

การวิเคราะห์ข้อมูลทวีตเตอร์เพื่อการเตือนภัยน้ำท่วม
ANALYSIS OF TWITTER DATA FOR EARLY FLOOD
WARNING

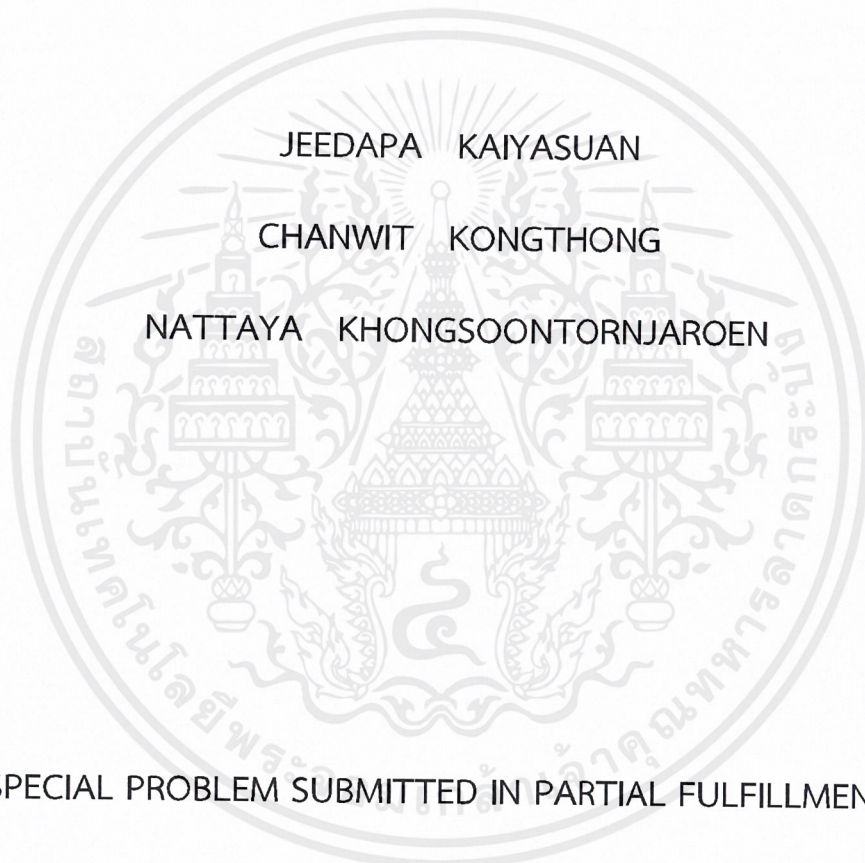


ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANALYSIS OF TWITTER DATA FOR EARLY FLOOD
WARNING



A SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2017



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ การวิเคราะห์ข้อมูลทวีตเตอร์เพื่อการเตือนภัยน้ำท่วม
Analysis of Twitter Data for Early Flood Warning

ชื่อนักศึกษา นางสาวจิตาภา ไกยสวน รหัสนักศึกษา 57050191
นายชาญวิทย์ คงทอง รหัสนักศึกษา 57050208
นายณัฐยะ คงสุนทรเจริญ รหัสนักศึกษา 57050225

ปริญญา วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2560
อาจารย์ที่ปรึกษา ดร.กุลสวัสดิ์ จิตขจรวานิช

คณะวิทยาศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้
ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการ
คอมพิวเตอร์) ประจำปีการศึกษา 2560

คณะกรรมการสอบ	ลายมือชื่อ
ดร.อัคเดช อุดมชัยพร ประธานกรรมการ	
ผศ.ดร.สายชล ใจเย็น กรรมการ	
ดร.กุลสวัสดิ์ จิตขจรวานิช กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การวิเคราะห์ข้อมูลทวิตเตอร์เพื่อการเตือนภัยน้ำท่วม		
ชื่อนักศึกษา	นางสาวจิตาภา ไกยสวน	รหัสนักศึกษา 57050191	
	นายชาญวิทย์ คงทอง	รหัสนักศึกษา 57050208	
	นายณัฐยะ คงสุนทรเจริญ	รหัสนักศึกษา 57050225	
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)		
ภาควิชา	วิทยาการคอมพิวเตอร์		
คณะ	วิทยาศาสตร์		
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)		
ปีการศึกษา	2560		
อาจารย์ที่ปรึกษา	ดร.กุลสวัสดิ์ จิตขจรวานิช		

บทคัดย่อ

ภัยธรรมชาติเป็นเรื่องใกล้ตัวดังที่เราเห็นได้จากข่าวสารต่าง ๆ ไม่ว่าจะเป็นปัญหาภัยแล้ง ไฟป่า แผ่นดินไหว และน้ำท่วม เป็นต้น ซึ่งล้วนแล้วแต่ส่งผลกระทบต่อการดำเนินชีวิตและทรัพย์สิน ทั้งในระยะสั้นและในระยะยาว งานวิจัยนี้มีวัตถุประสงค์เพื่อวิเคราะห์ข้อมูลจาก social media เพื่อลดผลกระทบที่เกิดจากภัยธรรมชาติ และเนื่องด้วยภัยน้ำท่วมเป็นปัญหาที่เกิดขึ้นบ่อยครั้ง และส่งผลกระทบต่อประเทศไทยตลอดทั้งปี เราจึงเลือกที่จะวิเคราะห์การเกิดภัยน้ำท่วมโดยอาศัยข้อมูลจากเครือข่ายสังคมออนไลน์ “ทวิตเตอร์” มาเป็นข้อมูลหลัก ร่วมกับการใช้เทคนิคทางคอมพิวเตอร์ อันได้แก่ การตัดคำ (เพื่อเปรียบเทียบข้อความที่ระบุถึงเหตุการณ์น้ำท่วมในฐานข้อมูลและทวิตเตอร์) การทำ Geo (en-/de-) coding (เพื่อหาสถานที่ที่เกิดเหตุการณ์น้ำท่วม) การทำ String Matching เพื่อยืนยันสถานที่ที่เกิดเหตุการณ์น้ำท่วมในกรณีที่ใช้พิมพ์สถานที่ที่เกิดเหตุการณ์น้ำท่วมผิดพลาด และในขั้นตอนสุดท้ายเราใช้ Geocoding API เพื่อสร้างแผนที่และแสดงผล (visualize) บริเวณที่เกิดเหตุการณ์น้ำท่วมในประเทศไทย ระบบที่พัฒนาขึ้นทำให้ผู้ใช้งานทราบได้ในทันที (semi-real time) ว่าบริเวณใดในประเทศไทยที่เกิดเหตุการณ์น้ำท่วมและมีระดับความรุนแรงมากน้อยเพียงใด

คำสำคัญ : Geocoding API, Geo (en-/de-) coding, String Matching

Title	Analysis of Twitter Data for Early Flood Warning	
Students	Jeedapa Kaiyasuan	Student ID 57050191
	Chanwit Kongthong	Student ID 57050208
	Nattaya Khongsoontornjaroen	Student ID 57050225
Degree	Bachelor of Science (Computer Science)	
Department	Computer Science	
Faculty	Science	
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)	
Academic Year	2017	
Advisor	Dr.Kulsawasd Jitkajornwanich	

Abstract

Natural disasters cause significant damage to the country as well as its citizens as we have seen in the news. Drought, wild fire, earthquake and flooding are some examples of the primary natural disasters occurred in Thailand. In this research, we focus on "flooding" and use data retrieved from a popular social media, "Twitter". The goal of this work is to analyze near real-time data (Tweets) for early flood warning. Traditional methods in processing, analyzing and reporting a flooding event take quite some time. In social medias, on the other hand, by harnessing crowdsourcing, a potential flooding can be predicted a lot faster--of course, with the price of reliability of retrieved tweets. In our research, several techniques are incorporated in order to maximize the accuracy, including, tokenization, geo-encoding and decoding, NLP via string matching (Levenshtein's algorithms), and Google APIs for visualization. Finally, the dynamic yet user-friendly map is produced with respect to the posted relevant tweets and their respective frequencies.

Keywords: Geocoding API, Geo-encoding and decoding, String Matching

กิตติกรรมประกาศ

การจัดทำปัญหาพิเศษเรื่องนี้ (การวิเคราะห์ข้อมูลทวิตเตอร์เพื่อการเตือนภัยน้ำท่วม) สามารถสำเร็จลุล่วงไปได้ ด้วยความกรุณาและความช่วยเหลือในด้านต่าง ๆ รวมทั้งคำปรึกษาที่มีประโยชน์จากบุคคลหลายท่าน คณะผู้จัดทำขอกราบขอบพระคุณสำหรับคำแนะนำอันมีค่าและการช่วยเหลือต่าง ๆ ที่ทำให้หัวข้อปัญหาพิเศษเล่มนี้สำเร็จสมบูรณ์ อันได้แก่

ดร.กุลสวัสดิ์ จิตขจรวานิช อาจารย์ที่ปรึกษาปัญหาพิเศษ ขอขอบคุณสำหรับการให้คำแนะนำและการช่วยเหลือในการทำหัวข้อพิเศษให้ดำเนินลุล่วงไปด้วยดี

ดร.อัคเดช อุดมชัยพร และ ผศ.ดร.สายชล ใจเย็น ที่กรุณาเป็นประธานกรรมการ และกรรมการสอบปัญหาพิเศษ รวมทั้งยังให้คำชี้แนะในการทำหัวข้อปัญหาพิเศษนี้

ดร.สยาม ลววิโรจน์วงศ์ นักวิจัยชำนาญการ จากสำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ (องค์การมหาชน) และ ผศ.ดร.พีรพล เวทีกุล จากจุฬาลงกรณ์มหาวิทยาลัย ที่ชี้แนะแนวทางในการทำหัวข้อปัญหาพิเศษให้ออกมาสมบูรณ์ตามที่คาดหวังไว้

นอกจากนี้ยังมีบุคคลท่านอื่นๆ ที่ไม่ได้กล่าวไว้ ณ ที่นี้จึงใคร่ขอขอบพระคุณทุกท่านที่ได้มีส่วนร่วมในการให้ความช่วยเหลือ ให้คำปรึกษา ให้คำแนะนำ ให้การสนับสนุน ตลอดจนกำลังใจในการจัดทำปัญหาพิเศษฉบับนี้ให้ผ่านลุล่วงไปได้ด้วยดี

จิตภา โภยสวน

ชาญวิทย์ คงทอง

ณัฐยะ คงสุนทรเจริญ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของปัญหาพิเศษ.....	1
1.3 ขอบเขตของปัญหาพิเศษ.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ข้อเสนอแนะ.....	2
1.6 เครื่องมือที่ใช้ในการพัฒนา.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 แอปพลิเคชันทวิตเตอร์ (Twitter).....	4
2.1.1 ข้อมูลที่ได้จากแอปพลิเคชันทวิตเตอร์.....	4
2.1.1.1 Tweet Objects.....	5
2.1.1.2 User Objects.....	7
2.1.1.3 Entities Objects.....	9
2.1.1.4 Extended Entities Objects.....	10
2.1.1.5 Geo Objects.....	10
2.1.1.5.1 Place Objects.....	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.1.1.5.2 Coordinates Objects	11
2.2 Word Tokenization.....	13
2.2.1 Convolution Neural Network (CNN).....	13
2.3 String Matching	14
บทที่ 3 วิธีการดำเนินงานและการออกแบบระบบ	16
3.1 โครงสร้างการออกแบบระบบ	16
3.2 การเลือกใช้เครือข่ายสังคมออนไลน์.....	18
3.3 Data Collection.....	18
3.4 Twitter Analysis	25
3.5 การ Visualize บนแผนที่ประเทศไทย	31
3.6 แผนภาพยูสเคส (Use Case Diagram).....	40
3.7 ตารางอธิบายแผนภาพยูสเคสของผู้ใช้ระบบ (Use Case Narrative)	41
3.8 แผนภาพแสดงปฏิสัมพันธ์ (Sequence Diagram).....	46
3.9 แผนภาพกิจกรรม (Activity Diagram).....	47
บทที่ 4 ผลการดำเนินงานและการอภิปรายผล	48
4.1 การดึงข้อความทวีตเตอร์ในช่วงเวลาที่เกิดและไม่เกิดเหตุการณ์และสถิติที่เกี่ยวข้อง	48
4.2 เวลาในการประมวลผลข้อมูลทวีตเตอร์.....	50
4.2.1 การประมวลผลเพื่อหาพิกัดสถานที่น้ำท่วมด้วยวิธีการ Levenshtein	50
4.2.2 เวลาในการประมวลผลโปรแกรมตั้งแต่เริ่มต้นจนแสดงสถานที่ที่เกิดเหตุการณ์น้ำท่วม .	50
4.3 หน้าจอการทำงานของเว็บไซต์.....	51
บทที่ 5 สรุปผลปัญหาพิเศษและข้อเสนอแนะ.....	54
5.1 สรุปผลปัญหาพิเศษ	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
5.2 ข้อจำกัดของปัญหาพิเศษ.....	54
5.3 ข้อเสนอแนะในการพัฒนาปัญหาพิเศษ.....	54
เอกสารอ้างอิง	55
ภาคผนวก.....	57
ภาคผนวก ก. วิธีการติดตั้ง XAMPP จำลอง WebServer	58
ภาคผนวก ข. ขั้นตอนการติดตั้งโปรแกรม ไพชาร์ม (Pycharm).....	63



สารบัญตาราง

ตารางที่	หน้า
2.1 Tweet Object Attributes	5
2.2 User Object Attributes	7
2.3 Entities Object Attributes.....	9
2.4 Place Object Attributes.....	10
2.5 Coordinates Object Attributes.....	11
3.1 ตารางอธิบายแผนภาพยูสเคสของยูสเซอร์ที่เข้าใช้ระบบ	41
3.2 ตารางอธิบายแผนภาพยูสเคส Visualize on a map	42
3.3 ตารางอธิบายแผนภาพยูสเคสการทำ String Matching	43
3.4 ตารางอธิบายแผนภาพยูสเคสการทำ Geocoder	44
3.5 ตารางอธิบายแผนภาพยูสเคสการทำ Word Tokenization.....	44
3.6 ตารางอธิบายแผนภาพยูสเคสการทำ Get Tweets	45
4.1 ตารางแสดงเวลาการประมวลผลพื้นที่ที่เกิดเหตุการณ์น้ำท่วมด้วยวิธีของ Levenshtein	50
4.2 ตารางแสดงเวลาการประมวลผลโปรแกรมตั้งแต่เริ่มต้นจนแสดงสถานที่ที่เกิดเหตุที่น้ำท่วม	50

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างทวิต 1 ทวิตจากแอปพลิเคชันทวิตเตอร์	4
2.2 ตัวอย่างข้อมูลที่ได้จากทวิตเตอร์โดยอ้างอิงจากรูปที่ 2.1	12
2.3 ข้อความที่เราจะนำมาทำการตัดคำ	13
2.4 ข้อความที่ AI ทำการตัดคำ	13
2.5 ข้อความที่ AI ทำการตัดคำโดยใช้เทคนิคเลื่อนซ้าย	14
2.6 ข้อความที่ AI ทำการตัดคำจนสมบูรณ์	14
3.1 กระบวนการทำงานการกรองข่าวสารโดยการนำข้อมูลจากทวิตเตอร์มาเพื่อกรองข่าวสาร น้ำท่วม	16
3.2 โค้ดที่ใช้ในการดึงทวิตจากทวิตเตอร์	19
3.3 โค้ดที่ใช้ในการดึงทวิตจากทวิตเตอร์	20
3.4 ตัวอย่างข้อมูลจากทวิตเตอร์ที่เก็บใน MongoDB	20
3.5 แผนภาพอีอาร์ของข้อมูลพื้นที่ในประเทศไทย ประกอบไปด้วย แขวง เขต และจังหวัด	21
3.6 ตัวอย่างข้อมูลดาต้าเบสของจังหวัดในประเทศไทย	21
3.7 ตัวอย่างข้อมูลดาต้าเบสของเขตในประเทศไทย	22
3.8 ตัวอย่างข้อมูลดาต้าเบสของแขวงในประเทศไทย	22
3.9 ตัวอย่างข้อมูลดาต้าเบสที่ได้จาก Geocoder	23
3.10 ตัวอย่างข้อมูลดาต้าเบส Dashboard ที่จะส่งไป Visualize บนแผนที่ประเทศไทย	23
3.11 ตัวอย่างข้อมูลดาต้าเบสที่จะส่งไป Visualize บนแผนที่ประเทศไทย	24
3.12 ตัวอย่างข้อมูลดาต้าเบสที่ได้จากการทำ Levenshtein	25
3.13 โค้ดที่ใช้ในการทำ Optimize Word	26
3.14 โค้ดที่ใช้ในการทำ Optimize Word	27
3.15 โค้ดที่ใช้ในการทำ Optimize Word	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.16 โค้ดที่ใช้ในการทำ Optimize Word.....	29
3.17 โค้ดที่ใช้ในการ Search คำในฐานข้อมูล.....	30
3.18 โค้ดที่ใช้ในการ Search คำในฐานข้อมูล.....	31
3.19 Marker แจ้งเตือนน้ำท่วมที่มีการกระพริบเพื่อบอกว่ากำลังเกิดเหตุการณ์น้ำท่วมอยู่.....	32
3.20 โค้ดที่ใช้ในการทำ Marker.....	32
3.21 โค้ดที่ใช้ในการทำ Marker.....	33
3.22 โค้ดที่ใช้ในการแสดงผลเว็บไซต์.....	34
3.23 โค้ดที่ใช้ในการแสดงผลเว็บไซต์.....	35
3.24 โค้ดที่ใช้ในการแสดงผลเว็บไซต์.....	35
3.25 โค้ดที่ใช้ในการแสดงผลเว็บไซต์.....	36
3.26 โค้ดที่ใช้ในการแสดงผลเว็บไซต์.....	36
3.27 โค้ดที่ใช้ในการแสดงผลเว็บไซต์.....	37
3.28 โค้ดที่ใช้ในการแสดงผลเว็บไซต์.....	38
3.29 โค้ดที่ใช้ในการแสดงผลเว็บไซต์.....	39
3.30 แผนภาพยูสเคส.....	40
3.31 แผนภาพแสดงปฏิสัมพันธ์ของระบบ.....	46
3.32 แผนภาพกิจกรรมของระบบ.....	47
4.1 กราฟเปรียบเทียบพฤติกรรมการทวีตข้อความของผู้ใช้ทวีตเตอร์ในช่วงที่เกิดเหตุการณ์ และช่วงที่ไม่ได้เกิดเหตุการณ์โดยใช้แฮชแท็กบัพเพสันนิวาส (#บัพเพสันนิวาส).....	48
4.2 กราฟเปรียบเทียบพฤติกรรมการทวีตข้อความของผู้ใช้ทวีตเตอร์ในช่วงที่เกิดเหตุการณ์ และช่วงที่ไม่ได้เกิดเหตุการณ์โดยใช้แฮชแท็กGabriel (#Gabriel).....	49
4.3 รูปภาพหน้าจอของเว็บไซต์แจ้งเตือนเหตุการณ์น้ำท่วม.....	51
4.4 หน้าจอแสดงฟังก์ชันระบุสถานที่ของผู้ใช้งาน.....	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 หน้าจอแสดงการกดใช้ฟังก์ชันระบุสถานที่ที่ผู้ใช้งานใช้งานอยู่.....	52
4.6 หน้าจอแสดงฟังก์ชันค้นหาสถานที่.....	52
4.7 หน้าจอแสดงการกดใช้ฟังก์ชันค้นหาสถานที่.....	53
4.8 หน้าจอแสดงแดชบอร์ด.....	53
ก.1 หน้าต่างการ Setup XAMPP.....	58
ก.2 หน้าต่างการเลือก Componenets ที่ต้องการ.....	59
ก.3 หน้าต่างการเลือก Path.....	59
ก.4 หน้าต่างการ Setup XAMPP.....	60
ก.5 หน้าต่างการ Setup XAMPP.....	60
ก.6 หน้าต่างการ Installing XAMPP.....	61
ก.7 หน้าต่างการ Installing XAMPP เสร็จสมบูรณ์.....	61
ก.8 หน้าต่างการจำลอง WebServer.....	62
ก.9 หน้าต่างการแสดง Path ที่ใช้เก็บโค้ด.....	62
ข.1 หน้าต่างการ Setup Pycharm.....	63
ข.2 หน้าต่างขณะกำลังติดตั้งโปรแกรม Pycharm.....	64
ข.3 หน้าต่างสำหรับการใช้งาน Pycharm ครั้งแรก.....	64
ข.4 หน้าต่างสำหรับการตั้งค่าเริ่มต้น.....	65
ข.5 หน้าต่างสำหรับการตั้งชื่อไฟล์.....	65
ข.6 หน้าต่างแสดงตัวอย่างหน้าจอ Pycharm.....	66

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ภัยธรรมชาติเป็นเรื่องใกล้ตัวดังที่เราเห็นได้จากข่าวสารต่าง ๆ ไม่ว่าจะเป็นปัญหาภัยแล้ง ไฟป่า แผ่นดินไหว และน้ำท่วม เป็นต้น ซึ่งล้วนแล้วแต่ส่งผลเสียต่อการดำเนินชีวิตและทรัพย์สิน ทั้งในระยะสั้นและในระยะยาว งานวิจัยนี้มีวัตถุประสงค์เพื่อวิเคราะห์ข้อมูลจาก social media เพื่อลดปัญหาของภัยธรรมชาติ เนื่องจากภัยน้ำท่วมเป็นปัญหาที่เกิดขึ้นบ่อยครั้ง และส่งผลกระทบต่อโดยตรงกับประเทศไทยตลอดทั้งปี เราจึงเลือกที่จะวิเคราะห์การเกิดภัยน้ำท่วมโดยอาศัยข้อมูลจากเครือข่ายสังคมออนไลน์ “ทวิตเตอร์” มาเป็นข้อมูลหลัก ร่วมกับการใช้เทคนิคทางคอมพิวเตอร์ต่าง ๆ ได้แก่ การตัดคำ (เพื่อเปรียบเทียบข้อความที่ระบุถึงเหตุการณ์น้ำท่วมในดาต้าเบสและทวิตเตอร์) การทำ Geo (en-/de-) coding (เพื่อหาสถานที่ที่เกิดเหตุการณ์น้ำท่วม) และการทำ String Matching เพื่อยืนยันสถานที่ที่เกิดเหตุการณ์น้ำท่วมในกรณีที่ผู้ใช้พิมพ์สถานที่ที่เกิดเหตุการณ์น้ำท่วมผิดพลาด

1.2 วัตถุประสงค์ของปัญหาพิเศษ

- 1) เพื่อประมวลผลข้อมูลจากทวิตเตอร์ในการประเมินสถานการณ์น้ำท่วมแบบ near real-time
- 2) เพื่อแสดงผลตำแหน่งและบริเวณที่เกิดน้ำท่วมบนแผนที่

1.3 ขอบเขตของปัญหาพิเศษ

- 1) ประมวลผลข้อมูลจากเครือข่ายสังคมออนไลน์ “ทวิตเตอร์” เท่านั้น
- 2) ข้อมูลทวิตเตอร์ที่นำมาใช้ต้องเป็นภาษาไทย
- 3) รองรับการแสดงผลผ่านทาง web browser บน personal computer เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เป็นอีกทางเลือกหนึ่งในการรับข่าวสารเกี่ยวกับเหตุการณ์น้ำท่วม
- 2) สามารถใช้เป็นเครื่องมือช่วยในการตัดสินใจวางแผนต่าง ๆ (เช่น การเดินทาง)
- 3) สามารถนำไปต่อยอดประยุกต์ใช้กับเหตุการณ์ฉุกเฉินอื่น ๆ

1.5 ข้อสันนิษฐาน

ต้องการทดสอบว่าระบบสามารถประเมินสถานการณ์น้ำท่วมได้อย่าง semi-real time และสอดคล้องกับความเป็นจริง

1.6 เครื่องมือที่ใช้ในการพัฒนา

- 1) ฮาร์ดแวร์ (Hardware)
 - เครื่องคอมพิวเตอร์ Spec : RAM 4 GB Intel core i5 Geforce GTX550
 - เครื่องคอมพิวเตอร์ Spec : RAM 16 GB Intel core i7 Geforce GT 940 M
 - เครื่องคอมพิวเตอร์ Spec : RAM 4 GB Intel core i5 Geforce GT720
- 2) ซอฟต์แวร์ (Software)
 - MongoDB เวอร์ชัน Windows Server 2008 R2-64bit and later, with SSL support x64
 - Microsoft SQL Server
 - Python เวอร์ชัน 3.6.3
 - Pycharm
 - Studio 3T
 - Atom
 - Sublime
 - xampp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- HTML
- PHP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

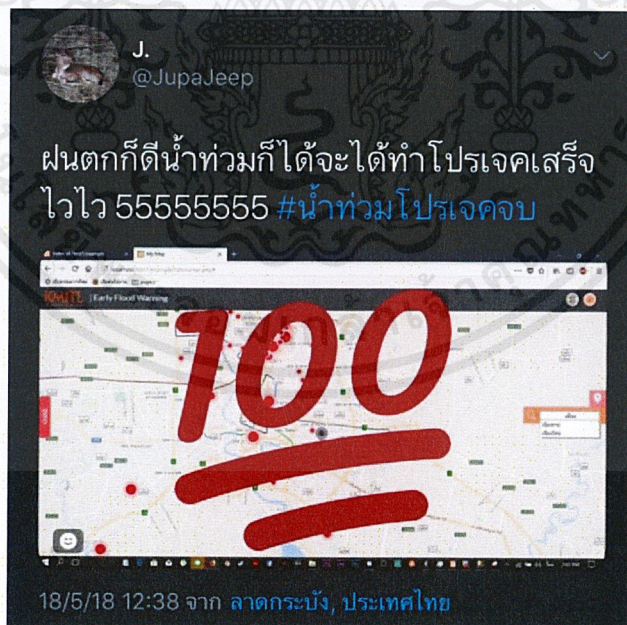
ทฤษฎีที่เกี่ยวข้อง

ในปัจจุบันข่าวสารมีการแพร่หลายอย่างรวดเร็วผ่านช่องทางเทคโนโลยีต่าง ๆ หนึ่งในแอปพลิเคชันที่มีการแพร่หลายของข่าวสารอย่างรวดเร็วก็คือ แอปพลิเคชันทวิตเตอร์ ในที่นี้จะกล่าวถึงการทำงานของแอปพลิเคชันทวิตเตอร์ที่เรานำมาใช้ในการกรองข่าวสาร และเครื่องมือต่างๆ ที่เข้ามามีส่วนในการกรองข่าวสารหลังจากที่เราได้ข้อมูลมาจากแอปพลิเคชันทวิตเตอร์ เพื่อเป็นทฤษฎีและเครื่องมือที่ใช้อ้างอิงในการประมวลผลข้อมูลทวิตเตอร์เพื่อการเตือนภัยน้ำท่วม

2.1 แอปพลิเคชันทวิตเตอร์ (Twitter)

ทวิตเตอร์ เป็นเครือข่ายสังคมออนไลน์ประเภทหนึ่ง ที่เน้นความเรียบง่ายและความรวดเร็ว ซึ่งทำให้เป็นจุดเด่นในการเผยแพร่ข้อความสั้นๆ และทำให้ได้รับความนิยมอย่างมากในปี 2016 จากการจัดอันดับจากงาน Thailand Zocial Award 2017 มีผู้ใช้ทวิตเตอร์ถึง 9 ล้านคน และมีอัตราการเติบโตขึ้น 70 เปอร์เซ็นต์เมื่อเทียบกับจากปี 2016 [<https://thailandzocialawards.com/download-slides/Social-Media-Movement-by-Pnern-Asavavipas.pdf>]

2.1.1 ข้อมูลที่ได้จากแอปพลิเคชันทวิตเตอร์



รูปที่ 2.1 ตัวอย่างทวิต 1 ทวิตจากแอปพลิเคชันทวิตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่แอปพลิเคชันทวีตเตอร์บริการให้จะอยู่ในรูปแบบของ JavaScript Object Notation (JSON) ในทุก ๆ ออบเจ็กต์ ออบเจ็กต์เหล่านี้จะใช้อธิบายแอตทริบิวต์ต่าง ๆ ทวีตแต่ละรายการจะมีการสร้างอ็อบเจ็กต์เอนทิตี ซึ่งเป็นอาร์เรย์ของเนื้อหาที่มีการแชร์กันเช่น hashtags กล่าวถึงสื่อและลิงค์ต่างๆ หากมีการเชื่อมโยงข้อมูลการให้บริการ JSON จะสามารถให้ข้อมูล metadata ได้เช่นกัน เช่น URL ที่สมบูรณ์ , ชื่อและคำอธิบายของเว็บเพจ ดังนั้นนอกเหนือจากเนื้อหาข้อความตัวเองแล้ว ทวีตสามารถมีแอตทริบิวต์ได้มากกว่า 140 รายการที่เกี่ยวข้อง (อ้างอิงจากทวีตดังรูป 2.1) ได้แก่

2.1.1.1 Tweet Object

ทวีตเป็นพื้นฐานของทวีตเตอร์ เรียกอีกอย่างว่า การอัปเดตสถานะ ออบเจ็กต์ทวีตจะประกอบไปด้วยแอตทริบิวต์พื้นฐานต่าง ๆ เช่น id, created_at, and text นอกจากนี้ออบเจ็กต์ทวีตยังเป็น parent ออบเจ็กต์ให้กับออบเจ็กต์อื่น ๆ เช่น user object , extended_entities เป็นต้น ใน 1 ทวีตที่ผู้ใช้ทวีตจะสามารถให้ข้อมูลดังตารางที่ 2.1

ตารางที่ 2.1 Tweet Object Attributes

แอตทริบิวต์	ประเภท	คำอธิบาย / ตัวอย่าง
created_at	String	จะเป็นเวลา UTC 'created_at': 'Fri May 18 05:47:19 +0000 2018'
id	Int64	เป็นการระบุทวีตในรูปแบบของ integer 64 บิต 'id': 997352915881771008
id_str	String	เป็นการระบุทวีตในรูปแบบการแทนด้วย String 'id_str': '997352915881771008'
Text	String	'text': 'RT @JupaJeep: ฟนตกก็ดีน้ำท่วมก็ได้จะได้ ทำโปรเจคเสร็จไวไว 5555555 #น้ำท่วมโปรเจคจบ https://t.co/Kfr7B07SFX'
Source	String	'source': 'Twitter for iPhone'
Truncated	Boolean	'truncated': False
in_reply_to_status_id	Int64	'in_reply_to_status_id': None
in_reply_to_status_id_str	String	'in_reply_to_status_id_str': None
in_reply_to_user_id	Int64	'in_reply_to_user_id': None
in_reply_to_user_id_str	String	'in_reply_to_user_id_str': None

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

in_reply_to_screen_name	String	'in_reply_to_screen_name': None
user	User object	ผู้ใช้ที่โพสต์ทวีต รายละเอียด (ตารางที่ 2.2)
coordinates	Coordinates	ตำแหน่งทางภูมิศาสตร์ของทวีตตามรายงานของแอปพลิเคชันผู้ใช้หรือไคลเอนต์ อาร์เรย์ของพิกัดด้านในมีรูปแบบเป็น GeoJSON (ลองจิจูด-ละติจูด) รายละเอียด (ตารางที่ 2.5)
place	Places	รายละเอียด (ตารางที่ 2.4)
quoted_status_id	Int64	"quoted_status_id":114749583439036416
quoted_status_id_str	String	"quoted_status_id_str": "114749583439036416"
is_quote_status	Boolean	"is_quote_status":false
quoted_status	Tweet	
retweeted_status	Tweet	จะได้ข้อมูลทั้งหมดของตาราง 2.1
quote_count	Integer	'quote_count': 0
reply_count	Int	'reply_count': 0
retweet_count	Int	'retweet_count': 1
favorite_count	Integer	'favorite_count': 0
entities	Entities	Entities ที่ถูกแยกออกจากข้อความทวีต รายละเอียด (ตารางที่ 2.3)
extended_entities	Extended Entities	ข้อมูลรูปภาพต่าง ๆ จะถูกสร้างเป็นอาร์เรย์ "สื่อ" แบบ metadata รายละเอียด 2.1.1.4
favorited	Boolean	'favorited': False
retweeted	Boolean	'retweeted': False
possibly_sensitive	Boolean	'possibly_sensitive': False
filter_level	String	'filter_level': 'low'
lang	String	'lang': 'th'
matching_rules	Array of Rule Objects	"matching_rules": "[{"tag": "rain Tweets", "id": 831566737246023680, "id_str": "831566737246023680"}]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		}, { "tag": "snow Tweet", "id": 831567402366218240, "id_str": "831567402366218240" }]"
--	--	--

2.1.1.2 User Object

ออกแบบเจ็ทต์ผู้ใช้งานสำหรับแอคเคาต์ที่เปิดใช้งานแบบสาธารณะ จะมีรายละเอียดของข้อมูลผู้ใช้งานและทวีตที่ผู้ใช้ทวีต แสดงดังตารางที่ 2.2

ตารางที่ 2.2 User Object Attributes

แอตทริบิวต์	ประเภท	คำอธิบาย / ตัวอย่าง
id	Int64	'id': 225362477
id_str	String	'id_str': '225362477'
name	String	'name': 'J.'
screen_name	String	'screen_name': 'JupaJeep'
location	String	'location': '@Bangkok THAILAND'
url	String	'url': None
description	String	'description': 'comsci kmitl สุราเมระยะมีชชป มาที่ภูฐานา ขอบคุณสิ่งศักดิ์สิทธิ์ where were you when i needed you most? '
derived	Arrays of Enrichment Objects	"derived": { "locations": [], "klout": [] }
protected	Boolean	'protected': False
verified	Boolean	'verified': False
followers_count	Int	'followers_count': 167
friends_count	Int	'friends_count': 294
listed_count	Int	'listed_count': 0
favourites_count	Int	'favourites_count': 1199

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

statuses_count	Int	'statuses_count': 39162
created_at	String	'created_at': 'Sat Dec 11 11:27:16 +0000 2010'
utc_offset	Int	'utc_offset': -18000
time_zone	String	'time_zone': 'Quito'
geo_enabled	Boolean	'geo_enabled': True
lang	String	'lang': 'en'
contributors_enabled	Boolean	'contributors_enabled': False
profile_background_color	String	'profile_background_color': 'D6A1F0'
profile_background_image_url	String	'profile_background_image_url': 'http://pbs.twimg.com/profile_background_images/511496547/548914_3357145541931_1667762325_2647299_452024326_n.jpg'
profile_background_image_url_https	String	'profile_background_image_url_https': 'https://pbs.twimg.com/profile_background_images/511496547/548914_3357145541931_1667762325_2647299_452024326_n.jpg'
profile_background_tile	Boolean	'profile_background_tile': True
profile_banner_url	String	'https://pbs.twimg.com/profile_banners/225362477/1515329960'
profile_image_url	String	'profile_image_url': 'http://pbs.twimg.com/profile_images/993556002661220352/YAe693-f_normal.jpg'
profile_image_url_https	String	'profile_image_url_https': 'https://pbs.twimg.com/profile_images/993556002661220352/YAe693-f_normal.jpg'
profile_link_color	String	'profile_link_color': 'E81C4F'
profile_sidebar_border_color	String	'profile_sidebar_border_color': '000000'
profile_sidebar_fill_color	String	'profile_sidebar_fill_color': 'E6D6CD'
profile_text_color	String	'profile_text_color': '5B0BC2'
profile_use_background_image	Boolean	'profile_use_background_image': False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

default_profile	Boolean	'default_profile': False
default_profile_image	Boolean	'default_profile_image': False
withheld_in_countries	String	When present, indicates a textual representation of the two-letter country codes this user is withheld from. "withheld_in_countries": "GR, HK, MY"
withheld_scope	String	When present, indicates whether the content being withheld is the "status" or a "user." "withheld_scope": "user"

หมายเหตุ : ตารางที่ไฮไลท์สีเทา หมายถึงข้อมูลที่ทวีตเตอร์เคย provide ให้เรา แต่ในปัจจุบันแอตทริบิวต์ที่อยู่ในตารางที่ไฮไลท์สีเทาได้ถูกยกเลิกไปแล้ว

2.1.1.3 Entities Objects

เอนทิตีที่ออบเจ็กต์ให้ข้อมูล metadata และข้อมูลเพิ่มเติมเกี่ยวกับข้อความที่โพสต์บนทวีตเตอร์ เช่น แฮชแท็ก , รูปภาพ , โพลสำรวจ เป็นต้น ตัวอย่างเอนทิตีที่ออบเจ็กต์ที่ทวีตเตอร์ให้บริการข้อมูลแสดงได้ดังตารางที่ 2.3

ตารางที่ 2.3 Entities Object Attributes

แอตทริบิวต์	ประเภท	คำอธิบาย / ตัวอย่าง
media	Array of Media Objects	สื่อที่อัปโหลด (ทวีต) ในทวีตเตอร์
hashtags	Array of Hashtag Objects	แฮชแท็กที่แยกอยู่ในข้อความ แทนด้วย # ตามด้วยข้อความ
urls	Array of URL Objects	URL ที่รวมอยู่ในข้อความที่ทวีต
user_mentions	Array of User Mention Objects	ผู้ใช้ทวีตเตอร์คนอื่น ๆ ที่กล่าวถึงในข้อความของทวีต
symbols	Array of Symbol Objects	สัญลักษณ์ต่างที่ใช้เช่น \$ ที่รวมอยู่ในข้อความของทวีต 'symbols': []
polls	Array of Poll Objects	ข้อความทวีตที่เป็นโพลสำรวจต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.4 Extended Entities Objects

สื่อต่าง ๆ ในทวีตเตอร์ ได้แก่ รูปภาพ วิดีโอ และไฟล์.GIF ทวีตเตอร์จะมีการสร้างออบเจกต์ขึ้นมาเพื่อเก็บข้อมูลแยกเป็น Extended entities object รายละเอียดข้อมูลที่เก็บแยกเป็น Extended entities object ตัวอย่างเช่น

```
'extended_entities': {'media': [ ] }
```

ตัวอย่างทวีตในรูปแบบของ JSON ของ Extended entities object

2.1.1.5 Geo Objects

ทวีตสามารถเชื่อมโยงกับตำแหน่งตำแหน่งทางภูมิศาสตร์ที่สร้างทวีต ตำแหน่งของทวีตสามารถกำหนดโดยผู้ใช้ หรือเมื่อโพสต์ทวีตโดยใช้ API ตำแหน่งที่ตั้งของทวีตจะสามารถเป็นตำแหน่ง 'จุดที่แน่นอน' หรือ Twitter Place พร้อมกับ bounding box ซึ่งอธิบายพื้นที่ขนาดใหญ่ตั้งแต่สถานที่ไปจนถึงภูมิภาคทั้งหมด มีออบเจกต์ 2 ตัวที่ใช้อธิบายตำแหน่งของทวีตได้แก่ Place object และ Coordinates object

2.1.1.5.1 Place Objects

เป็นสถานที่เฉพาะเจาะจง มีชื่อสถานที่ที่สอดคล้องกับพิกัดทางภูมิศาสตร์ แสดงข้อมูลได้ดังตาราง 2.4

ตารางที่ 2.4 Place Object Attributes

แอตทริบิวต์	ประเภท	คำอธิบาย / ตัวอย่าง
id	String	กำหนด id เพื่อระบุสถานที่ ใช้เป็น String ไม่ใช่ Integer
url	String	URL แสดงตำแหน่งข้อมูลเป็น metadata
place_type	String	ประเภทของสถานที่ เช่น 'place_type': 'city'
Name	String	แสดงชื่อสถานที่โดยย่อที่สามารถอ่านได้ เช่น 'name': 'Lat Krabang'
full_name	String	แสดงชื่อสถานที่แบบเต็ม เช่น 'full_name': 'Lat Krabang, Thailand'
country_code	String	แสดงตัวย่อของชื่อประเทศ 'country_code': 'TH'
Country	String	แสดงชื่อประเทศ 'country': 'Thailand'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

bounding_box	<u>Object</u>	bounding box ของ coordinates บริเวณที่ล้อมรอบกับสถานที่ 'bounding_box': {'type': 'Polygon', 'coordinates': [[[100.70832, 13.71536], [100.70832, 13.730883], [100.791425, 13.730883], [100.791425, 13.71536]]]},
--------------	---------------	--

2.1.1.5.2 Coordinates Objects

เป็นออบเจกต์พิกัด จะเป็นอาร์เรย์ [long, lat] กับพิกัดทางภูมิศาสตร์ แสดงข้อมูลได้ดังตาราง 2.5

ตารางที่ 2.5 Coordinates object Attributes

แอตทริบิวต์	ประเภท	คำอธิบาย / ตัวอย่าง
Coordinates	Collection of Float	แสดงลองจิจูดและละติจูดในรูปแบบ [longitude, latitude]. 'coordinates': None
Type	String	ชนิดของข้อมูลที่เข้ารหัสในพิกัดหรือพเพอร์ต์ ในที่นี้คือ "จุด" สำหรับเขตพิกัดที่ใช้ร่วมกัน

จากข้อมูลทั้งหมดที่ทวีตเตอร์ให้ข้อมูลกับเราใน 1 ทวีต ทวีตเตอร์ให้ข้อมูลมากกว่า 140 แอตทริบิวต์ตามที่ได้กล่าวไปเบื้องต้น แต่เนื่องจากไม่ใช่ทุกแอตทริบิวต์ที่จำเป็นในการใช้กรองข้อมูลข่าวสารน้ำท่วม เราจึงเลือกเก็บข้อมูลทวีตเตอร์เพียงบางส่วนที่จำเป็นในการกรองข้อมูลข่าวสารน้ำท่วมเท่านั้น เนื่องจากข้อมูลที่ใช้กรองข่าวสารน้ำท่วมจริง ๆ คือ ข้อความที่ผู้ใช้ทวีตเท่านั้น โดยข้อมูลที่เรานำมาเลือกเก็บในฐานข้อมูลจากทวีตแต่ละทวีต ได้แก่

- 1) idusser
- 2) ussrName
- 3) follower
- 4) text
- 5) hashtags
- 6) date
- 7) coordinates

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Word Tokenization

การตัดคำภาษาไทยต่างจากการตัดคำภาษาอังกฤษ ตรงที่ภาษาไทยไม่มีการเว้นวรรคในแต่ละคำของประโยค และมีคำอ่านที่กำกวม ทำให้การตัดคำภาษาไทยสามารถทำได้ยาก เราจึงเลือกใช้ Deep learning ของบริษัททรู (True) โดยบริษัททรูได้เลือกใช้โมเดล Convolution Neural Network แบบ 1 มิติ (CNN) มาทำการตัดคำด้วยวิธี Binary Classification และใช้ชุดข้อมูลจาก Nectec มาเป็นดาต้าเซตในการเทรนข้อมูล นอกจากนี้บริษัททรูยังมีไลบรารีชื่อ deepcut ที่พัฒนาด้วยภาษาไพธอนให้เราสามารถนำมาใช้ในการตัดคำได้

2.2.1 Convolution Neural Network (CNN)

Convolution Neural Network เป็นการเรียนรู้เชิงลึกซึ่งเป็นวิธีการหนึ่งของการเรียนรู้ของเครื่อง หลักการทำงานของ CNN คือ สมมติว่ามีประโยค “ตัดคำได้ดีมาก” ตัว AI จะทราบว่าจะข้อความประกอบไปด้วยตัวอักษรดังนี้

ต	ั	ด	ค	ำ	ไ	ด	ั	ด	ั	ม	า	ก
---	---	---	---	---	---	---	---	---	---	---	---	---

รูปที่ 2.3 ข้อความที่เราจะนำมาทำการตัดคำ

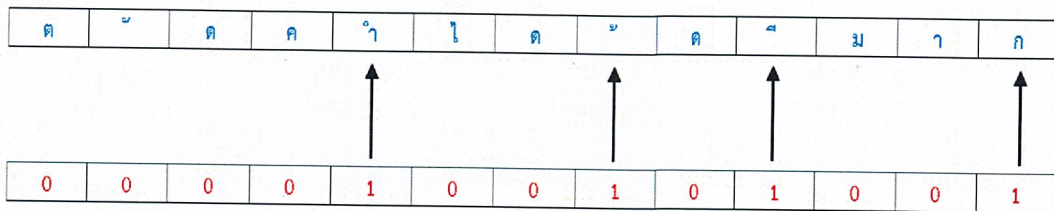
โดย AI จะมอง 1 ประโยค เป็นอาร์เรย์ของชุดตัวอักษร ในประโยค “ตัดคำได้ดีมาก” จะเห็นเป็น 13 ตัวอักษร จากนั้น AI ก็จะบอกว่า ตัวอักษรไหนเป็นตัวเริ่มต้นของคำ ถ้าใช่ก็จะให้คำตอบว่า 1

ต	ั	ด	ค	ำ	ไ	ด	ั	ด	ั	ม	า	ก
↑					↑			↑		↑		
1	0	0	0	0	1	0	0	1	0	1	0	0

รูปที่ 2.4 ข้อความที่ AI ทำการตัดคำ

ตัวอักษรสีแดง เป็นผลลัพธ์จากการทำนายของ AI ที่ประกอบด้วยชุดของอาร์เรย์ 13 ตัว โดยที่เลข 1 บอกว่าเป็นอักษรเริ่มต้น (0 คือไม่ใช่) แต่เพื่อความสะดวกตอนใช้คำสั่ง `deepcut.tokenize('ตัดคำได้ดีมาก')` ฟังก์ชันก็จะตัดแบ่งคำออกมาให้เลยอัตโนมัติ โดยเบื้องหลังการทำงานจะใช้เทคนิคเลื่อนเลข 1 ในอาร์เรย์ไปทางซ้าย (ส่วนอักษรตัวสุดท้ายของประโยคก็จะระบุว่าเป็น 1 ไปเลย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ข้อความที่ AI ทำการตัดคำโดยใช้เทคนิคเลื่อนซ้าย

จากนั้นภายในไลบรารีจะเริ่มไล่ตัวอักษร ตั้งแต่หมายเลข 0 กำกับ จนมาหยุดตรงตัวอักษรเลข 1 ก็จะได้ว่าเป็น 1 คำไทย ดังนั้นเมื่อใช้คำสั่ง `deepcut.tokenize('ตัดคำได้ดีมาก')` ก็จะทำให้ค่าออกมาเป็นอาร์เรย์ดังต่อไปนี้

```
[ 'ตัด', 'คำ', 'ได้', 'ดี', 'มาก' ]
```

รูปที่ 2.6 ข้อความที่ AI ทำการตัดคำจนสมบูรณ์

2.3 String Matching

การทำ String Matching สามารถทำได้หลากหลายวิธี เราเลือกวิธีของ Levenshtein edit distance ในงานวิจัยชิ้นนี้ Levenshtein edit distance เป็นขั้นตอนวิธีการวัดหาค่าความต่างกันของข้อความ 2 ชุด ระหว่างชุดแรกที่เป็นต้นแบบ และ ชุดที่สองที่เป็นชุดเปรียบเทียบ โดยค่าความต่างกันจะวัดจากจำนวนของการเปลี่ยนแปลงข้อความจากชุดเปรียบเทียบ (ที่ตัดตัวอักษรออก แทรกตัวอักษร และแทนที่ตัวอักษร) จนกระทั่งได้ข้อความที่เป็นชุดต้นแบบ

ขั้นตอนการทำงานของ Levenshtein edit distance จะเป็นการนำข้อความ 2 ชุด มาเปรียบเทียบจำนวนความแตกต่างกัน โดยพิจารณาดังนี้

- 1) การแทรก เป็นการนำเอาอักขระตัวใดๆมา เพื่อให้ชุดอักขระชุดนั้นเหมือนกับอีกชุดอักขระหนึ่งในภายหลัง เช่น กรุงเทพฯ → กรุงเทพ จะแทรกตัว ง ให้กับ กรุงเทพฯ เพื่อให้ กรุงเทพฯ กลายเป็น กรุงเทพ เป็นต้น
- 2) การตัดออก เป็นการตัดอักขระออกครั้งละ 1 ตัว จากชุดอักขระตัวหนึ่ง เพื่อให้ชุดอักขระชุดนั้นเหมือนกับอีกชุดอักขระหนึ่งในภายหลัง เช่น เชียงราย → เชียงราย จะตัดตัว ย ออก เพื่อให้ เชียงราย กลายเป็น เชียงราย เป็นต้น
- 3) การแทนที่ เป็นการนำอักขระของชุดอักขระหนึ่งไปแทนอักขระของอีกชุดอักขระหนึ่ง เพื่อให้ชุดอักขระชุดนั้นเหมือนกับอีกชุดอักขระหนึ่งในภายหลัง เช่น ระนอง → ระยอง จะแทนที่ น ด้วย ย เพื่อให้ ระนอง กลายเป็น ระยอง หรือ อาจมองในทางกลับกันก็ได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนการทำงานของ Levenshtein edit distance จะนำคำที่ตัดได้มาเทียบกับชุดของ
มูลในคำต้นเบส ของ แขวง เขต และจังหวัด คำใดที่มีค่า Levenshtein edit distance น้อยที่สุดจะ
ถือว่าคำนั้นเป็นพื้นที่ที่เกิดเหตุการณ์น้ำท่วม

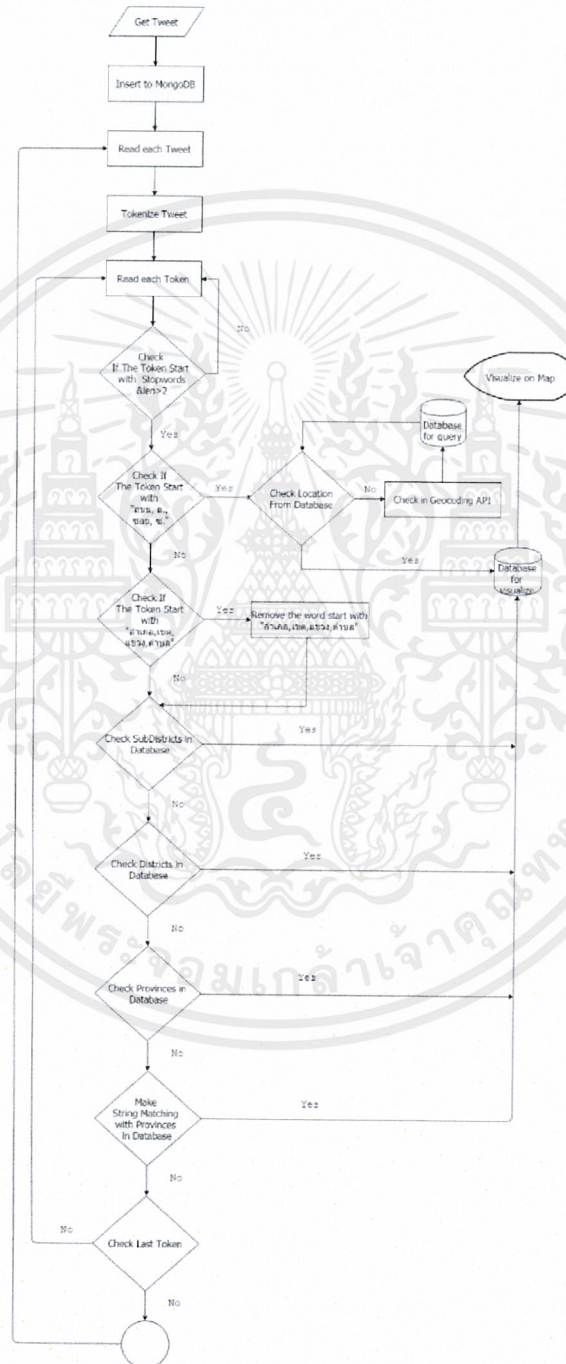


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงานและการออกแบบระบบ

3.1 โครงสร้างการออกแบบระบบ



รูปที่ 3.1 กระบวนการทำงานการกรองข่าวสารโดยการนำข้อมูลจากทวิตเตอร์มา
เพื่อกรองข่าวสารนี้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มจากการดึงทวิตโดยอาศัยแฮชแท็กน้ำท่วม (#น้ำท่วม) ข้อความที่ได้จากทวิตเตอร์เป็นรูปแบบของ JSON เราจึงนำข้อความที่ได้มาเก็บไว้ใน MongoDB จากนั้นจะเรียกแต่ละทวิตออกมาทำการตัดคำเป็นคำ ๆ เมื่อทำการตัดคำเสร็จแล้ว จะทำการแยก Stop Words ที่ไม่เกี่ยวข้องกับการกรองข่าวสารน้ำท่วม เช่น . , % \$ @ & เป็นต้น ส่วนคำอื่น ๆ จะถูกนำไปเช็ค ใน ขั้นตอนต่าง ๆ โดยเช็คจากหน่วยเล็กไปยังหน่วยใหญ่ เพราะ

1. ธรรมชาติการเขียน เขียนจาก ถ. แขวง เขต จังหวัด
2. ถ้าได้ location ของหน่วยย่อย = ได้ location ของพื้นที่
3. ในกรณี check หน่วยย่อย return > 1 เช่น ถ้าเจอ ถนนสุขุมวิท ซึ่งมีอยู่หลายจังหวัด ให้เช็คในหน่วยที่ใหญ่ถัดไป

ลำดับในการเช็คหลังจากทำการตัดคำเริ่มจาก ลำดับแรก เป็นการเช็คค่าแต่ละคำขึ้นต้นด้วย ถนน, ถ., ซอย และ ซ. หรือไม่ เนื่องจากในฐานข้อมูลดาต้าเบสของเราไม่มีข้อมูล latitude and longitude ของถนนและซอย คำที่ขึ้นต้นดังกล่าวจะถูกส่งเข้าไปหาตำแหน่งพิกัดใน Geocoder เมื่อพบตำแหน่งจะถูกนำไปเก็บในดาต้าเบสสำหรับ visualize บนแผนที่ประเทศไทย และเก็บพิกัดแยกอีกดาต้าเบส เพื่อสำรองพิกัดไว้ใช้ในอนาคต หากไม่มีค่าดังกล่าวคำทั้งหมดจะถูกส่งไปยัง ลำดับที่ 2 คือการส่งคำทั้งหมดไปเช็คในดาต้าเบสในคอลัมน์ของแขวงทั้ง 7364 records ว่ามีค่าเหล่านี้ในคอลัมน์แขวงหรือไม่ หากไม่พบก็จะส่งค่าเหล่านี้ไปยัง ลำดับที่ 3 เป็นการเช็คในคอลัมน์เขตซึ่งมี 928 records ว่าตรงกับค่าในคอลัมน์เขตหรือไม่ หากไม่พบจะส่งค่าไปเช็คใน ลำดับที่ 4 เป็นการเช็คในคอลัมน์จังหวัดซึ่งมี 77 records ว่าค่าเหล่านี้มีค่าใดตรงกับคอลัมน์จังหวัดหรือไม่ หากยังไม่พบ คำเหล่านี้จะถูกส่งไปยังลำดับสุดท้าย เป็นการทำ String Matching ด้วยวิธีของ Levenshtein ว่าค่าเหล่านี้ใกล้เคียงกับคำใดบ้างซึ่งอาจหมายถึงการใช้พิมพ์ข้อความผิดพลาด เช่นผู้ใช้พิมพ์ข้อความ กรุงเทพฯ ซึ่งไม่มีในฐานข้อมูลดาต้าเบสของเรา เราจึงต้องนำมาทำ String Matching ในขั้นตอนสุดท้าย ว่า คำว่ากรุงเทพฯ ที่ผู้ใช้พิมพ์มานั้นใกล้เคียงกับคำใดบ้างในฐานข้อมูลดาต้าเบส ซึ่งเมื่อทำ String Matching แล้วจะได้คำว่า กรุงเทพฯ หากในขั้นตอนใดขั้นตอนหนึ่งพบตำแหน่งสถานที่แล้วคำอื่นๆในประโยคจะหยุดการเช็คในทันที ซึ่งเรานำคำและพิกัดที่ได้ในลำดับต่างๆ มาเก็บไว้ในดาต้าเบส และทำการนับพื้นที่ซ้ำเพื่อคำนวณความถี่ของข้อความ และนำไปกำหนดระดับความรุนแรงของการเกิดเหตุการณ์น้ำท่วม จากนั้นจะส่งไปยัง Geocoding API เพื่อทำการ Visualize บนแผนที่ประเทศไทย

3.2 การเลือกใช้เครือข่ายสังคมออนไลน์

เครือข่ายสังคมออนไลน์ (Social network) ในปัจจุบันมีมากมายหลายเครือข่าย แต่ที่ได้รับความนิยมในประเทศไทย ได้แก่ Facebook Twitter Instagram Line และ Youtube เราไม่เลือกใช้ไลน์เนื่องจากไลน์เป็นแอปพลิเคชันแชทที่มีความเป็นส่วนตัวและไม่เปิดเผยข้อมูล เราไม่เลือกใช้ Instagram และ Youtube เพราะ ทั้ง 2 แอปพลิเคชันนี้เน้นให้ข้อมูลไปทางรูปภาพและวิดีโอมากกว่าข้อความข่าวสารที่เราต้องการ และเราไม่เลือกใช้ข้อมูล facebook ทั้ง ๆ ที่เป็นเครือข่ายสังคมออนไลน์ที่มีการอัปเดตข่าวสารแบบสาธารณะและมีความรวดเร็วในการเผยแพร่และกระจายข่าวสาร เนื่องจาก Facebook ไม่ได้มีการแสดงผลแบบ Real-Time และไม่ Support Developers ในการใช้ Facebook และเราได้เลือกใช้ Twitter ในการนำข้อมูล Twitter มา โดยอ้างอิงจากผลการประกาศรางวัล Thailand Social awards 2016 ว่าอัตราการเติบโตของ Twitter ในประเทศไทยมีมากถึง 70% นอกจากนั้น Twitter ยังมีการอัปเดตแบบ Real-Time และ Support Developers ในการให้ข้อมูลอีกด้วย

3.3 Data Collection

วิธีที่ใช้ในการดึงข้อมูลทวีตเตอร์มีหลายแบบเช่น การใช้ภาษาจาวา (Java) และการใช้ภาษาไพธอน (python) ในแรกเริ่มปัญหาพิเศษนี้เลือกใช้ภาษาจาวา แต่เนื่องจากภาษาจาวามีข้อจำกัดเกี่ยวกับทวีตเตอร์ในเรื่องของการนำข้อมูลทวีตเตอร์ออกมา ซึ่งในภาษาจาวา สามารถดึงข้อมูลทวีตเตอร์ออกมาได้เพียงครั้งละ 100 ทวีตเท่านั้น และต้องกำหนดระยะเวลาการดึงข้อมูลออกจากทวีตเตอร์ว่าจะให้นำข้อมูลออกมาทุก ๆ กี่นาทีอีกด้วย ต่างจากภาษาไพธอนที่เป็นแบบ listener ที่รองรับการดึงข้อมูลทางทวีตเตอร์มากกว่า โดยสามารถดึงข้อความได้ตลอดเวลาและไม่จำกัดจำนวนทวีตที่ดึงออกมา รวมถึงภาษาไพธอนยังมีไลบรารีต่าง ๆ ที่รองรับการทำงานที่เกี่ยวข้องกับข้อมูลจำนวนมากอีกด้วย ในขั้นตอนการดึงข้อมูลทวีตเตอร์จะอาศัยการดึงข้อมูลจากคีย์เวิร์ด “น้ำท่วม” เท่านั้น เนื่องจาก หากเลือกใช้คำอื่นร่วมด้วย เช่น ฝนตก สถานที่ที่เรานำไปพล็อตลงในแผนที่ว่าเกิดเหตุการณ์น้ำท่วมจะเป็นเพียงสถานที่ที่เกิดฝนตกเท่านั้น และนำข้อมูลที่ดึงมาได้ไปเก็บไว้ใน mongoDB เนื่องจากทวีตเตอร์ให้ข้อมูลมาในรูปแบบของ JSON และฐานข้อมูล mongoDB ก็รองรับการเก็บข้อมูลแบบ JSON โดยเลือกเก็บแค่ iduser usrName follower text hashtags date coordinates เพราะว่ามีสิ่งที่เรานำมาใช้ประมวลผลน้ำท่วมจริง ๆ คือข้อความที่ผู้ใช้ทวีตเตอร์ทวีตออกมา

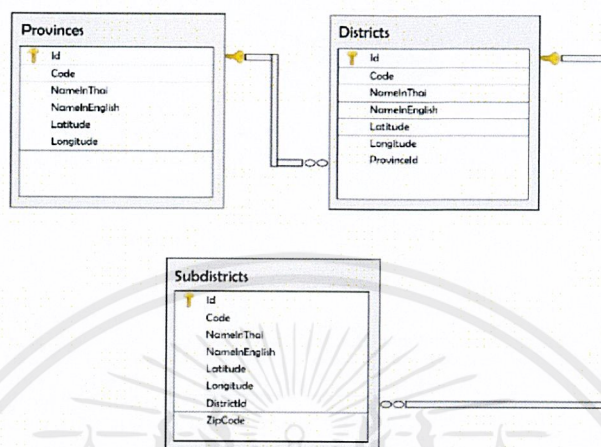
```

1  from pymongo import MongoClient
2  import json
3  from tweepy.streaming import StreamListener
4  from tweepy import OAuthHandler
5  from tweepy import Stream
6  import datetime
7  import logging
8
9  connection = MongoClient('localhost', 27017)
10 db = connection.TwitterStream
11 db.tweets.ensure_index('id', unique=True, dropDups=True)
12 collection = db.DataSET
13
14 keywords = ['บ้านท่วม']
15 language = ['th']
16
17 consumer_key = 'g08Rmn0XGUNHixbgPhgcSpeyp'
18 consumer_secret = 'XBOgsYRs3GnKwB7Grww7GDoO5bjUBt2keTeMIA3Hl8HvalcT4f'
19 access_token = '2989702846-M8greQrlamjmzzBOxG8u5l0SqCLwYRKQ37HEnr2'
20 access_token_secret = 'JNvw0iwiphxzSc lgoY84FeFaYOPVxpZeXEU6SJ5ifi2c0'
21 logger = logging.getLogger('myapp')
22 hdlr = logging.FileHandler('/FinalProject/StreamingTweet/myapp.log')
23 formatter = logging.Formatter('%(asctime)s %(levelname)s %(message)s')
24 hdlr.setFormatter(formatter)
25 logger.addHandler(hdlr)
26 logger.setLevel(logging.WARNING)
27 logger.error('We have a problem')
28 logger.info('While this is just chatty')
--

```

รูปที่ 3.2 โค้ดที่ใช้ในการดึงทวีตจากทวีตเตอร์

นอกจากนี้เราได้เพิ่มค่าเบสเพื่อเก็บ Latitude และ Longitude ของเขต แขวง และ จังหวัด เพื่อประกอบการวิเคราะห์ข่าวสารแล้วเก็บลงใน RDBMS Microsoft SQL Server



รูปที่ 3.5 แผนภาพอ็วาร์ของข้อมูลพื้นที่ในประเทศไทย ประกอบไปด้วย แขวง เขต และจังหวัด

Select * From dbo.Provinces

Id	NameInThai	NameInEnglish	Latitude	Longitude
1	กรุงเทพมหานคร	Bangkok	13.7278956	100.5241235
2	กระบี่	Krabi	8.0862997	98.9062835
3	กาญจนบุรี	Kanchanabun	14.0227797	99.5328115
4	กาฬสินธุ์	Kalasin	16.4314078	103.5058755
5	กำแพงเพชร	Kamphaeng Phet	16.4827798	99.5226617999999
6	ขอนแก่น	Khon Kaen	16.4419355	102.8359921
7	จันทบุรี	Chanthaburi	12.61134	102.1038546
8	ฉะเชิงเทรา	Chachoengsao	13.6904194	101.0779596
9	ชลบุรี	Chon Buri	13.3611431	100.9846717
10	ชัยนาท	Chai Nat	15.1851971	100.125125
11	ชัยภูมิ	Chaiyaphum	15.8068173	102.0315027
12	ชุมพร	Chumphon	10.4930496	99.1800196999999
13	เชียงราย	Chiang Rai	19.9071656	99.830955
14	เชียงใหม่	Chiang Mai	18.7877477	98.9931311
15	ตรัง	Trang	7.5593851	99.6110065
16	ตราด	Trat	12.2427563	102.5174734
17	ตาก	Tak	16.8839901	99.1258498
18	นครนายก	Nakhon Nayok	14.2069466	101.2130511
19	นครปฐม	Nakhon Pathom	13.8199206	100.0621676
20	นครพนม	Nakhon Phanom	17.392039	104.7695508
21	นครราชสีมา	Nakhon Ratchasima	14.9798997	102.0977693
22	นครศรีธรรมราช	Nakhon Si Thammarat	8.4303975	99.9631219
23	นครสวรรค์	Nakhon Sawan	15.6930072	100.1225595
24	นonthaburi	Nonthabun	13.8621125	100.5143528
25	นราธิวาส	Narathiwat	6.4254607	101.8253143
26	น่าน	Nan	18.7756318	100.7730417
27	บุรีรัมย์	Buri Ram	14.9930017	103.1029191
28	ปทุมธานี	Pathum Thani	14.0208391	100.5250276
29	ประจวบคีรีขันธ์	Prachuap Khiri Khan	11.812367	99.7973271

รูปที่ 3.6 ตัวอย่างข้อมูลค่าเบสของจังหวัดในประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Select * From dbo.Districts

100 %

Id	Code	NameInThai	NameInEnglish	Latitude	Longitude	ProvinceId
1	1001	พระนครศรีอยุธยา	Phra Nakhon	13.7564166666667	100.49925	1
2	1002	ดุสิต	Dust	13.7728	100.5154	1
3	1003	หนองจอก	Nong Chok	13.85125	100.856125	1
4	1004	บางรัก	Bang Rak	13.728	100.5232	1
5	1005	บางเขน	Bang Khen	13.867	100.628	1
6	1006	บางกะปิ	Bang Kapi	13.771	100.6475	1
7	1007	ปทุมวัน	Pathum Wan	13.7405	100.53175	1
8	1008	ป้อมปราบศัตรูพ่าย	Pom Prap Sattruphai	13.7512	100.5114	1
9	1009	พระโขนง	Phra Khanong	13.692	100.626	1
10	1010	มีนบุรี	Min Buri	13.808	100.7485	1
11	1011	ลาดกระบัง	Lat Krabang	13.743	100.789166666667	1
12	1012	ยานนาวา	Yannawa	13.6895	100.5385	1
13	1013	สัมพันธวงศ์	Samphanthawong	13.738	100.509333333333	1
14	1014	พญาไท	Phaya Thai	13.782	100.545	1
15	1015	ธนบุรี	Thon Buri	13.722	100.4848	1
16	1016	บางกอกใหญ่	Bangkok Yai	13.738	100.4805	1
17	1017	ห้วยขวาง	Huai Khwang	13.7723333333333	100.500666666667	1
18	1018	คลองสาน	Khlong San	13.7265	100.5025	1
19	1019	ตลิ่งชัน	Taling Chan	13.7675	100.439333333333	1
20	1020	บางกอกน้อย	Bangkok Noi	13.7634	100.4728	1
21	1021	บางขุนเทียน	Bang Khun Thian	13.58	100.4145	1
22	1022	ภาษีเจริญ	Phasi Charoen	13.7267142857143	100.445571428571	1
23	1023	หนองแขม	Nong Khaem	13.697	100.3545	1
24	1024	ราชบุรีบูรณะ	Rat Burana	13.6725	100.5005	1
25	1025	บางพลัด	Bang Phlat	13.787	100.49325	1
26	1026	ดินแดง	Di Daeng	13.778	100.567	1
27	1027	บึงกุ่ม	Bung Kum	13.808	100.65	1
28	1028	สาทร	Sa Thon	13.714	100.529	1
29	1029	บางยี่	Bang Su	13.82	100.529	1

รูปที่ 3.7 ตัวอย่างข้อมูลดาต้าเบสของเขตในประเทศไทย

Select * From dbo.Subdistricts

100 %

Id	Code	NameInThai	NameInEnglish	Latitude	Longitude	DistrictId	ZipCode
1	100101	พระบรมมหาราชวัง	Phra Borom Maha Ratchawang	13.751	100.492	1	10200
2	100102	วังบูรพาภิรมย์	Wang Burapha Phrom	13.744	100.499	1	10200
3	100103	วัดราชบพิธ	Wat Ratchabopit	13.75	100.499	1	10200
4	100104	สำราญราษฎร์	Samran Rat	13.751	100.503	1	10200
5	100105	ศาลเจ้าพ่อเสือ	San Chaopho Suea	13.754	100.497	1	10200
6	100106	เสาชิงช้า	Sao Chingcha	13.753	100.5	1	10200
7	100107	บวรนิเวศ	Bowon Newet	13.757	100.501	1	10200
8	100108	ตลาดยอด	Talat Yot	13.76	100.498	1	10200
9	100109	ชนะสงคราม	Chana Songkhram	13.762	100.495	1	10200
10	100110	บ้านหน้าถนน	Ban Phan Thom	13.762	100.503	1	10200
11	100111	บางขุนพรหม	Bang Khun Phrom	13.765	100.505	1	10200
12	100112	วัดสามพระยา	Wat Sam Phraya	13.768	100.499	1	10200
13	100201	ดุสิต	Dust	13.772	100.513	2	10300
14	100202	วชิรพยาบาล	Wachira Phayaban	13.778	100.505	2	10300
15	100203	สวนจิตรลดา	Suan Chit Lada	13.767	100.52	2	10300
16	100204	สี่แยกมหานาค	Si Yaek Maha Nak	13.758	100.517	2	10300
17	100206	ถนนนครไชยศรี	NULL	13.789	100.522	2	10300
18	100301	กระหม่อม	NULL	13.823	100.897	3	10530
19	100302	หนองจอก	NULL	13.869	100.885	3	10530
20	100303	คลองสิบ	NULL	13.914	100.82	3	10530
21	100304	คลองสิบสอง	NULL	13.914	100.88	3	10530
22	100305	โคกเม็ก	NULL	13.84	100.836	3	10530
23	100306	คูน้ำเหนือ	NULL	13.872	100.81	3	10530
24	100307	ลำผักชี	NULL	13.797	100.844	3	10530
25	100308	ลำไยฝั่ง	NULL	13.781	100.877	3	10530
26	100401	มหาพฤฒาราม	NULL	13.734	100.52	4	10500
27	100402	สีลม	NULL	13.73	100.525	4	10500
28	100403	สุริยวงศ์	NULL	13.724	100.53	4	10500
29	100404	บางรัก	NULL	13.727	100.527	4	10500

รูปที่ 3.8 ตัวอย่างข้อมูลดาต้าเบสของแขวงในประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

select * from dbo.DataFromGoogle

100 %

Results Messages

	Name	Latitude	Longitude
1	ก.พหลโยธิน	16.5852954	99.4371755
2	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
3	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
4	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
5	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
6	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
7	ก.พหลโยธิน	16.5852954	99.4371755
8	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
9	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
10	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
11	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
12	ก.งามวงศ์วาน	13.8524042	100.5525372
13	ก.งามวงศ์วาน	13.8524042	100.5525372
14	ก.สามัคคี	13.8826151	100.5308715
15	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
16	ก.งามวงศ์วาน	13.8524042	100.5525372
17	ก.พหลโยธิน	16.5852954	99.4371755
18	ก.งามวงศ์วาน	13.8524042	100.5525372
19	ชอยชินเขต	13.8636156	100.552757
20	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
21	ก.นสามัคคี	13.8826151	100.5308715
22	ก.งามวงศ์วาน	13.8524042	100.5525372
23	ชอยประชาชนนนทบุรี	13.8482607	100.5431315
24	ก.งามวงศ์วาน	13.8524042	100.5525372
25	ก.นสามัคคี	13.8826151	100.5308715

รูปที่ 3.9 ตัวอย่างข้อมูลดาต้าเบสที่ได้จาก Geocoder

select * from dbo.DataDashboard

100 %

Results Messages

	Counttweet	SumFrequency
1	22776	6685

รูปที่ 3.10 ตัวอย่างข้อมูลดาต้าเบส Dashboard ที่จะส่งไป Visualize บนแผนที่ประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
select * from dbo.DataVisual1
```

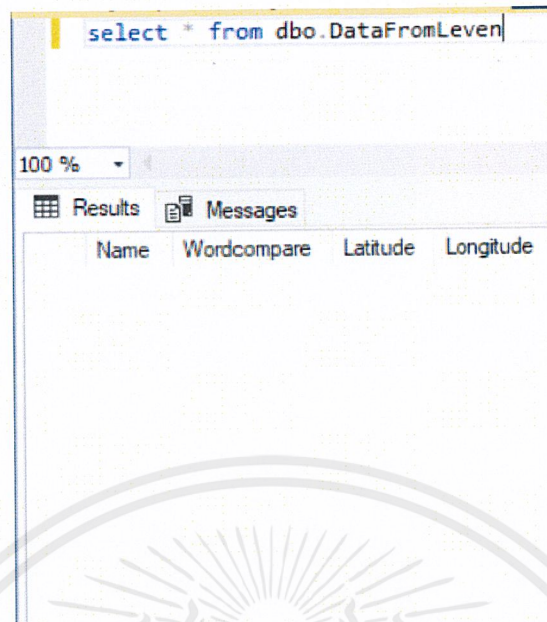
100 %

Results Messages

	Name	Latitude	Longitude	Frequency
1	ช.เทศมาลง	13.845003	100.4127288	11
2	ช.ลาดพร้าว	13.8042086	100.5858762	1
3	ช.สามัคคี	13.9022327	100.575679	16
4	ชอย	13.7326038	100.5570528	13
5	ชอย	13.8941365	100.5845739	3
6	ชอยไฮแสง	13.8577524	100.5378938	2
7	ชอยงามวงศ์	13.8524042	100.5525372	1
8	ชอยชินเชล	13.8636156	100.552757	1
9	ชอยบาง	13.6401623	100.3598766	1
10	ชอยประชาชนนนทบุรี	13.8482607	100.5431315	14
11	ชอยเปรม	13.7450662	100.5098357	1
12	ชอยพัฒนาการ	13.7311792	100.6286988	2
13	ชอยจัสติกรรมย์	14.0409822	100.6140184	5
14	ชอยสามัคคี	13.8856156	100.5139643	21
15	ชอยเสนา	12.6018454	99.9483289	1
16	ชอยเสือ	13.7403237	100.5133077	10
17	ชอยอารีย์	13.7823676	100.5431884	1
18	ชอยอินทามระ	13.7885928	100.5517995	3
19	ก.งามวงศ์วาน	13.8524042	100.5525372	7
20	ก.งามวงศ์วาน ๒	13.8535425	100.5504912	40
21	ก.แจ้งวัฒนะ	13.887298	100.5759382	11
22	ก.พหลโยธิน	16.5852954	99.4371755	103
23	ก.แพรภษา	13.5706236	100.6499472	11
24	ก.รัตนวิเศษ	13.8706607	100.4653703	29
25	ก.รัตนวิเศษ ๒	13.8662655	100.4929385	1
26	ก.ลาดพร้าว	13.7884168	100.6053308	1

รูปที่ 3.11 ตัวอย่างข้อมูลดาต้าเบสที่จะส่งไป Visualize บนแผนที่ประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ตัวอย่างข้อมูลดาต้าเบสที่ได้จากการทำ Levenshtein

3.4 Twitter Analysis

ในขั้นตอนนี้คือ Word Optimization เพื่อคัดคำว่าจะใช้คำใดมาประมวลผลเพื่อเพิ่มความเร็วของการทำงานในขั้นตอนต่างๆหลังจากการตัดคำ (Word Tokenization) โดยมีกระบวนการดังนี้

- 1) ตัดคำที่มีจำนวนพยางค์น้อยกว่า 2 ตัวอักษร เช่น ก็ ละ
- 2) ตัดอักขระพิเศษต่าง ๆ เช่น , % & แต่ไม่ตัดแฮชแท็ก (#) ออก เนื่องจากผู้ใช้อาจพิมพ์สถานที่ที่เกิดเหตุการณ์นี้ว่าท่วมลงไปหลังแฮชแท็ก
- 3) ตัดตัวเลขออกจากการทำ Word Tokenization
- 4) ตัดตัดอักขระภาษาอังกฤษออกจากการทำ Word Tokenization
- 5) เพิ่ม Row “กทม” ใน Table Provinces
- 6) คำที่ขึ้นต้นด้วย ซอย, ถนน, ซ. และ ถ. จะถูกส่งเข้าไปค้นหาตำแหน่งใน Geocoder แทนการส่งค่าเข้าไปในดาต้าเบสและการทำ String Matching เนื่องจากในดาต้าเบสไม่มีฟิลด์ที่เป็นตำแหน่งของถนน และซอย การส่งค่าเข้า Geocoder จะเป็นการช่วยลดระยะเวลาในการประมวลผล
- 7) ตัวอักษรที่น้อยกว่า 5 ตัวอักษรจะต้องพิมพ์ถูกต้องทุกตัวอักษรเมื่อเทียบในดาต้าเบสจึงจะถือว่าเป็นตำแหน่งที่ถูกต้องในแผนที่ที่จะพล็อต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

class Customize () :
def CusVil (Word) :
    #for h in x:
    if Word(0:3) in WordSelect1: #Check "nuu" in geocoder
        g = geocoder.google(Word)
        lat = g.lat
        long = g.lng
        #print(Word, lat, long)
        if lat and long != None:
            SQLCommand = ("Insert into dbo.DataFromGoogle(Name, Latitude, Longitude) VALUES (?,?,?) ")
            Param = (Word, lat, long)
            cursor1.execute(SQLCommand, Param)
            print(Param)
            print("{0} row inserted to DataFromGoogle successfully.".format(cursor1.rowcount))
            #print(Word, lat, long)
            DataReal.commit()
            return True
        else:
            #print("False")
            return False
    elif Word(0:2) in WordSelect1: #Check "อ.น." in geocoder
        g = geocoder.google(Word)
        lat = g.lat
        long = g.lng
        #print(Word, lat, long)
        if lat and long != None:
            SQLCommand = ("Insert into dbo.DataFromGoogle(Name, Latitude, Longitude) VALUES (?,?,?) ")
            Param = (Word, lat, long)
            cursor1.execute(SQLCommand, Param)
            print(Param)
            print("{0} row inserted to DataFromGoogle successfully.".format(cursor1.rowcount))
            DataReal.commit()
            return True
        else:
            #print("False")
            return False

```

รูปที่ 3.14 โค้ดที่ใช้ในการทำ optimize word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

elif Word(0:8) in WordSelectI: #Check 'จังหวัด' in geocoder
    g = geocoder.google(h)
    lat = g.lat
    long = g.lng
    #print(Word, lat, long)
    if lat and long != None:
        SQLCommand = ('Insert into dbo.DataFromGoogle(Name,Latitude,Longitude) VALUES (?,?,?) ')
        Param = (Word, lat, long)
        cursor1.execute(SQLCommand, Param)
        print(Param)
        print('{0} row inserted to DataFromGoogle successfully.'.format(cursor1.rowcount))
        #print(Word, lat, long)
        DataReal.commit()
        return True
    else:
        #print('False')
        return False
else:
    #print('False')
    return False

def CusDis (Word) :
    try:
        #for w in Word:
        if Word(0:5) in WordSelectSubDistrict and Word(5) == CutSpace: #Cut prefix 'จังหวัด' then check in DB
            word = Word(5:None)
            DB.SearchProvinces(word)
            DB.SearchSubdistricts(word)
            DB.SearchDistricts(word)
            return True
        elif Word(0:5) in WordSelectSubDistrict and Word(5) != CutSpace:
            word = Word(5:None)
            DB.SearchProvinces(word)
            DB.SearchSubdistricts(word)
            DB.SearchDistricts(word)
            return True

```

รูปที่ 3.15 โค้ดที่ใช้ในการทำ optimize word

```

elif Word(0:3) in WordSelectSubDistrict and Word(3) == CutSpace: #Cut prefix '100' then check in DB
    word = Word(4:None)
    DB.SearchProvinces(word)
    DB.SearchSubdistricts(word)
    DB.SearchDistricts(word)
    return True
elif Word(0:3) in WordSelectSubDistrict and Word(4) != CutSpace:
    word = Word(5:None)
    DB.SearchProvinces(word)
    DB.SearchSubdistricts(word)
    DB.SearchDistricts(word)
    return True
elif Word(0:4) in WordSelectSubDistrict and Word(4) == CutSpace: #Cut prefix '1000' then check in DB
    word = Word(5:None)
    DB.SearchProvinces(word)
    DB.SearchSubdistricts(word)
    DB.SearchDistricts(word)
    return True
elif Word(0:4) in WordSelectSubDistrict and Word(5) != CutSpace:
    word = Word(4:None)
    DB.SearchProvinces(word)
    DB.SearchSubdistricts(word)
    DB.SearchDistricts(word)
    return True
else:
    return False
except Exception:
    pass

```

รูปที่ 3.16 โค้ดที่ใช้ในการทำ optimize word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

class DB():
    def SearchSubdistricts(x): # Search token from DB subdistricts
        #SQLCommand = ('SELECT NameInThai, Latitude, Longitude FROM dbo.Subdistricts Where NameInThai LIKE '%'+ x + '%';')
        SQLCommand = ('SELECT NameInThai, Latitude, Longitude FROM dbo.Subdistricts Where NameInThai = "' + x + '"');
        cursor.execute(SQLCommand)
        results = cursor.fetchall()
        if results != {}:
            for b in results:
                Name = b(0)
                Lat = b(1)
                Long = b(2)
                Param = (Name, Lat, Long)
                SQLCommand1 = ('Insert into dbo.DataFromDB(Name, Latitude, Longitude) VALUES (?,?,?)')
                cursor1.execute(SQLCommand1, Param)
                print(Param)
                print('{0} row inserted to DataFromDB successfully.'.format(cursor1.rowcount))
                DataReal.commit()
            return True
        else:
            return False

    def SearchDistricts(x): # Search token from DB districts
        SQLCommand = ('SELECT NameInThai, Latitude, Longitude FROM dbo.Districts Where NameInThai = "' + x + '"');
        cursor.execute(SQLCommand)
        results = cursor.fetchall()
        if results != {}:
            for b in results:
                Name = b(0)
                Lat = b(1)
                Long = b(2)
                Param = (Name, Lat, Long)
                SQLCommand1 = ('Insert into dbo.DataFromDB(Name, Latitude, Longitude) VALUES (?,?,?)')
                cursor1.execute(SQLCommand1, Param)
                print(Param)
                print('{0} row inserted to DataFromDB successfully.'.format(cursor1.rowcount))
                DataReal.commit()
            return True
        else:
            return False

```

รูปที่ 3.17 โค้ดที่ใช้ในการ search คำในฐานข้อมูล

```

    return True
else:
    return False

def SearchProvinces (x) :# Search token from DB Provinces
SQLCommand = (*SELECT NameInThai,Latitude,Longitude FROM dbo.Provinces Where NameInThai = '*'+x+'*')
cursor.execute(SQLCommand)
results = cursor.fetchall()
if results != ():
    for b in results:
        Name = b(0)
        Lat = b(1)
        Long = b(2)
        Param = (Name, Lat, Long)
        SQLCommand1 = (*Insert into dbo.DataFromDB(Name,Latitude,Longitude) VALUES (?,?,:) *)
        cursor1.execute(SQLCommand1, Param)
        print(Param)
        print('{0} row inserted to DataFromDB successfully.'.format(cursor1.rowcount))
        DataReal.commit()
    return True
else:
    return False

```

รูปที่ 3.18 โค้ดที่ใช้ในการ search คำในฐานข้อมูล

3.5 การ Visualize บนแผนที่ประเทศไทย

สำหรับส่วนของการระบุตำแหน่งของสถานที่ที่เกิดเหตุการณ์น้ำท่วม Google Map API มี marker ให้เราใช้หมดระบุตำแหน่งได้ทันที แต่เนื่องจากเราต้องการความแปลกใหม่ในเรื่องของ marker ที่ marker ควรเป็นสิ่งที่สามารถบอกเล่าเรื่องราวได้มากกว่าที่ marker ปกติจะทำได้ ดังนั้นเราจึงได้คิดที่จะสร้าง marker ที่สามารถขยับได้เป็น animation และสามารถเปลี่ยนสภาพไปตามความถี่ของข้อมูลที่เรามีใน database รวมไปถึงตัว marker จะตรงกับความหมายที่เราต้องการสื่อคือ "การเตือนภัย" ทำให้ในขั้นตอนของ Map Visualization การปรับแต่ง marker เป็นส่วนสำคัญมาก เพราะการจะแสดงผลแบบ dynamic ในขั้นตอน implementation นั้นมีความซับซ้อนค่อนข้างมาก และตัวอย่างให้ศึกษานั้นมีน้อย นอกจากนั้นเราได้เพิ่มฟังก์ชันการปรับมุมมองให้สามารถเน้นไปในพื้นที่ที่ผู้ใช้งานต้องการเพื่อดูแผนที่โดยรวมว่าระยะทางที่ผู้ใช้อยู่กับระยะทางที่เกิดเหตุการณ์น้ำท่วมมีความใกล้/ไกลมากน้อยเพียงใด


```

347
348 .my-marker2 {
349
350     position: absolute;
351     z-index: 2;
352 }
353
354 .pulse2 {
355     width: 17px; /* ความกว้างของพลาสมา */
356     height: 17px; /* ความสูงของพลาสมา */
357     border: 6px solid #ff8b33; /* สีของพลาสมา */
358
359     -webkit-border-radius: 30px;
360     -moz-border-radius: 30px;
361     border-radius: 30px;
362     background-color: #ec8030;
363     z-index: 10;
364     position: absolute;
365 }
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

รูปที่ 3.21 โค้ดที่ใช้ในการทำ marker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1  var map;
2  var popup;
3  var Popup;
4  var marker;
5  var geocoder;
6  var ourRequest;
7  var infoWindow;
8  var infoBubble;
9
10 window.alert("เพื่อให้ข้อมูลเป็นแบบ Realtime รบกวนผู้ใช้ Refresh Website เป็นระยะ");
11
12
13 var date = (moment().format('DD-MM-YYYY'));
14 var d = (moment().format('DD-MM-YYYY, h:mm:ss a'));
15 console.log(date);
16 function init() {
17
18     //definesPopupClass();
19
20     //var mapCenter = new google.maps.LatLng(18.796143, 98.979263);
21     var mapCenter = new google.maps.LatLng(13.736717, 100.523186);
22
23     map = new google.maps.Map(document.getElementById('map'), {
24         zoom: 6,
25         gestureHandling: 'greedy',
26         center: mapCenter,
27         disableDefaultUI: true,
28         styles:
29     [
30     {
31         {
32             "featureType": "administrative.country",
33             "elementType": "geometry",
34             "stylers": [
35                 {
36                     "visibility": "simplified"
37                 }
38             ],
39             "hue": "#ff0000"
40         }
41     ]
42     }
43     });
44
45

```

รูปที่ 3.22 โค้ดที่ใช้ในการแสดงผลเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

48 ourRequest = new XMLHttpRequest();
49 ourRequest.open("GET", "http://log(host)/final/example/jsonlog/' + date + '.json');
50 ourRequest.onload = function() {
51     if (ourRequest.status >= 200 && ourRequest.status < 400) {
52         var ourData = JSON.parse(ourRequest.responseText);
53         console.log("YES IT WORKS");
54         //console.log(ourData.Result.location[0].Geometry);
55         //console.log(ourData.Result.location.Geometry);
56         //console.log(ourData.Result.location[0].Freq);
57         var counter = 0;
58         for (i=0;i<=ourData.Result.location.length;i++){
59             counter = counter + 1;
60             if (ourData.Result.location[i].Freq<=30) {
61                 addmarker(ourData.Result.location[i].Geometry,ourData.Result.location[i].PlaceName,ourData.Result.location[i].Freq);
62             }
63             else if (ourData.Result.location[i].Freq<=100) {
64                 addmarker2(ourData.Result.location[i].Geometry,ourData.Result.location[i].PlaceName,ourData.Result.location[i].Freq);
65             }
66             else {
67                 addmarker3(ourData.Result.location[i].Geometry,ourData.Result.location[i].PlaceName,ourData.Result.location[i].Freq);
68             }
69
70             var counttweet = ourData.Result.Sum[0].Counttweet;
71             var sumfreq = ourData.Result.Sum[0].Sumfreq;
72             console.log(counttweet);
73             var a = document.getElementById("peep4");
74             a.innerHTML = "Dashboard" + "<br>" + "จำนวนtweetทั้งหมด : "+counttweet + "<br>" + "ความถี่รวมของจุดทั้งหมด : "+sumfreq
75             + "<br>" + "จำนวนจุดทั้งหมด : "+ counter +
76             "<br>" + "Tweetล่าสุดมี : " + d;
77
78             var b = document.getElementById("peep5");
79             b.innerHTML = "Technique" + "<br>" + "หลักStringMatching : " + "Levenshtein distance"
80             + "<br>" + "การค้นหาค่า : " + "Deepout จาก TRUE ผ่านในผล CNN ตามลำดับ";
81
82         }
83     }
84     else {
85         console.log("We connected to the server, but it returned an error.");
86     }
87 }
88 };

```

รูปที่ 3.23 โค้ดที่ใช้ในการแสดงผลเว็บไซต์

```

89 ourRequest.send();
90
91
92 infowindow = new google.maps.InfoWindow;
93
94 function addmarker(coords,placename,freq){
95     marker = new RichMarker({
96         position:new google.maps.LatLng(coords),
97         map: map,
98         content: "<div class='my marker'> "
99         + "<div class='dot'><div>"
100         + "<div class='pulse'><div>"
101         + "<div>"
102         });
103     marker.addListener("mouseover",function(){
104         infobubble = new InfoBubble({
105             map: map,
106             height:200,
107             width:200,
108             content: "<div class='mylabel1'>" + "สถานที่ : " + placename + "<div>" + "<br>" +
109             "<div class='mylabel2'>" + "ค่าความถี่ : " + new google.maps.LatLng(coords) + "<div>" + "<br>" + "<div class='mylabel3'>" + "จำนวนtweet : " + freq + " + "ครั้ง" + "<div>";
110             position: new google.maps.LatLng(coords),
111             anchor: position,
112             disableAutoPan: true,
113             arrowPosition: 30,
114             arrowStyle: 2
115         });
116     });
117     infobubble.open(map);
118
119     });
120
121     marker.addListener("mouseout",function(){
122         infobubble.close(map);
123     });
124 }
125 }

```

รูปที่ 3.24 โค้ดที่ใช้ในการแสดงผลเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

135 |
136 | function addmarker2(coords, placename, freq){
137 |     marker = new RichMarker({
138 |         position: new google.maps.LatLng(coords),
139 |         map: map,
140 |         content: '<div class="my-marker2" id="markerlick">' +
141 |             '<div class="dot2"><div>' +
142 |             '<div class="pulse2"><div>' +
143 |                 '<div>'
144 |             </div>
145 |         });
146 |
147 |     marker.addListener("mouseover", function(){
148 |         infoBubble = new InfoBubble({
149 |             map: map,
150 |             height: 200,
151 |             width: 200,
152 |             content: '<div class="mylabel1">' + "ชื่อสถานที่: " + placename + '</div>' + '<br>' +
153 |                 '<div class="mylabel2">' + "จำนวน: " + new google.maps.LatLng(coords) + '</div>' + '<br>' + '<div class="mylabel3">' + "ความถี่: " + freq + " " + "ครั้ง" + '</div>',
154 |             position: new google.maps.LatLng(coords),
155 |             // disableAutoPan: true,
156 |             disableAutoPan: true,
157 |             arrowPosition: 39,
158 |             arrowStyle: 2
159 |         });
160 |
161 |         infoBubble.open(map);
162 |     });
163 |
164 |     marker.addListener("mouseout", function(){
165 |         infoBubble.close(map);
166 |     });
167 | }

```

รูปที่ 3.25 โค้ดที่ใช้ในการแสดงผลเว็บไซต์

```

174 - function addmarker3(coords, placename, freq){
175 -     marker = new RichMarker({
176 -         position: new google.maps.LatLng(coords),
177 -         map: map,
178 -         content: '<div class="my-marker3" id="markerlick">' +
179 -             '<div class="dot3"><div>' +
180 -             '<div class="pulse3"><div>' +
181 -                 '<div>'
182 -             </div>
183 -         });
184 -
185 -     marker.addListener("mouseover", function(){
186 -         infoBubble = new InfoBubble({
187 -             map: map,
188 -             height: 200,
189 -             width: 200,
190 -             content: '<div class="mylabel1">' + "ชื่อสถานที่: " + placename + '</div>' + '<br>' +
191 -                 '<div class="mylabel2">' + "จำนวน: " + new google.maps.LatLng(coords) + '</div>' + '<br>' + '<div class="mylabel3">' + "ความถี่: " + freq + " " + "ครั้ง" + '</div>',
192 -             position: new google.maps.LatLng(coords),
193 -             // disableAutoPan: true,
194 -             disableAutoPan: true,
195 -             arrowPosition: 39,
196 -             arrowStyle: 2
197 -         });
198 -
199 -         infoBubble.open(map);
200 -     });
201 -
202 -     marker.addListener("mouseout", function(){
203 -         infoBubble.close(map);
204 -     });
205 - }
206 -
207 -
208 -
209 -
210 -
211 -
212 -
213 -
214 -
215 -
216 -
217 -
218 -
219 -
220 -
221 -
222 -
223 -
224 -
225 -
226 -
227 -
228 -
229 -
230 -
231 -
232 -
233 -
234 -
235 -
236 -
237 -
238 -
239 -
240 -
241 -
242 -
243 -
244 -
245 -
246 -
247 -
248 -
249 -
250 -
251 -
252 -
253 -
254 -
255 -
256 -
257 -
258 -
259 -
260 -
261 -
262 -
263 -
264 -
265 -
266 -
267 -
268 -
269 -
270 -
271 -
272 -
273 -
274 -
275 -
276 -
277 -
278 -
279 -
280 -
281 -
282 -
283 -
284 -
285 -
286 -
287 -
288 -
289 -
290 -
291 -
292 -
293 -
294 -
295 -
296 -
297 -
298 -
299 -
300 -
301 -
302 -
303 -
304 -
305 -
306 -
307 -
308 -
309 -
310 -
311 -
312 -
313 -
314 -
315 -
316 -
317 -
318 -
319 -
320 -
321 -
322 -
323 -
324 -
325 -
326 -
327 -
328 -
329 -
330 -
331 -
332 -
333 -
334 -
335 -
336 -
337 -
338 -
339 -
340 -
341 -
342 -
343 -
344 -
345 -
346 -
347 -
348 -
349 -
350 -
351 -
352 -
353 -
354 -
355 -
356 -
357 -
358 -
359 -
360 -
361 -
362 -
363 -
364 -
365 -
366 -
367 -
368 -
369 -
370 -
371 -
372 -
373 -
374 -
375 -
376 -
377 -
378 -
379 -
380 -
381 -
382 -
383 -
384 -
385 -
386 -
387 -
388 -
389 -
390 -
391 -
392 -
393 -
394 -
395 -
396 -
397 -
398 -
399 -
400 -
401 -
402 -
403 -
404 -
405 -
406 -
407 -
408 -
409 -
410 -
411 -
412 -
413 -
414 -
415 -
416 -
417 -
418 -
419 -
420 -
421 -
422 -
423 -
424 -
425 -
426 -
427 -
428 -
429 -
430 -
431 -
432 -
433 -
434 -
435 -
436 -
437 -
438 -
439 -
440 -
441 -
442 -
443 -
444 -
445 -
446 -
447 -
448 -
449 -
450 -
451 -
452 -
453 -
454 -
455 -
456 -
457 -
458 -
459 -
460 -
461 -
462 -
463 -
464 -
465 -
466 -
467 -
468 -
469 -
470 -
471 -
472 -
473 -
474 -
475 -
476 -
477 -
478 -
479 -
480 -
481 -
482 -
483 -
484 -
485 -
486 -
487 -
488 -
489 -
490 -
491 -
492 -
493 -
494 -
495 -
496 -
497 -
498 -
499 -
500 -
501 -
502 -
503 -
504 -
505 -
506 -
507 -
508 -
509 -
510 -
511 -
512 -
513 -
514 -
515 -
516 -
517 -
518 -
519 -
520 -
521 -
522 -
523 -
524 -
525 -
526 -
527 -
528 -
529 -
530 -
531 -
532 -
533 -
534 -
535 -
536 -
537 -
538 -
539 -
540 -
541 -
542 -
543 -
544 -
545 -
546 -
547 -
548 -
549 -
550 -
551 -
552 -
553 -
554 -
555 -
556 -
557 -
558 -
559 -
560 -
561 -
562 -
563 -
564 -
565 -
566 -
567 -
568 -
569 -
570 -
571 -
572 -
573 -
574 -
575 -
576 -
577 -
578 -
579 -
580 -
581 -
582 -
583 -
584 -
585 -
586 -
587 -
588 -
589 -
590 -
591 -
592 -
593 -
594 -
595 -
596 -
597 -
598 -
599 -
600 -
601 -
602 -
603 -
604 -
605 -
606 -
607 -
608 -
609 -
610 -
611 -
612 -
613 -
614 -
615 -
616 -
617 -
618 -
619 -
620 -
621 -
622 -
623 -
624 -
625 -
626 -
627 -
628 -
629 -
630 -
631 -
632 -
633 -
634 -
635 -
636 -
637 -
638 -
639 -
640 -
641 -
642 -
643 -
644 -
645 -
646 -
647 -
648 -
649 -
650 -
651 -
652 -
653 -
654 -
655 -
656 -
657 -
658 -
659 -
660 -
661 -
662 -
663 -
664 -
665 -
666 -
667 -
668 -
669 -
670 -
671 -
672 -
673 -
674 -
675 -
676 -
677 -
678 -
679 -
680 -
681 -
682 -
683 -
684 -
685 -
686 -
687 -
688 -
689 -
690 -
691 -
692 -
693 -
694 -
695 -
696 -
697 -
698 -
699 -
700 -
701 -
702 -
703 -
704 -
705 -
706 -
707 -
708 -
709 -
710 -
711 -
712 -
713 -
714 -
715 -
716 -
717 -
718 -
719 -
720 -
721 -
722 -
723 -
724 -
725 -
726 -
727 -
728 -
729 -
730 -
731 -
732 -
733 -
734 -
735 -
736 -
737 -
738 -
739 -
740 -
741 -
742 -
743 -
744 -
745 -
746 -
747 -
748 -
749 -
750 -
751 -
752 -
753 -
754 -
755 -
756 -
757 -
758 -
759 -
760 -
761 -
762 -
763 -
764 -
765 -
766 -
767 -
768 -
769 -
770 -
771 -
772 -
773 -
774 -
775 -
776 -
777 -
778 -
779 -
780 -
781 -
782 -
783 -
784 -
785 -
786 -
787 -
788 -
789 -
790 -
791 -
792 -
793 -
794 -
795 -
796 -
797 -
798 -
799 -
800 -
801 -
802 -
803 -
804 -
805 -
806 -
807 -
808 -
809 -
810 -
811 -
812 -
813 -
814 -
815 -
816 -
817 -
818 -
819 -
820 -
821 -
822 -
823 -
824 -
825 -
826 -
827 -
828 -
829 -
830 -
831 -
832 -
833 -
834 -
835 -
836 -
837 -
838 -
839 -
840 -
841 -
842 -
843 -
844 -
845 -
846 -
847 -
848 -
849 -
850 -
851 -
852 -
853 -
854 -
855 -
856 -
857 -
858 -
859 -
860 -
861 -
862 -
863 -
864 -
865 -
866 -
867 -
868 -
869 -
870 -
871 -
872 -
873 -
874 -
875 -
876 -
877 -
878 -
879 -
880 -
881 -
882 -
883 -
884 -
885 -
886 -
887 -
888 -
889 -
890 -
891 -
892 -
893 -
894 -
895 -
896 -
897 -
898 -
899 -
900 -
901 -
902 -
903 -
904 -
905 -
906 -
907 -
908 -
909 -
910 -
911 -
912 -
913 -
914 -
915 -
916 -
917 -
918 -
919 -
920 -
921 -
922 -
923 -
924 -
925 -
926 -
927 -
928 -
929 -
930 -
931 -
932 -
933 -
934 -
935 -
936 -
937 -
938 -
939 -
940 -
941 -
942 -
943 -
944 -
945 -
946 -
947 -
948 -
949 -
950 -
951 -
952 -
953 -
954 -
955 -
956 -
957 -
958 -
959 -
960 -
961 -
962 -
963 -
964 -
965 -
966 -
967 -
968 -
969 -
970 -
971 -
972 -
973 -
974 -
975 -
976 -
977 -
978 -
979 -
980 -
981 -
982 -
983 -
984 -
985 -
986 -
987 -
988 -
989 -
990 -
991 -
992 -
993 -
994 -
995 -
996 -
997 -
998 -
999 -

```

รูปที่ 3.26 โค้ดที่ใช้ในการแสดงผลเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

215 ~ function show() {
216 ~     if (x.style.visibility === "hidden") {
217 ~         x.style.visibility = "visible";
218 ~         x.style.marginRight = "0px";
219 ~
220 ~     } else {
221 ~         x.style.visibility= "hidden";
222 ~         x.style.marginRight = "-200px";
223 ~     }
224 ~ }
225 ~
226 ~ document.getElementById('peep1').addEventListener('click', function() {
227 ~     showinfo();
228 ~ });
229 ~
230 ~ function showinfo(){
231 ~     if (navigator.geolocation) {
232 ~         navigator.geolocation.getCurrentPosition(function(position) {
233 ~             var pos = [
234 ~                 lat: position.coords.latitude,
235 ~                 lng: position.coords.longitude
236 ~             ];
237 ~
238 ~             var marker = new RichMarker({
239 ~                 position:new google.maps.LatLng(pos),
240 ~                 map: map,
241 ~                 content: '<div class="pulsenew"><div>' + '<div class="pinnew"><div>'
242 ~             });
243 ~
244 ~             //infoWindow.setPosition(pos);
245 ~             //infoWindow.setContent('You are here');
246 ~             //infoWindow.open(map);
247 ~
248 ~             map.setCenter(pos);
249 ~             map.setZoom(11);
250 ~             function() {
251 ~                 handleLocationError(true, infoWindow, map.getCenter());
252 ~             });
253 ~         } else {
254 ~             //Browser doesn't support Geolocation
255 ~             handleLocationError(false, infoWindow, map.getCenter());
256 ~         }
257 ~     }
258 ~ }

```

รูปที่ 3.27 โค้ดที่ใช้ในการแสดงผลเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

260 function handleLocationError(browserHasGeolocation, infoWindow, pos) {
261     infoWindow.setPosition(pos);
262     infoWindow.setContent(browserHasGeolocation ?
263         'Error: The Geolocation service failed.' :
264         'Error: Your browser doesn\'t support geolocation.'):
265     infoWindow.open(map);
266 }
267 }
268
269
270
271 geocoder = new google.maps.Geocoder;
272 var infowindow = new google.maps.InfoWindow;
273
274 document.getElementById("address")
275     .addEventListener("keyup", function(event) {
276     event.preventDefault();
277     if (event.keyCode === 13) {
278         document.getElementById("address").click();
279         setCamera();
280     }
281 });
282
283
284 // โคลด latlng
285 function geocodeLatLng(geocoder, map, infowindow) {
286     var input = document.getElementById('latlng').value;
287     var latlngStr = input.split(',', 2);
288     var latlng = {lat: parseFloat(latlngStr[0]), lng: parseFloat(latlngStr[1])};
289     geocoder.geocode({'location': latlng}, function(results, status) {
290         if (status === 'OK') {
291             if (results[0]) {
292                 map.setCenter(results[0].geometry.location);
293                 map.setZoom(8);
294
295                 // infowindow.setContent(results[0].formatted_address);
296                 infowindow.open(map, marker);
297             } else {
298                 window.alert('No results found');
299             }
300         } else {
301             window.alert('Geocoder failed due to: ' + status);
302         }
303     });
304 }
305
306 // โคลด รัศมีเป็นค่า
307

```

รูปที่ 3.28 โค้ดที่ใช้ในการแสดงผลเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

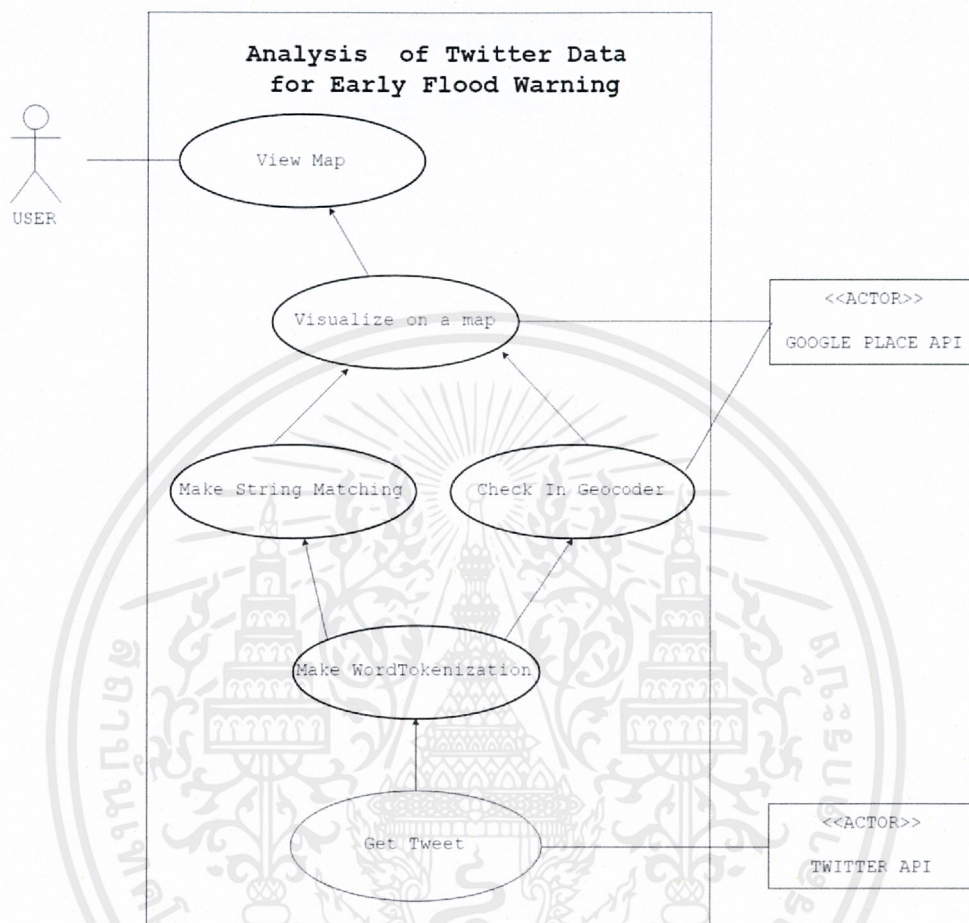
307
308 ✓ function codeAddress() {
309
310     var address = document.getElementById('address').value;
311
312 ✓     geocoder.geocode( { 'address': address}, function(results, status) {
313 ✓         if (status == 'OK') {
314             map.setCenter(results[0].geometry.location);
315             map.setZoom(8);
316
317 ✓         } else {
318             alert('Geocode was not successful for the following reason: ' + status);
319         }
320     });
321 }
322
323
324 //โค้ด set camera
325
326 ✓ function setCamera() {
327
328     var address = document.getElementById('address').value;
329
330 ✓     geocoder.geocode( { 'address': address}, function(results, status) {
331 ✓         if (status == 'OK') {
332             map.setCenter(results[0].geometry.location);
333             map.setZoom(12);
334
335 ✓         } else {
336             alert('Your place is not recognized: ' + status);
337         }
338     });
339 }
340
341
342
343 // โหมดเต็ม
344

```

รูปที่ 3.29 โค้ดที่ใช้ในการแสดงผลเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 แผนภาพยูสเคส (Use Case Diagram)



รูปที่ 3.30 แผนภาพยูสเคส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 ตารางอธิบายแผนภาพยูสเคสของผู้ใช้ระบบ (Use Case Narrative)

ตารางที่ 3.1 ตารางอธิบายแผนภาพยูสเคสของยูสเซอร์ที่เข้าใช้ระบบ

Use Case Name :	View map	
Trigger Event :	ผู้ใช้งานต้องเชื่อมต่ออินเทอร์เน็ต	
Description :	ผู้ใช้งานเข้าเว็บไซต์เพื่อดูตำแหน่งที่เกิดเหตุการณ์น้ำท่วม	
Actor :	User	
Related Use Case:	Visualize on a map	
Pre-condition :	เข้าเว็บไซต์	
Post-condition :	แสดงหน้าจอแผนที่พร้อมตำแหน่งที่เกิดน้ำท่วม	
Flow of Activity :	Actor	system
	1. กด/พิมพ์เข้าเว็บไซต์	2. ตรวจสอบตำแหน่งที่อยู่ของเว็บไซต์ถูกต้องหรือไม่
	3. ดูตำแหน่งที่เกิดเหตุการณ์น้ำท่วม	
Exception Condition :	หากผู้ใช้งานพิมพ์เว็บไซต์ผิด ระบบจะไม่สามารถแสดงแผนที่และตำแหน่งที่เกิดเหตุการณ์น้ำท่วมได้	

ตารางที่ 3.2 ตารางอธิบายแผนภาพยูสเคส Visualize on a map

Use Case Name :	Visualize on a map	
Trigger Event :	ต้องมีข้อมูลพิกัดที่ดึงจากทวีเตอร์และจับคู่กับดาด้าเบสแล้ว	
Description :	แสดงการ visualize ตำแหน่งที่เกิดเหตุการณ์น้ำท่วมบนแผนที่ประเทศไทย	
Actor :	Google place API	
Related Use Case:	Make String Matching, Check in Geocoder	
Pre-condition :	ระบบจะต้องทำการจับคู่ค่าที่ได้จากทวีเตอร์กับพิกัด Lat/Long เรียบร้อยแล้ว	
Post-condition :	ได้แผนที่ที่มีการแจ้งเตือนบริเวณที่เกิดเหตุการณ์น้ำท่วม	
Flow of Activity :	Actor	system
		1. ส่งค่าพิกัด Lat/long จากดาด้าเบสไปยัง Google place API
		2. นำพิกัดมาพล็อตในแผนที่ประเทศไทยบนเว็บไซต์
Exception Condition :	-	

ตารางที่ 3.3 ตารางอธิบายแผนภาพยูสเคสการทำ String Matching

Use Case Name :	Make String Matching	
Trigger Event :	ต้องมีค่าที่ตัดจากทวีเตอร์	
Description :	จับคู่ค่าที่ตัดแล้วกับดาต้าเบสเพื่อหาสถานที่ที่เกิดเหตุการณ์น้ำท่วม	
Actor :	-	
Related Use Case:	Make WordTokenization	
Pre-condition :	ต้องคัดคำที่ขึ้นต้นด้วย ซ., ถ., ถนน, ซอย ออกไปค้นหาใน Geocoder แล้วจึงจะสามารถทำ String Matching ได้	
Post-condition :	ได้สถานที่ที่เกิดเหตุการณ์น้ำท่วมจริงและได้พิกัดที่เกิดน้ำท่วม	
Flow of Activity :	Actor	system
		1. เช็คคำที่ขึ้นต้นด้วย แขวง, เขต, อำเภอ, ตำบล เพื่อตัดคำนำหน้าทิ้ง
		2. เช็คคำในดาต้าเบสแขวง
		3. เช็คคำในดาต้าเบสเขต
		4. เช็คคำในดาต้าเบสจังหวัด
Exception Condition :	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 ตารางอธิบายแผนภาพยูสเคสการทำ Geocoder

Use Case Name :	Check in Geocoder	
Trigger Event :	ต้องมีคำที่ตัดจากทวีตเตอร์	
Description :	จับคู่คำที่ตัดแล้วกับดาต้าเบสเพื่อหาสถานที่ที่เกิดเหตุการณ์น้ำท่วม	
Actor :	Google place API	
Related Use Case:	Make WordTokenization	
Pre-condition :	ต้องคัดคำที่ขึ้นต้นด้วย ซ., ถ., ถนน, ซอย ว่าขึ้นต้นด้วยคำพวกนี้หรือไม่	
Post-condition :	ได้สถานที่ที่เกิดเหตุการณ์น้ำท่วมจริงและได้พิกัดที่เกิดน้ำท่วม	
Flow of Activity :	Actor	system
		1. เช็คคำที่ขึ้นต้นด้วย ซ., ถ., ถนน, ซอย
		2. ส่งไปเช็คยัง Geocoder
Exception Condition :	-	

ตารางที่ 3.5 ตารางอธิบายแผนภาพยูสเคสการทำ WordTokenization

Use Case Name :	Make WordTokenization	
Trigger Event :	ต้องมีข้อมูลที่ดึงจากทวีตเตอร์	
Description :	ตัดคำที่ได้มาจากทวีตเตอร์ เพื่อส่งค่าไปยังขั้นตอนอื่น ๆ	
Actor :	-	
Related Use Case:	Get Tweets	
Pre-condition :	ดึงข้อมูลจากทวีตเตอร์มาเก็บไว้ในดาต้าเบส	
Post-condition :	ได้คำที่ตัดออกมาเพื่อส่งไปประมวลผลในขั้นตอนถัดไป	
Flow of Activity :	Actor	system
		1. ตัดคำออกจากประโยค
Exception Condition :	-	

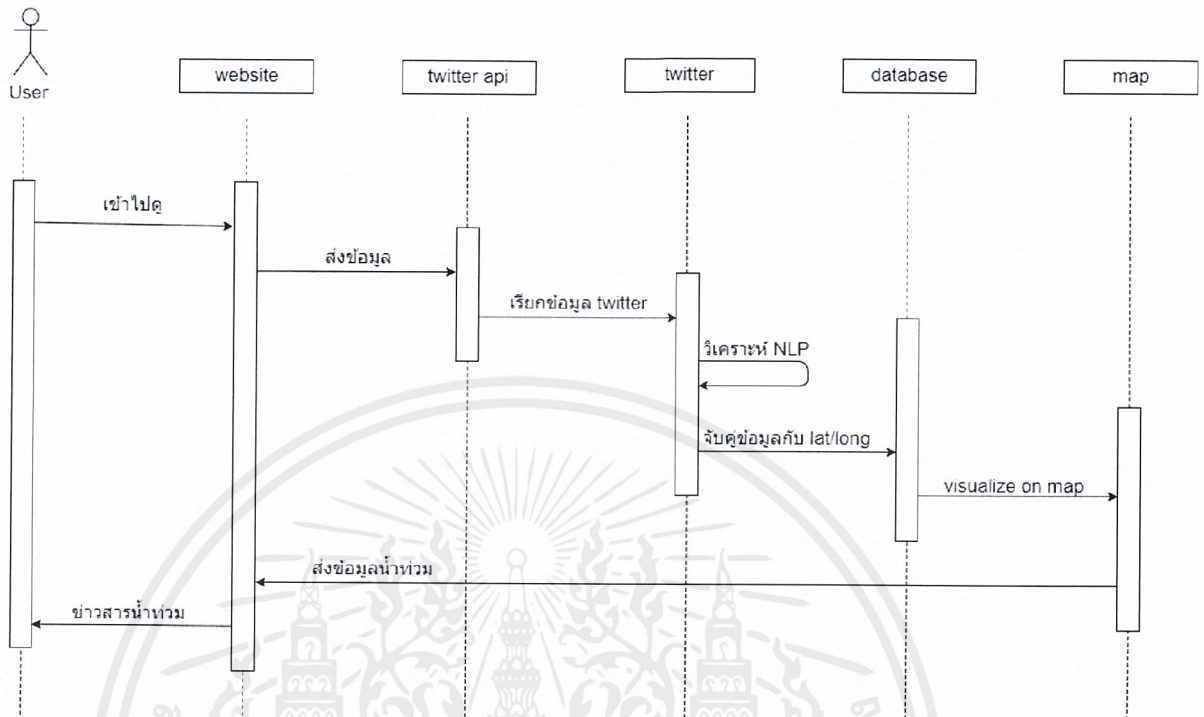
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 ตารางอธิบายแผนภาพยูสเคส Get tweets

Use Case Name :	Get tweets and Latitude / Longitude	
Trigger Event :	ต้องเป็นผู้พัฒนาแอปพลิเคชันทวิตเตอร์	
Brief Description :	ดึงข้อความจากทวิตเตอร์ออกมาเพื่อนำมาใช้ประมวลผลในขั้นตอนอื่น ๆ	
Actor :	Twitter API	
Related Use Case:	-	
Pre-condition :	ต้องสมัครเป็นผู้พัฒนาแอปพลิเคชันทวิตเตอร์ก่อน	
Post-condition :	สามารถดึงข้อมูลในทวิตเตอร์ออกมาได้	
Flow of Activity :	Actor	system
	1. TwitterAPI ให้ข้อมูลต่างๆที่เราต้องการจาก Twitter	2. ระบบเก็บข้อมูลที่ดึงออกมาได้ลงดาต้าเบสที่เป็น MongoDB
Exception Condition :	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

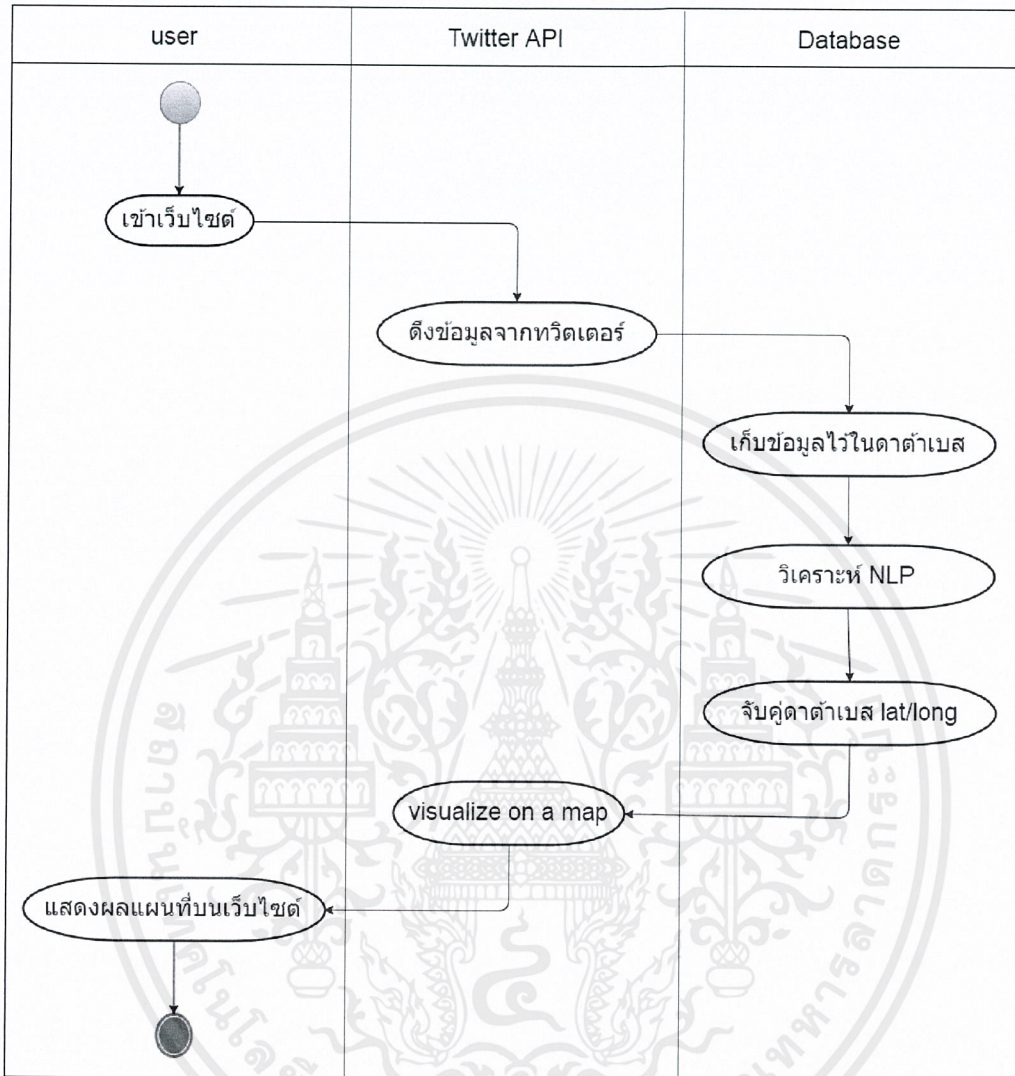
3.8 แผนภาพแสดงปฏิสัมพันธ์ (Sequence Diagram)



รูปที่ 3.31 แผนภาพแสดงปฏิสัมพันธ์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 แผนภาพกิจกรรม (Activity Diagram)



รูปที่ 3.32 แผนภาพกิจกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

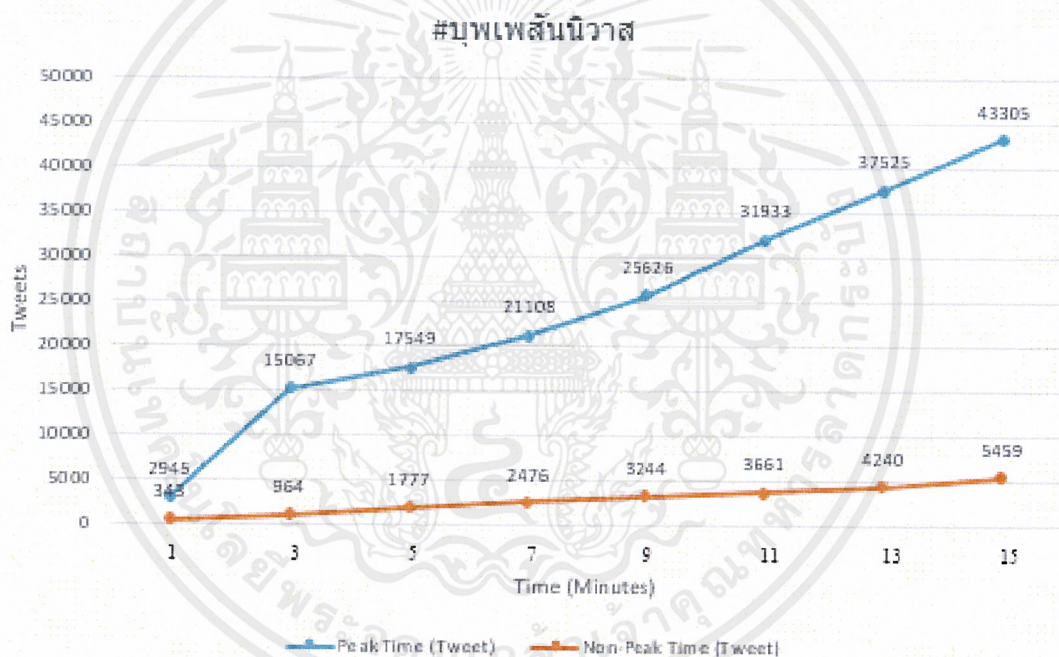
บทที่ 4

ผลการดำเนินงานและการอภิปรายผล

จากการทำวิจัยครั้งนี้ คณะผู้จัดทำได้ทำการทดลอง 3 ส่วนด้วยดังนี้

4.1 การดึงข้อความทวีตเตอร์ในช่วงเวลาที่เกิดและไม่เกิดเหตุการณ์และสถิติที่เกี่ยวข้อง

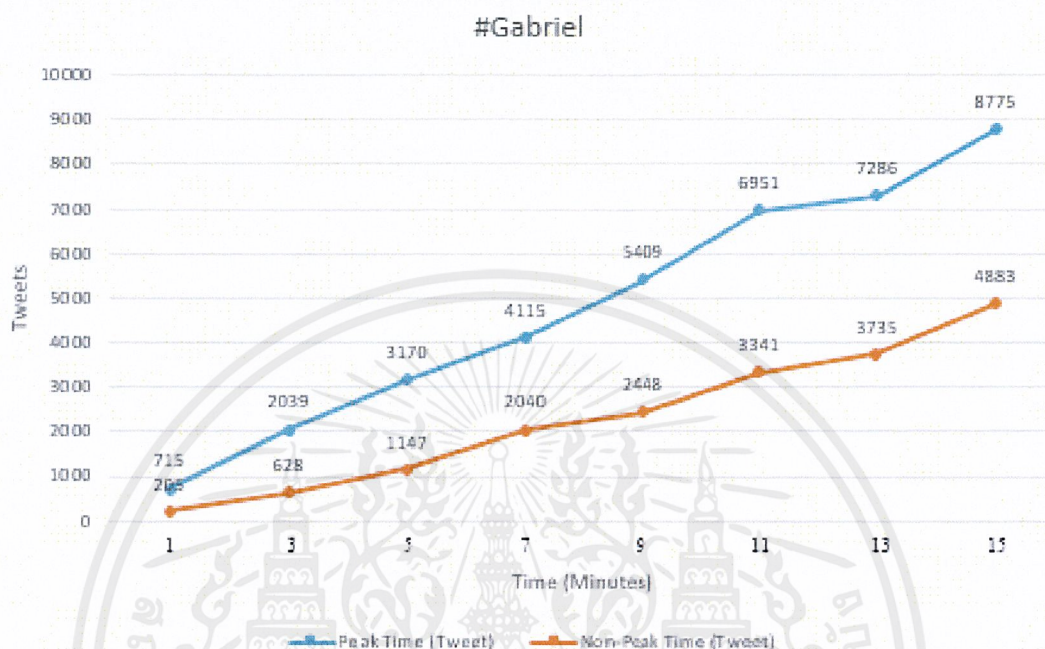
ในหัวข้อนี้จะแสดงจำนวนทวีตที่ผู้ใช้ทวีตในช่วงที่เกิดเหตุการณ์เทียบกับข้อความทวีตในช่วงที่ไม่ได้เกิดเหตุการณ์โดยหัวข้อนี้อาศัยแฮชแท็กบุพเพสันนิวาส (#บุพเพสันนิวาส) ซึ่งเป็น Top Trend in Thailand ณ วันที่ 8 มีนาคม 2018 ซึ่งเป็นช่วงเวลาที่ละครกำลังออกอากาศ เทียบกับ ช่วงเวลาที่ไม่ได้ออกอากาศ ณ วันที่ 12 มีนาคม 2018 ผลที่ได้แสดงดังกราฟด้านล่าง



รูปที่ 4.1 กราฟเปรียบเทียบพฤติกรรมกรทวีตข้อความของผู้ใช้ทวีตเตอร์ในช่วงที่เกิดเหตุการณ์ และช่วงที่ไม่ได้เกิดเหตุการณ์โดยใช้แฮชแท็กบุพเพสันนิวาส (#บุพเพสันนิวาส)

จากกราฟจะเห็นว่าเมื่ออยู่ในช่วงที่เกิดเหตุการณ์ (ช่วงที่ละครบุพเพสันนิวาสออนแอร์) ทวิตจะมีจำนวนเพิ่มมากขึ้นเรื่อยๆไม่จำกัด เทียบกับช่วงที่ไม่ได้เกิดเหตุการณ์ ซึ่งถือว่ามีผู้คนพูดถึงอยู่จำนวนหนึ่งเนื่องจากเป็นช่วงที่ละครเรื่องบุพเพสันนิวาสมีงานอีเว้นท์ทำให้ผู้คนกล่าวถึง #บุพเพสันนิวาส ในสังคมออนไลน์ แต่เมื่อเทียบกับช่วงที่เกิดเหตุการณ์ก็จะเห็นความแตกต่างของทวีตอย่างชัดเจน

และใช้แฮชแท็กGabriel (#Gabriel) ซึ่งเป็น Top Trend in Worldwide ณ วันที่ 11 มีนาคม 2018 ซึ่งเป็นช่วงเวลาเกิดเหตุการณ์เทียบ กับช่วงเวลาที่ไม่ได้เกิดเหตุการณ์ ณ วันที่ 12 มีนาคม 2018 ผลที่ได้แสดงดังกราฟด้านล่าง



รูปที่ 4.2 กราฟเปรียบเทียบพฤติกรรมทวิตข้อความของผู้ใช้ทวิตเตอร์ในช่วงที่เกิดเหตุการณ์ และช่วงที่ไม่ได้เกิดเหตุการณ์โดยใช้แฮชแท็กGabriel (#Gabriel)

จากกราฟจะเห็นว่าในช่วงเวลาที่เกิดเหตุการณ์การหายตัวไปของเด็กชาย Gabriel ซึ่งหายตัวนานถึง 1 ปี และถูกค้นพบว่าเสียชีวิตลงในรถยนต์ของเพื่อนสนิทของพ่อเป็นเรื่องที่ฮือฮาอย่างมากในสังคมออนไลน์หลังจากนั้น 1 วันจำนวนทวิตก็ลดลงตาลำดับเรื่อย ๆ จะเห็นได้ว่าพฤติกรรมทวิตข้อความของผู้ใช้ทวิตเตอร์ไม่ว่าจะเป็นในประเทศไทยหรือทั่วทั้งโลกจะเอนเอียงไปในทิศทางเดียวกันที่มียอดขึ้นสูงในช่วงแรกและลดลงตามระยะเวลาที่ผู้คนสนใจ

4.2 เวลาในการประมวลผลข้อมูลทวิตเตอร์

4.2.1 การประมวลผลเพื่อหาพิกัดสถานที่น้ำท่วมด้วยวิธีการ Levenshtein

เราได้นำชุดข้อมูลทำการทดสอบว่าทวิตทั้งหมดที่ดึงมาได้แต่ละครั้งจะใช้เวลาประมวลผลในการหาพิกัดที่เกิดเหตุการณ์น้ำท่วมจะใช้เวลาทั้งหมดแสดงดังตารางด้านล่าง

ตารางที่ 4.1 ตารางแสดงเวลาการประมวลผลพื้นที่ที่เกิดเหตุการณ์น้ำท่วมด้วยวิธีของ Levenshtein

คอลัมน์ที่ประมวลผล	จำนวน Records	จำนวนทวิต	เวลาที่ใช้ในการประมวลผลน้ำท่วม
Sub districts	7364 Records	732 ทวิต	30 นาที
Districts	928 Records	732 ทวิต	15 นาที
Provinces	77 Records	732 ทวิต	5 นาที

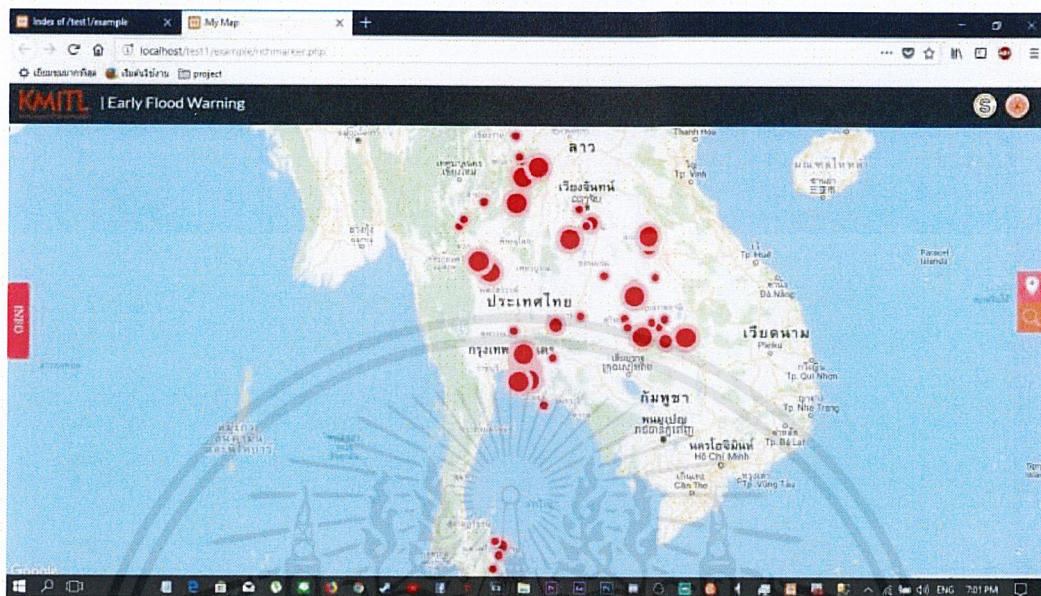
จากตาราง เนื่องจาก แขวง มีจำนวนมากที่สุด 7364 แขวงทำให้ระยะเวลาการ process ข้อมูลใช้เวลานานที่สุด ซึ่งจะเห็นได้ว่าการประมวลผลในระดับแขวงใช้เวลานานมากจึงเลือกที่จะใช้เทคนิค Levenshtein ในการประมวลแค่ระดับจังหวัดเท่านั้น อย่างไรก็ตาม หากมีฮาร์ดแวร์หรืออัลกอริทึม ที่รองรับการประมวลผลที่เร็วและมีประสิทธิภาพมากกว่าก็สามารถนำมาประยุกต์ใช้ได้เช่นกัน

4.2.2 เวลาในการประมวลผลโปรแกรมตั้งแต่เริ่มต้นจนแสดงสถานที่ที่เกิดเหตุการณ์น้ำท่วม

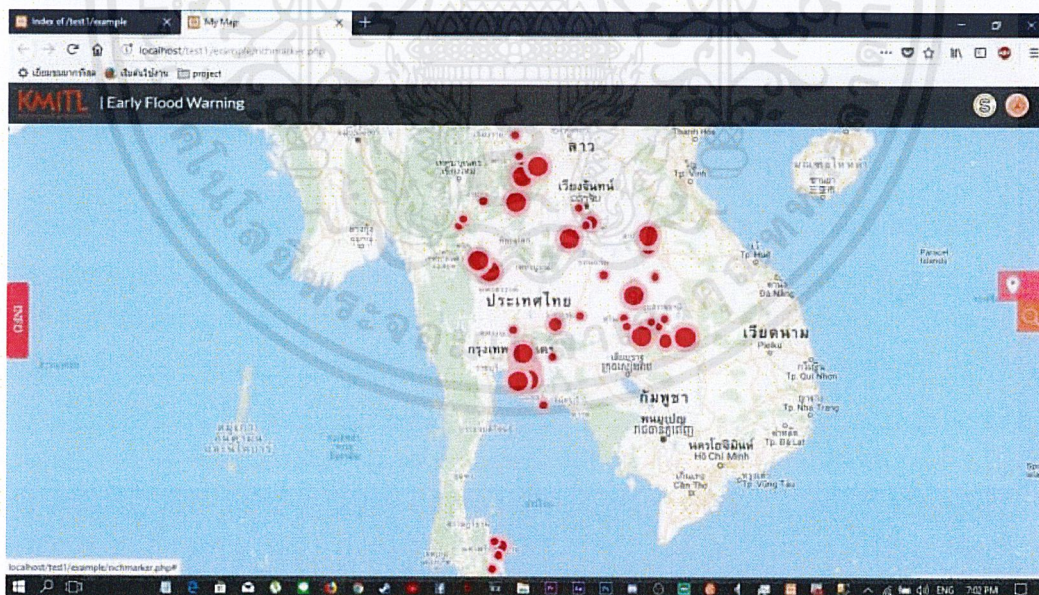
ตารางที่ 4.2 ตารางแสดงเวลาการประมวลผลโปรแกรมตั้งแต่เริ่มต้นจนแสดงสถานที่ที่เกิดเหตุการณ์น้ำท่วม

จำนวนทวิต	เวลาที่ใช้ในการประมวลผล
732 ทวิต	7 นาที
1386 ทวิต	10 นาที
22776 ทวิต	1 ชั่วโมง 5 นาที

4.3 หน้าจอการทำงานของเว็บไซต์

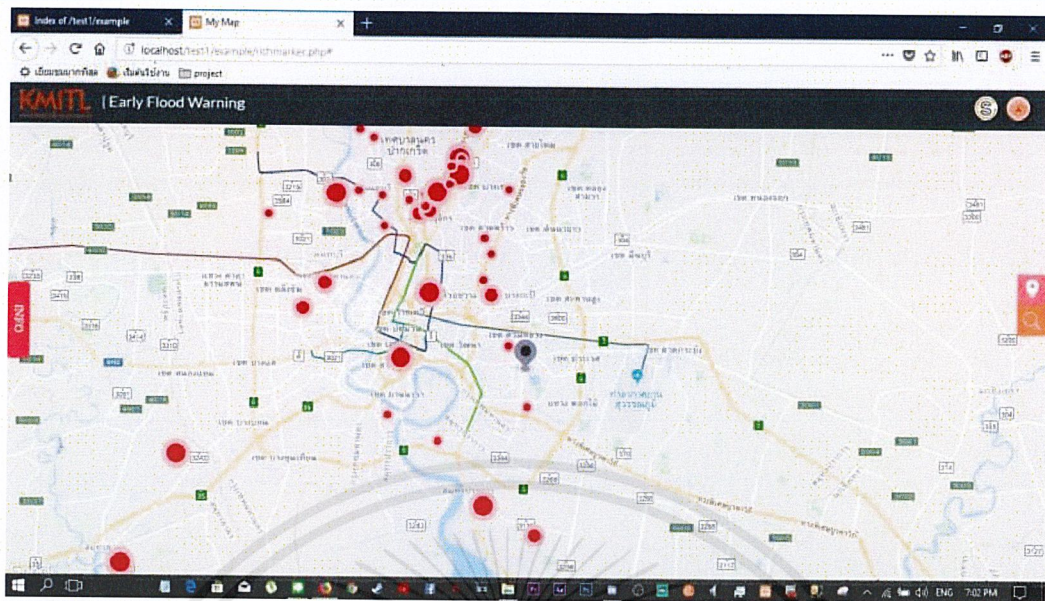


รูปที่ 4.3 รูปภาพหน้าจอของเว็บไซต์แจ้งเตือนเหตุการณ์น้ำท่วม

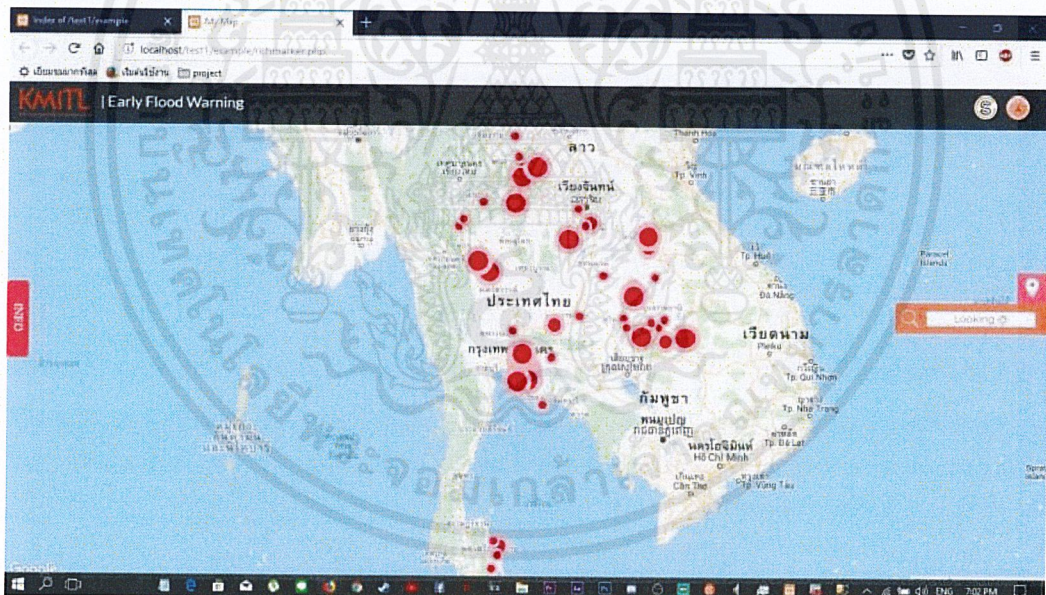


รูปที่ 4.4 หน้าจอแสดงฟังก์ชันระบุสถานที่ของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

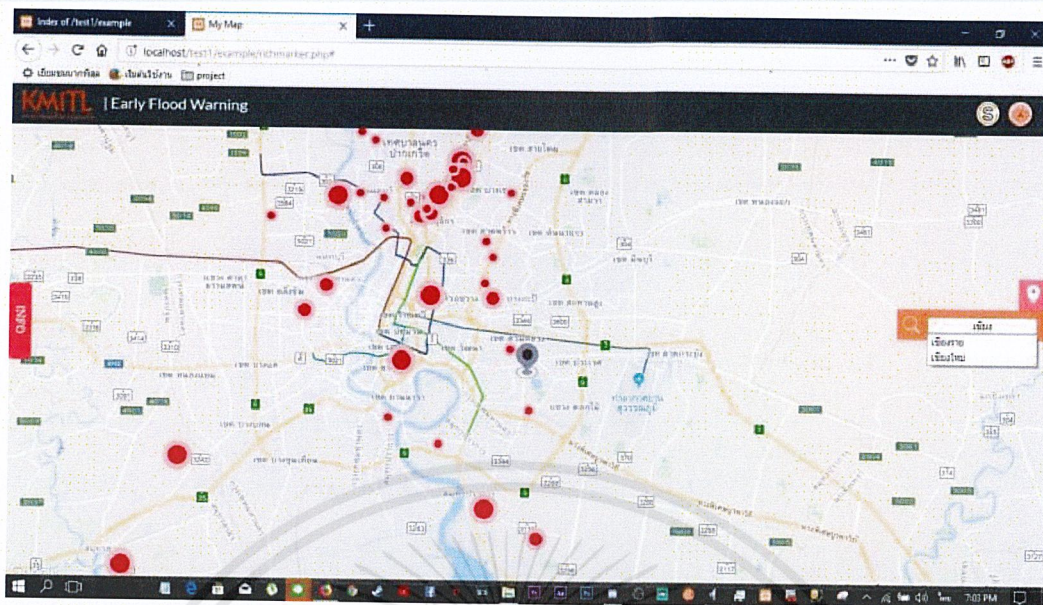


รูปที่ 4.5 หน้าจอแสดงการกดใช้ฟังก์ชันระบุสถานที่ที่ผู้ใช้งานใช้งานอยู่

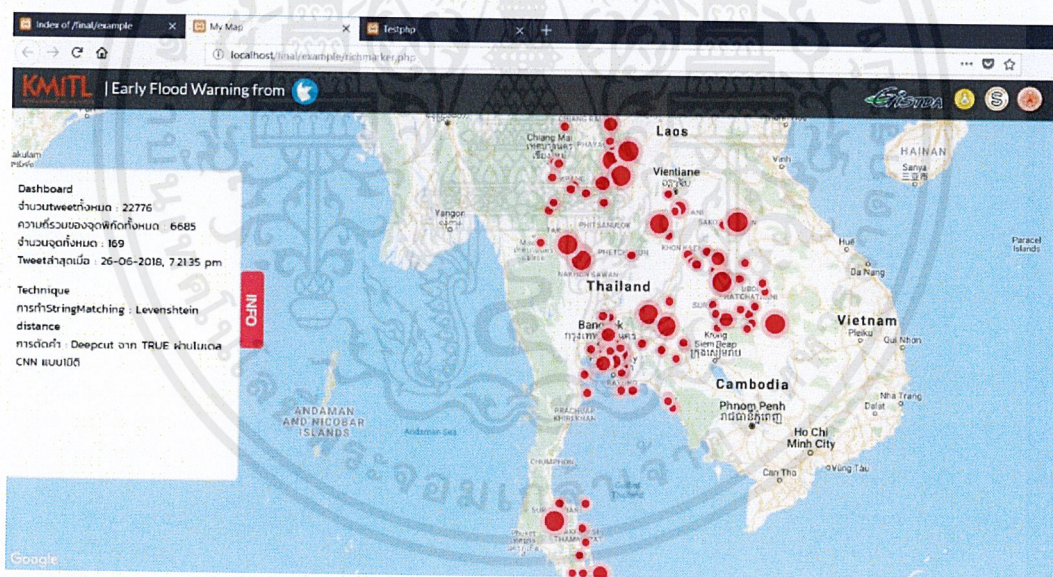


รูปที่ 4.6 หน้าจอแสดงฟังก์ชันค้นหาสถานที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 หน้าจอแสดงการกดใช้ฟังก์ชันค้นหาสถานที่



รูปที่ 4.8 หน้าจอแสดงแดชบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลปัญหาพิเศษและข้อเสนอแนะ

5.1 สรุปผลปัญหาพิเศษ

จากการดำเนินการพัฒนาระบบประมวลผลข้อมูลทวิตเตอร์เพื่อการเตือนภัยน้ำท่วม ผู้ใช้สามารถทราบเหตุการณ์และสถานที่เกิดเหตุการณ์น้ำท่วมได้อย่างรวดเร็วโดยไม่ต้องรอให้สำนักข่าวออกอากาศทางโทรทัศน์ ซึ่งใช้เวลาในการประมวลผล โดยเฉลี่ยประมาณ 2-3 ชั่วโมง นับจากเวลาเมื่อได้รับข้อมูลจริง ผลลัพธ์ที่ได้แสดงผลในรูปแบบของ map visualization แสดงตำแหน่งที่เกิดเหตุการณ์น้ำท่วมได้ทันท่วงที ทำให้ผู้ใช้งานสามารถหลีกเลี่ยงเส้นทางที่เกิดเหตุการณ์น้ำท่วม และช่วยในการวางแผนอื่น ๆ ได้

5.2 ข้อจำกัดของปัญหาพิเศษ

- 1) ประเทศไทยมีผู้ใช้ทวิตเตอร์น้อยเมื่อเทียบกับสื่อสังคมออนไลน์ชนิดอื่น (เช่น facebook, line, instagram)
- 2) ผู้ใช้ทวิตเตอร์ส่วนใหญ่ไม่เปิดใช้การระบุตำแหน่งทำให้ไม่สามารถระบุบริเวณที่เกิดน้ำท่วมโดยตรงได้
- 3) ในช่วงที่ทำการทดลองไม่ได้เกิดปัญหาน้ำท่วมขึ้นบ่อยครั้งในประเทศไทย ทำให้ข้อมูลที่นำมาทำการทดลองอาจยังไม่มากพอ
- 4) ภาษาไทยมีโครงสร้างและหลักการการใช้ภาษาที่ซับซ้อนทำให้ยากต่อการประมวลผล

5.3 ข้อเสนอแนะในการพัฒนาปัญหาพิเศษ

- 1) การประมวลผลข้อมูลทวิตเตอร์เพื่อการเตือนภัยน้ำท่วมมีข้อเสนอแนะในการปรับปรุงเพิ่มเติมคือ การพัฒนาการตัดคำให้มีความน่าเชื่อถือมากยิ่งขึ้น
- 2) การพัฒนาโปรแกรมให้มาสารถใช้บนมือถือได้เพื่อเพิ่มความสะดวกสบายให้แก่ผู้ใช้งาน
- 3) อาจมีการนำข้อมูลจากแหล่งอ้างอิงอื่น ๆ เช่น กรมอุตุนิยมวิทยา เพื่อใช้ในการประเมินและ/เพิ่ม ประสิทธิภาพ
- 4) สามารถเพิ่มคำที่เป็นคีย์เวิร์ดอื่น ๆ ได้ ที่เกี่ยวข้องกับน้ำท่วม
- 5) ให้ Input Specification เป็นข้อมูลสถานะจากผู้ใช้งานทวิตเตอร์ และ Output Specification เป็นข้อมูลที่เป็นข่าวสารบอกสถานการณ์น้ำท่วมและพื้นที่ที่น้ำท่วม
- 6) อัลกอริทึมของ Levenshtein ที่เราใช้อยู่ match แค่งจังหวัด เนื่องจากใช้เวลานาน ถ้าในอนาคตใช้เครื่อง server หรืออัลกอริทึม ที่ดีกว่า อาจจะ match กับแขวงกับเขตได้เร็วกว่าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Bernd Klein. 2554. **Levenshtein Distance**. [Online]. Available : https://www.python-course.eu/levenshtein_distance.php
- [2] Gennon. 2560. **Connecting to Microsoft SQL server using Python**. [Online]. Available : <https://stackoverflow.com/questions/33725862/connecting-to-microsoft-sql-server-using-python>
- [3] Dwight Merriman, Eliot Horowitz and Kevin Ryan – the team behind DoubleClick. 2550. **Connect to MongoDB via the Python Driver**. [Online]. Available : <https://docs.mongodb.com/tutorials/connect-to-mongodb-python/>
- [4] obskyr. 2558. **Tweepy Documentation**. [Online]. Available : <http://docs.tweepy.org/en/v3.5.0/>
- [5] Susan Ibach. 2558. **Python and Data : SQL Server as a data source for Python applications**. [Online]. Available : <https://blogs.msdn.microsoft.com/cdn devs/2015/03/11/python-and-data-sql-server-as-a-data-source-for-python-applications/>
- [6] Sam Delgado. 2557. **twitter-to-mongo**. [Online]. Available : <https://github.com/SamDelgado/twitter-to-mongo>
- [7] Patana Patanasongsivilai. 2560. **ตัดคำภาษาไทยโดยใช้ AI ของบริษัท True**. [Online]. Available : <http://www.patanasongsivilai.com/blog/tudkumthai/>
- [8] Titipat Achakulvisut. 2560. **Deepcut**. [Online]. Available : <https://github.com/rkcosmos/deepcut>
- [9] codesanook.com. 2560. **ฐานข้อมูล จังหวัด อำเภอ ตำบล ละติจูด ลองจิจูด รหัสไปรษณีย์ สำหรับ MySQL และ SQL server**. [Online]. Available : <https://www.facebook.com/codesanookpage/posts/1640641812624476>

เอกสารอ้างอิง (ต่อ)

- [10] Aaron Amm. 2560. Administrative division of Thailand in SQL file format. [Online]. Available : <https://github.com/aaronamm/thai-administrative-division-province-district-subdistrict-sql>
- [11] แอดมินโฮ โอน้อยออก. 2560. ตัดคำภาษาไทยโดยใช้ AI ของบริษัท True. [Online]. Available : <http://www.patanasongsivilai.com/blog/tudkumthai/>





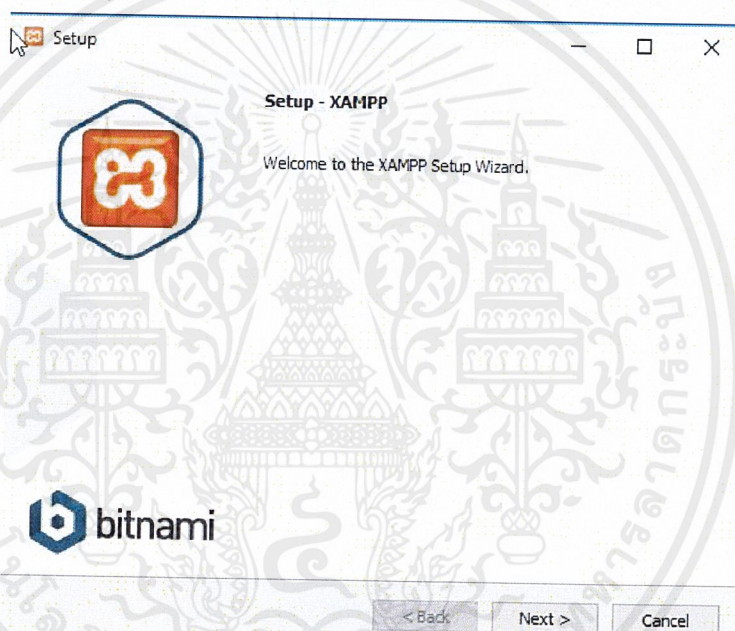
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

วิธีการติดตั้ง XAMPP จำลอง WebServer

ก.1 การติดตั้ง XAMPP

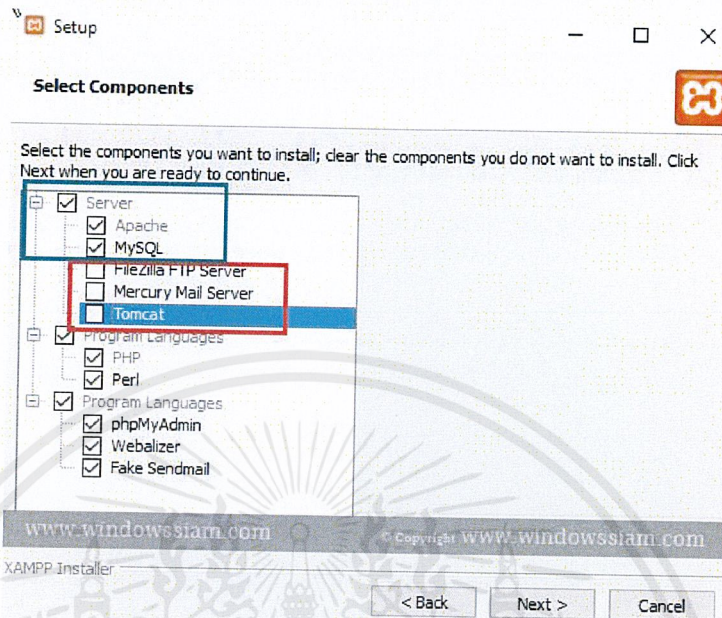
1. ทำการดาวน์โหลด XAMPP สำหรับ Windows
2. ทำการติดตั้ง คลิก Next



รูปที่ ก.1 หน้าต่างการ Setup XAMPP

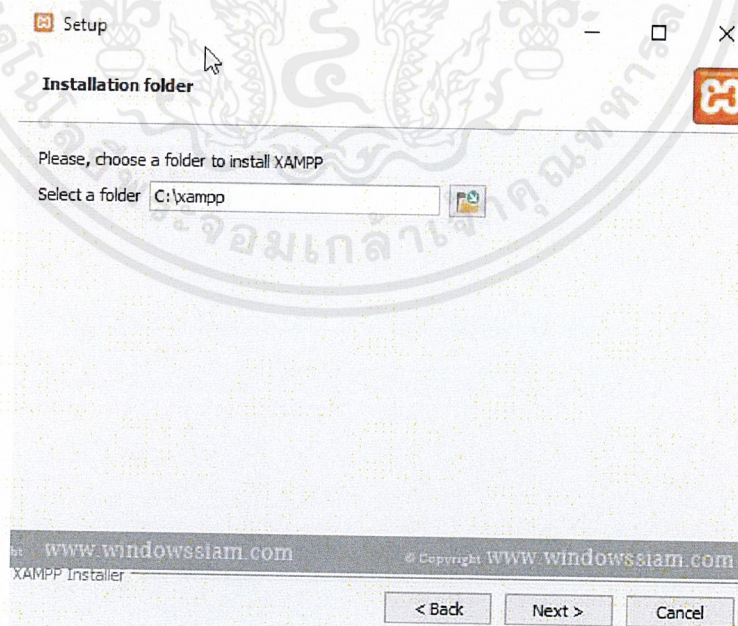
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการเลือก Components ที่เราต้องการ



รูปที่ ก.2 หน้าต่างการเลือก Components ที่ต้องการ

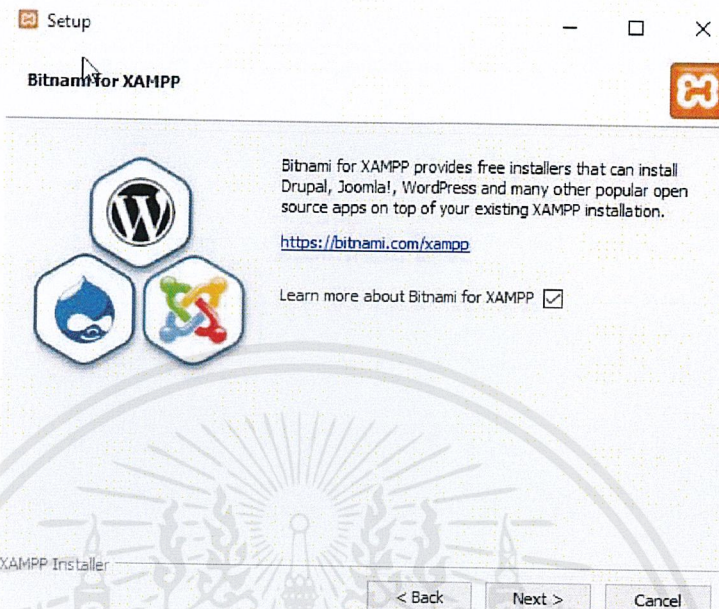
4. เลือก Path ในการติดตั้ง XAMPP



รูปที่ ก.3 หน้าต่างการเลือก Path

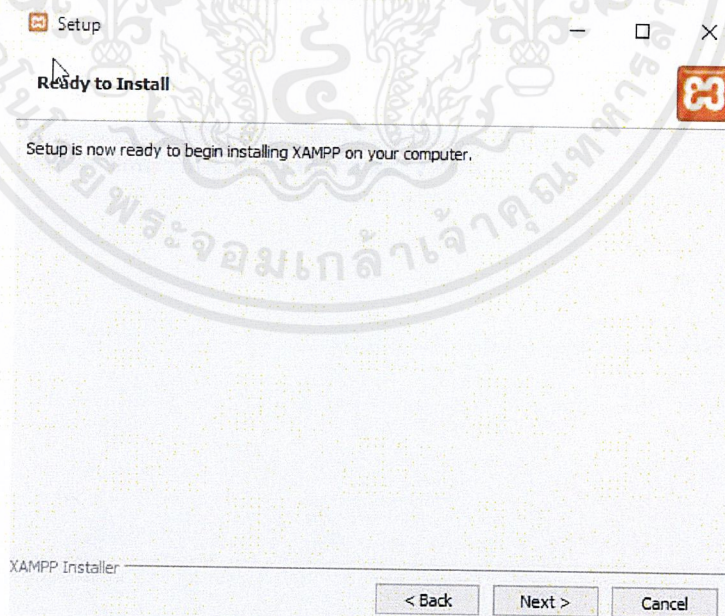
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คลิก Next



รูปที่ ก.4 หน้าต่างการ Setup XAMMP

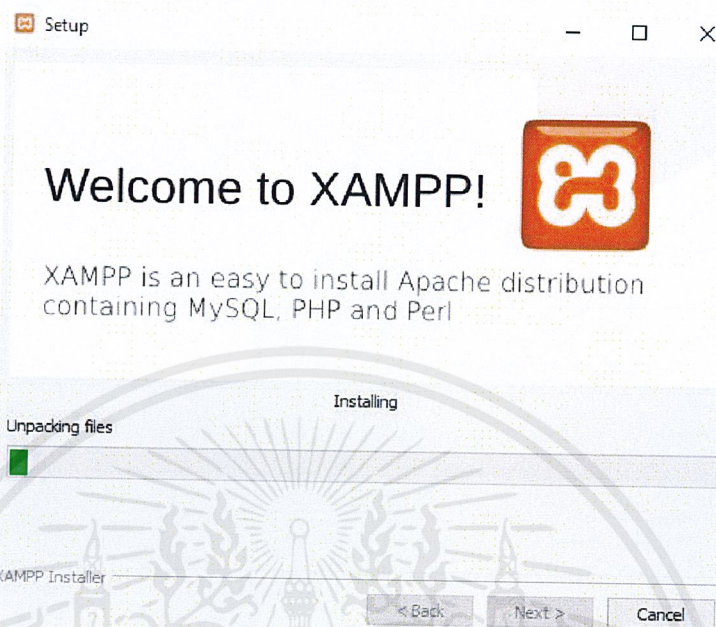
6. คลิก Next



รูปที่ ก.5 หน้าต่างการ Setup XAMMP

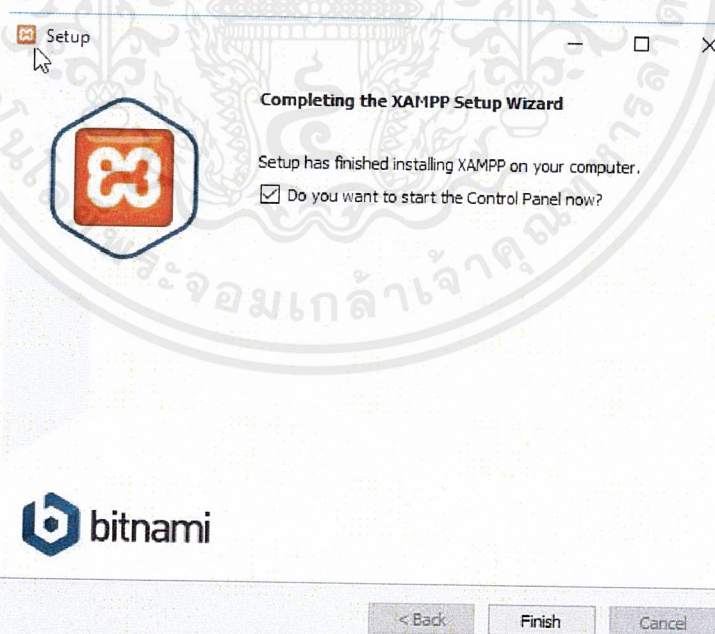
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. รอทำการติดตั้ง



รูปที่ ก.6 หน้าต่างการ Installing XAMMP

8. ทำการติดตั้งเรียบร้อย และทำการ Restart Computer 1 ครั้ง

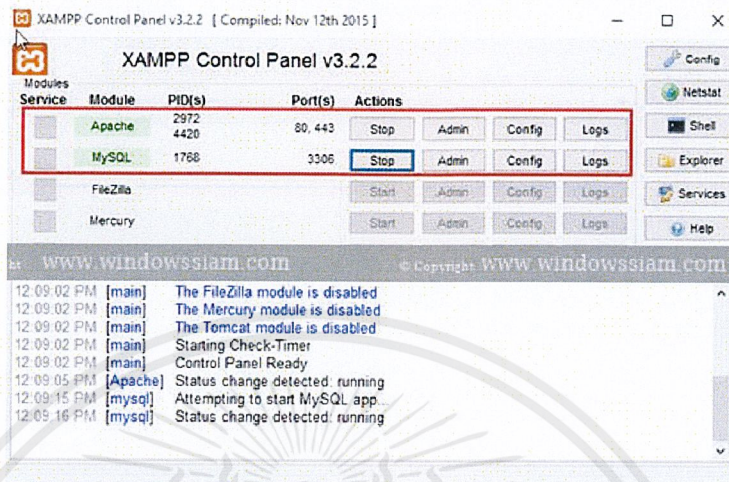


รูปที่ ก.7 หน้าต่างการ Installing XAMMP เสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

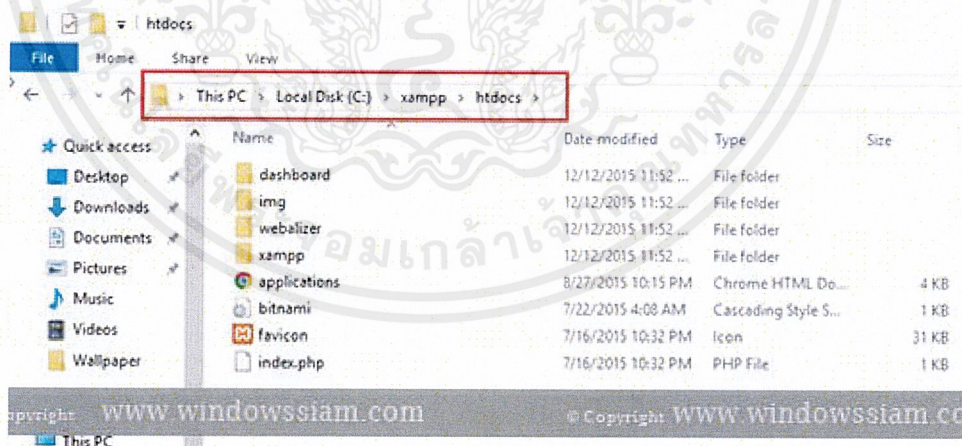
ก.2 จำลอง WebServer

1. เข้า Control Panel ของ XAMPP และทำการคลิก Start Apache และ Mysql ตามลำดับ



รูปที่ ก.8 หน้าต่างการจำลอง WebServer

2. จากนั้นเข้า Phpmyadmin และ localhost สามารถดูได้ที่ <http://localhost/phpmyadmin> และ <http://localhost>
3. สำหรับโค้ดต่างๆ ของ Php ในการทำ WebSite ให้เอามาไว้ที่ C:\xampp\htdocs\ และหลังจากที่เราลง xampp เราสามารถสร้าง Folders และไฟล์ต่างๆได้ใน htdocs



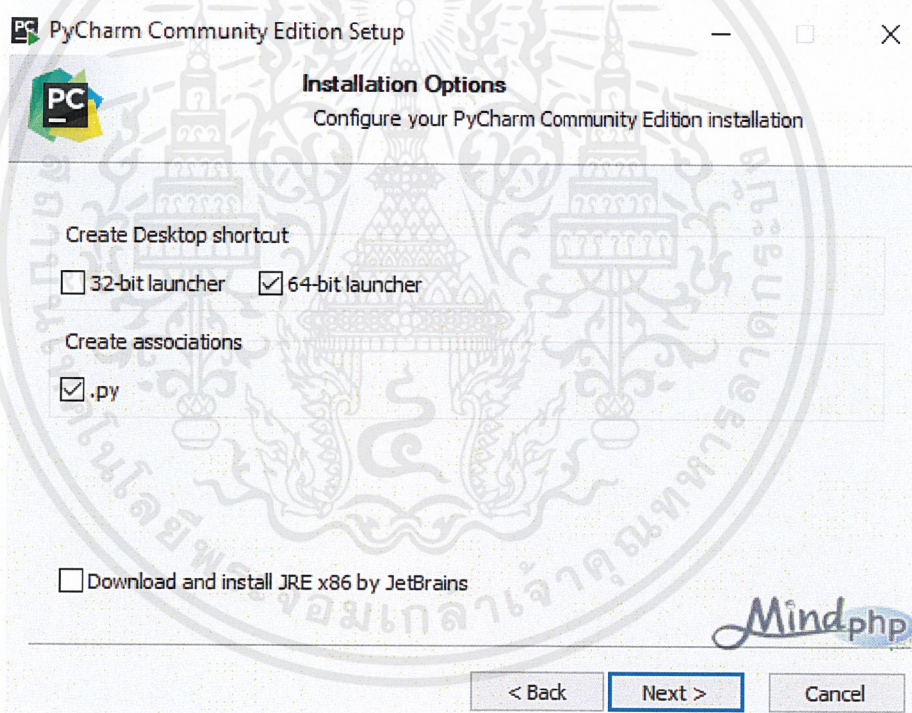
รูปที่ ก.9 Path ที่ใช้สำหรับเก็บโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ขั้นตอนการติดตั้งโปรแกรม ไพชาร์ม (Pycharm)

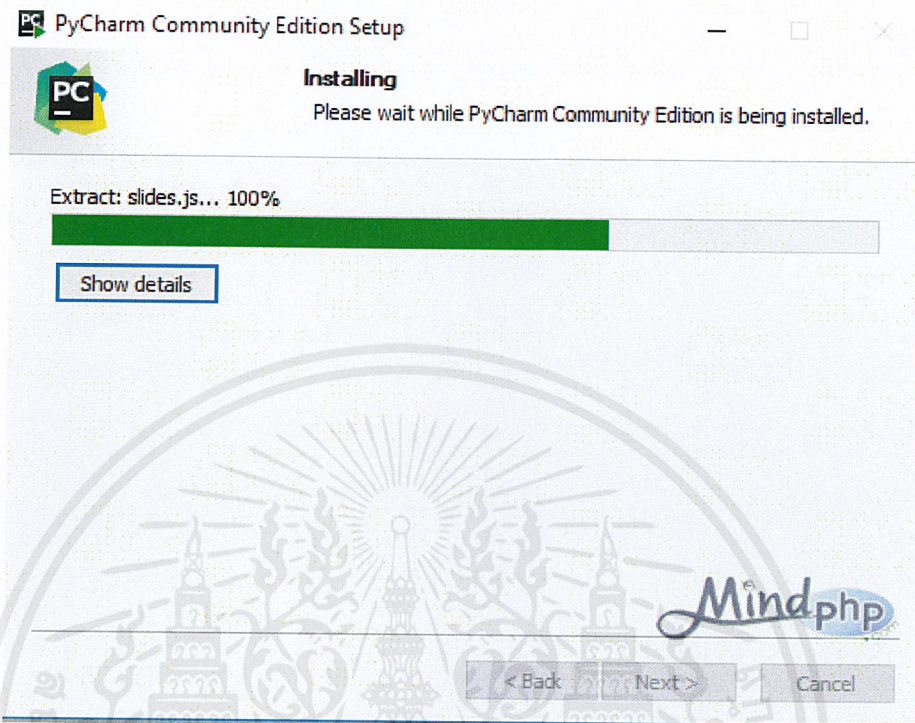
1. ดาวน์โหลดโปรแกรม Pycharm ได้ที่ <https://www.jetbrains.com/pycharm/download> โดยเลือกหมวดที่จะดาวน์โหลดเป็นหมวด Community
2. เมื่อทำการดาวน์โหลดโปรแกรมเสร็จแล้ว จะมีหน้าต่าง "Pycharm Community Edition Setup" ขึ้นมา ให้กด "Next" ไปตามขั้นตอนการติดตั้ง
3. เมื่อมาถึงหน้าต่าง "Installation Options" เราสามารถกำหนดค่าเองได้ว่าจะให้มี Shortcut บนหน้าจอเดสก์ท็อปหรือไม่ หลังจากนั้น ก็เป็นขั้นตอนการติดตั้ง Pycharm



รูปที่ ข.1 หน้าต่างการ Setup Pycharm

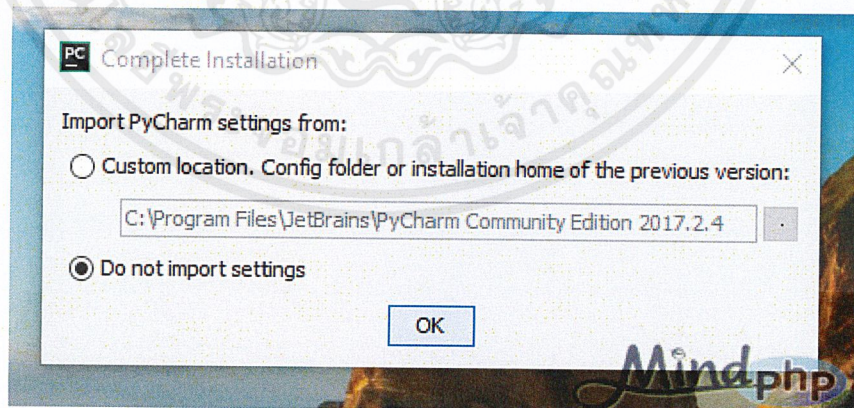
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หลังจากนั้น ก็เป็นขั้นตอนการติดตั้ง Pycharm



รูปที่ ข.2 หน้าต่างขณะกำลังติดตั้งโปรแกรม Pycharm

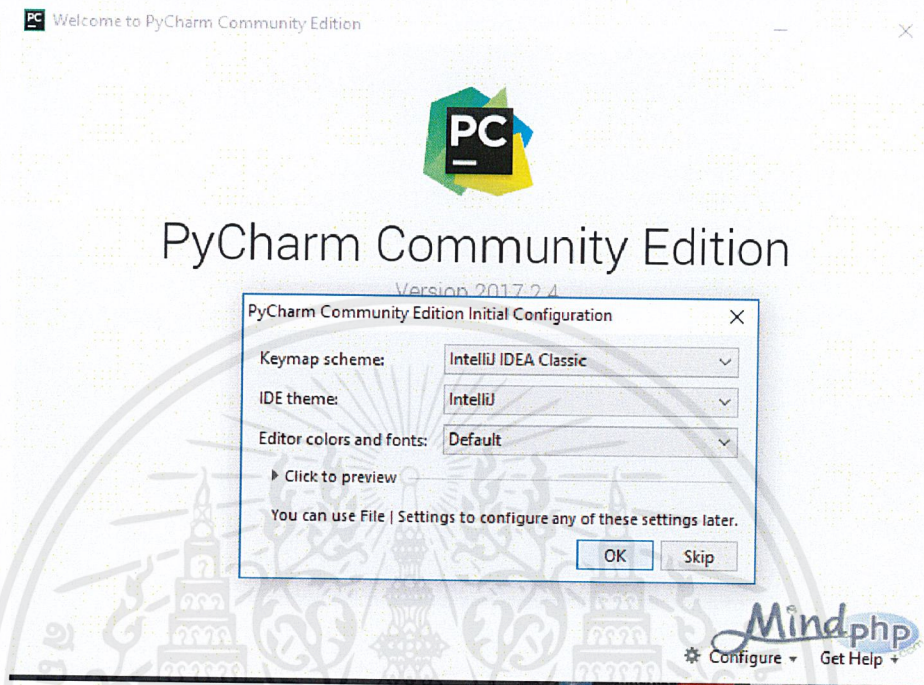
5. เมื่อโปรแกรมติดตั้งเสร็จเรียบร้อย ให้เราทดลองกดใช้งาน หากเป็นการใช้งาน Pycharm ครั้งแรก จะขึ้นหน้าต่างดังภาพ ให้เรากด "OK"



รูปที่ ข.3 หน้าต่างสำหรับการใช้งาน Pycharm ครั้งแรก

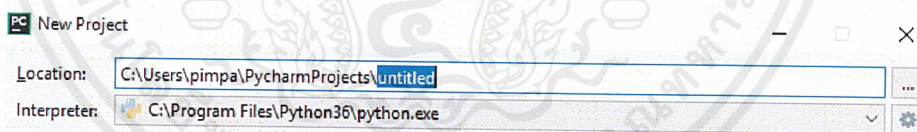
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หลังจากนั้นก็จะมีหน้าต่างสำหรับการตั้งค่าเริ่มต้น ให้เราตั้งค่าตามที่เราต้องการ แล้วกด "OK"



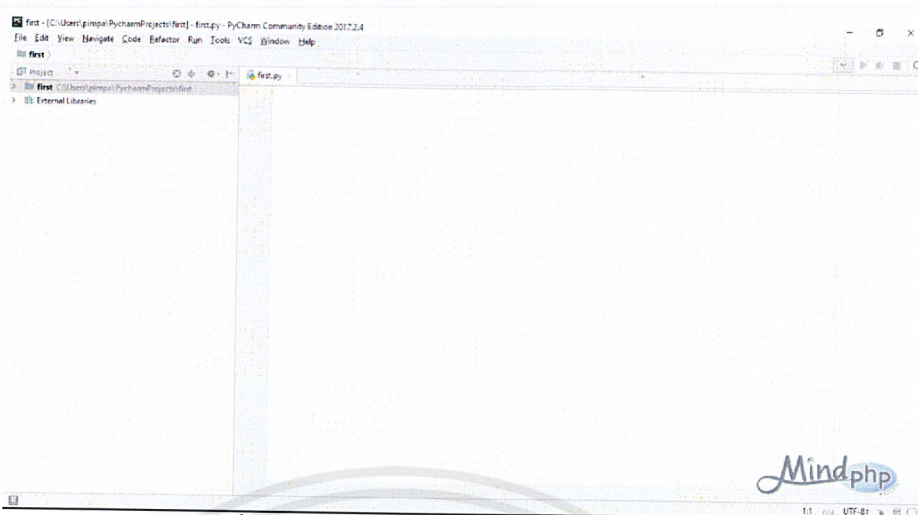
รูปที่ ข.4 หน้าต่างสำหรับการตั้งค่าเริ่มต้น

7. จากนั้น ก็จะเข้าสู่หน้าสำหรับการตั้งชื่อไฟล์ของเรา

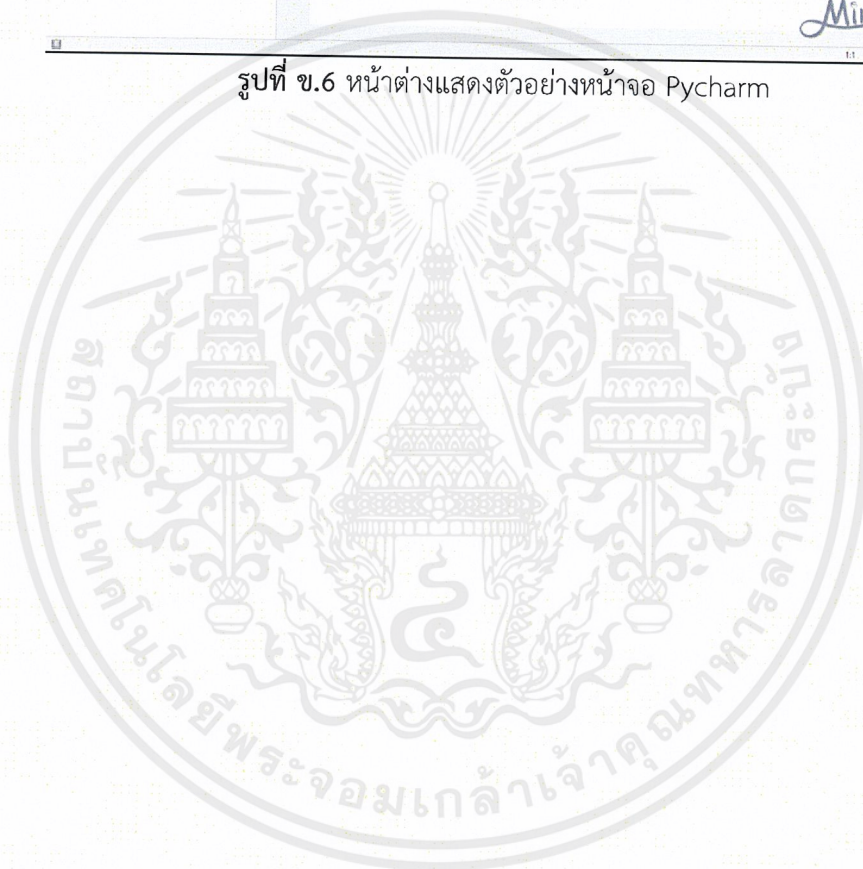


รูปที่ ข.5 หน้าต่างสำหรับการตั้งชื่อไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.6 หน้าต่างแสดงตัวอย่างหน้าจอ Pycharm



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้