

การบริการระดับต้นสำหรับผู้พิการทางสายตา  
Location service for the visually impaired



ปริญญาโท เป็นส่วนหนึ่งของกรศึกษาต่อหลักสูตรปริญญาโท สาขาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2555

การบริการระบุตำแหน่งสำหรับผู้พิการทางสายตา  
LOCATION SERVICE FOR THE VISUALLY IMPAIRED



จากรุวรรณ พันทอง  
ชนิสรา พงษ์พูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาก่อนจะนำออกจำหน่ายหรือเข้าถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
ปีการศึกษา 2555

ปริญญาานิพนธ์ปีการศึกษา 2555

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การบริการระบุตำแหน่งสำหรับผู้พิการทางสายตา

LOCATION SERVICE FOR THE VISUALLY IMPAIRED

ผู้จัดทำ

- |                   |         |              |          |
|-------------------|---------|--------------|----------|
| 1. นางสาวจรรววรรณ | พันทอง  | รหัสนักศึกษา | 52010152 |
| 2. นางสาวชนิสรา   | พงษ์พูล | รหัสนักศึกษา | 52010212 |



..... อาจารย์ที่ปรึกษา  
(ดร. วีชระ ฉัตรวิริยะ)

..... อาจารย์ที่ปรึกษา  
(อาจารย์ บัณฑิต พัสยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การบริการระบุตำแหน่งสำหรับผู้พิการทางสายตา

นางสาว จารุวรรณ	พันทอง	52010152
นางสาว ชนิสรา	พงษ์พูล	52010212
ดร. วัชรระ	ฉัตรวิริยะ	อาจารย์ที่ปรึกษา
อาจารย์ บัณฑิต	พัสยา	อาจารย์ที่ปรึกษาร่วม
ปีการศึกษา 2555		

### บทคัดย่อ

ในปัจจุบันนี้ยังไม่มีระบบที่เหมาะสมสำหรับผู้พิการทางสายตาที่อำนวยความสะดวกสบายในการบอกตำแหน่งด้วยเสียง ซึ่งมีการออกแบบในการใช้งานที่เหมาะสมกับผู้พิการทางสายตาโดยเฉพาะ โดยมีการประยุกต์นำสิ่งต่างๆ ทั้งความรู้ การแก้ปัญหา และเทคโนโลยีที่มีในปัจจุบันให้เกิดประโยชน์สูงสุด ซึ่งจะมี การคำนวณทางคณิตศาสตร์ วิศวกรรมศาสตร์ การวิเคราะห์ และคาดการณ์ถึงแนวทางในการบอกตำแหน่งที่ทำให้ผู้พิการทางสายตาเข้าใจ และก็ยังสามารถนำไปใช้กับผู้ใช้งานทั่วไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# LOCATION SERVICE FOR THE VISUALLY IMPAIRED

Ms. Jaruwan	Pannong	52010152
Ms. Chanisara	Pongpool	52010212
Dr. Watchara	Chatwiriya	Advisor
Mr. Bundit	Pasaya	Co-advisor

Academic Year 2012

## ABSTRACT

At present there is no proper system for the visually impaired in a comfortable position with the sound. It is designed to work with the visually impaired in particular. The applications bring both knowledge and technology solutions available today to achieve maximum benefit. This is a mathematical calculation. Engineering Analysis and Forecasts to guide the positioning of the visually impaired to understand. And it can also be applied to general users.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

โครงการ เรื่อง การบริการระบุตำแหน่งสำหรับผู้พิการทางสายตานิ้สำเร็จลุล่วงได้โดยความ  
กรุณาของดร. วชิระ ฉัตรวิริยะที่ปรึกษาโครงการและอาจารย์บัณฑิต พัสยาที่ปรึกษาร่วมโครงการใน  
การให้คำปรึกษา การแก้ปัญหา แนวคิดต่างๆในการออกแบบ พัฒนาระบบตลอดจนการแนะนำและ  
ตรวจทานความเหมาะสมและความถูกต้องของรายงาน

และเพื่อนๆที่ให้คำแนะนำในการแก้ปัญหาเรื่องการพัฒนาโปรแกรมคือ นายเฉลิม แซ่ลิ้ม  
นายณัฐกร โชติธนาการ และนายจිරพัฒน์ สันติวิโรทัย  
ผู้จัดทำจึงขอขอบคุณในความกรุณา มา ณ ที่นี้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 วิธีการดำเนินการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 ส่วนประกอบของปฏิญานิพนธ์.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	5
2.1 API.....	5
2.2 Google Map API.....	5
2.3 Geolocation.....	8
2.4 Geocoding.....	9
2.5 Direction.....	15
2.6 Google Text To Speech.....	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

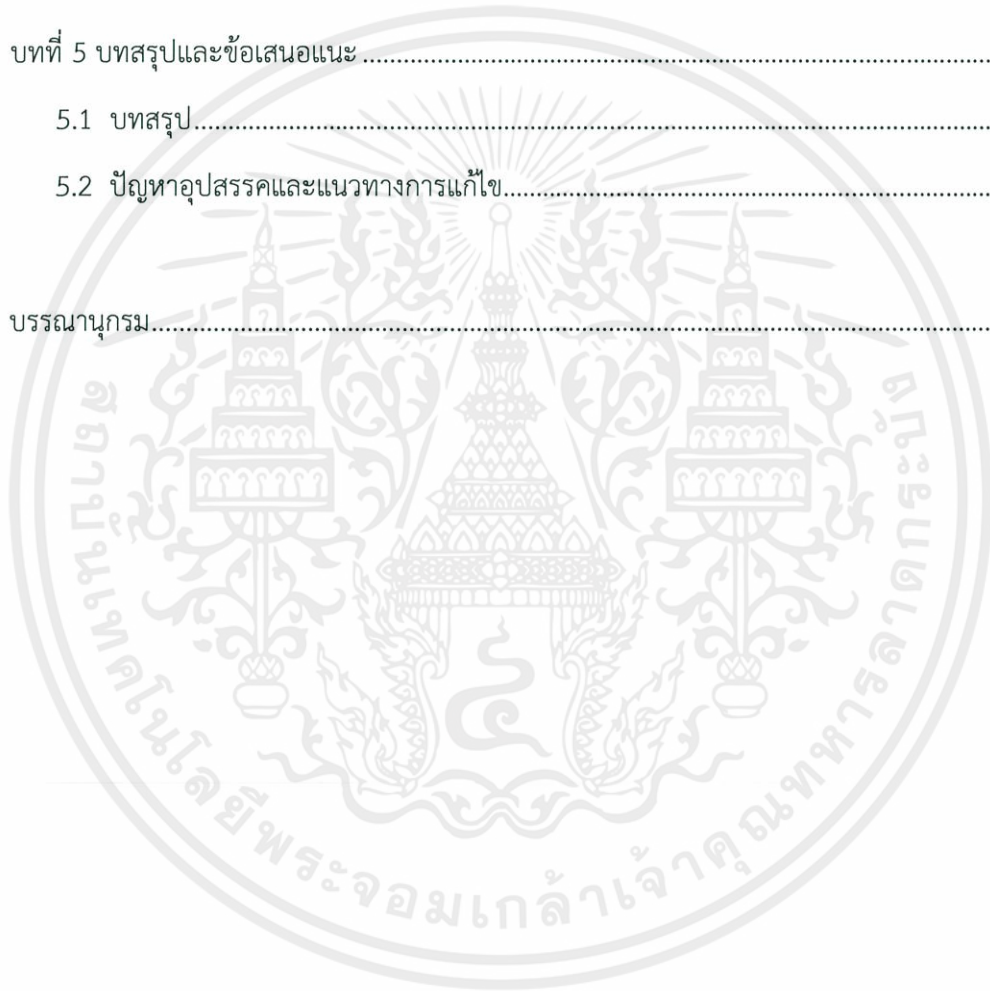
## สารบัญ(ต่อ)

	หน้า
2.7 Web Server.....	20
2.8 Apperve.....	21
2.9 SQL.....	21
2.10 MySQL.....	22
2.11 PHP.....	24
2.12 HTML.....	26
2.13 HTML5.....	26
2.14 JavaScript.....	27
2.15 Ajax.....	29
2.16 จุดสนใจ(POI).....	29
บทที่ 3 การออกแบบและพัฒนา.....	30
3.1. ออกแบบการทำงานของระบบโดยรวม.....	30
3.2 การออกแบบแผนผังการทำงานโปรแกรม (Program Flowchart).....	31
3.3 การออกแบบการไหลของข้อมูล (Data Flow).....	35
3.4 ออกแบบฐานข้อมูลสถานที่ใกล้เคียง (POI).....	41
บทที่ 4 การทดลองและผลการทดลอง.....	43
4.1 ส่วนการสร้างฐานข้อมูล.....	43
4.2 ส่วนพัฒนาแต่ละฟังก์ชันของระบบ.....	46
4.3 ส่วนพัฒนาระบบเมื่อนำฟังก์ชันต่างๆมารวมกัน.....	73
4.4 ส่วนการอัปเดตเว็บไซต์เข้าสู่เว็บเซิร์ฟเวอร์.....	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานวิจัยเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ต้องเปลืองเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
4.5.ส่วนทดสอบการทำงานของระบบ.....	80
บทที่ 5 บทสรุปและข้อเสนอแนะ .....	83
5.1 บทสรุป.....	83
5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข.....	84
บรรณานุกรม.....	85



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
ตาราง 3.1 รายละเอียดโปรแกรมย่อยในแผนผังการไหลของข้อมูล .....	37
ตาราง 3.2 รายละเอียดโปรแกรมย่อยในแผนผังการไหลของข้อมูล (ต่อ).....	38
ตาราง 3.3 ข้อมูลพิกัดตำแหน่งปัจจุบัน.....	39
ตาราง 3.4 ข้อมูลพิกัดละติจูดเดิม .....	39
ตาราง 3.5. ข้อมูลสถานที่ใกล้เคียง (POI).....	39
ตาราง 3.6 ข้อมูลสถานที่ใกล้เคียงที่สุด.....	39
ตาราง 3.7 ข้อความที่จะแปลงเป็นเสียง.....	40
ตาราง 3.8 ชื่อไฟล์เสียงการอธิบายการบอกตำแหน่ง.....	41
ตาราง 3.9 รายละเอียดตารางฐานข้อมูล .....	41
ตาราง 4.1 แสดงผลลัพธ์การทดสอบการทำงานของระบบบนเครื่องคอมพิวเตอร์.....	81
ตาราง 4.2 แสดงผลลัพธ์การทดสอบการทำงานของระบบบนโทรศัพท์มือถือ.....	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 แสดงแผนที่ google.....	7
รูปที่ 3.1 การออกแบบการทำงานของระบบ.....	30
รูปที่ 3.2 การออกแบบการทำงานของโปรแกรมทั้งหมด.....	31
รูปที่ 3.3 ออกแบบแผนผังการทำงานของโปรแกรมส่วนสร้างไฟล์เสียงการอธิบายตำแหน่ง.....	33
รูปที่ 3.4 แผนผังการไหลของข้อมูล ระดับ context.....	35
รูปที่ 3.5 แผนผังการไหลของข้อมูล ระดับ 0.....	36
รูปที่ 4.1 แสดงฐานข้อมูลที่สร้าง.....	44
รูปที่ 4.2 แสดงรายละเอียดของชนิดข้อมูลในตาราง.....	44
รูปที่ 4.3 แสดงขั้นตอนการนำข้อมูลที่จัดเก็บลงฐานข้อมูล.....	45
รูปที่ 4.4 φόρμที่สร้างเพื่อใช้ในการทดสอบ.....	47
รูปที่ 4.5 φόρμที่สร้างเพื่อใช้ในการทดสอบ (ต่อ).....	48
รูปที่ 4.6. การหาพิกัดตำแหน่งทางภูมิศาสตร์โดยลากสัญลักษณ์เพื่อกำหนดพิกัดที่ต้องการหรือกดปุ่ม ตรวจหาพิกัดปัจจุบัน.....	48
รูปที่ 4.7. แผนที่ที่ได้จากการสร้างและสัญลักษณ์ที่ใช้ในการกำหนดตำแหน่ง.....	50
รูปที่ 4.8. ผลลัพธ์ที่ได้จากการแปลงข้อมูลพิกัดภูมิศาสตร์ที่กำหนดเป็นรายละเอียดข้อมูลสถานที่ ..	51
รูปที่ 4.9. การกำหนดขอบเขตในการเลือกข้อมูลสถานที่ใกล้เคียง.....	51
รูปที่ 4.10. ผลลัพธ์ที่ได้จากการดึงข้อมูลสถานที่ใกล้เคียง.....	55
รูปที่ 4.11. การสร้างฟอร์มเพื่อจัดเก็บข้อมูลลงตัวแปรเพื่อนำมาใช้ในการประมวลผลต่อไป.....	56
รูปที่ 4.12 ผลลัพธ์การคำนวณระยะทาง.....	59
รูปที่ 4.13 การสร้างฟอร์มเพื่อจัดเก็บข้อมูลสถานที่ใกล้เคียงที่ใกล้ที่สุดลงตัวแปรเพื่อนำมาใช้ในการ ประมวลผลต่อไป.....	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 4.14 ผลลัพธ์การจัดลำดับข้อมูลตามระยะทาง.....	62
รูปที่ 4.15 การสร้างฟอร์มเพื่อทดสอบการประมวลผลในการใช้งาน google dirction .....	63
รูปที่ 4.16 ผลลัพธ์รายละเอียดเพื่ออธิบายตำแหน่งใกล้เคียงที่สุด 3 สถานที่ .....	64
รูปที่ 4.17 ผลลัพธ์การปรับปรุงข้อมูล.....	67
รูปที่ 4.18 การสร้างฟอร์มเพื่อทดสอบการแปลงข้อมูลเป็นเสียง.....	67
รูปที่ 4.19 ผลลัพธ์ที่ได้จากการแปลงข้อความเป็นเสียง .....	70
รูปที่ 4.20 ผลลัพธ์ที่ได้จากการรวมไฟล์เสียง .....	72
รูปที่ 4.21 ผลลัพธ์จากการรวมโปรแกรมย่อย.....	77
รูปที่ 4.22 การสร้างไซต์.....	78
รูปที่ 4.23 การกำหนด ftp เพื่อใช้ในการอัปโหลดไฟล์ระบบ.....	79
รูปที่ 4.24 การเชื่อมต่อกับ server ที่ต้องการอัปโหลดไฟล์ระบบ.....	79
รูปที่ 4.25 ผลลัพธ์ที่ได้จากการอัปโหลดไฟล์ระบบทั้งหมด .....	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของโครงการ

ในปัจจุบันนี้ สังคมประกอบด้วยคนปกติและผู้พิการ สำหรับคนปกติยังสามารถดำเนินชีวิตได้อย่างสะดวกสบายโดยมีสิ่งเอื้ออำนวยต่างๆเข้ามามีบทบาทกับชีวิต แต่สำหรับผู้พิการนั้น โดยเฉพาะผู้พิการทางสายตา ยังไม่ มีระบบที่เหมาะสมที่จะอำนวยความสะดวกสบายในการบอกตำแหน่งด้วยเสียง

ด้วยเหตุดังกล่าวๆทำให้สนใจที่จะศึกษาและพัฒนาระบบบอกตำแหน่งด้วยเสียงสำหรับผู้พิการทางสายตา โดย มีการประยุกต์นำสิ่งต่างๆทั้งความรู้การแก้ปัญหาและเทคโนโลยีที่มีในปัจจุบัน เพื่อให้เกิดประโยชน์สูงสุด ต่อผู้พิการทางสายตามากที่สุด ซึ่ง จะมีการคำนวณทางคณิตศาสตร์ วิศวกรรมศาสตร์ การวิเคราะห์และคาดการณ์ถึงแนวทางในการบอกตำแหน่งที่ทำให้ผู้พิการทางสายตาเข้าใจโดยมีการออกแบบที่เหมาะสมสำหรับผู้พิการโดยเฉพาะ และยังสามารถนำไปใช้กับผู้ที่ต้องการได้ทั่วไป

### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่ออำนวยความสะดวกสบายในการให้ข้อมูลสารสนเทศเกี่ยวกับตำแหน่งสำหรับผู้พิการทางสายตาหรือผู้ใช้งานทั่วไป
2. เพื่อนำเทคโนโลยีที่มีในปัจจุบันมาประยุกต์ให้เกิดประโยชน์สูงสุด
3. เพื่อศึกษาเกี่ยวกับการใช้งาน google service api

### 1.3 ขอบเขตของโครงการ

1. ศึกษา application ที่มีสำหรับผู้พิการในปัจจุบันรวมถึงที่คล้ายกับโครงการ
2. ศึกษาข้อมูลเอกสารของ google map development และทดสอบการทำงาน ดังนี้ Google map api version 3, services ต่างๆของ google map (Geocoding, Geolocation, Diriction)
3. ศึกษาการแปลงข้อความเป็นเสียงจาก Google text to speech (TTS)

เอกสารนี้เป็นเอกสารที่ 4. ศึกษาการเก็บข้อมูลจุดสนใจ (POI) ภาษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น 5. ศึกษาและทดลองการสร้าง server ด้วย Appserv ของเอกสารทุกครั้งที่มีการนำไปใช้

6. ภาษาที่ใช้ในทำโครงการประกอบด้วย ภาษา Java (ใช้พัฒนาแอปพลิเคชันแอนดรอยด์), ภาษา PHP (การติดต่อระหว่าง server กับ client), ภาษา html และ java script (ใช้ในการติดต่อดึงข้อมูลจาก google map) และภาษา SQL (จัดการและเรียกใช้ข้อมูลในฐานข้อมูล)

7. วิเคราะห์และแก้ปัญหาการบอกตำแหน่งบนแผนที่โดยการสร้างเครื่องมือมาทดสอบแล้ววางเงื่อนไขในการบอกตำแหน่ง

8. ทดสอบการทำงานของโปรแกรมโดยสร้างตำแหน่งแล้วตรวจสอบผลลัพธ์การบอกตำแหน่งรวมถึงระยะเวลาที่ได้จากการประมวลผลและดึงข้อมูลเพื่อเปรียบเทียบผลลัพธ์

#### 1.4 วิธีการดำเนินการ

1. ศึกษาวิเคราะห์ โดยแบ่งการวิเคราะห์ออกเป็น 3 ส่วน
  - 1) ส่วน Application ในระบบบอกตำแหน่งสำหรับผู้พิการทางสายตา ได้พัฒนาพัฒนาบนเว็บแอปพลิเคชัน
  - 2) ส่วน Server บริการข้อมูลสถานที่ใกล้เคียง (POI)
  - 3) ส่วน API (Application Programming Interface) เป็น library (ชิ้นส่วนของ function/module/utility) ที่ผู้ผลิตสร้างขึ้นมาให้ผู้ที่พัฒนา Application เรียกใช้งานได้ ในโครงการนี้ จะใช้ Google Map API ในการพัฒนาระบบ
2. แผนการทำงาน
 

Work breakdown
3. ออกแบบ
  - 1) โครงสร้างการทำงานของระบบ คือ Flowchart และ Data Flow
  - 2) หน้าจอการทำงานของแอปพลิเคชันและโปรแกรมทดสอบการทำงาน
4. พัฒนา แบ่งการพัฒนาออกเป็น 3 ส่วนใหญ่ๆ
  - 1) ส่วนการสร้างฐานข้อมูล
 

จัดเก็บข้อมูลจุดสนใจ (POI) ลงในฐานข้อมูล
  - 2) ส่วนพัฒนาแต่ละฟังก์ชันของระบบ
    - การแปลงตำแหน่ง latitude และ longitude เป็นสถานที่
    - การวิเคราะห์เลือกนำข้อมูลเพื่อส่งไปออกเสียง
    - การคำนวณระยะทาง
    - การแปลงข้อมูลเป็นเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การควบคุมการทำงานของโปรแกรม เช่น ปุ่ม play เพื่อฟังการอธิบายตำแหน่ง และปุ่ม pause เพื่อหยุดฟังการอธิบาย เป็นต้น
  - 3) ส่วนพัฒนาระบบเมื่อนำฟังก์ชันต่าง ๆ มารวมกัน
  - 4) ส่วนการอัปโหลดเว็บไซต์เข้าสู่เว็บเซิร์ฟเวอร์
  - 5) ส่วนทดสอบการทำงานของระบบ
5. ทดสอบ
- สร้างสถานการณ์จำลองเพื่อมาทดสอบ จำนวนไม่ต่ำกว่า 20 ชุด
6. จัดทำเอกสาร
- 1) รายงานความก้าวหน้าของโครงการทุกๆ 2 อาทิตย์
  - 2) รายงานของโครงการ
  - 3) เอกสารการนำเสนอ (Presentation)

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจเกี่ยวกับการทำงานของ Google Map API
2. ได้รับความรู้ความเข้าใจเกี่ยวกับการ ออกแบบและ สร้างเซิร์ฟเวอร์ (server) ด้วยโปรแกรม Appserv
3. ได้เว็บแอปพลิเคชันที่สามารถบอกตำแหน่งได้ตามเงื่อนไขที่กำหนดได้
4. ได้เว็บแอปพลิเคชันที่รองรับการใช้งานสำหรับผู้พิการทางสายตาได้

### 1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง API ,Google Map API, Geolocation , Geocoding , Direction, Google text to speech (TTS), Web Server, Apperve, SQL, MySQL, PHP, HTML, HTML5, Ajax, JavaScript, POIและข้อมูลสถานที่จุดสนใจ(POI)

บทที่ 3 การออกแบบและพัฒนา กล่าวถึง การออกแบบการทำงานของ ระบบ (Flowchart) , ออกแบบการไหลของข้อมูล (Data Flow) และออกแบบฐานข้อมูลการจัดเก็บข้อมูลสถานที่ใกล้เคียงที่สนใจ (POI)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึง ส่วนการสร้างฐานข้อมูล ,ส่วนพัฒนาแต่ละฟังก์ชันของระบบ,ส่วนพัฒนาระบบเมื่อนำฟังก์ชันต่างๆมารวมกัน ,ส่วนการอัปโหลดเว็บไซต์เข้าสู่เว็บเซิร์ฟเวอร์และส่วนทดสอบการทำงานของระบบ

บทที่ 5 บทสรุป กล่าวถึงบทสรุปของโครงการ วิจารณ์สิ่งที่ได้ร้ จากโครงการ ข้อจำกัด รวมถึงปัญหาอุปสรรคต่างๆ ของโครงการ และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีที่ใช้ในการพัฒนา

### 2.1 API

API ย่อมาจาก Application Programming Interface คือ ช่องทางการเชื่อมต่อ ที่จะเชื่อมต่อกับเว็บไซต์ผู้ให้บริการ API จากที่อื่น เป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น หรือเชื่อมการทำงานเข้ากับระบบปฏิบัติการ

ตัวอย่างเช่น Twitter มีหลายเว็บที่ทำการเชื่อมโยงข้อมูลกับ twitter ทั้งเป็นการอ่านข้อมูลจาก twitter หรือส่งข้อมูลเข้า twitter เองก็ตาม ซึ่งล้วนอาศัยการเชื่อมต่อแลกเปลี่ยนข้อมูลกัน ด้วย API นั้นเองและอีกหนึ่งตัวอย่างเช่น Google Maps API เป็นบริการของ Google ที่สามารถนำข้อมูลของ Google Maps มาใช้งานและพัฒนาได้

#### 2.1.1 ประโยชน์

- 1) สามารถรับส่งข้อมูลข้าม Server ได้
- 2) ไม่จำเป็นต้องเข้าหน้าเว็บหลักก็สามารถนำข้อมูลมาใช้งานได้ โดย API แบ่งเป็น
  - 2.1) เอพีไอที่ขึ้นกับภาษา (language-dependent API) คือ เอพีไอที่สามารถเรียกใช้จากโปรแกรมที่เขียนขึ้นด้วยภาษาเพียงภาษาใดภาษาหนึ่ง
  - 2.2) เอพีไอไม่ขึ้นกับภาษา (language-independent API) คือ เอพีไอที่สามารถเรียกได้จากโปรแกรมหลายๆภาษา

API ถือเป็นกลุ่มของฟังก์ชันหรือคลาส (Class) ที่ระบบปฏิบัติการ (OS) หรือผู้ให้บริการ สร้างขึ้นมา เพื่อรองรับการเรียกขอข้อมูลจากโปรแกรมอื่นๆทั้งนี้ API สามารถใช้งานได้กับภาษาในการเขียนโปรแกรมที่รองรับเท่านั้น ซึ่งมันจะถูกจัดทำให้อยู่ในรูปแบบ Syntax หรือ element ที่สามารถนำไปใช้ได้อย่างสะดวกสบาย

### 2.2 Google Map API

Google Maps API ช่วยให้นักพัฒนาสามารถพัฒนาโปรแกรมเพื่อใช้งานแผนที่จาก Google Maps สามารถแทรกองค์ประกอบเข้าไปในส่วนหนึ่งของเว็บเพจที่ต้องการได้โดยใช้ภาษา html และ javascript ซึ่งมีการจัดรูปแบบที่ไม่ซับซ้อน การบริการด้านแผนที่ของ Google นี้เริ่มต้นตั้งแต่กลางปี

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ Google Inc. ปี ค.ศ. 2005 และเป็นบริการฟรี ในการบริการแผนที่ที่สามารถแสดงข้อมูลได้หลักหลายซึ่ง มีแผนที่และภาพถ่ายดาวเทียม ที่คุณภาพดีซึ่งครอบคลุมทั่วพื้นผิวโลกในมาตราส่วนต่างๆ ทำให้การพัฒนาต่อผู้ใช้

ยอดจากสิ่งที่ถูกจัดไว้ให้แล้วเป็นงานที่น่าสนใจ ไม่ต้องลงทุนจัดหาทรัพยากรที่หายากและราคาแพง  
เองมาใช้ในโครงการอย่างที่เคยเป็นในอดีต

เนื่องจาก Google Maps API เป็นโปรแกรมรหัสเปิด (Open source programe) ในภาษา  
จาวาสคริปต์ จึงทำให้ผู้ใช้ที่เป็นนักพัฒนาโปรแกรมสามารถเข้าไปดูรายละเอียดของรหัสโปรแกรมได้  
สะดวก รวมทั้งสามารถปรับเปลี่ยนแก้ไขโปรแกรมได้ ทำให้ Google Maps API มีผู้ใช้กันอย่าง  
กว้างขวางประกอบกับแผนที่และภาพถ่ายดาวเทียมมีคุณภาพดี

Google Maps API จัดเป็นโปรแกรมประเภท Map mashup อย่างง่ายเพื่อเป็นตัวอย่างให้  
ผู้สนใจได้ทำขึ้นเพื่อเริ่มต้นทดลองใช้งาน หรือเรียนรู้ทำความเข้าใจในเทคโนโลยีและขีดความสามารถ  
ของ Google Maps API

### 2.2.1 Google Maps mashup

คำว่า “mashup” หรือ “mash-up” เป็นศัพท์เฉพาะในของการพัฒนา application แบบ  
client-server ประเภทที่ทำเดิมต่อขึ้นเองโดยใช้ software tool และทรัพยากรที่ทางผู้ให้บริการจัด  
ไว้ให้ ซึ่งการพัฒนาแอปพลิเคชันที่ใช้นับอินเทอร์เน็ตมีองค์ประกอบอย่างน้อย 2 ส่วน ส่วนแรกคือ  
application program interface (API) และทรัพยากรสนับสนุนจากเซิร์ฟเวอร์ของผู้ให้บริการ และ  
ส่วนที่ 2 คือโปรแกรมและทรัพยากรของผู้พัฒนาแอปพลิเคชันนั้น (ติดตั้งอยู่ที่เว็บไซต์ของผู้พัฒนา)  
ตัวอย่างกรณีของ Google Maps mashup หมายถึงเว็บเพจที่นักพัฒนาเว็บไซต์สร้างขึ้นให้มี  
ส่วนประกอบที่เป็นแผนที่ที่สามารถทำงานแบบโต้ตอบกับผู้ใช้ได้ โดยมีพื้นฐานเป็นแผนที่ที่เป็นส่วนที่  
Google จัดไว้ให้แล้ว และมีข้อมูลแผนที่ของตนเองซ้อนทับในลักษณะหมุดปัก (ที่สามารถแสดง  
ข้อความซึ่งแฝงอยู่เมื่อผู้ใช้คลิกที่หมุดเหล่านั้น) หรือลักษณะอื่นที่สลับซับซ้อนกว่านั้น

### 2.2.2 ภาษาที่ใช้ในการพัฒนาใน Google Maps API

ความรู้พื้นฐานเบื้องต้นที่จำเป็นในงานพัฒนา Google Maps mashup ซึ่งควรมี ดังนี้

- 1) Html / xhtml
- 2) javascript
- 3) ความรู้พื้นฐานวิชาภูมิศาสตร์ และการแผนที่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน... อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไข และต้องอ้างอิงถึงชื่อเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 แสดงแผนที่ google

### 2.2.3 ประโยชน์

- 1) สามารถเลือกประเภทแผนที่เพื่อนำมาใช้งานได้หลากหลายประเภทตามความเหมาะสมของการใช้งาน คือ
  - 1.1) ROADMAP แสดงแผนที่แบบสองมิติ ซึ่งถ้าไม่มีการตั้งค่าโดยปกติจะใช้ค่านี้นี้เสมอ (default)
  - 1.2) SATELLITE แสดงแผนที่เป็นภาพถ่ายดาวเทียม
  - 1.3) HYBRID แสดงแผนที่ที่มีผสมระหว่างภาพถ่ายดาวเทียม และเลเยอร์สำหรับแสดงลักษณะเฉพาะต่างๆ เช่น (ถนน, ชื่อเมือง)
  - 1.4) TERRAIN แสดงแผนที่ทางกายภาพที่แสดงคุณลักษณะระดับความสูงและพื้นน้ำ (ภูเขา, แม่น้ำ ฯลฯ)
- 2) มีการใช้งานที่ไม่ซับซ้อนง่ายต่อการนำไปพัฒนา
- 3) ข้อมูลที่ได้จากแผนที่มีประสิทธิภาพ เชื่อถือได้
- 4) มี Library และตัวอย่างการเขียนโปรแกรมที่จัดเป็นหมวดหมู่ที่ทำให้สะดวกในการพัฒนา
- 5) มีการออกแบบโครงสร้างของโดยใช้แนวคิดของการเขียนโปรแกรมเชิงวัตถุ (Object Oriented) ที่เข้าใจง่ายและมีประสิทธิภาพในการใช้งาน
- 6) สามารถใช้และเพิ่มข้อมูลเกี่ยวกับแผนที่ได้อย่างสะดวกจากหน้าเว็บได้ทุกที่
- 7) ข้อมูลมีความทันสมัย

### 2.2.4 การนำไปใช้งาน

- 1) ก่อนที่จะนำแผนที่ไปใช้งานจะต้องทำการสมัครเพื่อขอ API key กับทาง Google เนื่องจากทาง Google จะใช้ API ใช้การบริหารจัดการข้อมูลเพื่อไม่ให้เกิดปัญหาต่อการใช้งานในระบบโดยรวม
- 2) การเขียนโปรแกรมจะต้องมีลำดับการทำงานตามที่กำหนด ซึ่งการทำงานของ API จะมีการไหลแบบไม่เข้าจังหวะ (Asynchronous)
- 3) การใช้งานจะใช้งานผ่านการสร้างวัตถุ (Object) ตามการออกแบบคลาส (Class)
- 4) Google map API มีการทำงานที่หลากหลายสามารถนำมาเลือกใช้ตามความเหมาะสม ในที่นี้จะขอล่าเฉพาะ บริการ Geolocation และ Geocoding ซึ่งรายละเอียดของแต่ละบริการจะกล่าวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

    }
    // Browser doesn't support Geolocation
    else {
        browserSupportFlag = false;
        handleNoGeolocation(browserSupportFlag);
    }

    function handleNoGeolocation(errorFlag) {
        if (errorFlag == true) {
            alert("Geolocation service failed.");
            initialLocation = newyork;
        } else {
            alert("Your browser doesn't support geolocation. We've placed
you in Siberia.");
            initialLocation = siberia;
        }
        map.setCenter(initialLocation);
    }
}
}

```

### ระบุพารามิเตอร์เซ็นเซอร์

การใช้ Google Maps API จะต้องมีการใช้เซ็นเซอร์ (sensor) (เช่นระบุตำแหน่ง GPS) เพื่อตรวจสอบตำแหน่งของผู้ใช้ นี่เป็นสิ่งสำคัญอย่างยิ่งสำหรับอุปกรณ์มือถือ การใช้งานจะต้อง ใช้งานผ่านพารามิเตอร์ที่กำหนดการทำงานของเซ็นเซอร์ที่จำเป็นในแท็ก <script>

การใช้งานที่ตรวจสอบสถานที่ของผู้ใช้ผ่านเซ็นเซอร์จะต้องผ่านการเซ็นเซอร์ (sensor) = true

### โปรแกรมที่ 2.2 การกำหนดการใช้งานเซ็นเซอร์เพื่อหาตำแหน่งพิกัดทางภูมิศาสตร์

```

# Example using sensor when loading the Maps JavaScript API
<scripttype="text/javascript"
src="http://maps.googleapis.com/maps/api/js?sensor=true"></script>

```

หมายเหตุ ถ้าผู้ใช้งานไม่ต้องการใช้อุปกรณ์ตรวจจับพิกัดตำแหน่ง สามารถตั้งค่า พารามิเตอร์ให้เป็นเท็จ (false) ได้

## 2.4 Geocoding

### 2.3.1 ประโยชน์

เป็นบริการที่เกี่ยวกับการแปลงที่อยู่ ซึ่งเป็นรายละเอียด ที่อยู่ตามไปรษณีย์ (เช่น "1600 Amphitheatre Parkway, Mountain View, CA") ไปเป็นพิกัดทางภูมิศาสตร์ (เช่น 37.423021) ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละติจูดและลองจิจูด - 122.083739) และสามารถแปลงจากพิกัดทางภูมิศาสตร์ไปเป็นที่อยู่ได้ซึ่งจะเรียกว่า กระบวนการ Reverse Geocoding

### 2.3.2 การนำไปใช้งาน

#### 2.3.2.1 การร้องขอบริการของ Geocoding

สามารถใช้บริการของ Google Maps API geocoding ผ่านโปรแกรมที่ตนเองออกแบบผ่านทางวัตถุ (object) ของ google.maps.Geocoder ซึ่งวิธีการเริ่มต้นการขอใช้บริการ geocoding จะต้องทำผ่าน GeocodeRequest ซึ่งมีการบรรจุข้อมูล input และใช้วิธีการ callback ในการเรียกใช้งาน

วัตถุ GeocodeRequest ประกอบด้วยฟิลด์ข้อมูลที่เป็นตัวอักษรดังนี้

#### โปรแกรมที่ 2.3 วัตถุ GeocodeRequest

```
{
  address: string,
  latLng: LatLng,
  bounds: LatLngBounds,
  region: string
}
```

อธิบายแต่ละส่วนของข้อมูล ดังนี้

- 1) address (จำเป็น) คือข้อมูลที่อยู่ที่ต้องการแปลงเป็นข้อมูลประเภท string
- 2) latLng (จำเป็น) คือข้อมูล LatLng ที่ใกล้เคียงกับตำแหน่งปัจจุบันที่สุด
- 3) bounds (ไม่จำเป็น) คือขอบเขตการแสดงผลการทำงานของ geocode (สำหรับข้อมูลเพิ่มเติมโปรดดูที่Viewport Biasing )
- 4) region (ไม่จำเป็น) คือรหัสถิ่นฐานเป็นรหัสที่มีลักษณะเฉพาะตามรูปแบบของ IANA language region subtag ส่วนใหญ่แท้ของ map จะตามลำดับของ ccTLD ("top-level domain") โดยจะมี 2 ตัวอักษรในการบ่งบอกถึงถิ่นฐาน โดยสามารถศึกษาเพิ่มเติมได้ที่ Region Code Biasing
- 5) parameter bounds และ region มีผลต่อผลลัพธ์ที่เกิดขึ้น โดยจะส่งผลจะทำให้ได้รับข้อมูลที่ใดไม่ครบที่เกิดจากการทำงานของ Geocoder

หมายเหตุ สามารถใส่ ที่อยู่ (address) หรือ latLng เพื่อใช้ในการค้นหาซึ่ง จะเป็นกระบวนการ geocoder แบบ reverse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ (จำเป็น) ในที่นี้หมายถึง จะต้องมีการกำหนดองค์ประกอบเหล่านี้ไปในการร้องการคำ  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิ ใช้อ้างอิงข้อมูลเสมอ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2.2 การตอบสนองการร้องขอของ Geocoding

บริการ Geocoding ต้องใช้วิธีการ callback เพื่อดำเนินการโดยการดึงค่าของผล Geocoder ผ่านพารามิเตอร์ จำนวน 2 ค่าเพื่อเก็บผลและ statusCode ผลลัพธ์ที่ได้ GeocoderResults จะถูกเก็บในลักษณะของ array

วัตถุที่เก็บผลลัพธ์ของกระบวนการ Geocoding คือ GeocoderResults แสดงผลเป็นตามรูปแบบต่อไปนี้

#### โปรแกรมที่ 2.4 วัตถุ GeocoderResults

```
results[]: {
  types[]: string,
  formatted_address: string,
  address_components[]: {
    short_name: string,
    long_name: string,
    types[]: string
  },
  geometry: {
    location: LatLng,
    location_type: GeocoderLocationType
    viewport: LatLngBounds,
    bounds: LatLngBounds
  }
}
```

อธิบายแต่ละส่วนของข้อมูล ดังนี้

- 1) types[] เป็นการจัดเก็บข้อมูลแบบอาร์เรย์ (array) ซึ่งระบุชนิดของผลลัพธ์ ซึ่งมีการแบ่งหลายชนิด ตัวอย่างเช่น geocode ของ "Chicago" จะแสดงผลเป็นค่า type[] เป็น "locality" ซึ่งบ่งชี้ว่า "Chicago"
- 2) formatted\_address เป็นข้อมูลชนิดข้อความ (string) เป็นค่าของสถานที่ที่อยู่โดยส่วนมากมักจะเป็นค่าที่อยู่ตามไปรษณีย์ ซึ่งบางครั้งมีความแตกต่างกันในบางประเทศ ส่วนมากที่อยู่จะประกอบด้วยหนึ่งหรือหลายส่วนของข้อมูลประกอบกัน ตัวอย่างเช่นที่อยู่ "111 8th Avenue, New York, NY" แบ่งที่อยู่ออกเป็นส่วนประกอบย่อยๆดังนี้ "111 8th Avenue" (a street address), "นิวยอร์ก" (the city) และ "NY" (the US state) (สำหรับข้อมูลเพิ่มเติมเกี่ยวกับชนิดดูที่ชนิดด้านล่าง)
- 3) address\_component [] เป็นข้อมูลที่มีการจัดเก็บ แบบอาร์เรย์ (array) โดยจะทำ

เอกสารนี้เป็นเอกสารที่สงวนการเก็บส่วนย่อยๆของที่อยู่ตามที่อธิบายไว้ข้างต้น ญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีก4) geometry มีการบรรจุข้อมูลดังนี้: อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.1) location สถานที่ที่จะบรรจุ ค่าละติจูดและลองจิจูด หมายถึง เราจะส่งค่าสถานที่ LatLng object กลับซึ่งไม่ใช่ค่าที่เป็นข้อความ
- 4.2) location\_type เก็บข้อมูลเพิ่มเติมเกี่ยวกับการระบุสถานที่ ค่านี้เป็นปัจจุบัน
- google.maps.GeocoderLocationType.ROOFTOP แสดงข้อมูลผลลัพธ์ที่มีความแม่นยำมาก
  - google.maps.GeocoderLocationType.RANGE\_INTERPOLATED แสดงข้อมูลผลลัพธ์ที่มีการประมาณการซ้อนทับ (บนท้องถนนปกติ) ระหว่างจุดสองจุดได้อย่างแม่นยำ (เช่น แยกต่างๆ) ผลลัพธ์ของการซ้อนทับปกติแล้วจะส่งค่ากลับเมื่ออยู่บนชั้นตาดฟ้าที่ไม่ใช่อยู่บนถนน
  - google.maps.GeocoderLocationType.GEOMETRIC\_CENTER แสดงข้อมูลผลลัพธ์เป็นจุดศูนย์กลางทางของรูปเรขาคณิต เช่น เส้น (เช่นถนน) หรือรูปหลายเหลี่ยม (เขต)
  - google.maps.GeocoderLocationType.APPROXIMATE แสดงข้อมูลผลลัพธ์เป็นค่าการประมาณ
- 4.3) viewport เก็บคำแนะนำในการแสดงผลสำหรับการส่งค่าผลลัพธ์กลับ
- 4.4) bounds เก็บค่า LatLngBounds ซึ่งสามารถเก็บผลลัพธ์ที่ส่งกลับได้ทั้งหมด หมายถึง ขอบเขตเหล่านี้อาจไม่ตรงกับที่แนะนำใน viewport (ตัวอย่างเช่น San Francisco จะรวมหมู่เกาะฟาราลลอน (Farallon) ซึ่งเป็นส่วนหนึ่งของเมือง (city) แต่ไม่ควรจะส่งค่ากลับมาใน viewport.)
- ข้อมูลที่อยู่ของ Geocoder จะถูกส่งกลับตามการตั้งค่าภาษาของเบราว์เซอร์ที่ระบุไว้หรือภาษาที่ระบุเมื่อมีการโหลด API JavaScript มาใช้งานผ่านทางการใช้ languageparameter

### 2.2.2.3 Address Component Types

เป็นข้อมูลชนิดอาร์เรย์ (array) เพื่อระบุชนิดของส่วนประกอบที่อยู่โดยเฉพาะ ซึ่งที่อยู่ภายใน Geocoder อาจมีหลายประเภทชนิดโดยอาศัยการพิจารณา "แท็ก (tag)" ตัวอย่าง เช่น หลาย ๆ city จะติดแท็ก (tag) กับประเภท political andlocality

ชนิดต่อไปนี้จะได้รับการสนับสนุนและกลับได้โดยการ HTTP Geocoder

- 1) street\_address แสดงค่าที่อยู่ของถนน
- 2) route แสดงค่าชื่อของถนน (เช่น "US 101")
- 3) intersection แสดงค่าการซ้อนทับของข้อมูลขนาดใหญ่ ยกตัวอย่างเช่น ถนน 2 ถนน ซ้อนทับกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีจุดประสงค์เพื่อเผยแพร่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) political แสดงค่าของเมืองปกติจะแสดงเป็นรูปหลายเหลี่ยมของบาง การปกครอง (political)
  - 5) country แสดงค่าเมือง , และเป็นข้อมูลที่มีการส่งกลับที่เป็นลำดับใหญ่ที่สุดของ Geocoder
  - 6) administrative\_area\_level\_1 แสดงเอ็นทิตี (entity) ของเมืองลำดับแรก ภายใต้ระดับของประเทศ
  - 7) administrative\_area\_level\_2 แสดงเอ็นทิตีของเมืองลำดับที่ 2 ภายใต้ระดับของประเทศ
  - 8) administrative\_area\_level\_3 แสดงเอ็นทิตีของเมืองลำดับที่ 3 ภายใต้ระดับของประเทศ ชนิดนี้เป็นการแบ่งข้อมูลของเมืองที่มีขนาดเล็ก
  - 9) colloquial\_area แสดงเอ็นทิตีของทางเลือกอีกชื่อที่เป็นชื่อทั่วไปที่รู้จักกัน
  - 10) locality แสดงเอ็นทิตีของการรวมเมือง (city หรือ town)
  - 11) sublocality แสดงเอ็นทิตีของ civil ลำดับที่ 1 ภายใต้ระดับ locality
  - 12) neighborhood แสดงชื่อของเขต
  - 13) premise แสดงชื่อของสถานที่, ปกติจะเป็นตึกหรือองค์กร
  - 14) subpremise แสดงเอ็นทิตีของลำดับที่ 1 ภายใต้ระดับของชื่อสถานที่
  - 15) postal\_code คือ รหัสไปรษณีย์
  - 16) natural\_feature แสดงคุณลักษณะที่โดดเด่นทางธรรมชาติ
  - 17) airport แสดงชื่อสนามบิน
  - 18) park แสดงชื่อสวนสาธารณะ
- ข้างล่างนี้เป็น component เพิ่มเติม
- 1) post\_box แสดงตู้ไปรษณีย์
  - 2) street\_number แสดงหมายเลขถนนหลัก
  - 3) floor แสดงชั้นที่อยู่ของอาคาร
  - 4) room แสดงห้องของที่อยู่ในอาคาร

#### 2.2.2.4 สถานะของการร้องขอของ Geocoding

สถานะส่งค่ากลับมาจะมี 1 ค่าตามข้างล่างนี้

- 1) google.maps.GeocoderStatus.OK แสดงว่าการ geocode การร้องขอข้อมูลสำเร็จ
- 2) google.maps.GeocoderStatus.ZERO\_RESULTS แสดงว่าการ geocode สำเร็จ แต่ไม่มีการส่งค่าข้อมูลกลับ สิ่งนี้อาจเกิดขึ้นถ้า geocode ถูกให้ค่าด้วยข้อมูลที่อยู่ที่ไม่ใช่สิ่งที่มีอยู่จริงหรือข้อมูลละติจูดและลองจิจูด (latng) ในสถานที่ทางไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษามาก่อน ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามทำซ้ำโดยไม่ขออนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) google.maps.GeocoderStatus.OVER\_QUERY\_LIMIT แสดงว่ามีการร้องขอข้อมูลเกินที่จำกัดไว้
- 4) google.maps.GeocoderStatus.REQUEST\_DENIED แสดงว่าการร้องขอถูกละเลยด้วยเหตุผลบางอย่าง
- 5) google.maps.GeocoderStatus.INVALID\_REQUEST แสดงว่ามีการกรอกข้อมูลที่ไม่ครบ (address หรือ latLng)

### 2.2.2.5 Reverse Geocoding (การค้นหา address)

Geocoding หมายถึง การแปลที่อยู่ตำแหน่งบนแผนที่ แต่กระบวนการนี้จะทำตรงกันข้ามคือ จะทำการแปลงตำแหน่งบนแผนที่ไปเป็นที่อยู่

Geocoder สนับสนุนการ Reverse geocoding โดยรูปแบบที่ได้จะถูกค้นด้วยเครื่องหมายจุลภาคระหว่างละติจูดและลองจิจูดของแต่ละของคูในพารามิเตอร์ชื่อว่า latLng

ตัวอย่างต่อไปนี้ geocodes ค่าละติจูด / ลองจิจูดและศูนย์กลางแผนที่ที่ ซึ่งจะแสดงในหน้าต่างข้อมูลที่มีที่อยู่ในรูปแบบของการเขียนที่อยู่ เราส่งค่ากลับมาเป็นค่า 2 ค่า ซึ่งมีความเฉพาะเจาะจงน้อยกว่าครั้งแรก (ในกรณีนี้ ชื่อ เขต)

### โปรแกรมที่ 2.5 แสดงการดึงข้อมูลด้วยการ geocodes

```

var geocoder;
var map;
var infowindow = new google.maps.InfoWindow();
var marker;
function initialize() {
  geocoder = new google.maps.Geocoder();
  var latlng = new google.maps.LatLng(40.730885, -73.997383);
  var mapOptions = {
    zoom: 8,
    center: latlng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  }
  map = new google.maps.Map(document.getElementById("map_canvas"),
  mapOptions);
}

function codeLatLng() {
  var input = document.getElementById("latlng").value;
  var latlngStr = input.split(",",2);
  var lat = parseFloat(latlngStr[0]);
  var lng = parseFloat(latlngStr[1]);
  var latlng = new google.maps.LatLng(lat, lng);
  geocoder.geocode({'latLng': latlng}, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
      if (results[1]) {
        map.setZoom(11);
        marker = new google.maps.Marker({

```

```

        position: latlng,
        map: map
    });
    infowindow.setContent(results[1].formatted_address);
    infowindow.open(map, marker);
}
} else {
    alert("Geocoder failed due to: " + status);
}
});
}
}

```

ผลลัพธ์ของกระบวนการ Reverse Geocoding ทั้งหมดได้แสดงไว้ด้านล่าง

## โปรแกรมที่ 2.6 ผลลัพธ์การ Geocode

```

results[0].formatted_address: "275-291 Bedford Ave, Brooklyn, NY
11211, USA",
results[1].formatted_address: "Williamsburg, NY, USA",
results[2].formatted_address: "New York 11211, USA",
results[3].formatted_address: "Kings, New York, USA",
results[4].formatted_address: "Brooklyn, New York, USA",
results[5].formatted_address: "New York, New York, USA",
results[6].formatted_address: "New York, USA",
results[7].formatted_address: "United States"

```

## 2.5 Direction

### 2.5.1 ประโยชน์

เป็นบริการที่ใช้ในการคำนวณทิศทาง (ซึ่งมีวิธีการของการเดินทางหลากหลาย) โดยใช้วัตถุ DirectionsService วัตถุนี้จะสื่อสารกับบริการ Google Maps API การบริการทิศทางที่ได้รับมาจากการร้องขอและส่งค่าคำนวณกลับมา สามารถรับข้อมูลหรืออาจจะจัดการการรับข้อมูลด้วยตัวเองผ่านวัตถุ DirectionsRenderer

ทิศทางอาจจะระบุต้นกำเนิดและปลายทางในรูปแบบข้อความ (เช่น "Chicago, IL" หรือ "Darwin, NSW, Australia") หรือเป็นค่าละติจูด ลองจิจูด (LatLng values) การบริการทิศทางจะได้รับค่าหลายเส้นทางซึ่งจะใช้จุดในการจัดการ ทิศทางจะแสดงเป็นเส้น วาดบนแผนที่หรือชุดของคำอธิบายข้อความภายในองค์ประกอบ <div> (เช่น "เลี้ยวขวาไปตามทางลาดสะพานวิลเลียม")

### 2.5.2 การนำไปใช้งาน

#### 2.5.2.1 การร้องขอการใช้งาน Direction

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการสร้างวัตถุชนิด DirectionsService และเรียก DirectionsService.route() เพื่อเริ่มต้นการร้องขอไปยังผู้ให้บริการทิศทางผ่าน DirectionsRequest

### โปรแกรมที่ 2.7 วัตถุ DirectionsRequest

```
{
  origin: LatLng | String,
  destination: LatLng | String,
  travelMode: TravelMode,
  transitOptions: TransitOptions,
  unitSystem: UnitSystem,
  waypoints[]: DirectionsWaypoint,
  optimizeWaypoints: Boolean,
  provideRouteAlternatives: Boolean,
  avoidHighways: Boolean,
  avoidTolls: Boolean
  region: String
}
```

อธิบายแต่ละส่วนของข้อมูลได้ดังนี้:

- 1) Origin (ค่าที่จำเป็นในการคำนวณ) ระบุตำแหน่งเริ่มต้นจากการที่จะคำนวณทิศทาง ค่านี้อาจจะถูกระบุเป็น String (เช่น "Chicago, IL") หรือเป็นค่า LatLng
- 2) Destination (จำเป็น) ระบุตำแหน่งสิ้นสุดที่การคำนวณทิศทาง ค่านี้อาจจะถูกระบุเป็น String (เช่น "Chicago, IL") หรือเป็นค่า LatLng
- 3) travelMode (จำเป็น) ระบุโหมดของการขนส่งที่จะใช้เมื่อมีการคำนวณทิศทาง ค่าที่ถูกต้องที่ระบุไว้ในโหมดการท่องเที่ยวด้านล่าง
- 4) transitOptions (ไม่จำเป็น) ระบุค่าที่ใช้เฉพาะการร้องขอที่เป็น travelMode google.maps.TravelMode.TRANSIT ค่าที่ถูกต้องจะมีคำอธิบายในตัวเลือกการขนส่งด้านล่าง
- 5) unitSystem (ไม่จำเป็น) ระบุว่าระบบหน่วยที่จะใช้เมื่อการแสดงผล ค่าที่ถูกต้องมีการระบุไว้ในระบบหน่วยด้านล่าง
- 6) avoidHighways (ไม่จำเป็น) เมื่อตั้งค่าเป็นจริงจะแสดงเส้นทางการคำนวณ ที่หลีกเลี่ยงการทางหลวงหลักถ้าเป็นไปได้

เอกสารนี้เป็นเอกสารที่ 7) avoidTolls (ไม่จำเป็น) เมื่อตั้งค่าเป็นจริงจะแสดงเส้นทางการคำนวณ ที่หลีกเลี่ยงทาง  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) region (ไม่จำเป็น) ระบุรหัสพื้นที่ตามที่ระบุไว้ ccTLD ("โดเมนระดับบนสุด") เป็นตัวอักษร 2 ตัว

หมายเหตุ (จำเป็น) ในที่นี้หมายถึง จะต้องมีการกำหนดองค์ประกอบเหล่านี้ไปในการร้องขอข้อมูลเสมอ

ตัวอย่าง ของการ DirectionsRequest ด้านล่าง:

### โปรแกรมที่ 2.8 แสดงการร้องขอการใช้งาน Direction

```
{
  origin: "Chicago, IL",
  destination: "Los Angeles, CA",
  waypoints: [
    {
      location: "Joplin, MO",
      stopover: false
    }, {
      location: "Oklahoma City, OK",
      stopover: true
    }
  ],
  provideRouteAlternatives: false,
  travelMode: TravelMode.DRIVING,
  unitSystem: UnitSystem.IMPERIAL
}
```

การเลือกโหมดการเดินทางเพื่อใช้ในการพัฒนาซึ่งจำเป็นต้องระบุโหมดการเดินทาง

- 1) google.maps.TravelMode.DRIVING (Default) ระบุเส้นทางการขับรถมาตรฐานโดยการใช้เครือข่ายถนน
- 2) google.maps.TravelMode.BICYCLING ทิศทางจักรยานผ่านเส้นทางจักรยานและถนนที่ต้องการ
- 3) google.maps.TravelMode.TRANSIT ทิศทางผ่านเส้นทางการขนส่งสาธารณะ
- 4) google.maps.TravelMode.WALKING ร้องขอเส้นทางเดินเท้าผ่านเส้นทางเดินเท้าและทางเท้า
- 5) ทิศทางที่ไม่สามารถตอบสนองได้จะส่ง DirectionsStatus = "ZERO\_RESULTS" กลับมา

#### 2.5.2.2 ผลลัพธ์ของการทำงาน Direction

เอกสารนี้เป็นเอกสารที่เริ่มขอเส้นทางไปยัง DirectionsService และได้ผลลัพธ์ กลับเป็น DirectionsResult และการคำนวณระยะทางและเวลาเดินทาง และต้องอ้างอิงถึงค่าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างต่อไปนี้จะคำนวณเส้นทางระหว่างสองตำแหน่งบนถนน Route 66 จุดเริ่มต้นและปลายทางที่กำหนดให้ "จุดเริ่มเริ่มต้น" และค่า "ปลายทาง" แสดงในรายการแบบเลื่อนลง โดย DirectionsRenderer จะจัดการการแสดงผลของเส้นระหว่างสถานที่ที่ระบุและตำแหน่งของเครื่องหมายที่จุดเริ่มต้นของปลายทางและจุดใด ๆ ถ้ามี

### โปรแกรมที่ 2.9 แสดงการใช้งาน DirectionsRenderer

```
var directionsDisplay;
var directionsService = new google.maps.DirectionsService();
var map;

function initialize() {
  directionsDisplay = new google.maps.DirectionsRenderer();
  var chicago = new google.maps.LatLng(41.850033, -87.6500523);
  var mapOptions = {
    zoom:7,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    center: chicago
  }
  map = new google.maps.Map(document.getElementById("map_canvas"),
  mapOptions);
  directionsDisplay.setMap(map);
}

function calcRoute() {
  var start = document.getElementById("start").value;
  var end = document.getElementById("end").value;
  var request = {
    origin:start,
    destination:end,
    travelMode: google.maps.TravelMode.DRIVING
  };
  directionsService.route(request, function(result, status) {
    if (status == google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(result);
    }
  });
}
```

ใน HTML body:

### โปรแกรมที่ 2.10 แสดงผลลัพธ์การคำนวณเส้นทาง

```
<div>
<strong>Start: </strong>
<select id="start" onchange="calcRoute();">
  <option value="chicago, il">Chicago</option>
  <option value="st louis, mo">St Louis</option>
```

```

<option value="joplin, mo">Joplin, MO</option>
<option value="oklahoma city, ok">Oklahoma City</option>
<option value="amarillo, tx">Amarillo</option>
<option value="gallup, nm">Gallup, NM</option>
<option value="flagstaff, az">Flagstaff, AZ</option>
<option value="winona, az">Winona</option>
<option value="kingman, az">Kingman</option>
<option value="barstow, ca">Barstow</option>
<option value="san bernardino, ca">San Bernardino</option>
<option value="los angeles, ca">Los Angeles</option>
</select>
<strong>End: </strong>
<select id="end" onchange="calcRoute();" >
  <option value="chicago, il">Chicago</option>
  <option value="st louis, mo">St Louis</option>
  <option value="joplin, mo">Joplin, MO</option>
  <option value="oklahoma city, ok">Oklahoma City</option>
  <option value="amarillo, tx">Amarillo</option>
  <option value="gallup, nm">Gallup, NM</option>
  <option value="flagstaff, az">Flagstaff, AZ</option>
  <option value="winona, az">Winona</option>
  <option value="kingman, az">Kingman</option>
  <option value="barstow, ca">Barstow</option>
  <option value="san bernardino, ca">San Bernardino</option>
  <option value="los angeles, ca">Los Angeles</option>
</select>
</div>

```

### 2.5.2.3 สถานะของการร้องขอของ Direction

- 1) OK แสดงการตอบสนองที่ถูกต้องมี DirectionsResult
- 2) NOT\_FOUND บ่งชี้อย่างน้อยหนึ่งในสถานที่ที่ระบุไว้ในคำขอของแหล่งกำเนิดของปลายทางหรือจุดที่ไม่สามารถ geocoded
- 3) ZERO\_RESULTS ระบุไม่มีเส้นทางที่จะพบระหว่างต้นทางและปลายทาง
- 4) MAX\_WAYPOINTS\_EXCEEDED บ่งชี้ว่า DirectionsWaypoints สูงสุดในการใช้งาน โดยกำหนดจุดสูงสุดที่อนุญาตให้คือ 8 รวมกับจุดจุดเริ่มต้นและปลายทาง สำหรับลูกค้าทางธุรกิจจะได้รับอนุญาต 23 จุดรวมกับจุดกำเนิดและจุดปลายทาง
- 5) INVALID\_REQUEST บ่งชี้ว่า DirectionsRequest ไม่ถูกต้อง สาเหตุที่พบบ่อยที่สุดของรหัสข้อผิดพลาดนี้เกิดจากการร้องขอที่ขาดหายไป
- 6) OVER\_QUERY\_LIMIT แสดงหน้าเว็บมีการโหลดข้อมูลมากเกินไปที่ได้รับอนุญาต
- 7) REQUEST\_DENIED แสดงหน้าเว็บที่ไม่ได้รับอนุญาตให้ใช้บริการทิศทาง
- 8) UNKNOWN\_ERROR บ่งชี้ทิศทางการร้องขอไม่สามารถดำเนินการเนื่องจากเกิดข้อผิดพลาดเซิร์ฟเวอร์ การร้องขออาจจะประสบความสำเร็จถ้าคุณลองอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาใช้เพื่อวัตถุประสงค์ในการเรียนรู้เท่านั้น การนำเอกสารนี้ไปใช้เพื่อวัตถุประสงค์อื่นโดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การร้องขออาจจะมีให้ดาวน์โหลดฟรี แต่การนำเอกสารนี้ไปใช้เพื่อวัตถุประสงค์อื่นโดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

## 2.6 Google Text To Speech(tts)

ยังไม่มีการทำออกมาในรูปแบบของ เอกสาร (document) ที่ชัดเจนแต่มีการนำมาใช้ใน google translate ที่ใช้ในการออกเสียงคำหรือประโยคที่พิมพ์ลงไป แต่ในการเขียนโปรแกรมเราจะนำเฉพาะบางส่วนของที่เราต้องการเท่านั้นมาใช้ โดยนำโปรแกรมส่วนที่ใช้ในการป้อน text และประมวลผลออกมาเป็นเสียง

รองรับการแปลงข้อความเป็นเสียงได้หลากหลายภาษาและยังมีประสิทธิภาพในการใช้งาน ซึ่งในอนาคตทาง google จะมีการทำเป็นเอกสารออกมาให้นำไปพัฒนาต่อได้

### โปรแกรมที่ 2.11 แสดงการแปลงจากข้อความเป็นเสียง

```
function read(){
    var au=document.createElement("audio");
    var txt= encodeURI(document.getElementById("txtTTS").value);
    var lang="th"; //ไทย th อังกฤษ en
    au.setAttribute("src","http://translate.google.com/translate_
tts?ie=UTF-8&q="+txt+"&tl="+lang);
    au.play();
}
```

อธิบายการทำงาน

ที่ [http://translate.google.com/translate\\_tts?ie=UTF-8&q="+txt+"&tl="+lang](http://translate.google.com/translate_tts?ie=UTF-8&q=)

ข้อความที่ต้องการให้แปลงเป็นเสียงที่ q= “ ข้อความที่ต้องการแปลงเป็นเสียง ”

กำหนดภาษาในการแปลงที่เป็นภาษาเดียวกับข้อความที่ tl = “ภาษาที่ใช้”

## 2.7 Web Server

Web server คือ โปรแกรมที่อยู่และที่ทำงานบนเครื่องฝั่ง Server (Host) ทำหน้าที่ในการรับคำสั่งจากการร้องขอของฝั่ง Client (โดยผ่านทาง เบราร์เซออร์) และประมวลผลการทำงานจากการร้องขอดังกล่าว แล้วส่งข้อมูลกลับไปยังเครื่องของ Client ที่ร้องขอ และจะต้องมีตัวแปรภาษานั้นๆ อีกทีหนึ่ง ดังนั้นถ้าเราต้องการให้เครื่องของเราสามารถ Run CGI Script ต่างๆนั้นได้เช่น ASP, PHP และ Perl เป็นต้น ได้เราจะต้องจำลองเครื่องคอมพิวเตอร์ของเราให้เป็น Web Server และลงตัวแปรภาษาที่เราต้องการเขียนนั้นเสียก่อน

### 2.7.1 Dynamic DNS

ในความเป็นจริงแล้วหากเราเชื่อมต่ออินเทอร์เน็ตแบบ ADSL นั้นเราจะได้ Public IP ที่แตกต่างกันเนื่องจาก จะมีการหมุนเวียน IP ไปให้ผู้อื่นใช้งานหลังจากที่เลิกการเชื่อมต่อแล้ว

เพราะฉะนั้นคงจะเป็นเรื่องยากที่ใครจะเข้ามาดูเว็บเราได้ตลอดเวลา เพราะหากเชื่อมต่อเน็ตใหม่ก็จะได้ IP Address ใหม่ด้วย ระบบ Dynamic DNS เป็นระบบมาช่วยในการอัปเดตข้อมูล IP เมื่อมีการเปลี่ยนแปลง

## 2.8 Apperve

AppServ คือ โปรแกรม ใช้ในการสร้างเซฟเวอร์บนพีซี ที่รวบรวมเอา Open Source Software หลากๆ Software มารวมกัน เมื่อติดตั้งในเครื่องคอมพิวเตอร์ (แบบ local) แล้วสามารถจำลองเป็น Server หรือ Hosting เสมือน ในเครื่องคอมพิวเตอร์เพื่อทดลองสร้างเว็บไซต์ก่อนอัปโหลดขึ้น Server หรือ Hosting จริง

โดยมี Package หลักๆ ดังนี้

1. Apache Web Server คือโปรแกรมที่ทำหน้าที่เป็น Web Server
2. MySQL Database คือโปรแกรมที่ทำหน้าที่เป็น Database Server
3. PHP Script Language คือภาษา PHP ที่เอาไว้เขียนโปรแกรมเกี่ยวกับเว็บ
4. phpMyAdmin คือตัวควบคุม MySQL Database ผ่านเว็บไซต์

ซึ่งโดยทั่วไปแล้วหากเราต้องการติดตั้ง Apache Web Server และเครื่องคอมพิวเตอร์เราสามารถใช้งาน PHP ได้ และต้องใช้ฐานข้อมูล MySQL ด้วย เมื่อลงโปรแกรมสมบูรณ์แล้วเครื่องคอมพิวเตอร์เราก็เปรียบเสมือน Web Server

## 2.9 SQL

SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรมเพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) ซึ่งเราสามารถใช้คำสั่ง sql กับฐานข้อมูลชนิดใดก็ได้และใช้คำสั่งนี้กับการสั่งงานผ่านระบบฐานข้อมูล ที่แตกต่างกันจะได้ผลลัพธ์ที่ได้จะเหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูลชนิดใดก็ได้โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้ยังเป็นภาษาแล้วก็ยังโปรแกรม ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง ซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

- 1) Select query ใช้สำหรับดึงข้อมูลที่ต้องการ

เอกสารนี้เป็นเอกสารที่ 2) Update query ใช้สำหรับแก้ไขข้อมูล

ไม่ว่ากรณีใดๆทั้งสิ้น 3) Insert query ใช้สำหรับการเพิ่มข้อมูล

4) Delete query ใช้สำหรับลบข้อมูลออกไป

ปัจจุบันมีซอฟต์แวร์ระบบจัดการฐานข้อมูล (DBMS) ที่สนับสนุนการใช้คำสั่ง SQL เช่น Oracle, DB2, MS-SQL, MS-Access นอกจากนี้ภาษา SQL ถูกนำมาใช้เขียนร่วมกับโปรแกรมภาษาต่างๆ เช่น ภาษา C/C++, VisualBasic และ Java

### 2.9.1 ประโยชน์ของภาษา SQL

- 1) สร้างฐานข้อมูลและตาราง
- 2) สนับสนุนการจัดการฐานข้อมูล ซึ่งประกอบด้วย การเพิ่ม การปรับปรุง และการลบข้อมูล
- 3) สนับสนุนการเรียกใช้หรือ ค้นหาข้อมูล

### 2.9.2 ประเภทของคำสั่งภาษา SQL

1) ภาษานิยามข้อมูล (Data Definition Language : DDL) เป็นคำสั่งที่ใช้ในการสร้างฐานข้อมูล กำหนดโครงสร้างข้อมูลว่ามี Attribute ใด ชนิดของข้อมูลรวมทั้งการเปลี่ยนแปลงตาราง และการสร้างดัชนี คำสั่ง : CREATE, DROP, ALTER

2) ภาษาจัดการข้อมูล (Data Manipulation Language : DML) เป็นคำสั่งที่ใช้ในการเรียกใช้ เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลในตาราง คำสั่ง : SELECT, INSERT, UPDATE, DELETE

3) ภาษาควบคุมข้อมูล (Data Control Language : DCL) เป็นคำสั่งที่ใช้ในการกำหนดสิทธิ์การอนุญาต หรือ ยกเลิก การเข้าถึงฐานข้อมูล เพื่อป้องกันความปลอดภัยของฐานข้อมูล คำสั่ง : GRANT, REVOKE

## 2.10 MySQL

MySQL (มายเอสคิวแอล) คือ ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) โดยใช้ภาษา SQL แม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจ

### 2.10.1 รุ่นของผลิตภัณฑ์

รุ่นของผลิตภัณฑ์นั้นแบ่งออกมาได้ 3 สายการผลิต ได้แก่ เวอร์ชันใช้ฟรี เวอร์ชันการค้า และเวอร์ชันที่สนับสนุนกับผลิตภัณฑ์ SAP (MAX DB) ความแตกต่างคือเวอร์ชันคอมมิวนิตี้นั้นสามารถนำไปใช้งานได้ฟรีแต่ขาดการสนับสนุนหรือการช่วยเหลือเมื่อมีปัญหาเกิดขึ้น, เวอร์ชันที่เป็นคอมเมอร์เชียลนั้นให้บริการด้านความสนับสนุนเมื่อมีปัญหา (ค่าบริการ) สรุปคร่าวๆ ประเภทดาต้าเบสที่ให้เลือกใช้ดังนี้

- 1) MySQL Enterprise

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุผลบางเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) MySQL Cluster
- 3) MySQL Embedded
- 4) MySQL Community (เวอร์ชันopensource )

#### 2.10.2 ประเภทการจัดเก็บข้อมูล (Database Storage Engine) ที่สนับสนุน

- 1) MyISAM ค่าปกติ (default)
- 2) InnoDB สนับสนุนการทำ ทรานแซคชั่น (transaction) แบบ ACID
- 3) Memory การจัดเก็บในหน่วยความจำ ใช้เป็นตารางชั่วคราวเพื่อความรวดเร็ว เนื่องจากเก็บไว้ในหน่วยความจำ ทำให้มีความเร็วในการทำงานสูงมาก
- 4) Merge เป็นการรวม Table หลาย ๆ ตัวให้แสดงผล หรือแก้ไข เสมือนเป็นข้อมูลจาก Table เดียว
- 5) Archive เหมาะสำหรับการจัดเก็บข้อมูลพวก log file, ข้อมูลที่ไม่ต้องมีการ คิวรี (query) หรือใช้บ่อยๆ เช่น log file เพื่อประโยชน์ในการตรวจสอบย้อนหลัง (Security Audit Information)
- 6) Federated สำหรับการจัดเก็บแบบปลายทาง (remote server) แทนที่จะเป็นการจัดเก็บแบบ local เหมือนการจัดเก็บ (Storage) แบบอื่นๆ
- 7) NDB สำหรับการจัดเก็บแบบ คลัสเตอร์ (cluster)
- 8) CSV เก็บข้อมูลจาก Text ไฟล์โดยอาศัยเครื่องหมาย คอมา (comma) เป็นตัวแบ่งฟิลด์
- 9) Blackhole

#### 2.10.3 ชนิดของข้อมูลที่สนับสนุน

- 1) ชนิดข้อมูลที่ MySQL สนับสนุนแบ่งเป็น 3 ประเภทหลักคือ
- 2) ชนิดข้อมูลที่เป็นตัวเลข  
BIT (มีใช้ได้กับ MyISAM, InnoDB, Memory), TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, Float
- 3) ชนิดข้อมูลที่เกี่ยวข้องกับวันที่และเวลา  
DATETIME, DATE, TIMESTAMP, TIME, YEAR
- 4) ชนิดข้อมูลที่เกี่ยวข้องกับตัวอักษร  
CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM, SET

#### 2.10.4 การนำไปใช้งาน

MySQL เป็นที่นิยมใช้กันมากสำหรับฐานข้อมูลสำหรับเว็บไซต์ นิยมใช้งานร่วมกับ ภาษา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า โปรแกรม PHP โปรแกรมช่วยในการจัดการฐานข้อมูล และ ทำงานกับฐานข้อมูลสามารถใช้โปรแกรม ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมี เหตุผลเบื้องหลังเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ command-line เพื่อจัดการฐานข้อมูล (โดยใช้คำสั่ง mysql และ mysqladmin เป็นต้น) หรือจะดาวน์โหลดโปรแกรมจัดการฐานข้อมูลแบบ GUI จากเว็บไซต์ของ MySQL

### 2.10.5 ส่วนเชื่อมต่อกับภาษาการพัฒนาอื่น (database connector)

มีส่วนติดต่อ (interface) เพื่อเชื่อมต่อกับภาษาในการพัฒนา อื่นๆ เพื่อให้เข้าถึงฟังก์ชันการทำงานกับฐานข้อมูล MySQL ได้เช่น ODBC (Open Database Connector) อันเป็นมาตรฐานกลางที่กำหนดมาเพื่อให้ใช้เป็นสะพานในการเชื่อมต่อกับโปรแกรมหรือระบบอื่นๆ เช่น MyODBC อันเป็นไดรเวอร์เพื่อใช้สำหรับการเชื่อมต่อในระบบปฏิบัติการวินโดวส์, JDBC คลาสส่วนเชื่อมต่อสำหรับ Java เพื่อใช้ในการติดต่อกับ MySQL และมี API (Application Programming Interface) ต่างๆมีให้เลือกใช้มากมายในการที่เข้าถึง MySQL โดยไม่ขึ้นอยู่กับภาษาการพัฒนาใดภาษาหนึ่ง

นอกเหนือจาก ตัวเชื่อมต่อกับภาษาอื่น (Connector) ที่ได้กล่าวมาแล้ว ยังมี API ที่สนับสนุนในขณะนี้คือ

- 1) DBI สำหรับการเชื่อมต่อกับ ภาษา perl
- 2) Ruby สำหรับการเชื่อมต่อกับ ภาษา ruby
- 3) Python สำหรับการเชื่อมต่อกับภาษา python
- 4) .NET สำหรับการเชื่อมกับภาษา .NET framework
- 5) MySQL++ สำหรับเชื่อมต่อกับภาษา C++
- 6) Ch สำหรับการเชื่อมต่อกับ Ch (C/C++ interpreter)
- 7) PHP สำหรับการเชื่อมต่อกับภาษาPHP

ยังมีโปรแกรมอีกตัว เป็นโปรแกรมบริหารพัฒนาโดยผู้อื่น ซึ่งใช้กันอย่างแพร่หลายและนิยมกันเขียนในภาษาพีเอชพี เป็นโปรแกรมเว็บแอปพลิเคชัน ชื่อ phpMyAdmin

## 2.11 PHP

พีเอชพี (PHP) ย่อมาจาก PHP Hypertext Preprocessor คือ ภาษาคอมพิวเตอร์ ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะ โอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจาก ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีความรวดเร็วในการโต้ตอบ

ภาษาพีเอชพี จะเป็นส่วนประกอบภายในเว็บเพจ โดยคำสั่งจะปรากฏระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างควบคุมของ PHP จะมีความคล้ายคลึงกับ C/C++ มาก เช่น if , for , switch และ มีบางส่วนที่คล้าย Perl สามารถกำหนดตัวแปรโดยไม่ต้องกำหนดชนิดของตัวแปรว่าจะเป็น int, float และ boolean เป็นต้น

### 2.11.1 คุณสมบัติ

การแสดงผลของพีเอชพี จะปรากฏในลักษณะ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะ โคเลนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ ความสามารถการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจาก ดาต้าเบส ความสามารถจัดการกับ คุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะ CGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่านพีเอชพี พาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเซิร์ฟเวอร์หรือเบราว์เซอร์ ซึ่งมีลักษณะเหมือนกับ Cron (ใน ยูนิกซ์หรือลินุกซ์) หรือ Task Scheduler (ใน วินโดวส์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบ Simple text processing tasks ได้

การแสดงผลของพีเอชพี ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผล HTML แต่ยังสามารถสร้าง XHTML หรือ XML ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลัก PDF แฟลช (โดยใช้ libswf และ Ming) พีเอชพีมีความสามารถอย่างมากในการทำงานเป็นประมวลผลข้อความ จาก POSIX Extended หรือ รูปแบบ Perl ทั่วไป เพื่อแปลงเป็นเอกสาร XML ในการแปลงและเข้าสู่เอกสาร XML เรารองรับมาตรฐาน SAX และ DOM สามารถใช้รูปแบบ XSLT ของเราเพื่อแปลงเอกสาร XML

เมื่อใช้พีเอชพีในการทำ อีคอมเมิร์ซ สามารถทำงานร่วมกับโปรแกรมอื่น เช่น Cybercash payment, CyberMUT, VeriSign Payflow Pro และ C CVS functions เพื่อใช้ในการสร้างโปรแกรมทำธุรกรรมทางการเงิน

### 2.11.2 การรองรับพีเอชพี

คำสั่งของพีเอชพี สามารถสร้างผ่านทางโปรแกรมแก้ไขข้อความทั่วไป เช่น Notepad หรือ vi ซึ่งทำให้การทำงานพีเอชพีสามารถทำงานได้ในระบบปฏิบัติการหลักเกือบทั้งหมด โดยเมื่อเขียนคำสั่งแล้วนำมาประมวลผลใน Apache, Microsoft Internet Information Services (IIS) , Personal Web Server, Netscape และ iPlanet servers, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd, และอื่นๆ อีกมากมาย. สำหรับส่วนหลักของ PHP ยังมี Module ในการรองรับ CGI มาตรฐาน ซึ่ง PHP สามารถทำงานเป็นตัวประมวลผล CGI ได้ซึ่ง PHPคุณสมบัติในการเลือก ระบบปฏิบัติการ และ เว็บเซิร์ฟเวอร์ นอกจากนี้ยังสามารถใช้สร้าง โปรแกรม โครงสร้าง สร้างโปรแกรมเชิงวัตถุ (OOP) หรือสร้างโปรแกรมที่รวมทั้งสองอย่างเข้าด้วยกัน แม้ว่า

ความสามารถของคำสั่ง OOP มาตรฐานในเวอร์ชันนี้ยังไม่สมบูรณ์ แต่ตัวไลบรารีทั้งหลายของโปรแกรม และตัวโปรแกรมประยุกต์ (รวมถึง PEAR library) ได้ถูกเขียนขึ้นโดยใช้รูปแบบการเขียนแบบ OOP

พีเอชพีสามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับ ได้แก่ Oracle ,dBase, PostgreSQL, IBM, DB2, MySQL, Informix, ODBC โครงสร้างของฐานข้อมูลแบบ DBX ซึ่งทำให้พีเอชพีใช้กับฐานข้อมูลอะไรก็ได้ที่รองรับรูปแบบนี้แล้ว ะPHPยังรองรับ การทำงานของ ODBC(Open Database Connection) ซึ่งเป็นมาตรฐานการเชื่อมต่อฐานข้อมูลที่ใช้กันแพร่หลาย

พีเอชพียังสามารถรองรับการสื่อสารกับการบริการใน โพรโทคอล ต่างๆ เช่น LDAP IMAP SNMP NNTP POP3 HTTP COM (บนวินโดวส์)

## 2.12 HTML

HTML คือ ภาษาที่ใช้ในการเขียนเว็บเพจ ย่อมาจากคำว่า Hypertext Markup Language โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิงค์ Markup หมายถึง วิธีในการเขียนข้อความ language หมายถึงภาษา ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ในการเขียนข้อความ ลงบนเอกสารที่ต่างก็เชื่อมถึงกันใน cyberspace ผ่าน Hyperlink นั้นเอง

HTML เริ่มขึ้นเมื่อ ปี 1990 เพื่อตอบสนองความต้องการในการสื่อสารแลกเปลี่ยนข้อมูลกันของนักวิทยาศาสตร์ระหว่างสถาบันและมหาวิทยาลัยต่างๆทั่วโลก โดย Tim Berners-Lee นักพัฒนาของ CERN ได้พัฒนาภาษาที่มีรากฐานมาจาก SGML ซึ่งเป็นภาษาที่ซับซ้อนและยากต่อการเรียนรู้ จนมาเป็นภาษาที่ใช้ได้ง่ายและสะดวกในการแลกเปลี่ยนเอกสารทางวิทยาศาสตร์ผ่านการเชื่อมโยงกันด้วยลิงค์ในหน้าเอกสาร เมื่อ World Wide Web เป็นที่แพร่หลาย HTML จึงถูกนำมาใช้จนเกิดการแพร่หลายออกไปยังทั่วโลก จากความง่ายตายในการใช้งาน

## 2.13 HTML5

HTML5 เป็นรุ่นภาษามาร์กอัปสำหรับเว็ลด์ไวด์เว็บรุ่นต่อไปของ HTML โดยตราฟต์แรกได้ปรากฏออกมาเมื่อ 22 มกราคม พ.ศ. 2551 โดย HTML5 นี้มีการกล่าวถึงรูปแบบสองแบบที่รวมกันคือ แบบดั้งเดิม HTML (text/html) ที่รู้จักในชื่อ HTML5 และแบบที่สองคือ XHTML ที่รู้จักในชื่อ XHTML5 โดยทั้งสองระบบได้มีการพัฒนาในเวลาพร้อมกัน และเมื่อสิ้นสุดปี 2552 ทางทีมงานของ XHTML จะมาร่วมพัฒนา HTML5 การเปลี่ยนแปลงสำคัญใน HTML5 คือ element ใหม่ๆ ที่ใช้งานเราจะ

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อการเรียนการสอนของศูนย์พัฒนาระบบงานวิจัย สำนักงานการวิจัยแห่งชาติ  
ไม่ว่ากรณีใดๆ ห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐานใหม่จะมีคุณลักษณะเด่นที่สำคัญได้แก่

- 1) การใช้งานวิดีโอ
- 2) การแสดงตำแหน่งทางภูมิศาสตร์
- 3) การเก็บไฟล์ในลักษณะออฟไลน์
- 4) การแสดงกราฟิกส์
- 5) input types แบบใหม่ เช่น search, number, range, color, tel, url, email, date, month, week, time, datetime, datetime-local

โดยคุณสมบัติเด่นหลายอย่างไม่จำเป็นต้องติดตั้งซอฟต์แวร์เพิ่ม เช่น เกียรติส พลซ หรือ ซิลเวอร์ไลต์เหมือนที่ผ่านมาในการใช้งาน HTML4

HTML5 ได้มีการแนะนำ เอเลเมนต์ใหม่หลายตัวเพื่อตอบสนองการใช้งานของเว็บไซต์รุ่นใหม่ โดยเอเลเมนต์ใหม่ส่วนหนึ่งเป็นซีแมนติกทดแทนการใช้งานของบล็อกทั่วไป (<div>) และเอเลเมนต์อินไลน์ (<span>) ยกตัวอย่างเช่น <nav> (บล็อกสำหรับเมนู บอกทาง) และ <footer> (ส่วนด้านล่างของเว็บเพจ) เอเลเมนต์ส่วนอื่นแสดงถึงการใช้งาน เช่น เอเลเมนต์ทางด้านสื่อ <audio> และ <video> เอเลเมนต์บางตัวที่ตกฐานสำหรับ HTML 4.01 ได้ถูกยกเลิก เช่น <font> และ <center> ซึ่งถูกทดแทนด้วยการทำงานผ่านซีเอสเอส

#### จุดเด่นของ HTML5

1. element สำหรับการวาดภาพการตกแต่งภาพ ไม่ต้องพึ่ง flash หรือไม่ต้องพึ่งปลั๊กอินภายนอก
2. element วิดีโอและเสียง ไม่ต้องพึ่งปลั๊กอินภายนอก เช่น window media player ฯลฯ เป็นต้น
3. สนับสนุนในเรื่องการเก็บไฟล์ในลักษณะออฟไลน์ (Better support for local offline storage)
4. element ใหม่สำหรับเนื้อหาที่แบ่งเป็นส่วนๆหรือระบุความหมายของแต่ละส่วน article, footer, header, nav, section
5. เพิ่มความสะดวกในส่วนของ form เช่น มี input สำหรับ ปฏิทิน, วันที่, เวลา, อีเมล, URL, การค้นหา เป็นต้น

## 2.14 JavaScript

JavaScript เป็นภาษายุคใหม่สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง เราสามารถเขียน โปรแกรม JavaScript เพิ่มเข้าไปในเว็บเพจเพื่อใช้ประโยชน์

สำหรับงานด้านต่าง ๆ ทั้งการคำนวณ การแสดงผล การรับ-ส่งข้อมูล และที่สำคัญคือ สามารถโต้ตอบกับผู้ใช้ได้อย่างทันที นอกจากนี้ยังมีความสามารถด้านอื่น ๆ อีกหลายประการที่ช่วยสร้างความน่าสนใจให้กับเว็บเพจของเราได้อย่างมาก ภาษาจาวาสคริปต์ถูกพัฒนาโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript

#### 2.14.1 ลักษณะการทำงานของ JavaScript

JavaScript เป็นภาษาสคริปต์เชิงวัตถุ หรือเรียกว่า อ็อบเจ็กต์โอเรียนเตด (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนเอกสารด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ทำงานร่วมกับ ภาษา HTML และ ภาษาจาวาได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server) โดยมีลักษณะการทำงาน ดังนี้

1. Navigator JavaScript เป็น Client-Side JavaScript ซึ่งหมายถึง JavaScript ที่ถูกแปลทางฝั่งไคลเอนต์ (หมายถึงฝั่งเครื่อง คอมพิวเตอร์ของผู้ใช้ ไม่ว่าจะเป็นเครื่องพีซี เครื่องแมคอินทอช หรือ อื่น ๆ) จึงมีความเหมาะสมต่อการใช้งานของผู้ใช้ทั่วไปเป็นส่วนใหญ่
2. LiveWire JavaScript เป็น Server-Side JavaScript ซึ่งหมายถึง JavaScript ที่ถูกแปลทางฝั่งเซิร์ฟเวอร์ (หมายถึงฝั่งเครื่อง คอมพิวเตอร์ของผู้ให้บริการเว็บ โดยอาจจะเป็นเครื่องของซันซิลิ คอมมูราฟิกส์ หรือ อื่น ๆ) สามารถใช้ได้เฉพาะกับ LiveWire ของเน็ตสเคป โดยตรง

#### 2.14.2 JavaScript กับ HTML

การเขียน JavaScript เราอาจเขียนรวมอยู่ในไฟล์เดียวกันกับ HTML ได้ ซึ่งแตกต่างจากการเขียนโปรแกรมภาษา Java ที่ต้อง เขียนแยกออกเป็นไฟล์ต่างหาก ไม่สามารถเขียนรวมอยู่ในไฟล์เดียวกับ HTML ได้ วิธีการเขียน JavaScript เพื่อสั่งให้เว็บเพจทำงาน มีอยู่ด้วยกัน 2 วิธี ดังนี้ - เขียนด้วยชุดคำสั่งและฟังก์ชันของ JavaScript เอง หรือ - เขียนตามเหตุการณ์ที่เกิดขึ้นตามการใช้งานจากชุดคำสั่งของ HTML เมื่อเริ่มใช้งาน โปรแกรมบราวเซอร์ จะอ่านข้อมูลจากส่วนบนของเพจ HTML และทำงานไปตามลำดับจาก บนลงล่าง (top-down) โดยเริ่มที่ส่วน < HEAD >...< /HEAD > ก่อน จากนั้นจึงทำงานในส่วน < BODY >...< /BODY > เป็นลำดับต่อมา การทำงานของ JavaScript ดูไม่แตกต่างไปจาก HTML เท่าใดนัก แต่ HTML จะวางเลย์เอาต์โครงสร้างของอ็อบเจ็กต์ภายใน และส่วนเชื่อมโยงกับเว็บเพจเท่านั้น ในขณะที่ JavaScript สามารถเพิ่มเติมส่วนของการเขียนโปรแกรมและลอจิกเข้าไป

## 2.15 Ajax

Ajax (Asynchronous JavaScript and XML) เป็นวิธีการสร้างโปรแกรมประยุกต์ ปฏิสัมพันธ์สำหรับเว็บที่ประมวลผลคำขอของผู้ใช้ทันที Ajax รวมเครื่องมือโปรแกรมหลายอย่าง รวมถึง JavaScript, dynamic HTML (DHTML), Extensible Markup Language (XML), cascading style sheets (CSS), the Document Object Model (DOM) และ Microsoft object, XMLHttpRequest โดย Ajax ยอมให้เนื้อหาบนเว็บเพจเพื่อปรับปรุงอย่างทันที เมื่อผู้ใช้ทำการกระทำ ต่างจาก HTTP request ซึ่งผู้ใช้ไม่โหลดทั้งหมด ตัวอย่าง การใช้พยากรณ์อากาศ สามารถแสดงเงื่อนไขท้องถิ่นบนด้านหนึ่งของเพจ

Google Maps เป็นโปรแกรมประยุกต์ที่ใช้ Ajax อินเทอร์เน็ตที่ให้ผู้เปลี่ยนมุมมองและควบคุมแผนที่ในเวลาจริง โปรแกรมประยุกต์ Ajax ไม่ต้องมีการติดตั้ง plug-in แต่ทำงานโดยตรงกับ browser

## 2.16 จุดสนใจ(POI)

POI ย่อมาจาก Point Of Interest หรือ "จุดสนใจ" หมายถึง ค่าพิกัดตำแหน่งสถานที่ต่างๆที่มีผู้สำรวจได้บันทึกไว้ และนำมารวบรวมด้วยโปรแกรมเพื่อให้ได้ POI เพื่อนำไปใช้ในโปรแกรมนำทางต่างๆ ซึ่งจะมาในรูปแบบของไฟล์เดี่ยวที่รวมทุกพิกัดไว้ ซึ่งข้อมูล POI อาจประกอบไปด้วย ข้อมูลพิกัดทางภูมิศาสตร์ ชื่อสถานที่ ประเภทของสถานที่ เป็นต้น

ยกตัวอย่างเช่น ร้านค้า สถานที่ท่องเที่ยว หรือจุดที่สนใจอื่นๆ โดย POI นี้ จะแสดงในรูปแบบของพิกัด ตัวอย่างเช่น "N19 23.905 E98 35.311" เป็นต้น

### 2.16.1 การใช้งาน

สามารถนำข้อมูล POI ที่จัดทำนำมาใส่ในเครื่องมือต่างๆ เช่น นำมาใส่ในเครื่องนำทาง โปรแกรมจะคำนวณเส้นทางและนำทางไปยังค่าพิกัดใน POI

### 2.16.2 การเก็บค่า POI

ผู้ใช้เพิ่ม POI ด้วยตัวเอง ด้วยการหาพิกัดด้วยวิธีการต่างๆแล้วบันทึกเข้าโปรแกรม แต่ต้องบันทึกด้วยมือที่ละพิกัด

#### ตัวอย่างไฟล์ POI

ไฟล์นามสกุล .gpi หรือ .kml ที่แตกต่างกันไปตามชนิดของอุปกรณ์ที่ใช้จัดเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

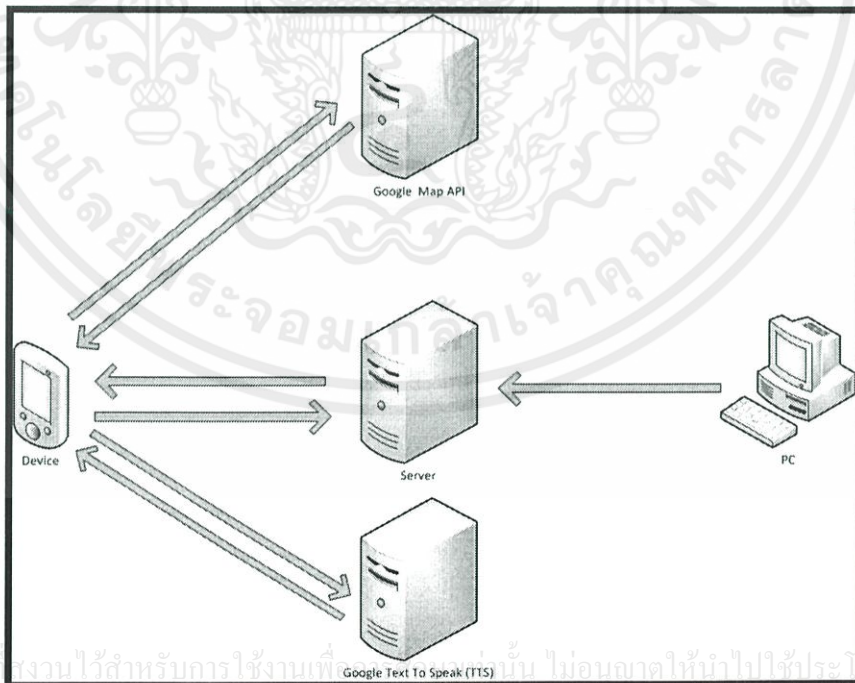
### บทที่ 3

## การออกแบบระบบ

### 3.1 ออกแบบการทำงานของระบบโดยรวม

การติดต่อของระบบจากภายนอกจะมีอยู่ 3 ระบบคือ

- 1) Google Map API เพื่อร้องขอข้อมูลเกี่ยวกับแผนที่ เส้นทาง การคำนวณระยะทาง รวมถึงลักษณะของแผนที่
  - 2) Google Text To Speech (TTS) เพื่อร้องขอการแปลงข้อความที่เป็นเสียง
  - 3) Server เพื่อร้องขอข้อมูลจุดสนใจ (POI) ในบริเวณตำแหน่งที่ผู้ใช้งานอยู่
- จากที่กล่าวไปแล้วข้างต้น 1) และ 2) เป็นส่วนการพัฒนาของ Google แล้วสามารถเรียกใช้ได้เลยแต่จะต้องมีการเขียนโปรแกรมเพื่อติดต่อเพื่อร้องขอข้อมูล และใน 3) จะเป็นส่วนผู้พัฒนาจะทำการออกแบบและสร้าง server เพื่อใช้ในการจัดเก็บข้อมูลจุดสนใจ (POI) ในการจัดเตรียมข้อมูลสำหรับการร้องขอจากผู้ใช้งาน

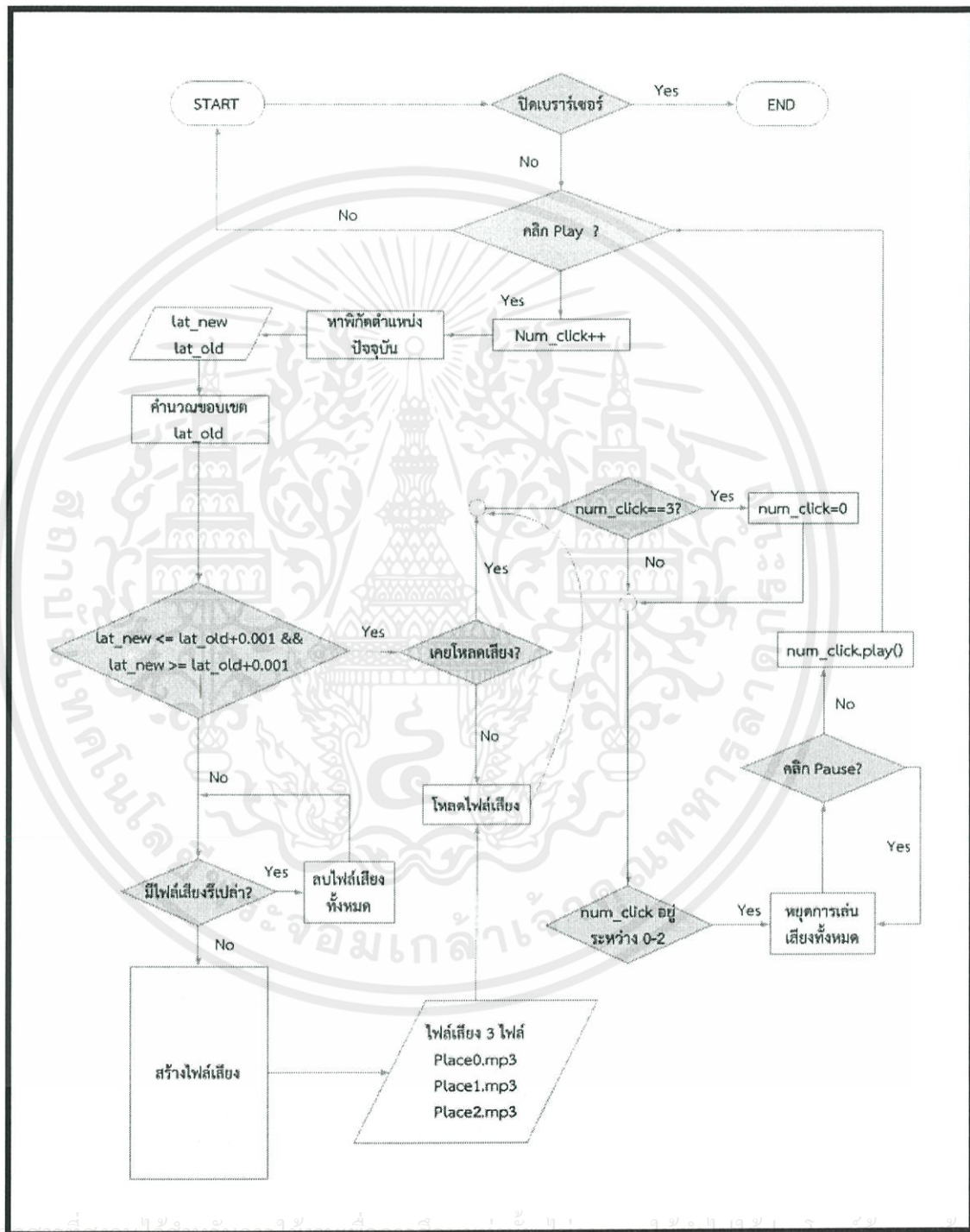


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่เอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 การออกแบบการทำงานของระบบ

### 3.2 การออกแบบแผนผังการทำงานโปรแกรม (Program Flowchart)

#### 3.2.1 การออกแบบแผนผังการทำงานโปรแกรมทั้งหมด



เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
 รูปที่ 3.2 การออกแบบการทำงานของโปรแกรมทั้งหมด

## อธิบาย

จากแผนภาพที่แสดง แสดงการทำงานของระบบ จะมีปุ่มอยู่ 2 ปุ่มคือ ปุ่มเริ่มการทำงานหรือเปิดฟังเสียง (Play) และปุ่มหยุดการทำงานหรือปุ่มหยุดฟัง ซึ่งออกแบบมาเพื่อให้มีความสะดวกและใช้งานง่ายต่อการใช้งานสำหรับผู้พิการทางสายตา ที่อธิบายไปข้างต้นในบทที่ 1 โปรแกรมที่สร้างขึ้นนี้จะทำงานได้จะต้องมีการเชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ตเสียก่อนเพราะต้องมีการใช้งานผ่านเบราว์เซอร์ เป็นโปรแกรมที่ช่วยในการอธิบายตำแหน่งปัจจุบันที่ผู้ใช้งานอยู่โดยมีการอธิบายอ้างอิงกับสถานที่ใกล้เคียงที่สนใจ (POI) ซึ่งอยู่ในฐานข้อมูลที่ระบบเราสร้างขึ้นมาด้วยเสียง เพื่อให้มีการเริ่มการทำงานผู้ใช้งานจะต้องกดหรือคลิกที่ปุ่ม play

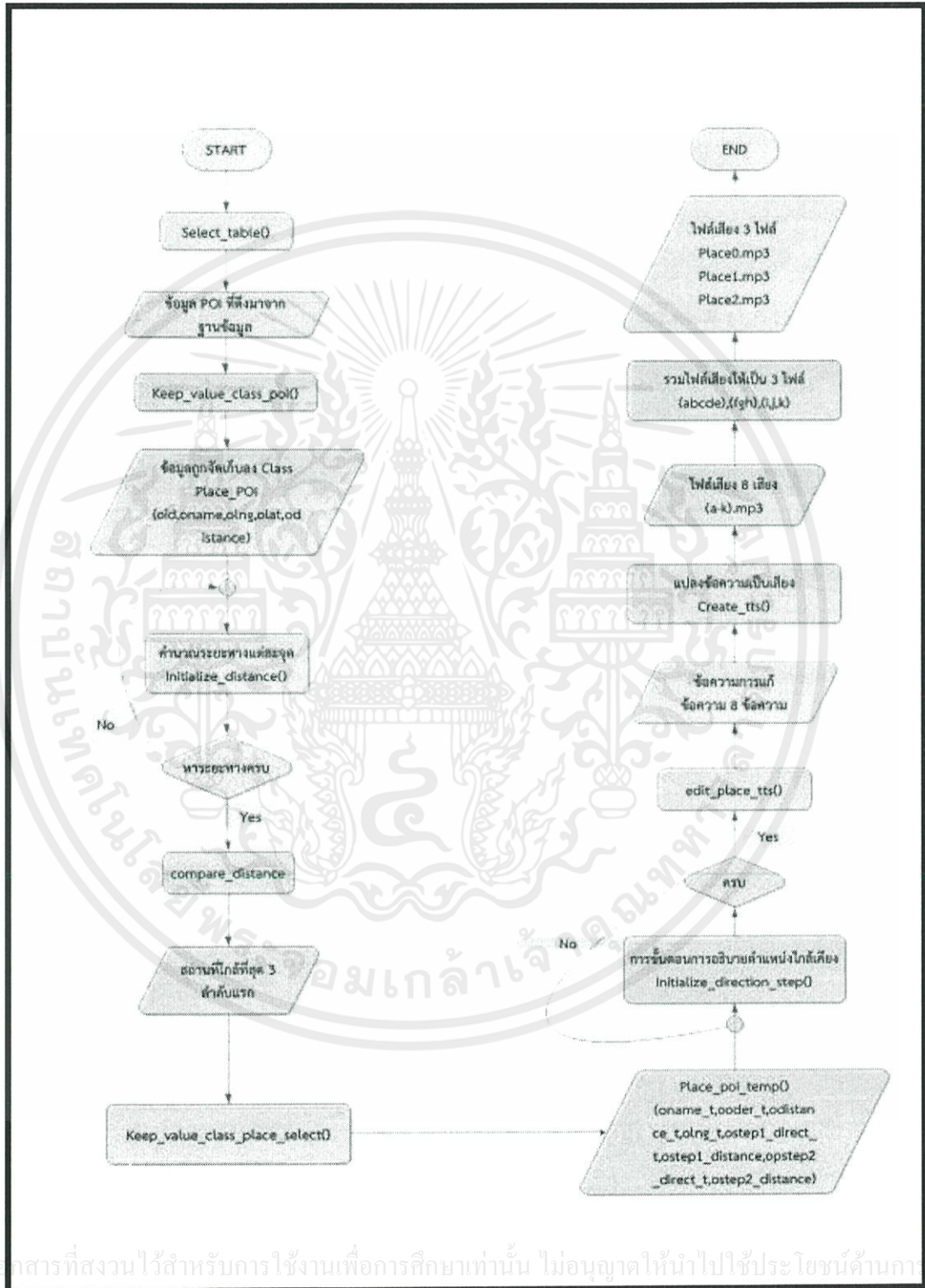
ขั้นตอนแรก (อธิบายในด้านซ้ายมือ) โปรแกรมจะทำหาพิกัดที่ผู้ใช้งานอยู่ จากนั้นเปรียบเทียบกับข้อมูลพิกัดเดิมที่มีการเก็บไว้ก่อนหน้านี้ ถ้าพิกัดปัจจุบันอยู่ในขอบเขตพิกัดเดิมจะทำการเล่นไฟล์เสียงเดิมจากไฟล์ place0 ,play1 ถึง play2 แต่ถ้าไม่ ระบบจะทำการลบไฟล์เสียงที่ใช้อธิบายเดิม แล้วทำการสร้างไฟล์เพื่อใช้ในการอธิบายตำแหน่งปัจจุบันที่ผู้ใช้งานอยู่ใหม่ ซึ่งก็คือ place0 ,play1 ถึง play2 (การสร้างไฟล์เสียงจะอธิบายในลำดับถัดไป)

ส่วนทางด้านขวามือของรูป จะเป็นส่วนในการควบคุมการเปิดหรือหยุดการฟังเสียงหลังจากที่มีไฟล์เสียงการอธิบายตำแหน่งที่อยู่ในขอบเขตที่กำหนด ระบบจะทำการเล่นไฟล์เสียง place0.mp3 เป็นลำดับแรก เมื่อมีการกดอีกครั้งก็จะเล่นไฟล์ place1.mp3 และกดอีกครั้งก็จะเป็นการเล่นไฟล์ place2.mp3 ซึ่งจะวนการเล่นไฟล์เสียงอย่างนี้ไปเรื่อยๆ ถ้ากดปุ่ม pause จะทำการหยุดเสียงที่เล่นในขณะนั้น

จะสังเกตว่า โปรแกรมการทำงาน จะรอเหตุการณ์การกดปุ่ม play จากผู้ใช้งานตลอดเวลา โปรแกรมจะหยุดการทำงานหรือเริ่มใหม่อีกรอบเมื่อมีการปิดเบราว์เซอร์หรือทำงานโหลดเบราว์เซอร์ที่ใช้ขณะนั้นใหม่ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1. การออกแบบแผนผังการทำงานของโปรแกรมส่วนสร้างไฟล์เสียงการอธิบายตำแหน่ง



รูปที่ 3.3 ออกแบบแผนผังการทำงานของโปรแกรมส่วนสร้างไฟล์เสียงการอธิบายตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นๆ  
ไม่ว่ากรณีใดๆทั้งสิ้น ผู้ใช้ทุกท่านมีให้คิดปรุงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีโอกาสไปใช้

## อธิบาย

การทำงานของโปรแกรมมีการทำงาน ไม่ซับซ้อนแยกการทำงานแบ่งเป็นหลายฟังก์ชันที่มีจุดประสงค์และการทำงานที่แตกต่างกัน ในส่วนนี้จะเป็นการอธิบายขั้นตอนของการสร้างไฟล์เสียง ซึ่งประกอบด้วยขั้นตอนที่เริ่มตั้งแต่ การหาพิกัดทางภูมิศาสตร์ นำพิกัด ที่ได้มาหาสถานที่ใกล้เคียง (select\_table()) จากฐานข้อมูล (POI) เพื่อให้สามารถอธิบายถึงสถานที่ใกล้เคียงได้ว่าอยู่ ตรงจุดไหน เราก็จะใช้ ข้อมูล ระยะ (initialize\_distance()) มาใช้ในการเลือกสถานที่ใกล้เคียง ใกล้ที่สุด (compare\_distance()) จากนั้น หาว่าสถานที่ใกล้ที่สุด 3 ลำดับแรกอยู่ที่ไหน และมีระยะทางเท่าไร (initialize\_direction\_step()) เพื่อนำข้อมูลที่ได้มาอธิบายตำแหน่งที่ผู้ใช้งานอยู่

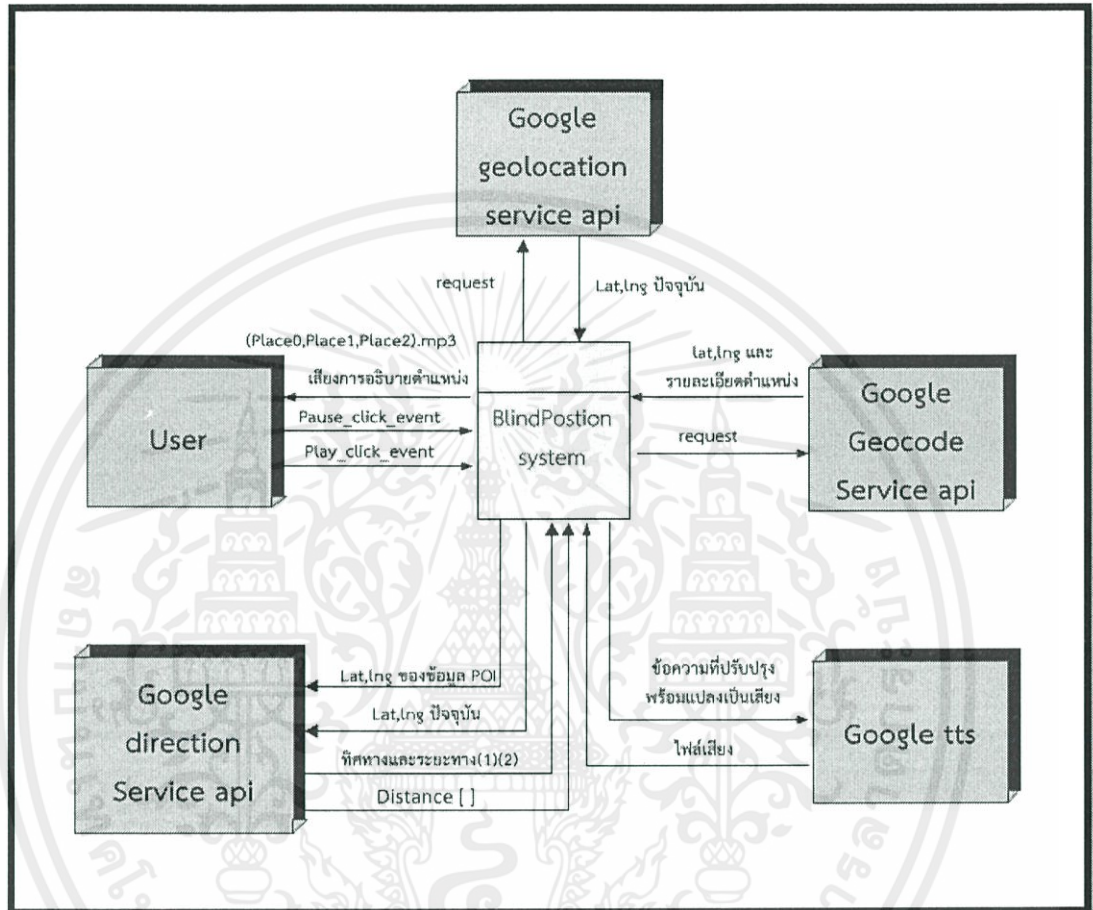
เมื่อได้ข้อมูลทั้งหมดซึ่งประกอบด้วยข้อมูลตำแหน่งปัจจุบันที่ผู้ใช้งานอยู่และข้อมูลสถานที่ใกล้เคียง 3 แห่ง (2ขั้นตอนแรกของการที่จะไปสถานที่ใกล้เคียงแต่ละที่ซึ่งแต่ละขั้นประกอบด้วยทิศทางและระยะทาง ) เราจะนำข้อมูลที่ได้ทั้งหมดมาทำการตัดแต่งเพื่อให้เหมาะสมก่อนนำไปใช้ในการแปลงเป็นเสียง (edit\_place\_tts()) ซึ่งจะได้ข้อความทั้งหมด 8 ข้อความ

จากนั้นแต่ละข้อความจะถูกนำไปแปลงเป็นข้อความเสียง (create\_tts()) ซึ่งเป็นการแปลงผ่าน google tts(translate) เพื่อให้มีเหมาะสมในการประมวลผลเราจะนำไฟล์เสียงไปรวมจาก 8 ไฟล์คือ a,b,c,...,k ให้เหลือเพียง 3 ไฟล์ซึ่งมีการแบ่งดังนี้ ไฟล์ place0.mp3 เกิดจากการรวมไฟล์ a.mp3,b.mp3,c.mp3,d.mp3 และ e.mp3 ไฟล์ place0.mp3 เกิดจากการรวมไฟล์ f.mp3,g.mp3 และ h.mp3 และไฟล์ place0.mp3 เกิดจากการรวมไฟล์ i.mp3, j.mp3 และ k.mp3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3. การออกแบบการไหลของข้อมูล (Data Flow)

#### 3.3.1 การออกแบบการไหลของข้อมูล ระดับ Context



รูปที่ 3.4 แผนผังการไหลของข้อมูล ระดับ context

#### อธิบาย

ระบบที่พัฒนาที่ชื่อว่า BlindPosition อยู่ตำแหน่งตรงกลางของแผนภาพซึ่ง ข้อมูลติดต่อกับระบบภายนอกทั้งเข้าและออกกับระบบต่างๆดังนี้

User คือ ผู้ใช้งานเพื่อฟังการอธิบายตำแหน่ง จากการควบคุมโดยการกดปุ่ม play หรือ pause

Google geolocation service api เพื่อรับข้อมูลพิกัดตำแหน่งปัจจุบัน

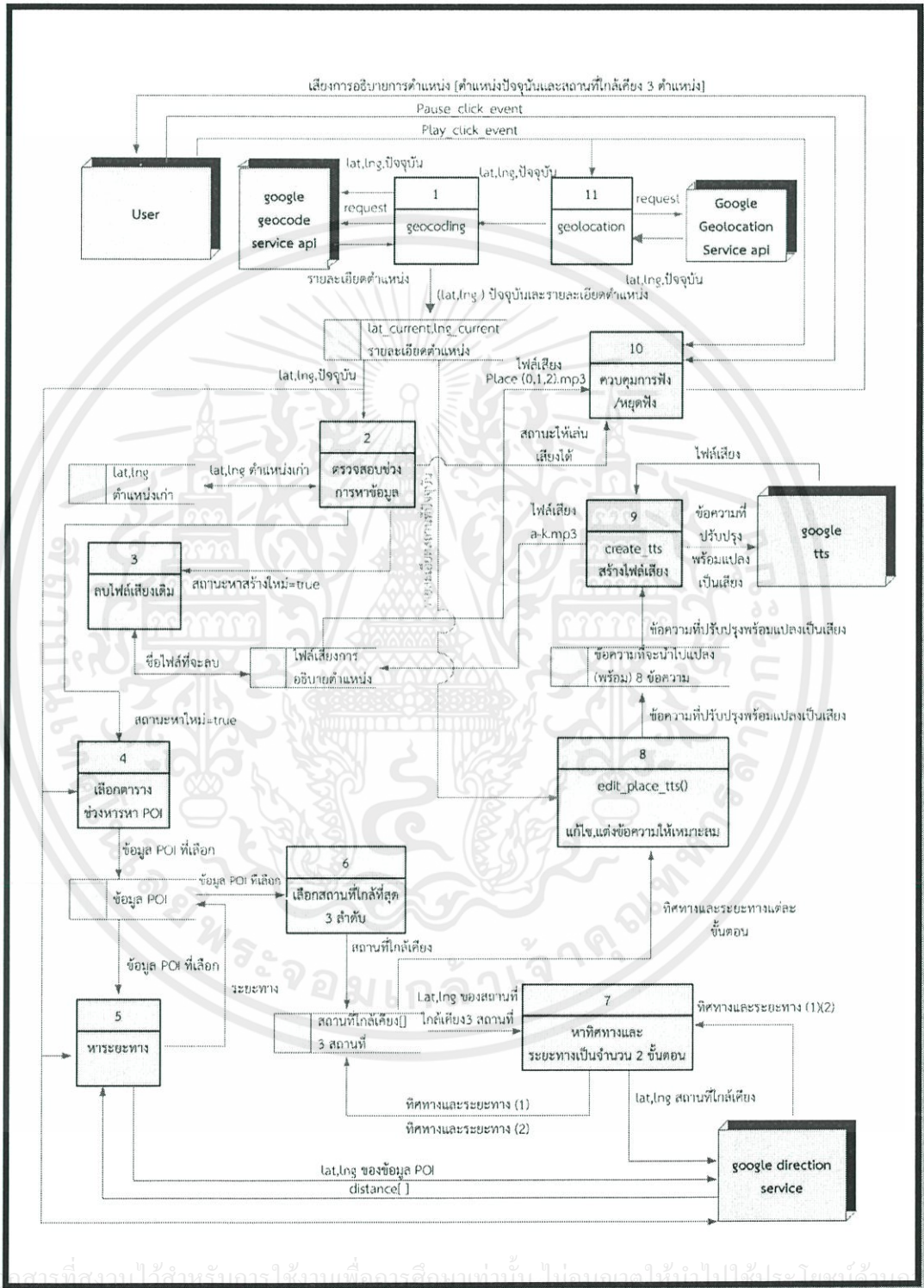
Google geocoding service api เพื่อแสดงรายละเอียดชื่อสถานที่พิกัดตำแหน่งปัจจุบัน

Google direction service api เพื่อหารายละเอียดตำแหน่งสถานที่ใกล้เคียง

Google tts เพื่อนำข้อความที่ต้องการใช้ในการอธิบายไปแปลงเป็นข้อความเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 การออกแบบการไหลของข้อมูล ระดับ 0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการเรียนการสอนเท่านั้น ไม่สงวนลิขสิทธิ์  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือทำซ้ำโดยไม่ได้รับอนุญาต  
รูปที่ 3.5 แผนผังการไหลของข้อมูล ระดับ 0

### 3.3.2.1 โปรแกรมย่อย (Process) ประกอบไปด้วย 11 โปรแกรม คือ

ตาราง 3.1 รายละเอียดโปรแกรมย่อยในแผนผังการไหลของข้อมูล

ชื่อโปรแกรมย่อย	ชื่อฟังก์ชัน	อธิบาย
Geolocation	initialize_local()	โปรแกรมหาพิกัดภูมิศาสตร์ตำแหน่งปัจจุบันของผู้ใช้งาน
Geocoding	Initialize()	โปรแกรมแปลงพิกัดภูมิศาสตร์เป็นชื่อสถานที่ เช่น ชื่อถนน, ชื่ออำเภอ, ชื่อจังหวัดและชื่อประเทศ เป็นต้น
ตรวจสอบช่วงการหาข้อมูล		โปรแกรมตรวจสอบช่วงพิกัดภูมิศาสตร์ปัจจุบันเปรียบเทียบกับพิกัดภูมิศาสตร์เดิม ถ้าอยู่ในขอบเขตจะได้ไม่ต้องมีการสร้างไฟล์เสียงใหม่
ลบไฟล์เสียงเดิม	Fdelete()	โปรแกรมลบไฟล์เสียงเดิมเพื่อรองรับไฟล์เสียงใหม่ที่จะสร้างขึ้น อีกทั้งยังช่วยในการบริหารเนื้อที่หน่วยความจำไม่ให้ออกเกินไปได้
เลือกตารางการหา POI	select_table()	โปรแกรมกำหนดขอบเขตในการดึงข้อมูลสถานที่ใกล้เคียง (POI) จากฐานข้อมูล ซึ่งภายจามีการกำหนดจำนวนข้อมูลตำแหน่งใกล้เคียงเพื่อไม่ให้มากเกินไปที่จะส่งผลการประมวลผลของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.2 รายละเอียดโปรแกรมย่อยในแผนผังการไหลของข้อมูล (ต่อ)

ชื่อโปรแกรมย่อย	ชื่อฟังก์ชัน	อธิบาย
หาระยะทาง	initialize_distance()	โปรแกรมหาระยะทางระหว่างพิกัดภูมิศาสตร์ตำแหน่งปัจจุบันกับพิกัดภูมิศาสตร์ของสถานที่ใกล้เคียง(POI) แต่ละจุด
เลือกสถานที่ใกล้ที่สุด 3 สถานที่	compare_distance()	โปรแกรมจัดเรียงสถานที่ตามระยะทางที่ใกล้ที่สุดมาก่อนเป็นจำนวน 3 สถานที่ เพื่อนำมาใช้ในการอธิบายตำแหน่งสถานที่ใกล้เคียง
หาทิศทางและระยะทางเป็นจำนวน 2 ขั้นตอน	initialize_distance()	โปรแกรมหาทิศทางและระยะทางระหว่างพิกัดภูมิศาสตร์ตำแหน่งปัจจุบันกับพิกัดภูมิศาสตร์ของสถานที่ใกล้เคียงที่สุด 3 สถานที่ที่ได้จากการเลือก
แก้ไข,แต่งข้อความให้เหมาะสม	edit_place_tts()	โปรแกรมแก้ไข ปรับปรุงข้อมูลที่ได้ทั้งหมด เพื่อให้ข้อมูลที่มีความน่าสนใจ เข้าใจง่าย
สร้างไฟล์เสียง	create_tts()	โปรแกรมแปลงข้อความเป็นเสียง
ควบคุมการฟัง/หยุดเสียง		โปรแกรมควบคุมการเปิดฟัง/หยุดเสียงที่ใช้ในการอธิบายการบอกตำแหน่ง
รวมไฟล์เสียง	mer_mp3()	โปรแกรมรวมไฟล์เสียง

### 3.3.2.2 การจัดเก็บข้อมูล (Data store)

ตาราง 3.3 ข้อมูลพิกัดตำแหน่งปัจจุบัน

ชื่อข้อมูล	ชนิดข้อมูล	อธิบาย
Lat	Float	ข้อมูลพิกัดละติจูดปัจจุบัน
Lng	Float	ข้อมูลพิกัดลองจิจูดปัจจุบัน

ตาราง 3.4 ข้อมูลพิกัดละติจูดเดิม

ชื่อข้อมูล	ชนิดข้อมูล	อธิบาย
Lat_old	Float	ข้อมูลพิกัดละติจูดเดิม ที่เคยมีการสร้างไฟล์เสียงการอธิบายตำแหน่ง

ตาราง 3.5. ข้อมูลสถานที่ใกล้เคียง (POI)

ชื่อข้อมูล	ชนิดข้อมูล	อธิบาย
Oid	Int	หมายเลขประจำตัวของข้อมูล
Oname	String	ชื่อสถานที่ใกล้เคียง
Olat	Float	พิกัดละติจูดของข้อมูลใกล้เคียง
Olng	Float	พิกัดลองจิจูดของข้อมูลใกล้เคียง
odistance	Int	ระยะทางระหว่างพิกัดตำแหน่งปัจจุบันกับพิกัดสถานที่ใกล้เคียง

ตาราง 3.6 ข้อมูลสถานที่ใกล้เคียงที่สุด

ชื่อข้อมูล	ชนิดข้อมูล	อธิบาย
Oid_t	Int	หมายเลขประจำตัวของข้อมูล
Oname_t	String	ชื่อสถานที่ใกล้เคียง
Olat_t	Float	พิกัดละติจูดของข้อมูลใกล้เคียง
Olng_t	Float	พิกัดลองจิจูดของข้อมูลใกล้เคียง

Odistance_t	Int	ระยะทางระหว่างพิกัดตำแหน่ง
ชื่อข้อมูล	ชนิดข้อมูล	อธิบาย
		ปัจจุบันกับพิกัดสถานที่ใกล้เคียง
Ostep1_direct_t	String	ทิศทางของตำแหน่งสถานที่ใกล้เคียงที่สุดขั้นที่1
Ostep1_distance_t	Int	ระยะทางระหว่างพิกัดตำแหน่งปัจจุบันกับพิกัดสถานที่ใกล้เคียงที่สุดขั้นที่1
Ostep2_direct_t	String	ทิศทางของตำแหน่งสถานที่ใกล้เคียงที่สุดขั้นที่2
Ostep2_distance_t	int	ระยะทางระหว่างพิกัดตำแหน่งปัจจุบันกับพิกัดสถานที่ใกล้เคียงที่สุดขั้นที่2

ตาราง 3.7 ข้อความที่จะแปลงเป็นเสียง

ชื่อข้อมูล	ชนิดข้อมูล	อธิบาย
Placetts0_1	String	ข้อความสถานที่ปัจจุบันครั้งแรก
Placetts0_2	String	ข้อความสถานที่ปัจจุบันครั้งหลัง
Placetts1_1	String	ข้อความสถานที่ใกล้เคียงที่สุดลำดับที่ 1 ส่วนแรก
Placetts1_2	String	ข้อความสถานที่ใกล้เคียงที่สุดลำดับที่ 1 ส่วนกลาง
Placetts1_3	String	ข้อความสถานที่ใกล้เคียงที่สุดลำดับที่ 1 ส่วนหลัง
Placetts2_1	String	ข้อความสถานที่ใกล้เคียงที่สุดลำดับที่ 2 ส่วนแรก
Placetts2_2	String	ข้อความสถานที่ใกล้เคียงที่สุดลำดับที่ 2 ส่วนกลาง
Placetts2_3	String	ข้อความสถานที่ใกล้เคียงที่สุดลำดับที่ 2 ส่วนหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษเท่านั้น ไม่ควรนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อข้อมูล	ชนิดข้อมูล	อธิบาย
Placetts3_1	String	ข้อความสถานที่ใกล้เคียงที่สุด ลำดับที่ 3 ส่วนแรก
Placetts3_2	String	ข้อความสถานที่ใกล้เคียงที่สุด ลำดับที่ 3 ส่วนกลาง
Placetts3_3	String	ข้อความสถานที่ใกล้เคียงที่สุด ลำดับที่ 3 ส่วนหลัง

ตาราง 3.8 ชื่อไฟล์เสียงการอธิบายการบอกตำแหน่ง

ชื่อข้อมูล	ชนิดข้อมูล	อธิบาย
a (b,c,..,k)	char	ไฟล์เสียงที่ได้จากการแปลง ข้อความสถานที่ใกล้เคียง (Placetts1 – Placetts8)
Place0(Place1,place2)	string	ชื่อไฟล์เสียงที่เกิดจากการรวม ไฟล์เสียง(a-k) โดย Place0 เกิดจาก a,b,c,d,e Place1 เกิดจาก f,g,h Place2 เกิดจาก i,j,k

### 3.4 ออกแบบฐานข้อมูลสถานที่ใกล้เคียง (POI)

จากการศึกษาข้อมูลพิกัดภูมิศาสตร์ของประเทศไทยจะมีขอบเขตโดยประมาณตั้งแต่ละติจูดที่ 5 ถึง 21 และลองจิจูดที่ 97 ถึง 105 ทั้งนี้จะสังเกตว่า ขอบเขตที่กล่าวไปข้างต้นจะมากกว่าข้อมูลจริงๆ ทั้งนี้เพื่อให้การจัดเก็บและการออกแบบที่ง่ายต่อการจัดการ ในการออกแบบเราจะใช้ข้อมูลพิกัดละติจูดเป็นข้อมูลหลักในกำหนดจำนวนตารางฐานข้อมูลของระบบ

จากการวิเคราะห์ถึงความเหมาะสมได้ทำการออกแบบจำนวนตารางทั้งหมด 16 ตารางคือ lat5 ถึง lat ซึ่งแต่ละตารางประกอบด้วย 5 คอลัมน์คือ id,name,lat,lng และ lat\_lng

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.9 รายละเอียดตารางฐานข้อมูล

ชื่อคอลัมน์	ชนิดข้อมูล	อธิบาย
id	int (8)	หมายเลขประจำตัวของข้อมูล สถานที่ใกล้เคียง
name	varchar(100)	ชื่อสถานที่ใกล้เคียง
lat	float	พิกัดละติจูดของสถานที่ใกล้เคียง
lng	float	พิกัดลองจิจูดของสถานที่ ใกล้เคียง
lat_lng	text	ช่วงพิกัดละติจูดและลองจิจูดของ สถานที่ใกล้เคียง กรณีสถานที่นั้น มีขนาดใหญ่มาก เช่น สนามบิน สุวรรณภูมิ โรงงานขนาดใหญ่ มหาวิทยาลัย เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การพัฒนาระบบ

#### Location service for the visually impaired

การพัฒนาระบบจะแบ่งการพัฒนออกเป็น 5 ส่วน คือ ส่วนการสร้างฐานข้อมูล ส่วนพัฒนาแต่ ละฟังก์ชันของระบบ ส่วนพัฒนาระบบเมื่อนำฟังก์ชันต่างๆมารวมกัน ส่วนนำระบบที่พัฒนาไปลงที่ web service และส่วนสุดท้ายคือส่วนทดสอบการทำงานของระบบ

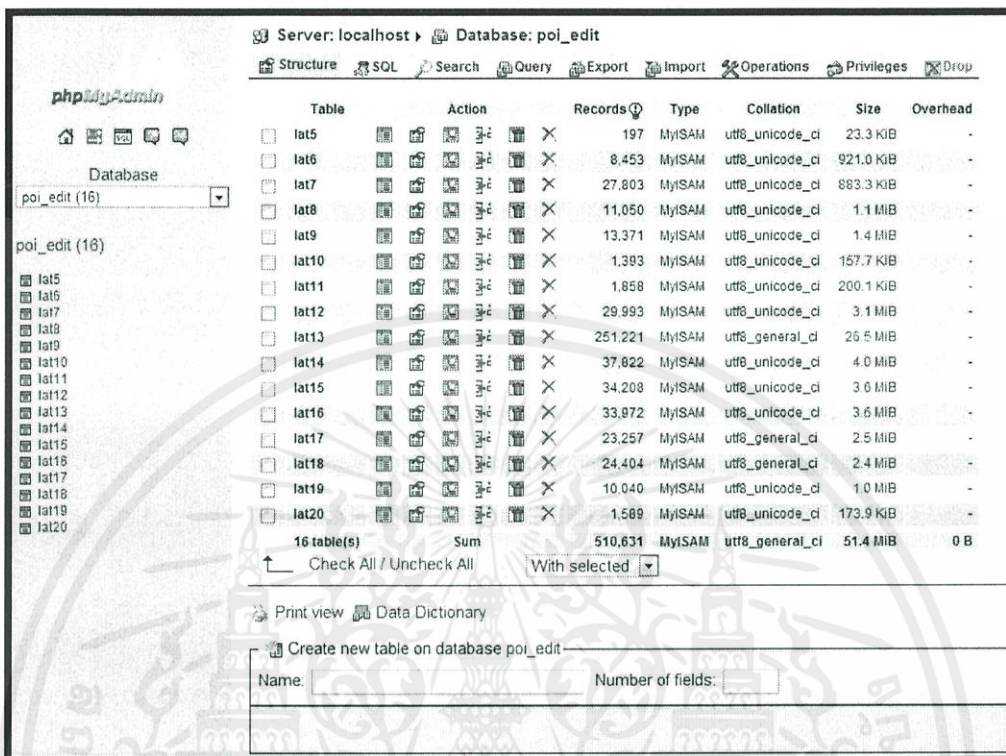
##### 4.1 ส่วนการสร้างฐานข้อมูล

การสร้างฐานข้อมูลในส่วนนี้ คือ ข้อมูล POI ซึ่งมีมากกว่า 500,00 จุดทั่วประเทศไทย โดยทางผู้จัดทำได้แบ่งตารางไฟล์ Excel ตามละติจูดของประเทศไทยซึ่งมีค่าตั้งแต่องศาที่ 5 ถึงองศาที่ 20 ซึ่งแบ่งได้ทั้งหมด 16 ตารางไฟล์ Excel เช่น องศาที่ 5 ก็จะให้ชื่อว่า lat5 องศาที่ 17 ก็ให้ชื่อ lat17 เป็นต้น

ในการนำข้อมูลเราใช้งานของ Apperve เวอร์ชัน 2.5.9 โดยจะใช้ในส่วนของ MySQL Database คือโปรแกรมที่ทำหน้าเป็น Database Server และ phpMyAdmin คือตัวควบคุม MySQL Database ผ่านเว็บไซต์ ซึ่งมีขั้นตอนดังต่อไปนี้

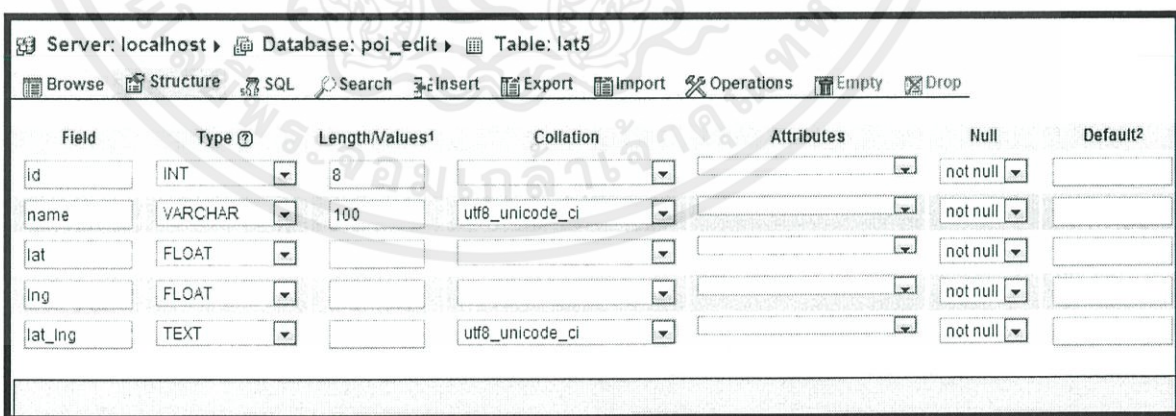
1. เข้า url = <http://localhost/phpmyadmin/> ผ่านเว็บเบราว์เซอร์ และกรอก Username และ Password ที่ได้ทำการตั้งค่าตอนติดตั้ง Appserve
2. ทำการสร้างชื่อฐานข้อมูล และ สร้าง Table (ตาราง) ไว้ทั้งหมด 16 ตารางตามที่เราแบ่งไฟล์ Excel ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงฐานข้อมูลที่สร้าง

- ซึ่งโครงสร้าง (Structure) แต่ละ Table จะประกอบด้วย Field 5 field ซึ่งมีรายละเอียด type (ชนิด) ดังรูปภาพ



รูปที่ 4.2 แสดงรายละเอียดของชนิดข้อมูลในตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อทำการสร้างฐานข้อมูล และ สร้าง Table 16 ตารางเรียบร้อยแล้ว แต่ละ Table จะต้อง import ข้อมูลตาราง Excel ที่แบ่งตามค่าละเอียด โดยไฟล์ที่จะนำเข้าต้องเป็นสกุล .csv และจะต้องเป็นการบันทึก Format Encoding (การเข้ารหัส) เป็น UTF-8 ดังรูป ซึ่งข้อจำกัดของบริการ Appserve คือการ import ไฟล์ขนาดจะไม่เกิน 2,048 KB
5. การนำข้อมูล เข้าแต่ละ Table โดยการ import จะต้องทำการเลือกไฟล์ และ Character set of the file (ชุดอักขระของไฟล์) เป็น utf8 และ Format of imported file จะต้องเลือก CSV using LOAD DATA และตั้งค่า จบฟิลด์ด้วย (Fields terminated by) เป็น , (comma) และ ครอบฟิลด์ด้วย (Fields enclosed by) เป็น " (double quote) ดังรูปภาพ

รูปที่ 4.3 แสดงขั้นตอนการนำข้อมูลที่จัดเก็บลงฐานข้อมูล

6. ทำการ import ตารางให้ครบทั้งหมด 16 ตาราง ตามข้อ 5 ก็เป็นอันเสร็จสิ้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

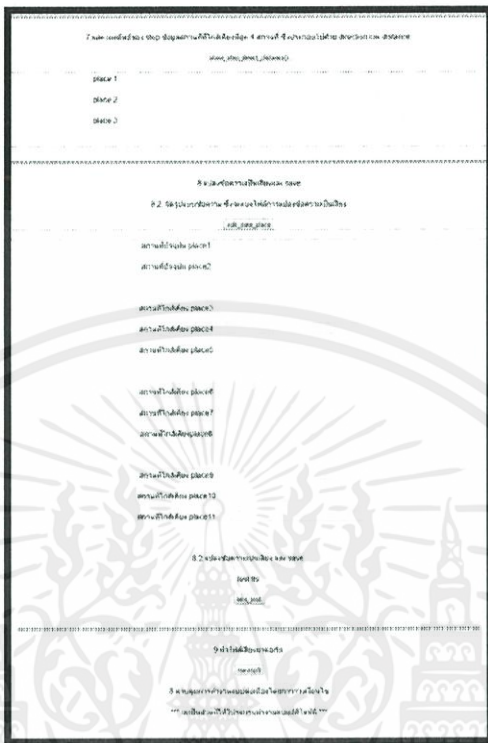
## 4.2 ส่วนพัฒนาแต่ละฟังก์ชันของระบบ

การพัฒนาจะพัฒนาโดยมีพื้นฐานเป็นอยู่บน เว็บแอปพลิเคชัน จะใช้ภาษา html , javascript ทำการประมวลผลที่ฝั่งไคลเอนท์ (Client) และใช้ภาษา PHP และ mySQL ทำการประมวลผลที่ฝั่งเซิร์ฟเวอร์ (Server) ซึ่งในการพัฒนาโปรแกรมจะใช้ภาษา javascript เป็นหลักซึ่งเป็นภาษาแบบ script ซึ่งเขียนง่ายและใช้เวลาในการทำความเข้าใจค่อนข้างน้อยโดยเฉพาะอย่างยิ่งมีความใกล้เคียงกับภาษาซีที่คุ้นเคยกัน การอธิบายการพัฒนาโปรแกรมในส่วนนี้อาจมีการอธิบายภาษาโปรแกรมมากกว่า 1 ภาษาซึ่งผู้อ่านสามารถศึกษาในบทที่ 2 ก่อนซึ่งได้จัดเตรียมความรู้พื้นฐานในการเขียนโปรแกรมของแต่ละภาษาไว้แล้ว การพัฒนาโปรแกรมจะแบ่งการพัฒนาออกเป็นส่วนๆ เพื่อสะดวกในการบริหารจัดการและแก้ไขเมื่อต้องการปรับปรุงโครงสร้างต่อในอนาคต โดยภาพรวมเป็นการทำงานของระบบจะเริ่มตั้งแต่การ เก็บข้อมูลตำแหน่ง ทางภูมิศาสตร์ จนไปถึงนำข้อความการอธิบายตำแหน่งไปแปลงเป็นเสียง ซึ่งจะแบ่งออกเป็น 11 โปรแกรมย่อยหรือส่วน โดยมีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้างส่วนทดสอบการเขียนโปรแกรม เพื่อแสดงผลลัพธ์โดยใช้ ภาษา HTML ในการกำหนดฟอร์ม(form) เพื่อรับ และแสดงข้อมูลโดยมีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

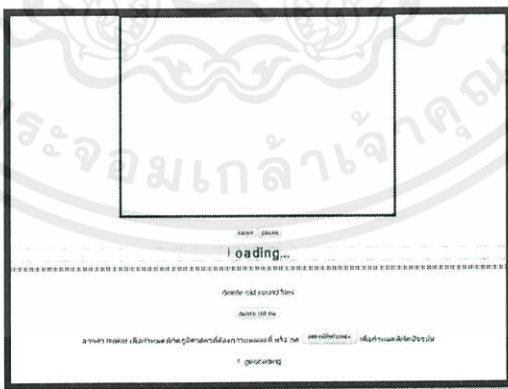




รูปที่ 4.5 รูปแบบที่สร้างเพื่อใช้ในการทดสอบ (ต่อ)

การพัฒนาโปรแกรมในส่วนต่างๆมีดังนี้

4.2.1 ส่วนที่ 1 จัดเก็บตำแหน่งพิกัดปัจจุบันของผู้ใช้งาน



รูปที่ 4.6 การหาพิกัดตำแหน่งทางภูมิศาสตร์โดยลากสัญลักษณ์เพื่อกำหนดพิกัดที่ต้องการหรือกด

ปุ่มตรวจหาพิกัดปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**จุดประสงค์** คือ เมื่อผู้ใช้งานกดปุ่ม โปรแกรมจะทำงานประมวลผลแล้วทำการเก็บข้อมูลตำแหน่งพิกัดปัจจุบัน เพื่อนำข้อมูลพิกัดที่ได้มาใช้ในการกำหนดจุดอ้างอิงในการข้อมูลสถานที่ใกล้เคียงต่อไป

**วิธีการดำเนินงาน** มีดังนี้

1. ทำการร้องขอบริการจาก google map api.2 ซึ่งในส่วนนี้จะใช้ บริการในส่วนของ geocoding ซึ่งมีประโยชน์ในการแปลงข้อมูลพิกัดเป็นรายละเอียดสถานที่ ก่อนอื่นเราจะต้องได้รับหมายเลข key api ที่ได้จาก google เป็นอันดับแรกเพื่อที่เราจะสามารถใช้บริการนี้ได้ ซึ่งสามารถศึกษาเพิ่มเติมได้ในบทที่ 2 ในเรื่อง geocoding
2. สร้างตัวแปร GGM ซึ่งใช้เก็บข้อมูลแผนที่สำหรับโปรแกรมของ

#### โปรแกรม 4.1 การสร้างแผนที่และกำหนดคุณลักษณะต่างๆ

```
<script type="text/javascript">
var map; var GGM;
var geocoder;
lat = 13.00;
lng = 100.00;
function initialize()
{
  GGM=new Object(google.maps);
  var my_Latlng = new GGM.LatLng(13.7257, 100.7846);
  geocoder = new GGM.Geocoder();
  var my_mapTypeId=GGM.MapTypeId.ROADMAP;
  var my_DivObj=$("#map_canvas")[0];
```

3. กำหนดองค์ประกอบของแผนที่

#### โปรแกรม 4.2 กำหนดองค์ประกอบของแผนที่

```
var myOptions =
{
  zoom: 10,
  center: my_Latlng ,
  mapTypeId:'roadmap'
};

map = new GGM.Map(my_DivObj,myOptions);
var my_Marker = new GGM.Marker
{ position: my_Latlng,map: map,draggable:true, title:"คลิกลากเพื่อหาตำแหน่งจุดที่ต้องการ!});
```

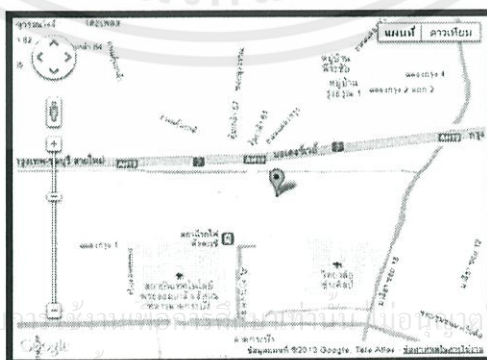
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ผลลัพธ์ที่ได้จะถูกจัดเก็บใน result ซึ่งเราสามารถนำมาดัดแปลงได้ตามความต้องการ ซึ่งเป็นผลลัพธ์ที่ได้จากการประมวลผลที่ฝั่ง server ของ google ซึ่งระบบของเราเป็นเพียงระบุตำแหน่ง และเรียกใช้บริการจากทาง google map เท่านั้น

#### โปรแกรม 4.3 ผลลัพธ์จากการการแปลงข้อมูลพิกัดภูมิศาสตร์เป็นรายละเอียดข้อมูลสถานที่

```
GGM.event.addListener(my_Marker, 'dragend', function()
{
    var my_Point = my_Marker.getPosition();
    var latlng = new google.maps.LatLng(lat, lng);
    geocoder.geocode({'latLng': latlng}, function(results,
status)
{
    if (status == google.maps.GeocoderStatus.OK)
    {
        if (results[0])
        {
            placetts_0_1 = results[0].address_components[0].long_name+" "
+results[0].address_components[1].long_name;
            placetts_0_2 = results[0].address_components[2].long_name+" "
+results[0].address_components[3].long_name;
        }
        else
        {
            alert('No results found');
        }
    } else
    {
        alert('Geocoder failed due to: ' + status);
    }
    });
    $("#lat_value").val(my_Point.lat());
    $("#lng_value").val(my_Point.lng());
});
```

#### ผลลัพธ์



รูปที่ 4.7 แผนที่ที่ได้จากการสร้างและสัญลักษณ์ที่ใช้ในการการกำหนดตำแหน่ง

1. geocoding

ข้อมูลตำแหน่งปัจจุบัน

Latitude

Longitude

address\_value

รูปที่ 4.8 ผลลัพธ์ที่ได้จากการแปลงข้อมูลพิกัดภูมิศาสตร์ที่กำหนดเป็นรายละเอียดข้อมูลสถานที่

#### 4.2.2 ส่วนที่ 2 ส่วนเลือกตาราง

2. เลือกตาราง

ระยะที่ใช้ในการหาสถานที่ใกล้เคียง

ระยะตาม lat  (ระยะตามพิกัด ไม่ใช่ระยะทางเป็น กม.)

ระยะตาม lng  (ระยะตามพิกัด ไม่ใช่ระยะทางเป็น กม.)

ขอบเขตบน  ขอบเขตล่าง

ขอบเขตซ้าย  ขอบเขตขวา

จะต้องค้นหาในตาราง จำนวน

ตาราง คือ ตารางที่ lat13 หรือ/และ undefine

รูปที่ 4.9 การกำหนดขอบเขตในการเลือกข้อมูลสถานที่ใกล้เคียง

จุดประสงค์ คือ ต้องการกำหนดชื่อตารางเพื่อใช้ในการหาข้อมูลสถานที่ใกล้เคียงจากฐานข้อมูล poi ที่ได้ออกแบบไว้ ของระบบที่เราสร้างขึ้นมาซึ่งเป็นฐานข้อมูลที่ได้สร้างไปในเรื่องการสร้างฐานข้อมูล โดยในส่วนนี้จะมีการกำหนดจำนวนและขอบเขตที่ใช้ในการดึงข้อมูลจากฐานข้อมูลเพื่อไม่ให้ มีจำนวน ข้อมูลมากจนทำให้การประมวลผลในขั้นต่อไปช้าเกินไป

วิธีการดำเนินงาน มีดังนี้

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ในโปรแกรม select\_table.js มีฟังก์ชัน select\_table() ซึ่งเป็นส่วนกำหนดค่าเริ่มต้นในการกำหนดขอบเขตโดย จะใช้พิกัดละติจูดเป็นหลัก ข้อสังเกต ถ้าตำแหน่งปัจจุบันอยู่ในช่วงละติจูดที่ 12 ถึง 14 จะมีการกำหนดช่วงที่มีความละเอียดกว่าช่วงอื่น เนื่องจากข้อมูลในช่วงนี้จะมีจำนวนข้อมูลที่ค่อนข้างมาก ซึ่งมีผลต่อการประมวลผล

#### โปรแกรม 4.4 การกำหนดเงื่อนไขในการกำหนดขอบเขต

```
if (lat_s >= 12 && lat_s < 16 )
{
    lat_value_up      = parseFloat(lat_s) + 0.01;
    lat_value_down    = parseFloat(lat_s) - 0.01;
    bound_left        = parseFloat(dlng_value) - 0.01;
    bound_right       = parseFloat(dlng_value) + 0.01;
}
else
{
    lat_value_up      = parseFloat(lat_s) + 0.1 ;
    lat_value_down    = parseFloat(lat_s) - 0.1 ;
    bound_left        = parseFloat(dlng_value) - 0.1;
    bound_right       = parseFloat(dlng_value) + 0.1;
}
```

2. ใน select\_table(server).php จะทำการดึงข้อมูลจากฐานข้อมูล โดยจะต้องมีการ สร้างการเชื่อมต่อกับฐานข้อมูล เลือกตาราง และคำสั่งที่ใช้ในการดึงข้อมูล

#### โปรแกรม 4.5 การเชื่อมต่อกับฐานข้อมูลและค่าที่ใช้ในการกำหนดขอบเขตการดึงข้อมูล

```
@mysql_connect("localhost","root","root") or die ("Connection Error
</body></html>");
mysql_select_db("poi_edit");
mysql_query("SET NAMES UTF8");
$x_u = $abound_up;
$x_d = $abound_down;
$y_r = $abound_right;
$y_l = $abound_left;
$sdd = $order_table1;
$result = mysql_query("SELECT * FROM $sdd WHERE lat between $x_d and
$x_u and lng between $y_l and $y_r ");
```

การกำหนดจำนวนข้อมูลที่ใช้ในการดึงจะต้องไม่ต่ำกว่า 4 และไม่มากกว่า 20 ข้อมูล จากการทดลอง ถือว่าเป็นจำนวนที่เหมาะสมและได้ตรงกับที่ออกแบบไว้

#### โปรแกรม 4.6 การกำหนดเงื่อนไขในการกำหนดจำนวนข้อมูลที่ดึงจากฐานข้อมูล

```

$num_rows= mysql_num_rows($result);
while($num_rows < 4 || $num_rows > 20)
{
    if ($num_rows <4)
    {
        $x_d = $x_d - 0.01;
        $x_u = $x_u + 0.01;
        $y_l = $y_l - 0.01;
        $y_r = $y_r + 0.01;
        $result = mysql_query("SELECT * FROM $sdd WHERE lat between $x_d and
        $x_u and lng between $y_l and $y_r ");
        $num_rows= mysql_num_rows($result);
    }
    else if ($num_rows > 20)
    {
        $x_d = $x_d + 0.001;
        $x_u = $x_u - 0.001;
        $y_l = $y_l + 0.001;
        $y_r = $y_r - 0.001;
        $result = mysql_query("SELECT * FROM $sdd WHERE lat between $x_d and
        $x_u and lng between $y_l and $y_r ");
        $num_rows= mysql_num_rows($result);
    }
    else if($num_rows > 200)
    {
        $x_d = $x_d + 0.003;
        $x_u = $x_u - 0.003;
        $y_l = $y_l + 0.003;
        $y_r = $y_r - 0.003;
        $result = mysql_query("SELECT * FROM $sdd WHERE lat between $x_d and
        $x_u and lng between $y_l and $y_r ");
        $num_rows= mysql_num_rows($result);
    }
}
}

```

3. ใน select\_table(server).php การส่งข้อมูลผลลัพธ์ที่ได้ไปที่ผู้ใช้งานโดยมีการเพิ่ม " |" ระหว่างข้อมูลย่อยในแต่ละอัน ซึ่งจะมีประโยชน์มากเพื่อนำมาใช้ในการจัดการประมวลผลที่ฝั่งผู้ใช้งาน

#### โปรแกรม 4.7 การแสดงผลลัพธ์ที่ได้จากการดึงข้อมูลจากฐานข้อมูล

```

for ($i = 0; $i<$num_rows;$i++)
{
    $id = mysql_result($result,$i,"id");
    $name = mysql_result($result,$i,"name");
    $lat_data = mysql_result($result,$i,"lat");
    $lng_data = mysql_result($result,$i,"lng");
    echo "{ $i } | { $id } | { $lat_data } | { $lng_data } | { $name } | ";
    echo "
}

```

4. ในโปรแกรม show\_select\_po.js จะทำการรับผลลัพธ์ที่ได้จาก select\_table() แล้วทำการส่งข้อมูลไปที่ server ทำงานประมวลผลตาม select\_table(server).php เพื่อให้ทำการดึงข้อมูลและส่งกลับมาที่ในส่วน div id = place\_poi ซึ่งในส่วนนี้จะใช้กระบวนการดึงข้อมูลเทคนิค ajax เพื่อไม่ให้มีการโหลดหน้าใหม่และมีการทำงานที่เหมาะสม

โปรแกรม 4.8 การนำผลลัพธ์มาแสดงที่เบราว์เซอร์ผู้ใช้งานผ่านการทำงานของ ajax เพื่อไม่ให้มีการโหลดหน้าใหม่

```

var xmlhttp;
function createXMLHttpRequest()
{
    if(window.ActiveXObject)
    {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    else if(window.XMLHttpRequest)
    {
        xmlhttp = new XMLHttpRequest();
    }
}
function show_select_poi()
{
    createXMLHttpRequest();
    var lat_current = document.getElementById("lat_value").value;
    var lng_current = document.getElementById("lng_value").value;
    var order_table1_current =
document.getElementById("order_table1").value;
    var abound_up_current =
document.getElementById("abound_up").value;
    var abound_down_current =
document.getElementById("abound_down").value;
    var abound_right_current =
document.getElementById("abound_right").value;
    var abound_left_current =
document.getElementById("abound_left").value;

    var url =
"http://localhost//poi//3_select_table(server).php?lat_value="+lat_cu
rrent+"&lng_value="+lng_current+
"&order_table1="+order_table1_current+"&abound_up="+abound_up
_current+"&abound_down="+abound_down_current+"&abound_right="+abound_
right_current+"&abound_left="+abound_left_current;

    xmlhttp.open("get", url, true);
    xmlhttp.onreadystatechange = handleStateChange;
    xmlhttp.send();
}
function handleStateChange()

```

```

{
    if(xmlHttp.readyState ==4 )
    {
        if(xmlHttp.status ==200)
        {
            document.getElementById("place_poi").innerHTML =
            xmlHttp.responseText;
        }
    }
}

```

## ผลลัพธ์

3. ข้อมูล POI ที่ได้จากรุ่นข้อมูล

0	180631	13.7255	100.781	คัชเชน
1	182244	13.7266	100.789	วิทยาลัยช่างศิลป์
2	182870	13.7269	100.785	โรงเรียนพรตพิทยพยัต
3	185456	13.7278	100.786	หอสมุดแห่งชาติเขตลาดกระบังเฉลิมพระเกียรติ
4	186412	13.7283	100.783	สถานีรถไฟหัวตะเข้
5	190005	13.7303	100.786	สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ
6	190262	13.7304	100.781	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าลาดกระบัง
7	190668	13.7306	100.781	สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ
8	190827	13.7308	100.789	สถานีรับสัญญาณดาวเทียม ศูนย์เทคโนโลยีอวกาศ
9	192672	13.7326	100.783	ร้านกวีวรรณ
10	193426	13.7331	100.783	บริษัท เค.ซี จำกัด
11	193601	13.7333	100.783	บ้านมิสซ

รูปที่ 4.10 ผลลัพธ์ที่ได้จากการดึงข้อมูลสถานที่ใกล้เคียง

### 4.2.3 ส่วนที่ 3 ส่วนการนำข้อมูล poi ที่ได้มาจัดเก็บลงในคลาส

จุดประสงค์ คือ นำข้อมูลที่ได้จากฝั่ง เซิร์ฟเวอร์ซึ่งเป็นเพียงข้อความเท่านั้นซึ่งไม่สะดวกในการนำมาประมวล จึงจะต้องมีกระบวนการในการเก็บข้อมูลที่ได้ในแต่ละชนิดจัดเก็บลงในคลาส place\_poi ที่ได้ออกแบบไว้ใน keep\_value\_class\_place\_poi.js

## วิธีการดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.หาระยะทางสถานที่ใกล้เคียงกับตำแหน่งปัจจุบัน

##### 4.1. keep\_value\_class\_place\_poi

รูปที่ 4.11 การสร้างฟอร์มเพื่อจัดเก็บข้อมูลลงตัวแปรเพื่อนำมาใช้ในการประมวลผลต่อไป

1. สร้างคลาส place\_poi ซึ่งมีส่วนประกอบดังนี้

โปรแกรม 4.9 การสร้างคลาสเพื่อใช้ในการจัดเก็บข้อมูล

```
Place_poi = function( oid, oname, olng, olat, odistance)
{
    this.oid_      = oid;
    this.oname_   = oname;
    this.olong_   = olng;
    this.olat_    = olat;
    this.odistance_ = odistance;
}
```

2. จัดเก็บข้อมูลลงคลาส place\_poi ที่ได้ทำการจัดเก็บข้อมูลตามที่ ออกแบบไว้โดยใช้ฟังก์ชัน split() เป็นหลักในการทำงาน

โปรแกรม 4.10 การจัดเก็บข้อมูลสถานที่ใกล้เคียง

```
function keep_value_class_place_poi()
{
    aPlace_poi = new Array();
    var i = 0;
    var nPlace_poi=0;
    var dplace_poi = document.getElementById("place_poi").value;
    var parts_dplace_poi = dplace_poi.split("|");
    nPlace_poi = parts_dplace_poi.length;
    nPlace_poi = parseInt(nPlace_poi) - 1;
    nPlace_poi_dist = nPlace_poi;

    for(var k = 0; k < nPlace_poi ; k = (parseInt(k)+5))
    {
        aPlace_poi[i] = new Place_poi();
        aPlace_poi[i].oid_      = parts_dplace_poi[(parseInt(k)+1)];
        aPlace_poi[i].olat_    = parts_dplace_poi[(parseInt(k)+2)];
        aPlace_poi[i].olong_   = parts_dplace_poi[(parseInt(k)+3)];
        aPlace_poi[i].odistance_ = 0;
        aPlace_poi[i].oname_   = parts_dplace_poi[(parseInt(k)+4)];
        i = parseInt(i) + 1;
    }
}
```

```

    }
    nArrayDistance = i;
}

```

#### 4.2.4 ส่วนที่ 4 ส่วน คำนวณระยะทางโดยใช้บริการของ Direction Service

จุดประสงค์ คือ คำนวณระยะทางระหว่างตำแหน่งปัจจุบันกับสถานที่ poi แต่ละจุด เพื่อที่จะนำผลลัพธ์ระยะทางแต่ละสถานที่มาเปรียบเทียบ เพื่อนำมาเลือกระยะทางที่ใกล้ที่สุดต่อไป

##### วิธีการดำเนินการ

1. ใน initialize\_distance.js ทำการกำหนดบริการที่เรียกใช้คือ Direction Service

##### โปรแกรม 4.11 การกำหนดการทำงานในการหาระยะทาง

```

function initialize_distance()
{
    var rendererOptions =
    {
        draggable: true
    };
    directionsDisplay_ = new
    google.maps.DirectionsRenderer(rendererOptions);
    directionsService_ = new google.maps.DirectionsService();
    var australia_ = new google.maps.LatLng(-25.274398, 133.775136);
    mapOptions_distance =
    {
        zoom: 7,
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        center: australia_
    };
    var map_distance = new
    google.maps.Map(document.getElementById('map_canvas_distance'),
    mapOptions_distance);
    directionsDisplay_.setMap(map_distance);
    before_com_dist();
}

```

2. ใน initialize\_distance.js กำหนดพิกัดตำแหน่งปัจจุบัน กับ ตำแหน่ง poi ที่ได้มา

##### โปรแกรม 4.12 การส่งข้อมูลพิกัดตำแหน่งสถานที่ใกล้เคียงไปคำนวณระยะทาง

```

function before_com_dist()
{
    destination_distance = new Array();
}

```

```

    for (var i_bcom_dist =0;      i_bcom_dist < nArryDistance ) ;
i_bcom_dist =parseInt(i_bcom_dist)+1 )
    {
        var lat_bcom_dist = aPlace_poi[ i_bcom_dist].olat_ ;
        var lng_bcom_dist = aPlace_poi[ i_bcom_dist].olng_ ;
destination_distance[i_bcom_dist]= lat_bcom_dist+","+lng_bcom_dist;
        calcRoute_distance(i_bcom_dist);
    }
}

```

3. ใน initialize\_distance.js การร้องขอบริการในการ คำนวณระยะทางจากที่มีการกำหนดไว้ มาเก็บใน aPlace\_poi[i\_bcom\_dist].odistance\_

**โปรแกรม 4.13** การคำนวณระยะทางระหว่างพิกัดตำแหน่งที่ต้องการหรือตำแหน่งปัจจุบันกับพิกัดตำแหน่งใกล้เคียงต่างๆ

```

function calcRoute_distance(i_bcom_dist)
{
var lat_value_distance = document.getElementById("lat_value").value;
var lng_value_distance = document.getElementById("lng_value").value;
var original_distance = lat_value_distance +","+lng_value_distance
;
var request_distance =
    {
        origin: original_distance,
        destination: destination_distance[i_bcom_dist],
        travelMode: google.maps.DirectionsTravelMode.DRIVING
    };
directionsService_.route(request_distance, function(response, status)
    {
        if (status == google.maps.DirectionsStatus.OK)
        {
            directionsDisplay_.setDirections(response);
            aPlace_poi[i_bcom_dist].odistance_ =
            response.routes[0].legs[0].distance.value;
        }
    });
}

```

4. ใน show\_initialize\_distance.js จะทำการแสดงผลลัพธ์ที่ได้จากการ คำนวณระยะทางใน div id = tpla\_adding

**โปรแกรม 4.14** การแสดงผลลัพธ์การคำนวณระยะทาง

```

function show_initialize_distance()
    var info_add_distance_ = "" ;

```

```

for ( var i_add_dist = 0 ; i_add_dist < nArryDistance ;
i_add_dist=parseInt(i_add_dist)+1 )
{
    info_add_distance_ = info_add_distance_ + "lat =
"+aPlace_poi[i_add_dist].olat_+" lng =
"+aPlace_poi[i_add_dist].olng_+" name =
"+aPlace_poi[i_add_dist].oname_+" dist =
"+aPlace_poi[i_add_dist].odistance_ + "
";
}
document.getElementById("tpla_adding").value = info_add_distance_ ;
}

```

## ผลลัพธ์

4.2. initialize\_distance

4.3. show\_initialize\_distance

ผลลัพธ์ที่ได้จากการหาระยะทาง

lat = 13.7255	lng = 100.781	name = คิมเขน	dist = 1340
lat = 13.7266	lng = 100.789	name = วิทยาลัยช่างศิลป์	dist = 2100
lat = 13.7269	lng = 100.785	name = โรงเรียนพรตพิทยพยัต	dist = 1535
lat = 13.7278	lng = 100.786	name = หอสมุดแห่งชาติเขตลาดกระบังเฉลิมพระเกียรติ	dist = 1646
lat = 13.7283	lng = 100.783	name = สถานีรถไฟหัวตะเข้	dist = 1260
lat = 13.7303	lng = 100.786	name = สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ	dist = 2050
lat = 13.7304	lng = 100.781	name = มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าลาดกระบัง	dist = 261
lat = 13.7306	lng = 100.781	name = สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ	dist = 941
lat = 13.7308	lng = 100.789	name = สถานีรับสัญญาณดาวเทียม ศูนย์เทคโนโลยีอวกาศ	dist = 2090
lat = 13.7326	lng = 100.783	name = ร้านเจริญวรรณ	dist = 6057
lat = 13.7331	lng = 100.783	name = บริษัท เค.ซี. จำกัด	dist = 6353
lat = 13.7333	lng = 100.783	name = บ้านมีสุข	dist = 6353

รูปที่ 4.12 ผลลัพธ์การคำนวณระยะทาง

4.2.5 ส่วนที่ 5 ส่วน การสร้างคลาสนำมาใช้ในการจัดเก็บข้อมูลเพื่อนำมาใช้ในการแปลงเป็นเสียง

จุดประสงค์ คือ ใช้ในการจัดเก็บข้อมูล ในกรณีที่มีการเลือกสถานที่ที่ใกล้เคียงที่สุดแล้ว อีกทั้งมีการเพิ่มเติม ส่วนประกอบในการอธิบายตำแหน่ง ร่องรับไว้ เพื่อที่ต่อมามีการอธิบายถึงตำแหน่งของสถานที่ใกล้เคียง ซึ่งประกอบด้วย

ขั้นที่ 1 ostep1\_direct\_t และ ostep\_distance\_t

ขั้นที่ 2 ostep2\_direct\_t และ ostep2\_distance\_t

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## วิธีการดำเนินการ

5. จัดลำดับระยะทางจากน้อยไปมาก (4 ลำดับ)

5.1 `keep_value_class_place_select`

รูปที่ 4.13 การสร้างฟอร์มเพื่อจัดเก็บข้อมูลสถานที่ใกล้เคียงที่ใกล้ที่สุดลงตัวแปรเพื่อนำมาใช้ในการประมวลผลต่อไป

1. สร้างคลาส Place\_poi\_temp ซึ่งประกอบไปด้วย

โปรแกรม 4.15 สร้างคลาสเพื่อจัดเก็บข้อมูลสถานที่ที่ใกล้เคียงที่สุด

```
Place_poi_temp = function(
  oname_t, oorder_t, odistance_t, olat_t, olng_t, ostep1_direct_t, ostep2_direct_t,
  ostep1_distance_t, ostep2_distance_t)
{
  this.oname_t_ = oname_t;
  this.oorder_t_ = oorder_t;
  this.odistance_t_ = odistance_t;
  this.olat_t_ = olat_t;
  this.olng_t_ = olng_t;
  this.ostep1_direct_t_ = ostep1_direct_t;
  this.ostep2_direct_t_ = ostep2_direct_t;
  this.ostep1_distance_t_ = ostep1_distance_t;
  this.ostep2_distance_t_ = ostep2_distance_t;
}
```

2. การนำข้อมูลมาจัดเก็บลงในคลาสที่ออกแบบไว้

โปรแกรม 4.16 การนำข้อมูลมาจัดเก็บ

```
aselect_neary = new Array();
function keep_value_class_place_select()
{
  for(var i_select_neary = 0; i_select_neary <= nArrayDistance ;
  i_select_neary = parseInt(i_select_neary) +1)
  {
    aselect_neary[i_select_neary] = new Place_poi_temp();
    aselect_neary[i_select_neary].oorder_t_ =
    aPlace_poi[i_select_neary].oorder_ ;
    aselect_neary[i_select_neary].oname_t_ =
    aPlace_poi[i_select_neary].oname_ ;
    aselect_neary[i_select_neary].odistance_t_ =
    aPlace_poi[i_select_neary].odistance_ ;
    aselect_neary[i_select_neary].olat_t_ =
    aPlace_poi[i_select_neary].olat_ ;
```

```

aselect_neary[i_select_neary].olng_t_ =
aPlace_poi[i_select_neary].olng_t_
}
}

```

#### 4.2.6 ส่วนที่ 6 ส่วนเปรียบเทียบระยะทาง

**จุดประสงค์** เปรียบเทียบระยะทางเพื่อเลือกระยะทางที่ใกล้ที่สุด 3 ลำดับแรกมาใช้ ในการอธิบายตำแหน่งปัจจุบัน

##### วิธีการดำเนินการ

1. กำหนดให้ระยะทางของข้อมูลแรกเป็นข้อมูลระยะทางที่น้อยที่สุด
2. จากนั้นเปรียบเทียบระยะทางนั้นกับทุกระยะทางของ สถานที่ ถ้า ระยะทางของสถานที่ที่ถูกนำมาเปรียบเทียบให้ทำการสลับ
3. จัดเก็บข้อมูลระยะทางที่น้อยที่สุดใน temp\_min
4. เลื่อนตำแหน่งที่เป็นตัวเปรียบเทียบไปที่ตำแหน่งถัดไป
5. ทำตามขั้นที่ 2 ถึง 4 จนกว่าจะได้ข้อมูลระยะทางที่ใกล้ที่สุด 3 ลำดับ

#### โปรแกรม 4.17 การจัดลำดับข้อมูลตามระยะทางที่น้อยที่สุด

```

function compare_distance()
{
    var A=0;
    var i_temp =1;
    var temp_dist;
    var temp_min = aselect_neary[0].odistance_t_ ;
    var i_min = 0;
    while(A <= 3) {
        if(temp_min >= aselect_neary[i_temp].odistance_t_ )
        {
            temp_min =
            aselect_neary[i_temp].odistance_t_ ;
            i_min = i_temp;

            i_temp = parseInt(i_temp) +1;
        }
        else
        {
            i_temp = parseInt(i_temp) +1;
        }

        if(i_temp > parseInt(nArrayDistance) -1 && A <= 3 )
        {
            aselect_neary[i_min].odistance_t_ = aselect_neary[A].odistance_t_ ;
            aselect_neary[A].odistance_t_ =
            aselect_neary[i_min].odistance_t_ ;
            aselect_neary[A].oname_t_ = aselect_neary[i_min].oname_t_ ;
        }
    }
}

```

```

aselect_neary[A].olat_t_           = aselect_neary[i_min].olat_t_
;
aselect_neary[A].olng_t_           =
aselect_neary[i_min].olng_t_ ;
aselect_neary[A].oorder_t_ = i_min;
A = parseInt(A)+ 1;
temp_min = aselect_neary[A].odistance_t_ ;
i_temp = A;
i_min = i_temp ;
    }
}

```

## ผลลัพธ์

5.2

5.3

การจัดรูปแบบการอธิบายตำแหน่งแสดงข้อมูลสถานที่ที่ใกล้ที่สุด 3 ลำดับ

lat = 13.7304	lng = 100.781	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี	dist = 261
lat = 13.7306	lng = 100.781	สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ	dist = 941
lat = 13.7283	lng = 100.783	สถานีรถไฟหัวตะเข้	dist = 1260

รูปที่ 4.14 ผลลัพธ์การจัดลำดับข้อมูลตามระยะทาง

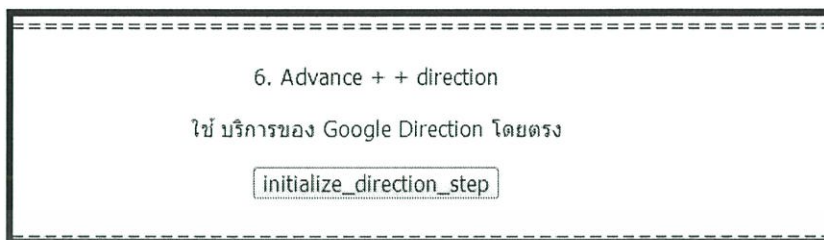
### 4.2.7 ส่วนที่ 7 ส่วนการอธิบายตำแหน่งปัจจุบันเพิ่มเติม

จุดประสงค์ คือ อธิบายรายละเอียดสถานที่ใกล้ เคียงเพิ่มเติมโดยมีการ อธิบายถึงทิศทาง และ ระยะทางในแต่ละขั้นตอน ซึ่งในการพัฒนาโปรแกรมจะมีการอธิบายเพียง 2 ขั้นตอนในแต่ละสถานที่ ใกล้เคียง หมายเหตุ การทำงานในส่วนนี้จะคล้ายกับส่วนที่ 4

แต่ที่มีการเรียกการใช้งานอีกครั้งเนื่องจาก การทำงานในส่วนอธิบายขั้นตอนสถานที่ที่จะใช้การ ประมวลผลที่ค่อนข้างใช้เวลา ดังนั้น ถ้าลดการประมวลผลให้มีการประมวลผลกับจำนวนข้อมูลที่ถูก เลือกแล้วจะช่วยลดเวลาได้มาก

### วิธีการดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 การสร้างฟอร์มเพื่อทดสอบการประมวลผลในการใช้งาน google dirction

1. มีขั้นตอนเหมือน ส่วนที่ 4 แต่ในส่วนจะใช้ประมวลผลกับข้อมูลที่ถูกรวบรวมแล้วเท่านั้น (สถานที่ใกล้เคียง 3 ลำดับแรก)

#### โปรแกรม 4.18 การกำหนดพิกัดตำแหน่งใกล้เคียงที่สุดเพื่อส่งไปหาระยะทาง

```
function com_dist()
{
    destination_direct = new Array();
    for (var i_com_dist =0 ; i_com_dist <= 2 ;i_com_dist
    =parseInt(i_com_dist)+1 )
    {
        var lat_com_dist = aselect_neary[i_com_dist].olat_t_
    ;
        var lng_com_dist = aselect_neary[i_com_dist].olng_t_
    ;
        destination_direct[i_com_dist]= lat_com_dist+","+lng_com_dist;
        calcRoute(i_com_dist);
    }
}
```

2. จัดเก็บข้อมูลที่รับมาจากการร้องขอ โดยข้อมูลจะอยู่ใน response.routes[ ]

#### โปรแกรม 4.19 การคำนวณระยะทางในแต่ละช่วงเมื่อมีการเปลี่ยนทิศทางจำนวน 2 ทิศทาง

```
function calcRoute(i_com_dist)
{
    var lat_value_direct = document.getElementById("lat_value").value;
    var lng_value_direct = document.getElementById("lng_value").value;
    var original_direct = lat_value_direct +","+lng_value_direct ;
    var request_direction = {
        origin: original_direct, /***
        destination: destination_direct[i_com_dist], // ***
        travelMode: google.maps.DirectionsTravelMode.WALKING
    };
    directionsService.route(request_direction, function(response, status)
    {
        if (status == google.maps.DirectionsStatus.OK) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุตบเบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

directionsDisplay.setDirections(response);

//step 1
aselect_neary[i_com_dist].ostep1_direct_t_ =
response.routes[0].legs[0].steps[0].instructions;
aselect_neary[i_com_dist].ostep1_distance_t_ =
response.routes[0].legs[0].steps[0].distance.text;
//step 2
aselect_neary[i_com_dist].ostep2_direct_t_ =
response.routes[0].legs[0].steps[1].instructions;
aselect_neary[i_com_dist].odistance_t_ =
response.routes[0].legs[0].distance.value;

aselect_neary[i_com_dist].odistance_t_ =
response.routes[0].legs[0].distance.value;
}
});
}

```

### ผลลัพธ์

```

7.แสดงผลลัพธ์ของ step ชื่อสถานที่ใกล้เคียงที่สุด 3 สถานที่ ซึ่งประกอบไปด้วย direction และ distance
show_step_direct_distance()
place 1 name = มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าลาดกระบัง step1 : direct1 = มุ่งไปทางทิศ <b>ตะวันตก</b>
place 2 name = สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ step1 : direct1 = มุ่งไปทางทิศ <b>ตะวันออก</b>
place 3 name = สถานีรถไฟหัวตะเภา step1 : direct1 = มุ่งไปทางทิศ <b>ตะวันตก</b> ไปตาม <b>ถนนเลลวงกรุง</b>

```

รูปที่ 4.16 ผลลัพธ์รายละเอียดเพื่ออธิบายตำแหน่งใกล้เคียงที่สุด 3 สถานที่

### 4.2.8 ส่วนที่ 8 ส่วนปรับแต่งข้อมูลที่ใช้ในการอธิบาย

จุดประสงค์ คือ เพื่อให้การอธิบาย ตำแหน่งมีความน่าสนใจ เข้าใจง่าย และมีประโยชน์ในการใช้งานมากขึ้น จะเห็นว่าตอนนี้เรามีข้อมูลเรียบร้อยแล้วซึ่งส่วนนี้มีความสำคัญมาก

#### วิธีการดำเนินการ

1. ข้อมูลชั้นตอนของแต่ละสถานที่ที่รับมาจะอยู่ใน < > เราจะต้องเอาข้อความที่ได้ออกมาจาก < >

### โปรแกรม 4.20 การปรับปรุงข้อมูลที่ได้โดยการตัด <> ออก

```

function edit_place_tts()
{
if (placetts_1_1 != "")
{placetts_1_1 = placetts_1_1.replace(/(<[^>+)>/ig, "");}
}

```

```

if (placetts_1_2 != "")
{placetts_1_2 =placetts_1_2.replace(/(<([>]+)>)/ig, "");}
if (placetts_1_3 != "")
{placetts_1_3 =placetts_1_3.replace(/(<([>]+)>)/ig, "");}
//2
if (placetts_2_1 != "")
{placetts_2_1 =placetts_2_1.replace(/(<([>]+)>)/ig, "");}
if (placetts_2_2 != "")
{placetts_2_2 =placetts_2_2.replace(/(<([>]+)>)/ig, "");}
if (placetts_2_3 != "")
{placetts_2_3 =placetts_2_3.replace(/(<([>]+)>)/ig, "");}
//3
if (placetts_3_1 != "")
{placetts_3_1 =placetts_3_1.replace(/(<([>]+)>)/ig, "");}
if (placetts_3_2 != "")
{placetts_3_2 =placetts_3_2.replace(/(<([>]+)>)/ig, "");}
if (placetts_3_3 != "")
{placetts_3_3 =placetts_3_3.replace(/(<([>]+)>)/ig, "");}

```

2. ตรวจสอบอาร์เรย์บางอาร์เรย์ที่ไม่มีข้อมูลอยู่ ซึ่งจะให้มีผลกระทบต่อการแปลงข้อความ เป็นเสียง โดยเราทำการปรับแต่งที่ทางฝั่งไคแอน โดยถ้าพบอาร์เรย์ใดที่ว่างหรือไม่มีข้อมูลให้ ทำการ แทนที่ด้วย "-" แทน

โปรแกรม 4.21 การตรวจสอบบางข้อมูลที่ไม่มีให้แทนด้วย “-” เพื่อไม่ให้เกิดข้อผิดพลาดในการ แปลงเป็นเสียง

```

var edplace = new Array ();
var i_ch_nu = 0;
iupl_6 = 1;
while(iupl_6 <= 6) /// **** จำนวนข้อมูล
{
    if( typeof(aselect_neary[i_ch_nu].ostep1_direct_t_) ==
'undefined' || typeof(aselect_neary[i_ch_nu].ostep1_distance_t_) ==
"undefined" )
    {
        edplace[iupl_6] = "-";
        iupl_6 = parseInt(iupl_6)+1;
    }
    else
    {
edplace[iupl_6] = " เดิน"+aselect_neary[i_ch_nu].ostep1_direct_t_+" เป็น
ระยะทาง "+aselect_neary[i_ch_nu].ostep1_distance_t_ ;
        iupl_6 = parseInt(iupl_6)+1;
    }
    if( typeof(aselect_neary[i_ch_nu].ostep2_direct_t_) ==
'undefined' || typeof(aselect_neary[i_ch_nu].ostep2_distance_t_) ==
"undefined" )
    {
edplace[iupl_6] = "-";
        iupl_6 = parseInt(iupl_6)+1;
    }
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            edplace[iup1_6] = " เดิน"+
aselect_neary[i_ch_nu].ostep2_direct_t_+" เป็นระยะทาง "+
aselect_neary[i_ch_nu].ostep2_distance_t_ ;
            iup1_6 = parseInt(iup1_6)+1;
        }
        i_ch_nu = parseInt(i_ch_nu)+1;
    }
//1
    if ( edplace[1] == "-" && edplace[2] == "-" )
    {placetts_1_1 = "สถานที่ 1 คือ "+aselect_neary[0].oname_t_ + " เป็นระยะทาง "+
aselect_neary[0].odistance_t_ ;} /**
    else {placetts_1_1 = "สถานที่ 1 คือ "+aselect_neary[0].oname_t_ ; }
//2
    if ( edplace[3] == "-" && edplace[4] == "-" )
    {placetts_2_1 = "สถานที่ 2 คือ "+aselect_neary[1].oname_t_ + " เป็นระยะทาง "+
aselect_neary[1].odistance_t_ ;} /**
    else {placetts_2_1 = "สถานที่ 2 คือ "+aselect_neary[1].oname_t_ ; }
//1
    if ( edplace[5] == "-" && edplace[6] == "-" )
    {placetts_3_1 = "สถานที่ 3 คือ "+aselect_neary[2].oname_t_ + " เป็นระยะทาง "+
aselect_neary[2].odistance_t_ ;} /**
    else {placetts_3_1 = "สถานที่ 3 คือ "+aselect_neary[2].oname_t_ ; }
//-----
-
placetts_1_2 = edplace[1] ;
placetts_1_3 = edplace[2] ;
placetts_2_2 = edplace[3] ;
placetts_2_3 = edplace[4] ;
placetts_3_2 = edplace[5] ;
placetts_3_3 = edplace[6] ;
}

```

### 3. การเพิ่มคำหรือข้อมูล และข้อมูล พร้อมทั้งการจัดลำดับข้อความเพื่อง่ายต่อการเข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ผลลัพธ์

8.2. จัดรูปแบบข้อความ ซึ่งจะแบ่งไฟล์การแปลงข้อความเป็นเสียง

สถานที่ปัจจุบัน place1

สถานที่ปัจจุบัน place2

สถานที่ใกล้เคียง place3

สถานที่ใกล้เคียง place4

สถานที่ใกล้เคียง place5

สถานที่ใกล้เคียง place6

สถานที่ใกล้เคียง place7

สถานที่ใกล้เคียง place8

สถานที่ใกล้เคียง place9

สถานที่ใกล้เคียง place10

สถานที่ใกล้เคียง place11

รูปที่ 4.17 ผลลัพธ์การปรับปรุงข้อมูล

## 4.2.9 ส่วนที่ 9 ส่วนแปลงข้อความเป็นเสียง

จุดประสงค์ คือ นำข้อความที่ได้จากการดัดแปลงแล้วมาแปลงเป็นเสียง โดยมีหลักการคือส่งข้อความที่ต้องการแปลงไปประมวลผลที่ google translate แล้วทำการดึงผลลัพธ์ที่ได้ซึ่งเป็นไฟล์นามสกุล .mp3 มาจัดเก็บที่ server ของระบบที่ได้สร้างขึ้น

วิธีการดำเนินการ

8.2 แปลงข้อความเป็นเสียง และ save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาร่วมกัน ไม่ขอสงวนสิทธิ์ในไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามรูปที่ 4.18 การสร้างฟอร์มเพื่อทดสอบการแปลงข้อมูลเป็นเสียงซึ่งมีการนำไปใช้

1. ใน create\_tts.js นำข้อความที่ได้จัดเก็บลงในตัวแปร p1\_c ถึง p11\_c

#### โปรแกรม 4.22 การส่งข้อมูลที่ปรับปรุงแล้วไปแปลงเป็นเสียงผ่านการทำงานของ ajax

```
function create_tts()
{
    createXMLHttpRequest_tts();

    var p1_c      = document.getElementById("data_p1").value ;
    var p2_c      = document.getElementById("data_p2").value ;
    //1
    var p3_c      = document.getElementById("data_p3").value ;
    var p4_c      = document.getElementById("data_p4").value ;
    var p5_c      = document.getElementById("data_p5").value ;
    //2
    var p6_c      = document.getElementById("data_p6").value ;
    var p7_c      = document.getElementById("data_p7").value ;
    var p8_c      = document.getElementById("data_p8").value ;
    //3
    var p9_c      = document.getElementById("data_p9").value ;
    var p10_c     = document.getElementById("data_p10").value;
    var p11_c     = document.getElementById("data_p11").value ;

    var url_tts   =
    "http://localhost/poi//13_tts(server).php?p1="+p1_c+"&p2="+p2_c+"&p3="
    +"p3_c+"&p4="+p4_c+"&p5="+p5_c+"&p6="+p6_c+"&p7="+p7_c+"&p8="+p8_c+"&
    &p9="+p9_c+"&p10="+p10_c+"&p11="+p11_c;

    xmlhttp_tts.open("get",url_tts,true);
    xmlhttp_tts.onreadystatechange = handleStateChange_tts;
    xmlhttp_tts.send();
}

```

2. ใน create\_tts.js ส่งข้อความไปที่ server เราได้สร้างขึ้นเป็นตัวจัดการประมวลผล เป็นตัวแทนส่งไปประมวลผลที่ google translate ผ่าน url\_tts
3. ใน create\_tts.js จัดเก็บผลลัพธ์ที่ฝั่ง server

#### โปรแกรม 4.23 สถานะในการส่งข้อมูล

```
function handleStateChange_tts()
{
    if(xmlHttpRequest_tts.readyState ==4 ) // readyState = 4 แสดงส่งข้อเรียบร้อย
    {
        if(xmlHttpRequest_tts.status ==200) // stats = 200 ข้อมูลถูกส่งมา และ
        สมบูรณ์
        document.getElementById("tts_tts").innerHTML =
        xmlhttp_tts.responseText;
    }
}

```

4. ใน tts(server).php จะทำงานในฝั่ง server ทำการส่งข้อความที่ได้รับจากฝั่งไคลเอนต์ ส่งไปประมวลผลที่ google translate

#### โปรแกรม 4.24 การแปลงข้อความเป็นเสียงจากการประมวลผลของ google tts

```

$info_place = ta array();
$name_file_place = array();
$info_place[1] = $p1;
$info_place[2] = $p2;
$info_place[3] = $p3;
$info_place[4] = $p4;
$info_place[5] = $p5;
$info_place[6] = $p6;
$info_place[7] = $p7;
$info_place[8] = $p8;
$info_place[9] = $p9;
$info_place[10] = $p10;
$info_place[11] = $p11;
$sch_name = 'a';
for($i_tts_save = 1; $i_tts_save <= 11; $i_tts_save++)
{
$name_file_place[$i_tts_save] = $sch_name.'.mp3';
if ($info_place[$i_tts_save] != "")
{
$rege = "http://translate.google.com/translate_tts?ie=utf-8&tl=th&q=".rawurlencode($info_place[$i_tts_save]); //*****
if ($soundfile = file_get_contents($rege))
{
file_put_contents($name_file_place[$i_tts_save], $soundfile);
}
}
}

```

5. ใน tts(server).php เมื่อส่งไปแล้วทำการดึงข้อมูลที่ได้ออกมาเป็นไฟล์นามสกุล .mp3 แล้วทำการบันทึก โดยไฟล์ที่ได้จะถูกจัดเก็บไม่แยกโฟลเดอร์

6. ใน tts(server).php เพื่อให้มีการจัดเก็บไฟล์ที่บันทึกที่มีการจัดเก็บที่เป็นระบบ จะการย้ายไฟล์ไปเก็บไว้ในโฟลเดอร์ sound ซึ่งก่อนย้ายจะต้องการจัดเก็บไฟล์เสมอทำการตรวจสอบว่า

#### โปรแกรม 4.25 การดึงผลลัพธ์ที่เป็นไฟล์เสียงที่แปลงแล้วกลับมาจากการประมวลผล

```

if ($soundfile = file_get_contents($rege))
{
file_put_contents($name_file_place[$i_tts_save], $soundfile);
$oldfile = "C:/AppServ/www/POI/".$name_file_place[$i_tts_save];
$newfile = "C:/AppServ/www/POI/sound/".$name_file_place[$i_tts_save];
if (file_exists($oldfile)) //การตรวจสอบการมีอยู่ของไฟล์

```

```

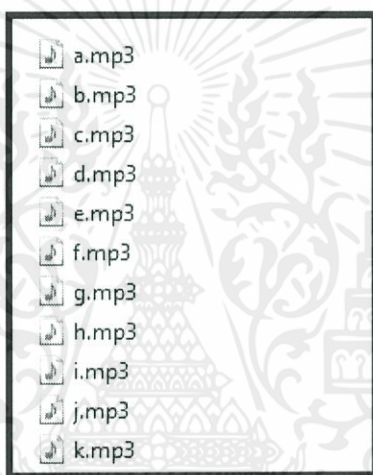
{
    $movefile = (dirname($oldfile)== dirname($newfile)) ;

    rename($oldfile,$newfile);
}
}

```

### ผลลัพธ์

ได้ไฟล์เสียง ชื่อ a ถึง k ที่มีนามสกุล .mp3 และถูกจัดเก็บที่โฟลเดอร์ sound []



รูปที่ 4.19 ผลลัพธ์ที่ได้จากการแปลงข้อความเป็นเสียง

#### 4.2.10 ส่วนที่ 10 ส่วนรวมไฟล์เสียง

จุดประสงค์ คือ เพื่อให้การเมื่อเปิดฟังการอธิบายการบอกตำแหน่งมีความต่อเนื่องกัน และให้ตรงกับการแก้ปัญหาที่ได้ออกแบบไว้

##### วิธีการดำเนินการ

1. การพัฒนาส่วนนี้จะมีการ include ไลบรารี lib\_merge.php มาเพิ่มเติม การดำเนินการในส่วนนี้จะประมวลผลที่ฝั่ง server โดยต้องการรวมไฟล์ให้ได้ ทั้งหมด 3 ไฟล์คือ ไฟล์แรก เป็นข้อมูลตำแหน่งปัจจุบัน และตำแหน่งใกล้เคียงที่ 1 เกิดจากการรวมไฟล์เสียง a.mp3 ,b.mp3 , c.mp3 และ e.mp3

ไฟล์ที่ 2 เป็นข้อมูลตำแหน่งใกล้เคียงที่ 2 เกิดจากการรวมไฟล์เสียง f.mp3 , g.mp3 และ h.mp3

ไฟล์ที่ 3 เป็นข้อมูลตำแหน่งใกล้เคียงที่ 3 เกิดจากการรวมไฟล์เสียง i.mp3 , j.mp3 และ k.mp3

2. ทำการตรวจสอบว่ามีไฟล์หรือไม่ ก่อนรวมไฟล์

### โปรแกรม 4.26 การรวมไฟล์เสียง

```
include ("lib_merge.php");
$state_outloop = 0;
for ( $i_mer = 0;$i_mer <=2;$i_mer ++)
{
    $state_outloop = 0;

    while($state_outloop != 1){
    if($i_mer == 0 && file_exists("C:/AppServ/www/POI/sound/a.mp3") &&
file_exists("C:/AppServ/www/POI/sound/b.mp3") &&
file_exists("C:/AppServ/www/POI/sound/c.mp3") &&
file_exists("C:/AppServ/www/POI/sound/d.mp3") &&
file_exists("C:/AppServ/www/POI/sound/e.mp3") )
    {$petersword = "abcde"; $state_outloop = 1;}
    else if($i_mer == 1 && file_exists("C:/AppServ/www/POI/sound/f.mp3")
&& file_exists("C:/AppServ/www/POI/sound/g.mp3") &&
file_exists("C:/AppServ/www/POI/sound/h.mp3"))
    {$petersword = "fgh";$state_outloop = 1;}

    else if($i_mer == 2 && file_exists("C:/AppServ/www/POI/sound/i.mp3")
&& file_exists("C:/AppServ/www/POI/sound/j.mp3") &&
file_exists("C:/AppServ/www/POI/sound/k.mp3"))
    {$petersword = "ijk";$state_outloop = 1;}
    }

    $word_count = strlen($petersword);

    // Set up the first file
    if ($word_count > 0)
    {
        $mp3 = new mp3($cas_fontpath . 'sound/' . substr($petersword, 0, 1)
        . '.mp3');
        $mp3->striptags();
    }

    // Generate the mp3 file from each letter in the word
    for ($i = 1; $i < $word_count; ++$i)
    {
        $cas_character = $cas_fontpath . 'sound/' . substr($petersword, $i,
        1);
        $cas_mp3equivalent = new mp3($cas_character . '.mp3');
        $mp3->mergeBehind($cas_mp3equivalent);
        $mp3->striptags();
    }
}
```

### 3. จัดเก็บผลลัพธ์ไฟล์เสียงที่รวมได้ ลงในโฟลเดอร์ sound

### โปรแกรม 4.27 การจัดเก็บไฟล์เสียงลงในโฟลเดอร์ sound

```
for ($i = 1; $i < $word_count; ++$i)
{
    $cas_character = $cas_fontpath . 'sound/' . substr($petersword, $i,
    1);
```

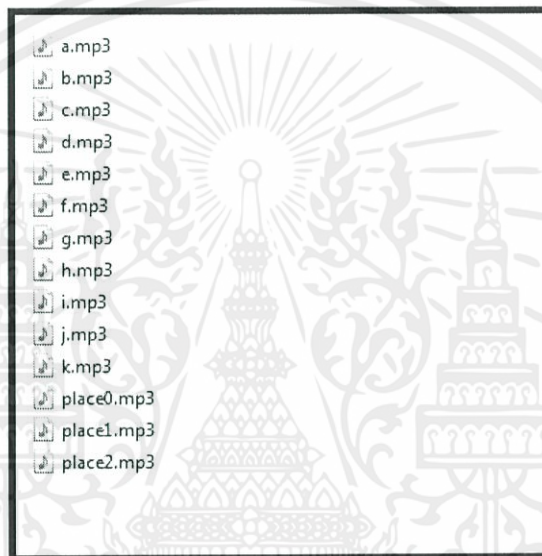
```

$cas_mp3equivalent = new mp3($cas_character . '.mp3');
$mp3->mergeBehind($cas_mp3equivalent);
$mp3->striptags();
}

// Spit out the audio file!
$mp3->output('sound/place'.$i_mer.'.mp3');

```

## ผลลัพธ์



รูปที่ 4.20 ผลลัพธ์ที่ได้จากการรวมไฟล์เสียง

### 4.2.11 ส่วนที่ 11 ลบไฟล์เสียงทั้งหมด

จุดประสงค์ คือ เพื่อเตรียมพื้นที่ให้ว่างสำหรับการสร้างไฟล์เสียงใหม่ ป้องกันการเกิดข้อผิดพลาดในการประมวลผล

#### วิธีการดำเนินการ

1. ทำการตรวจสอบว่าไฟล์ที่จะลบมีอยู่จริง
2. ลบไฟล์ทั้งหมดที่มี

### โปรแกรม 4.28 การลบไฟล์เสียงในโฟลเดอร์ sound

```

$place_file_del = 'place';
for($i_test = 0;$i_test <=2;$i_test++)
    $fname_place = $place_file_del.$i_test.".mp3" ;
    if(!file_exists("C:/AppServ/www/POI/sound/".$fname_place))
    {

```

```

        unlink("C:/AppServ/www/POI/sound/".$fname_place); // ฟังก์ชันลบไฟล์ที่มี
พาร images/test.jpg
    }
}
$char_file_del = 'a' ;
//delete file ก่อนสร้าง
for($i_test = 1;$i_test <=11;$i_test++)
{
    $fname_place = $char_file_del.".mp3" ;
    if (file_exists("C:/AppServ/www/POI/sound/".$fname_place))
    {
        unlink("C:/AppServ/www/POI/sound/".$fname_place); // ฟังก์ชันลบไฟล์ที่มี
พาร images/test.jpg
    }
    $char_file_del++;
}
}

```

#### 4.3 ส่วนพัฒนาระบบเมื่อนำฟังก์ชันต่างๆมารวมกัน

การนำโปรแกรมย่อยแต่ละส่วนมารวมกัน โดยมี การ ควบคุมทำงานโดยมีปุ่ม Play และปุ่ม Pause ซึ่งมีการวางเงื่อนไขในการประมวลผล

จุดประสงค์ คือ จะใช้ปุ่ม Play และปุ่ม Pause เป็นปุ่มที่ใช้ในการควบคุมการทำงาน ซึ่งการทำงานภาพรวมคือ เมื่อกดปุ่ม play จะมีการวางเงื่อนไขในการตรวจสอบว่าตำแหน่งพิกัดผู้ใช้งานกดนั้นอยู่ในบริเวณตำแหน่งที่กำหนดหรือไม่ เพื่อจะได้ไม่ต้องมีการสร้างไฟล์เสียงใหม่ จากการพัฒนาลำดับที่ 2 ผลลัพธ์ที่จะเป็นไฟล์เสียงอธิบายการบอกตำแหน่งจำนวน 3 ไฟล์คือ place0.mp3 , place1.mp3 และ place2.mp3 ถ้าอยู่ในขอบเขตจะทำการเรียกไฟล์ place0.mp3 ฟัง และถ้าตรวจสอบว่าเงื่อนไขนี้เป็นจริงตลอดก็จะเล่นไฟล์เสียงต่อไป แต่เมื่อกดปุ่ม Pause จะทำการหยุดไฟล์เสียงที่ฟังอยู่ในขณะนั้น

วิธีการดำเนินการ

1. สร้างปุ่มเพื่อควบคุมในการทำงาน คือปุ่ม Play และ ปุ่ม Pause

#### โปรแกรม 4.29 การสร้างปุ่ม play และปุ่ม pause

```

<button name="listen" type="button" href="#" id="play" value
="listen"></button>
<input name="pause" type="button" href="#" id="pause" value ="pause">

```

2. เขียนโปรแกรมเงื่อนไขเมื่อกดปุ่ม Pause

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### โปรแกรม 4.30 การกำหนดเงื่อนไขเมื่อกดปุ่ม pause

```

$('#pause').click(function(event)
{
    for( var i_pause = 2 ; i_pause >= 0 ; i_pause =
    parseInt(i_pause)-1)
    {
        music[i_pause].pause();
    }
});

```

### 3. เขียนโปรแกรมเงื่อนไขเมื่อกดปุ่ม Play

#### โปรแกรม 4.31 การกำหนดเงื่อนไขเมื่อกดปุ่ม play

```

ch_pause_ = 0;
num_click = -1;
lat_compare = 0;
$('#play').click (function(event)
{
    if (lat <= parseFloat(lat_compare) +0.001 && lat >=
    parseFloat(lat_compare) - 0.001 )
    {
        alert("n = "+num_click);
        playMusic();
        num_click = parseInt(num_click )+1;
        alert (" ใจของเดิมเจ้า !!");
        if(num_click== 3)
        {
            num_click = 0;
        }
        if(num_click != -1 && ch_pause_ == 1 )
        {
            if( num_click > 0 && num_click <= 2)
            {
                ch_pause_ = 1; // ถ้ามีการกดมากกว่า 1 ครั้ง
                pause_ = parseInt(num_click) - 1;
                music[pause_].pause();
            }
            else if(num_click == 0)
            {
                pause_ = 2;
                music[pause_].pause();
            }
        }
        alert(num_click);
        music[num_click].play();
    }
});

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ alert(num\_click); ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. วางเงื่อนไขเมื่อตำแหน่งปัจจุบันอยู่ในขอบเขตที่กำหนด จะทำการเปิดไฟล์เสียงเสียงที่สร้างไว้ก่อนหน้านี้

5. วางเงื่อนไขเมื่อตำแหน่งปัจจุบันหรือตำแหน่งที่กำหนดไม่ได้อยู่ในขอบเขตที่กำหนด จะทำการสร้างไฟล์เสียงเพื่อใช้ในการอธิบายตำแหน่งปัจจุบัน

โปรแกรม 4.32 การกำหนดเงื่อนไขเมื่อตำแหน่งปัจจุบันหรือตำแหน่งที่กำหนดไม่ได้อยู่ในขอบเขตที่กำหนด

```

else if(lat_compare == 0 || lat > parseFloat(lat_compare) + 0.001 ||
lat < parseFloat(lat_compare) - 0.001 )
{
    state_block = 14;
    if(state_block == 14 )
    {
        fsound_delete() ;
        state_block = 0;
    }
    lat_compare = lat ;
    control_lisen( );
};

function control_lisen( )
{
    if(state_block == 0 )
    {
        select_table();
        state_block = 1;
    }
    if(state_block == 1)
    document.getElementById("num_table").value != "" )
    {
        show_select_poi();
        state_block = 2;
    }
    if(state_block == 2 )
    {
        window.setTimeout(" keep_value_class_place_poi();" , 500) ;
        state_block = 3;
    }
    if(state_block == 3 )
    {
        window.setTimeout(" initialize_distance();" ,1000) ;
        state_block = 4;
    }
    if(state_block == 4 )
    {
        window.setTimeout("show_initialize_distance();" ,2000);
        state_block = 5;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(state_block == 5)
{
window.setTimeout("keep_value_class_place_select();" ,3000)
;

state_block = 6;
}
if(state_block == 6)
{
window.setTimeout("compare_distance();" ,3500) ;
state_block = 7;
}
if(state_block == 7)
{
window.setTimeout("show_compare_distance();" ,3800) ;
state_block = 8;
}
if(state_block == 8)
{
window.setTimeout("initialize_direction_step();" , 4500) ;
state_block = 9;
}
if(state_block == 9)
{
window.setTimeout("show_step_direct_distance();" , 5500) ;
state_block = 10;
}
if(state_block == 10)
{
window.setTimeout("edit_place_tts();" , 5700) ;
state_block = 11;
}
if(state_block == 11)
{
window.setTimeout("create_tts();" ,6000) ;
state_block = 12;
}
if(state_block == 12)
{
window.setTimeout("mer_mp3();" ,16000) ;
state_block = 13;
}
if(state_block == 13)
{
alert("play_music");
window.setTimeout("playMusic();" ,20000) ;

num_click = 0;
window.setTimeout("music[num_click].play();" ,30000) ;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ผลลัพธ์



รูปที่ 4.21 ผลลัพธ์จากการรวมโปรแกรมย่อย

## 4.4 ส่วนการอัปโหลดเว็บไซต์เข้าสู่เว็บเซิร์ฟเวอร์

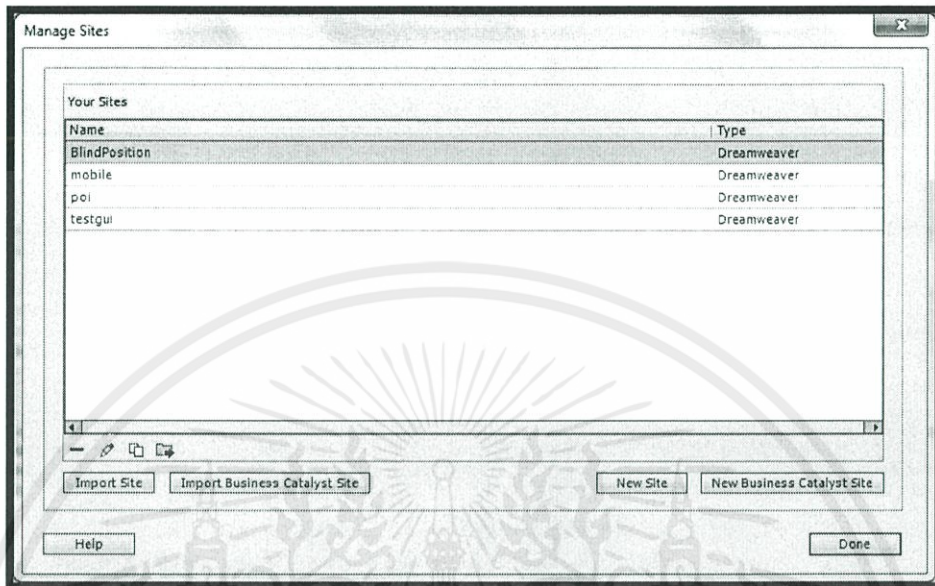
การสร้างเว็บไซต์เมื่อเสร็จสมบูรณ์แล้วจะต้องอัปโหลด (Upload) ข้อมูลทั้งหมดเข้าสู่อินเทอร์เน็ตโดยนำไปฝากหรือเก็บไว้บนเว็บเซิร์ฟเวอร์ เพื่อเผยแพร่ไปสู่สาธารณะชน ขณะที่ข้อมูลถูกเผยแพร่อยู่นั้น ผู้ดูแลหรือเจ้าของเว็บไซต์สามารถปรับปรุง แก้ไข เปลี่ยนแปลงเนื้อหา หรืออัปเดตข้อมูลใหม่ ๆ ได้ตลอดเวลา การจัดการเว็บไซต์มีขั้นตอนสำคัญ 3 ขั้นตอน คือ

1. จดทะเบียนโดเมนเนม (Domain Name)
2. จัดหาพื้นที่ฝากเว็บไซต์บนเว็บเซิร์ฟเวอร์
3. อัปโหลดไฟล์เข้าสู่เว็บเซิร์ฟเวอร์

ในที่นี้โครงการได้ใช้บริการของ <http://www.bangkokstatus.com> มีวิธีดังต่อไปนี้

1. เปิดโปรแกรม Dreamweaver (อาจใช้โปรแกรมสำหรับอัปโหลดโดยตรง เช่น CuteFTP, FTP Voyager ก็ได้) และเรียกไฟล์ index.html ขึ้นมา
- คลิกที่คำสั่ง Site >> Manage Sites จะได้หน้าต่าง Manages Sites คลิกที่ชื่อไซต์ที่จะอัปโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 การสร้างไซต์

- คลิกที่แท็บ Server เลือก ชื่อ Sever name ที่ได้สร้างไว้ และตั้งค่าให้ Connecting Using เป็น FTP และ ใส่ Login และ Password ตามที่เราได้ลงทะเบียนไว้แล้ว คลิกปุ่ม Test เพื่อทดสอบการต่อเชื่อมกับเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Basic Advanced

Server Name:

Connect using:

FTP Address:  Port:

Username:

Password:   Save

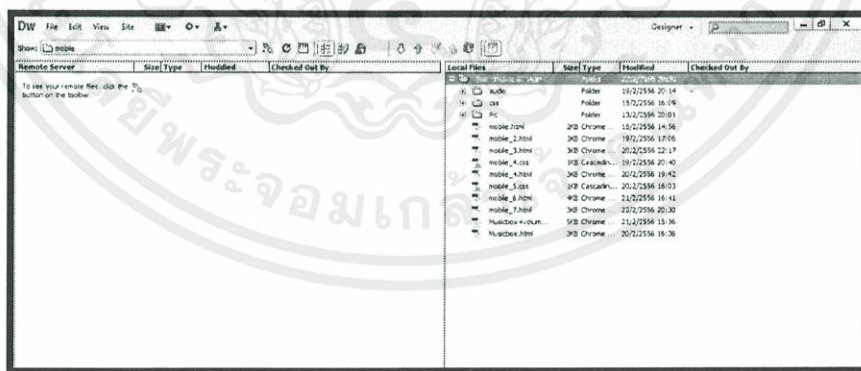
Root Directory:

Web URL:

▶ More Options

รูปที่ 4.23 การกำหนด ftp เพื่อใช้ในการอัปโหลดไฟล์ระบบ

3. ติดต่อกับเซิร์ฟเวอร์แล้วคลิกที่  ปุ่มพาเนลไฟล์ จะได้ดังนี้



รูปที่ 4.24 การเชื่อมต่อกับ server ที่ต้องการอัปโหลดไฟล์ระบบ

4. คลิกที่ปุ่ม  Put file เพื่อเลือกไฟล์จาก Local files ไปไว้ที่ Remote Site จะแสดงสถานะการอัปโหลดไฟล์

5. เมื่ออัปโหลดเสร็จแล้วไฟล์ทั้งหมดจะปรากฏที่หน้าต่าง Remote Site หรือบนเซิร์ฟเวอร์



ตาราง 4.1 แสดงผลลัพธ์การทดสอบการทำงานของระบบบนเครื่องคอมพิวเตอร์

ครั้งที่ทดสอบ	ระยะเวลาการตอบสนอง	ความถูกต้อง
1	10.54	ถูกต้อง
2	10.59	ถูกต้อง
3	10.30	ถูกต้อง
4	10.70	ถูกต้อง
5	10.00	ถูกต้อง
6	10.03	ถูกต้อง
7	10.35	ถูกต้อง
8	10.09	ถูกต้อง
9	10.24	ถูกต้อง
10	10.81	ถูกต้อง
11	10.49	ถูกต้อง
12	10.57	ถูกต้อง
13	10.80	ถูกต้อง
14	10.39	ถูกต้อง
15	10.90	ถูกต้อง
16	10.23	ถูกต้อง
17	10.56	ถูกต้อง
18	10.12	ถูกต้อง
19	10.53	ถูกต้อง
20	10.10	ถูกต้อง

ผลลัพธ์ มีความถูกต้องประมาณ 100 เปอร์เซ็นต์ และแสดงการคำนวณ

ระยะเวลาในการตอบสนองโดยประมาณ 10.41 วินาที ต้องใช้เวลาในการประมวลผลพอสมควร ทั้งนี้

ยังขึ้นกับความเร็วของอินเทอร์เน็ตและประสิทธิภาพของเซิร์ฟเวอร์ด้วย

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์การทดสอบการทำงานของระบบบนโทรศัพท์มือถือ

ตาราง 4.2 แสดงผลลัพธ์การทดสอบการทำงานของระบบบนโทรศัพท์มือถือ

ครั้งที่ทดสอบ	ระยะเวลาการตอบสนอง	ความถูกต้อง
1	14.13	ถูกต้อง
2	15.64	ถูกต้อง
3	15.30	ถูกต้อง
4	16.50	ถูกต้อง
5	14.20	ถูกต้อง
6	15.40	ถูกต้อง
7	16.06	ถูกต้อง
8	16.35	ถูกต้อง
9	16.17	ถูกต้อง
10	15.84	ถูกต้อง
11	15.70	ถูกต้อง
12	17.00	ถูกต้อง
13	15.45	ถูกต้อง
14	16.91	ถูกต้อง
15	14.89	ถูกต้อง
16	16.51	ถูกต้อง
17	17.83	ถูกต้อง
18	17.92	ถูกต้อง
19	16.15	ถูกต้อง
20	15.36	ถูกต้อง

ผลลัพธ์ มีความถูกต้องประมาณ 100 เปอร์เซ็นต์ และแสดงการคำนวณ

ระยะเวลาในการตอบสนองโดยประมาณ 15.97 วินาที ซึ่งใช้เวลาในการประมวลผลนานกว่าบน

คอมพิวเตอร์ เป็นเพราะประสิทธิภาพของอุปกรณ์ประกอบกับความเร็วของอินเทอร์เน็ตและ

ประสิทธิภาพของเซิร์ฟเวอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 บทสรุป

ผลลัพธ์ที่ได้จากการพัฒนาระบบคือ

ได้โปรแกรมตรงตามที่ได้ออกแบบไว้ ซึ่งสามารถอธิบายตำแหน่งปัจจุบันโดยมีกรอ้างอิงสถานที่ใกล้เคียงมาช่วยการอธิบายออกมาเป็นเสียง

ผลลัพธ์ การสามารถนำไปใช้อธิบายได้คร่าวๆแต่ยังไม่ละเอียดมากนัก แต่เชื่อว่าน่าจะประโยชน์กับผู้ใช้งานที่มีความบกพร่องทางสายตาสายตาเพราะออกแบบมาให้เหมาะสม ใช้งานง่าย

โดยระบบพัฒนาออกมาแบ่งออกเป็น 2 เวอร์ชันคือ

1. เวอร์ชันสำหรับ การทดลองและทดสอบการทำงาน ซึ่งสามารถนำไปพัฒนาต่อยอดได้สำหรับผู้ที่มีความสนใจ สามารถเพิ่มเติมฟังก์ชันต่างๆให้ระบบมีความน่าสนใจมากขึ้นโดยจะมีการแสดงข้อมูลอย่างละเอียด พร้อมปุ่มทดสอบในแต่ละขั้นตอน

2. เวอร์ชันสำหรับการใช้งาน มีการทำงานทุกอย่างเหมือนเวอร์ชันที่ใช้ทดลอง แต่มีการแสดงผลเฉพาะส่วนที่จำเป็นเท่านั้น คือจะมีเพียง ส่วนแสดงแผนที่และปุ่มควบคุมการทำงานเพียง Play และ Pause เท่านั้น

ได้รับความรู้ในการใช้งาน api ต่างๆโดยเฉพาะอย่างยิ่ง google map api ซึ่งถือว่ามีความน่าสนใจ และมีประโยชน์อย่างมาก

ได้รับความรู้ในการพัฒนาเว็บแอปพลิเคชัน

ได้รับความรู้ในการสร้างฐานข้อมูล การเชื่อมต่อกับระบบที่สร้างขึ้น

ได้เรียนรู้เทคโนโลยีในปัจจุบัน ยกตัวอย่างเช่น ajax , html5 และ jquery เป็นต้น ซึ่งมีประโยชน์ในพัฒนาเว็บแอปพลิเคชันให้มีความสะดวกและมีประสิทธิภาพมากขึ้น

ได้เรียนรู้ภาษาสคริป ซึ่งเป็นภาษาที่มีประสิทธิภาพอย่างมากโดยเฉพาะอย่างยิ่ง ภาษา

PHP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

1. ปัญหาในการตรวจสอบการทำงานของโปรแกรมที่เขียนโดยภาษา javascript

### แนวทางการแก้ไข

ทำการแทรก alert ในบรรทัดที่คิดว่ามีปัญหา ตรวจสอบการประมวลผลในแต่ละบรรทัด ซึ่งสามารถแก้ปัญหานี้ได้อย่างมาก เพราะการประมวลผลของ javascript จะทำการประมวลผลทีละบรรทัดต่อกัน

2 . ปัญหาการเชื่อมต่อข้อมูลระหว่าง ผู้ใช้งานและผู้ server ทำการรับส่งข้อมูล โดยไม่สูญเสียข้อมูล ไม่ต้องเปลี่ยนหน้าเวลาโหลดข้อมูล

### แนวทางการแก้ไข

ในการเขียนโปรแกรมโดยใช้เทคโนโลยีของ ajax

3. ปัญหาการส่งข้อความที่เป็นภาษาไทยส่งไปแปลงเป็นเสียงที่ google translate

### แนวทางการแก้ไข

ทำการเข้ารหัสข้อความก่อนที่จะส่งไปที่ server ของ google ก่อน

## 5.3 ข้อเสนอแนะ

1. เพื่อให้ได้ผลลัพธ์ในการอธิบายถึงสถานที่ใกล้เคียงให้มีความละเอียดมากขึ้น ควรมีการจัดเก็บเก็บข้อมูลสถานที่จุดสนใจ (POI) ให้มากขึ้น
2. สามารถนำแนวคิดและวิธีการแก้ปัญหานี้ไปใช้ในการพัฒนาแอปพลิเคชันบนมือถือได้ ซึ่งจะช่วยเพิ่มความสะดวกสบายในการใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- AMHN. "Google Map Api." [Online]. Available :  
<http://www.coastalaqua.com/webboard/index.php?topic=3360.0>. 2009.
- Google Developer. "Geolocation." [Online]. Available :  
<https://developers.google.com/maps/articles/geolocation>. 2012
- Google Developer. "Geocoding Service." [Online]. Available :  
<https://developers.google.com/maps/documentation/javascript/geocoding>. 2012.
- Google Developer. "Direction." [Online]. Available :  
<https://developers.google.com/maps/documentation/javascript/directions>. 2012.
- Mathieu Imbert. "How to play a Google Text-To-Speech URL." [Online]. Available :  
<http://www.appservnetwork.com/modules.php?name=Content&pa=showpage&pid=3>. 2006
- ryuozero. "Web Server คืออะไร ?." [Online]. Available :  
<http://ryuozero.exteen.com/20070612/web-server>. 2007.
- Support. "SQL คือ อะไร ?." [Online]. Available :  
<http://www.softwaresiam.com/index.php/access/11-sql>. 1999.
- Wikipedia. "มายเอสคิวเอล." [Online]. Available : <http://th.wikipedia.org/wiki/Mysql>. 2012.
- Wikipedia. "ภาษาพีเอชพี." [Online]. Available : <http://th.wikipedia.org/wiki/PHP>. 2012.
- Wikipedia. "Point of interest." [Online]. Available :  
[http://en.wikipedia.org/wiki/Point\\_of\\_interest](http://en.wikipedia.org/wiki/Point_of_interest). 2008.
- Web Server. "เว็บเซิร์ฟเวอร์" [Online]. Available : <http://www.mindphp.com/บทความ/66-server-hosting/1848-web-server-คืออะไร-web-server-หมายถึง.html>. 2010.
- Anonymous. "HTML" [Online]. Available : <http://www.codingbasic.com/html.html>. 2011.
- Anonymous. "HTML5" [Online]. Available :

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การคัดลอกหรือเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย  
ไม่ว่ากรณีใดก็ตาม อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Anonymous. “Java Script” [Online]. Available :

[www2.cvc.ac.th/trsai/it51/39012009/JavaScript.doc](http://www2.cvc.ac.th/trsai/it51/39012009/JavaScript.doc). 2011.

Anonymous. “Ajax” [Online]. Available : <http://www.com5dow.com\ไขปัญหาศัพท์-IT/655-Ajax-คืออะไร.html>. 2012

กาญพงศ์ ปัญญาดี. “Appserve” [Online]. Available :

<http://www.appservnetwork.com/modules.php?name=Content&pa=showpage&pid=3>. 2009.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้