimate.append(math.floor((timeindatabase[e] / 60)))
            for e in range(a, 47):
                timeestimate.append(0)
            sql = ("INSERT INTO estimatetime275(Bangkok, Phayathai, Makkasan,
Asoke, khlongtan, Sukhumvit, Haumak, Banthapchang, Soiwatlanboon, Ladkrabang,
Prachomklao,Hautakhe, Klongluangphang, Klongudomchon, Preng, Klongkwaeng,
Klongbangphra, Bangtoey, Chachoengsao, Phrongakat, Bangnamprieo,
Khlongsipkao, Khlongyisipet,Yothaka, Bansang, Nongnamkhao, Banpakphli,
Prachinburi, Khokmakok, Prachantakham, Nongsaeng, Bandongbang, Banphromsaeng,
Bankodaeng, Kabinburi, Kabinkao,Nongsang, Phraprong, Bankaeng, Salalamduan,
Sakaeo, Thakasem, Huaichot, Vatthananakhon, Banpongkom, Huaidua,
Aranyaprathet) VALUES('%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s',
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s',
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s','%s', '%s', '%s',
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')" %
(timeestimate[0], timeestimate[1], timeestimate[2],
timeestimate[3],timeestimate[4], timeestimate[5], timeestimate[6],
timeestimate[7], timeestimate[8], timeestimate[9], timeestimate[10],
timeestimate[11], timeestimate[12], timeestimate[13],timeestimate[14],
timeestimate[15], timeestimate[16], timeestimate[17], timeestimate[18],
timeestimate[19], timeestimate[20], timeestimate[21], timeestimate[22],
timeestimate[23], timeestimate[24], timeestimate[25], timeestimate[26],
timeestimate[27], timeestimate[28], timeestimate[29], timeestimate[30],
timeestimate[31], timeestimate[32], timeestimate[33], timeestimate[34],
timeestimate[35], timeestimate[36], timeestimate[37], timeestimate[38],
timeestimate[39], timeestimate[40], timeestimate[41], timeestimate[42],
timeestimate[43], timeestimate[44], timeestimate[45], timeestimate[46]))
            cur.execute(sql)
            conn.commit()
            break

while True:
    conn =
pymysql.Connect(host='localhost',user='root',passwd='ttsproj',db='tts')
    cur = conn.cursor()
    selectdate = ('SELECT DATE_FORMAT(day, "%a-%d-%m-%y") FROM track2 ORDER
BY day DESC LIMIT 1')
    cur.execute(selectdate)
    selectdate1 = cur.fetchone()
    for selectdate2 in selectdate1:
        Datedatabase = selectdate2
    Datesplit = Datedatabase.split('-')
    DatedatabaseDay = Datesplit[0]
    selecttime = ('SELECT DATE_FORMAT(day, "%H:%i:%s") FROM track2 ORDER BY
day DESC LIMIT 1')
    cur.execute(selecttime)
    selecttime1 = cur.fetchone()
    for selecttime2 in selecttime1:
        Timedatabase = selecttime2
    Timesplit = Timedatabase.split(':')
    timedatabaseHo = Timesplit[0]
    timedatabaseH = int(timedatabaseHo)
    timedatabaseMi = Timesplit[1]
    timedatabaseM = int(timedatabaseMi)
    timedatabaseSe = Timesplit[2]
    timedatabaseS = int(timedatabaseSe)
    selectstaforward = ('SELECT sta FROM track2 WHERE No=(SELECT max(No) FROM
track2)')
    cur.execute(selectstaforward)
    forward = cur.fetchone()
    for staforward in forward:
        stanow = staforward
```

```
    selectstabackward = ('SELECT sta FROM track2 WHERE No=(SELECT (max(No)-
90) FROM track2)')
    cur.execute(selectstabackward)
    backward = cur.fetchone()
    for stabackward in backward:
        staninety = stabackward
    selects = ('SELECT s FROM track2 WHERE No=(SELECT max(No) FROM track2)')
    cur.execute(selects)
    ss = cur.fetchone()
    for s in ss:
        s275 = s
    if stanow != 0 or stanow - staninety != 0:
        # outbound
        if (timedatabaseH >= 5 and timedatabaseH <= 10) or stanow >
staninety:
            selectla = ('SELECT L FROM track2 WHERE No=(SELECT max(No) FROM
track2)')
            cur.execute(selectla)
            la = cur.fetchone()
            for La in la:
                latitude = La
            selectlong = ('SELECT n FROM track2 WHERE No=(SELECT max(No) FROM
track2)')
            cur.execute(selectlong)
            long = cur.fetchone()
            for Long in long:
                longitude = Long
            if DatedatabaseDay == "Mon":
                with open('DatasetMonday.csv') as csvfile:
                    estimatetoAran(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
            elif DatedatabaseDay == "Tue":
                with open('DatasetTuesday.csv') as csvfile:
                    estimatetoAran(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
            elif DatedatabaseDay == "Wed":
                with open('DatasetWednesday.csv') as csvfile:
                    estimatetoAran(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
            elif DatedatabaseDay == "Thu":
                with open('DatasetThursday.csv') as csvfile:
                    estimatetoAran(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
            elif DatedatabaseDay  == "Fri":
                with open('DatasetFriday.csv') as csvfile:
                    estimatetoAran(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
            elif DatedatabaseDay  == "Sat":
                with open('DatasetSaturday.csv') as csvfile:
                    estimatetoAran(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
            else:
                with open('DatasetSunday.csv') as csvfile:
                    estimatetoAran(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)

    else:
        # inbound
        selectla = ('SELECT L FROM track2 WHERE No=(SELECT max(No) FROM
track2)')
        cur.execute(selectla)
        la = cur.fetchone()
        for La in la:
            latitude = La
```

```
        selectlong = ('SELECT n FROM track2 WHERE No=(SELECT max(No) FROM
track2)')
        cur.execute(selectlong)
        long = cur.fetchone()
        for Long in long:
            longitude = Long
        if DatedatabaseDay == "Mon":
            with open('DatasetMondayBK.csv') as csvfile:
                estimatetoBangkok(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
        elif DatedatabaseDay == "Tue":
            with open('DatasetTuesdayBK.csv') as csvfile:
                estimatetoBangkok(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
        elif DatedatabaseDay == "Wed":
            with open('DatasetWednesdayBK.csv') as csvfile:
                estimatetoBangkok(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
        elif DatedatabaseDay == "Thu":
            with open('DatasetThursdayBK.csv') as csvfile:
                estimatetoBangkok(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
        elif DatedatabaseDay  == "Fri":
            with open('DatasetFridayBK.csv') as csvfile:
                estimatetoBangkok(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
        elif DatedatabaseDay  == "Sat":
            with open('DatasetSaturdayBK.csv') as csvfile:
                estimatetoBangkok(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
        else:
            with open('DatasetSundayBK.csv') as csvfile:
                estimatetoBangkok(latitude, longitude, station, location,
timedatabaseH, timedatabaseM, timedatabaseS, s275)
    if stanow <= 2 and (timedatabaseH >= 21 or timedatabaseH <= 4):
        break
    refresh()
cur.close()
conn.close()
```