การทำ map matching บนระบบฐานข้อมูลขนาดใหญ่

LARGE SCALE MAP MATCHING USING DISTRIBUTED DATABASE SYSTEM

สุภทรา อยู่คง

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การทำ map matching บนระบบฐานข้อมูลขนาดใหญ่

LARGE SCALE MAP MATCHING USING DISTRIBUTED DATABASE SYSTEM

T149191

สุภทรา อยู่คง

ปพ.
ส833ก
2558
เลขหมู่
เลขทะเบียน 149191
วัน.เดือน.ปี 29 มี.ค. 2561
b. 12881521
l.

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LARGE SCALE MAP MATCHING USING DISTRIBUTED DATABASE SYSTEM

SUPATARA YOOKONG

COOPERATIVE EDUCATION SUBMITTED IN
PARTIAL FULFILLMENTOF THE REQUIREMENTS 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 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงานพิเศษ	การทำ map matching บนระบบฐานข้อมูลขนาดใหญ่ Large scale map matching using distributed database system
ชื่อนักศึกษา	นางสาว สุภทรา อยู่คง รหัสนักศึกษา 55050510
ปริญญา	วิทยาศาสตร์บัณฑิต
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผศ.กฤษฎา บุศรา
อาจารย์ที่ปรึกษาร่วม	ดร.ไพรัตน์ ธรเจริญศรี

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.กฤษฎา บุศรา ประธานกรรมการและอาจารย์ที่ปรึกษา	
ดร.ไพรัตน์ ธรเจริญศรี กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงานพิเศษ	การทำ map matching บนระบบฐานข้อมูลขนาดใหญ่
ชื่อนักศึกษา	นางสาวสุภทรา อยู่คง รหัสนักศึกษา 55050510
ปริญญา	วิทยาศาสตร์บัณฑิต
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผศ.กฤษฎา บุศรา
อาจารย์ที่ปรึกษาร่วม	ดร.ไพรัตน์ ธรเจริญศรี

บทคัดย่อ

ในปัจจุบันปริมาณจำนวนข้อมูลที่จัดเก็บในฐานข้อมูลมีอัตราการเติบโตที่สูง เมื่อข้อมูลที่จัดเก็บมีจำนวนมากในการจัดเก็บข้อมูลลงฐานข้อมูลแบบเชิงสัมพันธ์นั้นจะทำให้การจัดการกับข้อมูลไม่มีประสิทธิภาพ เท่าที่ควรเนื่องจากรูปแบบการจัดเก็บข้อมูลมีลักษณะโครงสร้างที่ซับซ้อนทำให้ต้องใช้เวลานานในการจัดการ เข้าถึงข้อมูลที่มีปริมาณมาก จึงมีการนำเทคโนโลยีฐานข้อมูลโนเอสคิวแอลเข้ามาใช้ในการจัดเก็บและจัดการข้อมูลที่มีจำนวนปริมาณมากในฐานข้อมูลโนเอสคิวแอลนั้นจะเป็นการเก็บข้อมูลในลักษณะ ไม่มีความสัมพันธ์เชิงโครงสร้าง ซึ่งฐานข้อมูลโนเอสคิวแอลไม่รองรับการเชื่อมความสัมพันธ์ของข้อมูล จึงได้นำมองโกดีบีมาใช้ในการจัดเก็บข้อมูลของจุดจีพีเอส และนำข้อมูลนี้มาแมพแมทเพื่อหาลิ้งไอดี

คำสำคัญ : มองโกดีบี แมพแมท ลิ้งไอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Large Scale Map Matching Using Distributed Database System
Students	Mrs.Supatara Yookong Student ID 55050510
Degree	Bachelor of Science
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2015
Advisor	Asst. Prof. Kridsada Budsara
Co-advisor	Dr. Pirat Thorncharoensri

Abstract

At present, the amount of data stored in databases is growing at a high rate. When a lot of data stored in the data storage into relational databases to manipulate data to make it ineffective. Because the format of data storage structures are complex, they require a long time to deal with large volumes of data. It is widely NOSQL database to use to store and manage large volumes of data with it. To store data in NOSQL databases, will store data in a manner no structural relationships. Therefore, Mongodb used to store data of the GPS. Its mapmatch find link_id.

Keywords: Mongodb, mapmatch, link_id

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่ข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษา ณ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.) ห้องปฏิบัติการวิจัยระบบขนส่งและการจราจร (ITS) ตั้งแต่วันที่ 15 มิถุนายน 2558 ถึงวันที่ 27 พฤศจิกายน 2558 ส่งผลให้ข้าพเจ้าได้รับความรู้ แนวทางการศึกษาค้นคว้านอกเหนือจากการศึกษาทฤษฎีภายในสถาบันการศึกษา การนำความรู้จากการศึกษาภายในสถาบันมาประยุกต์ใช้กับทำงานในชีวิตจริงเพิ่มมากขึ้น ซึ่งเป็นสิ่งที่มีคุณค่าอย่างมาก การปฏิบัติงานสหกิจศึกษาครั้งนี้ สำเร็จลุล่วงไปด้วยดี จากความช่วยเหลือและความสนับสนุนจากหลายฝ่าย ดังนี้

1) ดร.วสันต์ ภัทรอธิคม (นักวิจัย)
2) ดร.มนศักดิ์ โซ่เจริญธรรม (ผู้ช่วยวิจัย)
3) นาย ณพงศ์ วาณิชยพงศ์ (ผู้ช่วยวิจัย)
4) นาย วัชร ฟุ้งวัชรากร (ผู้ช่วยวิจัย)
5) นาย ณรงค์ อินทร์ธิรักษ์ (ผู้ช่วยวิจัย)
6) นาง สุภรณ์ เหมือนหนู (ผู้ช่วยวิจัย)

ผู้มีส่วนร่วมในการให้ข้อมูล คำแนะนำ ช่วยเหลือทั้งในเรื่องของการทำงานและเรื่องของการใช้ชีวิต ที่เป็นประโยชน์ในการปฏิบัติสหกิจศึกษาในครั้งนี้ ข้าพเจ้าขอขอบพระคุณไว้ ณ ที่นี้ด้วย

ขอขอบพระคุณ ผศ.กฤษฎา บุศรา อาจารย์ที่ปรึกษาในรายวิชา Cooperative Education ที่กรุณาให้คำปรึกษา คำแนะนำการเตรียมตัวสู่การฝึกงาน ประสบการณ์ทำงาน และแนวทางการแก้ปัญหาในชีวิตจริงที่อบรมและให้ความรู้ต่างๆ

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดา และสมาชิกครอบครัวทุกท่านที่ช่วยให้การสนับสนุนและให้กำลังใจ รวมถึงเพื่อนๆ ทุกคนที่คอยให้คำปรึกษาในเรื่องต่างๆ เป็นอย่างดีเสมอมา

สุภทรา อยู่คง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญรูป ..... ฉ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขต 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 3
2.1 การแสดงผล (Display) 3
2.1.1 Google Maps API 3
2.2 การประมวลผล (Processing) 4
2.2.1 Quantum GIS (QGIS) 4
2.2.2 Putty 4
2.2.3 WinSCP 5
2.2.4 FileZilla 5
2.3.1 PHP 6
2.3.2 SQL 6
2.3 ฐานข้อมูล (Database) 7
2.3.1 PostgreSQL 7
2.3.2 Mongodb 7
บทที่ 3 วิธีการดำเนินงาน 9
3.1 การ mapmatch โดยเรียกฟังก์ชันจากฐานข้อมูล PostgreSQL 10
3.1.1 ติดตั้ง PostgreSQL และ Postgis 10
3.1.2 ศึกษา Mongodb 11
3.1.3 Insert ข้อมูล และ update link_id 12
3.1.4 โปรแกรม mapmatch 14
3.1.5 โปรแกรมหาจำนวนของแต่ละ vid 15
3.1.6 โปรแกรมเรียกรัน mapmatch แบบ parallel 16
3.2 การ mapmatch โดยสร้างฟังก์ชันขึ้นมาใหม่ใน Mongodb 20
3.2.1 Insert ข้อมูล PostgreSQL มาไว้ในฐานข้อมูล Mongodb 20
3.2.2 โปรแกรมหา link_id จากตาราง link 23
บทที่ 4 ผลสหกิจศึกษาและการอภิปรายผล 25

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 5 สรุปผลสหกิจศึกษาและข้อเสนอแนะ 26

5.1 สรุปผลสหกิจศึกษา 26

5.2 ข้อเสนอแนะ 26

เอกสารอ้างอิง 27

ภาคผนวก 28

ภาคผนวก ก. การหา link_id เพื่อไป update ในฐานข้อมูล 29

ภาคผนวก ข. การเรียกรันโปรแกรม mapmatch 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 Google Maps API	3
2.2 โปรแกรม Quantum GIS	4
2.3 โปรแกรม Putty	5
2.4 โปรแกรม WinSCP	5
2.5 โปรแกรม FileZilla	6
2.6 โครงสร้างของ PostgreSQL	7
2.7 โครงสร้างของ Mongodb	8
3.1 การ mapmatch	9
3.2 Show database	12
3.3 Show collection	12
3.4 Create collection mapmatch	12
3.5 Show collection mapmatch	12
3.6 Insert ข้อมูล	13
3.7 ข้อมูลที่ Insert แล้ว	13
3.8 Update link_id	13
3.9 link_id ที่เพิ่มเข้ามา	13
3.10 Flowchart mapmatch	14
3.11 Flowchart count	15
3.12 ภาพรวมการ mapmatch แบบ parallel	19
3.13 Flowchart insert ข้อมูลจากตาราง link	22
3.14 Flowchart หา link_id	24
4.1 กราฟเปรียบเทียบเวลาในการ mapmatch	25
ก.1 พื้นที่โดยรอบรัศมี 30 เมตร ของ "latitude": 13.8623, "longitude": 100.5776	29
ก.2 เส้นทางภายในพื้นที่โดยรอบจุด	30
ก.3 Direction ของแต่ละเส้นทางที่พบ	31
ก.4 ข้อมูลที่ Update link_id แล้ว	31
ข.1 เชื่อมต่อ server	32
ข.2 เรียกรันโปรแกรม maplink	33
ข.3 โปรแกรม maplink เริ่มรัน	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.) เป็นหน่วยงานในกำกับของกระทรวงวิทยาศาสตร์และเทคโนโลยี จัดตั้งขึ้นเมื่อปี พ.ศ. 2534 ตาม พ.ร.บ.พัฒนาวิทยาศาสตร์และเทคโนโลยี พ.ศ. 2534 เพื่อเป็นหน่วยงานที่บริหารกองทุนพัฒนาวิทยาศาสตร์และเทคโนโลยี โดยอยู่ภายใต้การกำกับดูแลของคณะกรรมการพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (กวทช.) ประกาศสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ เรื่อง โครงสร้างและการจัดองค์กร อำนาจหน้าที่ วิธีการดำเนินงาน และสถานที่ติดต่อเพื่อขอรับข้อมูลข่าวสารของสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ประกาศในราชกิจจานุเบกษา เล่ม ๑๓๒ ตอนที่ ๗๙ ง วันที่ ๑๓ สิงหาคม ๒๕๕๘ สวทช. มุ่งผลักดันให้ประเทศไทยแข็งแกร่งและเจริญรุ่งเรืองบนเวทีเศรษฐกิจระดับโลก โดยการนำความสามารถอันเหนือชั้นด้านวิทยาศาสตร์และเทคโนโลยีมาช่วยให้ภาคการ เกษตรและภาคอุตสาหกรรมสามารถดำเนินงานได้ดี มีประสิทธิภาพสูงขึ้น ซึ่ง สวทช. ได้ดำเนินงานผ่านการทำงานร่วมกันของศูนย์ทั้ง 5 ศูนย์ ได้แก่

1) ศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ (BIOTEC)
   มุ่งพัฒนางานด้านเทคโนโลยีชีวภาพ
2) ศูนย์เทคโนโลยีโลหะและวัสดุแห่งชาติ (MTEC)
   มุ่งพัฒนางานด้านเทคโนโลยีที่เกี่ยวข้องกับวัสดุต่างๆ
3) ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)
   มุ่งพัฒนางานด้านอิเล็กทรอนิกส์และเทคโนโลยีคอมพิวเตอร์
4) ศูนย์นาโนเทคโนโลยีแห่งชาติ (NANOTEC)
   มุ่งพัฒนางานด้านนาโนเทคโนโลยี
5) ศูนย์บริหารจัดการเทคโนโลยี (TMC)
   มุ่งให้ความช่วยเหลือนักวิจัยและบริษัทต่างๆ ในการนำผลงานการ ค้นพบและเทคโนโลยีต่างๆ มาใช้ประโยชน์ในเชิงพาณิชย์ [1]

ซึ่งศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (National Electronics and Computer Technology Center : NECTEC หรือเนคเทค) ได้เปลี่ยนแปลงสถานะเป็นศูนย์แห่งชาติเฉพาะทาง และเปลี่ยนการจัดรูปแบบองค์กรใหม่ เพื่อให้มีความคล่องตัวขึ้นกว่าเดิม ตามพระราชบัญญัติพัฒนาวิทยาศาสตร์และเทคโนโลยี พ.ศ. 2534 พ.ร.บ. ฉบับนี้ก่อให้เกิดการรวมตัวกันขององค์กรต่างๆ 4 องค์กรที่มีอยู่ขณะนั้น ได้แก่ คณะกรรมการพัฒนาวิทยาศาสตร์และเทคโนโลยี (Science and Technology Development Board: STDB หรือ กพวท.) ศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ ศูนย์เทคโนโลยีโลหะและวัสดุแห่งชาติ และศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ขึ้นเป็นสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (National Science and Technology Development Agency: NSTD หรือ สวทช.) สังกัดกระทรวงวิทยาศาสตร์เทคโนโลยีและสิ่งแวดล้อม [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องปฏิบัติการวิจัยระบบขนส่งและการจราจร (ITS) นี้เป็นหน่วยวิจัยภายใต้เนคเทค ที่ใช้เทคโนโลยีสารสนเทศด้านการเดินทาง ขนส่ง และจราจร ได้เล็งเห็นพฤติกรรมการขับขี่รถบนช่วงถนนของคนเราจะแตกต่างกันตามช่วงเวลา และวันต่างๆ โดยเฉพาะในกรุงเทพมหานครที่มีผู้คนเดินทางเข้ามาทำงานเป็นจำนวนมาก ก็จะส่งผลให้เกิดการจราจรติดขัดและอาจเกิดอุบัติเหตุขึ้นได้ ก็ต้องมีการลดความเร็วลงให้เหมาะสมเพื่อความปลอดภัยทางถนนในการลดอุบัติเหตุ จึงได้มีการนำข้อมูลทางสถิติบนท้องถนนในอดีตนี้มาคำนวณหาอัตราความเร็วที่เหมาะสมในแต่ละช่วงถนนของวันและเวลาต่างๆ และคำนวณหาระยะเวลาการเดินทางจากความเร็วและระยะทางบนช่วงถนนที่ต่างกัน เมื่อนำข้อมูลความเร็วและระยะเวลาการเดินทางทั้งหมดนี้มาเชื่อมโยงและเก็บรวบรวมไว้ในฐานข้อมูลให้ผู้ใช้ได้ค้นหาก็จะแสดงผลลัพธ์เฉลี่ยในแต่ละช่วงเวลาและวันต่างๆของอดีตให้ เพื่อความสะดวกรวดเร็วในการศึกษาหาข้อมูลการเดินทาง และสามารถนำข้อมูลนี้ไปประมาณการณ์คาดการณ์ระยะเวลาการเดินทางในอนาคต

เนื่องจากการที่จะคำนวณระยะทาง ความเร็วในช่วงเวลา ช่วงถนนต่าง และเส้นทางการเดินทางได้นั้น จะต้องนำจุด gps แต่ละจุดมา mapmatch ลงบนถนนต่อกันเป็น link เพื่อหาเส้นทางให้กับ gps แต่ละจุด ว่าอยู่บนถนนอะไรและตำแหน่งใด การที่จะคาดการณ์ระยะเวลาการเดินทางในอนาคต หรือเส้นทางที่ดีที่สุดได้นั้นก็จำเป็นจะต้องอาศัยข้อมูลในอดีตและปัจจุบันจำนวนมากเพื่อเพิ่มประสิทธิภาพและความแม่นยำ ซึ่งจุด gps ในฐานข้อมูลมีจำนวนมากกว่า 122,051,818 เรคคอร์ด ระยะเวลาตั้งแต่ปี 1970 - ปัจจุบัน ซึ่งมีจำนวนมากและจะเพิ่มขึ้นในทุกๆวัน โดยมีข้อมูลจากรถ 2 ประเภทคือ รถแท็กซี่ และรถขนของ ในการจัดการกับข้อมูลจำนวนมากนี้ จำเป็นจะต้องมีเครื่องมือที่มีประสิทธิภาพ ซึ่งแต่เดิมเก็บข้อมูลไว้ในฐานข้อมูล PostgreSQL แต่มีข้อจำกัดทางด้านข้อมูล ทำให้ไม่สามารถ query ได้ จึงได้มีการเก็บข้อมูลไว้ในฐานข้อมูล Mongodb ที่มีขนาดใหญ่กว่า และยืดหยุ่น สามารถเพิ่มข้อมูลเส้นทางลงในฐานข้อมูลให้กับ GPS แต่ละจุดนี้ได้จึงเหมาะสำหรับข้อมูลจำนวนมากและได้มีการนำ parallel มาใช้เพื่อแบ่งงานให้แต่ละ server mapmatch gps ที่มีในฐานข้อมูลได้พร้อมๆกัน และเสร็จได้รวดเร็วทันต่อการใช้งานมากยิ่งขึ้น

1.2 วัตถุประสงค์

1) เพื่อออกแบบระบบให้สามารถทำงานได้รวดเร็วขึ้น โดยพร้อมกันอัตโนมัติ
2) เพื่อหาเส้นทางให้กับ gps แต่ละจุด
3) เพื่อนำจุด gps ที่มีในฐานข้อมูล มา mapmatch ลงบน link

1.3 ขอบเขต

1) gps ที่มีในฐานข้อมูล Mongodb
2) จุด gps ในจังหวัดกรุงเทพมหานคร
3) เส้นทางภายในจังหวัดกรุงเทพมหานคร

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ระบบสามารถทำงานพร้อมกันได้โดยอัตโนมัติ
2) ได้เส้นทางของจุด gps ที่มีอยู่ในฐานข้อมูล
3) สามารถนำ link_id ไปหาเส้นทางที่ดีที่สุด และระยะเวลาในการเดินทางได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

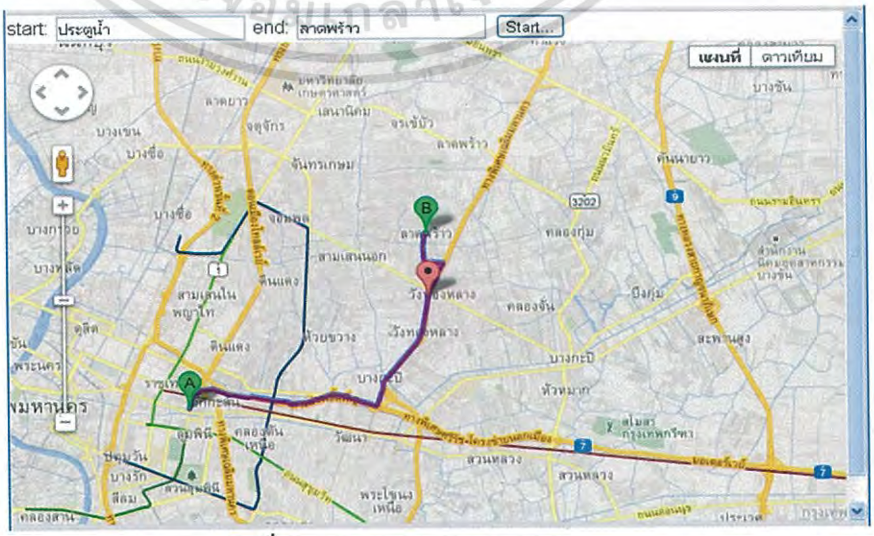
บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

Map matching เป็นเทคนิคในระบบสารสนเทศทางภูมิศาสตร์ที่หาตำแหน่งของยานพาหนะบนเส้นทางในแผนที่ มีวัตถุประสงค์เพื่อติดตามการจราจรของยานพาหนะ วิเคราะห์หาจุดเริ่มต้นและจุดสิ้นสุดของเส้นทางการขับรถ โดยวิธีการกำหนดตำแหน่งลงบนแผนที่นี้สามารถทำได้โดยการรับจุด GPS นี้มาหาเส้นทางที่ใกล้เคียงในพื้นที่ปิดบนแผนที่ก็จะได้เส้นทางของแต่ละจุด ซึ่งประกอบด้วย Three-tier architecture ดังนี้

2.1 การแสดงผล (Display)

2.1.1 Google Maps API เป็นชุด API ของ Google สำหรับพัฒนา web application และ mobile application (Android, iOS) ไว้สำหรับเรียกใช้แผนที่และชุด service ต่าง ๆ ของ Google เพื่อพัฒนา Application ได้เหมือนกับที่ Google โดยแผนที่ยัง features ต่าง ๆ มากมายให้เรียกใช้

– การปรับแต่งแผนที่ (Styled Map)
– ชุดควบคุมแผนที่ (Map Control)
– ชุดเครื่องมือวาดภาพบนแผนที่ (Drawing)
– การนำทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง (Directions Service)
– การคำนวณความสูงของจุดพิกัด (Elevation Service)
– การแปลงที่อยู่เป็นพิกัด Lattitude และ Longtitude (GeoCoding Service)
– การดึงข้อมูล POI (Point of Interest) คือข้อมูลสถานที่ต่าง ๆ ที่ Google รวบรวมไว้ให้ เช่น โรงแรม ห้างสรรพสินค้า โรงเรียน สถานที่ราชการต่างๆ และอื่นๆ อีกมากมาย (Places API) มาใช้งานใน application
– Street View [3] ภาพ Google Maps API ดังรูปที่ 2.1



รูปที่ 2.1 Google Maps API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การประมวลผล (Processing)

2.2.1 Quantum GIS (QGIS) เป็นโปรแกรม Desktop GIS ประเภทหนึ่งที่มีประสิทธิภาพในการนำมาใช้บริหารจัดการข้อมูลปริภูมิ โดยจัดอยู่ในกลุ่มซอฟต์แวร์รหัสเปิด (Free and Open Source Software: FOSS) ที่มีลักษณะระบบใช้งานง่าย ลักษณะการใช้งานเป็นแบบ Graphic User Interface (GUI) ซึ่งสะดวกต่อการใช้งาน ไม่ว่าจะเป็นการเรียกใช้ข้อมูลภาพ ข้อมูลตาราง การแสดงผลตาราง การแสดงผลกราฟ ตลอดจนสามารถสืบค้นข้อมูล วิเคราะห์ข้อมูลและนำเสนอข้อมูลได้ในรูปแบบแผนที่ สามารถเรียกใช้ข้อมูลเวกเตอร์ ราสเตอร์ ในรูปแบบที่เป็นมาตรฐานแพร่หลาย เช่น Shape file และ Geo Image ซอฟต์แวร์ QGIS สามารถแก้ไข Shape File format ได้ซึ่งเป็นที่ต้องการมากในเวลานี้ QGIS พัฒนาบนพื้นฐานของ Qt ที่ เป็นไลบรารี่สำหรับ Graphical User Interface (GUI) ที่ใช้งานได้ทั้ง UNIX, Window และ Mac การพัฒนาใช้ภาษา C++ เป็นหลัก นอกจากนั้น QGIS ยังเชื่อมต่อกับ Geospatial RDBMS เช่น PostGIS/PostgreSQL สามารถอ่านและเขียนฟีเจอร์ที่จัดเก็บใน PostGIS ได้โดยตรง สามารถเชื่อมต่อกับ GRASS ได้ทำให้สามารถเรียกดูข้อมูลที่จัดเก็บใน GRASS โดยตรง และสามารถเรียกใช้ฟังก์ชันต่างๆของ GRASS ได้ สนับสนุนการวิเคราะห์เชิงพื้นที่ (Spatial Analysis) ในเบื้องต้นและการแสดงผลข้อมูลเชิงตำแหน่งในรูปแบบของแผนที่การสร้างและการแก้ไขข้อมูลเชิงตำแหน่ง (Spatial Data) และข้อมูลตาราง (AttributeData สามารถจัดการข้อมูลได้ง่ายโดยใช้เครื่องมือตาม GUI ที่กำหนด [4] ภาพโปรแกรม Quantum GIS ดังรูปที่ 2.2

รูปที่ 2.2 โปรแกรม Quantum GIS

2.2.2 Putty เป็นโปรแกรมที่มีหน้าที่เอาไว้ ใช้ในการ Remote หรือติดต่อ เชื่อมต่อ ไปยังเครื่องคอมพิวเตอร์อีกเครื่องหนึ่ง โดยวิธีการ Telnet หรือ Secure Shell (SSH) จากเครื่องลูก (Client) เข้าไปจัดการ พิมพ์คำสั่ง หรือส่งคำสั่ง ในเครื่องแม่ (Server) ด้วยระบบ Command-line Interface (ใช้พิมพ์คำสั่ง) โดยให้ความรู้สึกเหมือนนั่งอยู่บนหน้าเครื่องเซิร์ฟเวอร์ (เครื่องแม่) จริงๆ โดยไม่ต้องเดินทางไปนั่งหน้าเครื่องนั้นจริงๆ อยู่ที่ไหนก็สามารถเข้าไปได้ และทำการเชื่อมต่อ (Connect) เข้าไปยังเครื่องแม่เซิร์ฟเวอร์ ได้ง่ายๆ เพียงแค่พิมพ์ชื่อโฮสต์ (Hostname) พอร์ตที่ต้องการเชื่อมต่อ (Port) ซึ่งโดยส่วนใหญ่ค่าปกติคือ 22 ซึ่งก็จะใส่รอเอาไว้ให้เราแล้ว และเมื่อใส่ค่าป้อนค่าต่างๆ ที่จำเป็นเสร็จสิ้นแล้ว ก็สามารถเชื่อมต่อเข้าไปได้เลยทันที โดย โปรแกรม Telnet ตัวนี้ถือว่าขนาดเล็กกะทัดรัด ไม่ถึงครึ่ง MB. [5] ภาพโปรแกรม Putty ดังรูปที่ 2.3

เอกสารนี้ ... ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 โปรแกรม Putty

2.2.3 WinSCP Portable (โปรแกรม FTP SFTP โปรแกรมรับส่งไฟล์ ผ่าน โปรแกรม Putty) สำหรับโปรแกรมนี้มีชื่อว่า โปรแกรม WinSCP เป็นโปรแกรมประเภทที่เอาไว้ใช้รับส่งไฟล์ หรือที่เรียกว่า โปรแกรม FTP หรือ SFTP SCP มันเป็น โปรแกรมรับส่งไฟล์ ขนาดจิ๋ว พกพาได้ที่ดีอีกตัวหนึ่ง ที่รวมเอาคุณสมบัติของ การเชื่อมต่อข้อมูล หรือใช้ในการ รับส่งไฟล์ ต่างๆ ระหว่างเครื่องลูก (Client) ไปยังเครื่องแม่ข่าย (Server) แต่ว่ามันต้องทำงานร่วมกับ โปรแกรม Telnet ชื่อดังอย่าง โปรแกรม Putty ซึ่งไม่ได้ให้มันเป็นเพียงแค่ โปรแกรมส่งคำสั่ง (Command) ไปยังเครื่องแม่ข่ายอย่างเดียว แต่ยังทำให้มันสามารถเป็น โปรแกรมรับส่งไฟล์ ที่เอาไว้อัพโหลด ดาวน์โหลดไฟล์ ระหว่างเครื่องลูกกับ เครื่องแม่ ได้อีกด้วย [6] ภาพโปรแกรม WinSCP ดังรูปที่ 2.4

รูปที่ 2.4 โปรแกรม WinSCP

2.2.4 FileZilla เป็นโปรแกรมที่ใช้ติดต่อกับ FTP server เพื่อดาวน์โหลดหรืออัปโหลดไฟล์ โดยเฉพาะกับเว็บไซต์ ทำให้เหมาะสำหรับนักออกแบบเว็บ นอกจากนี้ FileZilla ยังรองรับการถ่ายไฟล์อย่างปลอดภัยผ่าน SSH (SFTP) อีกด้วย FileZilla รองรับการกลับมาถ่ายไฟล์ต่อในกรณีที่อัปโหลดหรือดาวน์โหลดล้มเหลว และทำงานได้ดีผ่านไฟร์วอล และพรอกซี FileZilla มีวิธีใช้งานที่สะดวก แบ่งวินโดว์ออกเป็นสองส่วนคือแสดงฝั่ง local กับ remote จากนั้นก็สามารถ drag and drop ไฟล์ระหว่างสองฝั่งเพื่ออัปโหลดหรือดาวน์โหลดได้ทันที สามารถ login อย่างรวดเร็วโดยป้อนข้อมูลของไซต์คือที่อยู่ ผู้ใช้ และรหัสผ่าน ได้บนทูลบาร์เลยทันที นอกจากนี้ FileZilla ยังมี host manager ที่เก็บรายละเอียดของ เซิร์ฟเวอร์ที่ใช้เป็นประจำ เพื่อให้เชื่อมต่อได้อย่างง่ายและรวดเร็ว FileZilla ทำงานได้เฉพาะระบบ Windows เท่านั้น [7] ภาพโปรแกรม FileZilla ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 โปรแกรม FileZilla

2.2.5 PHP ภาษาคอมพิวเตอร์ประเภทโอเพนท์ซอร์ทสำหรับพัฒนาเว็บเพจเมื่อเครื่องบริการได้รับคำร้องจากผู้ใช้ก็จะส่งให้กับ ตัวแปล ภาษา ทำหน้าที่ประมวลผลและส่งข้อมูลกลับไปยังเครื่องของผู้ใช้ที่ร้องขอ ในรูป HTML ภาพ หรือแฟ้ม digital อื่นๆ ลักษณะของภาษามี คำสั่งมาจากภาษาซี ภาษาจาวาและ ภาษาเพิร์ล ซึ่ง ภาษา PHPนั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีความตอบโต้ได้อย่างรวดเร็ว PHP เป็นภาษาจำพวกภาษาสคริปคำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่าสคริปต์ และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่งตัวอย่างของภาษาสคริปเช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถ สอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ [8]

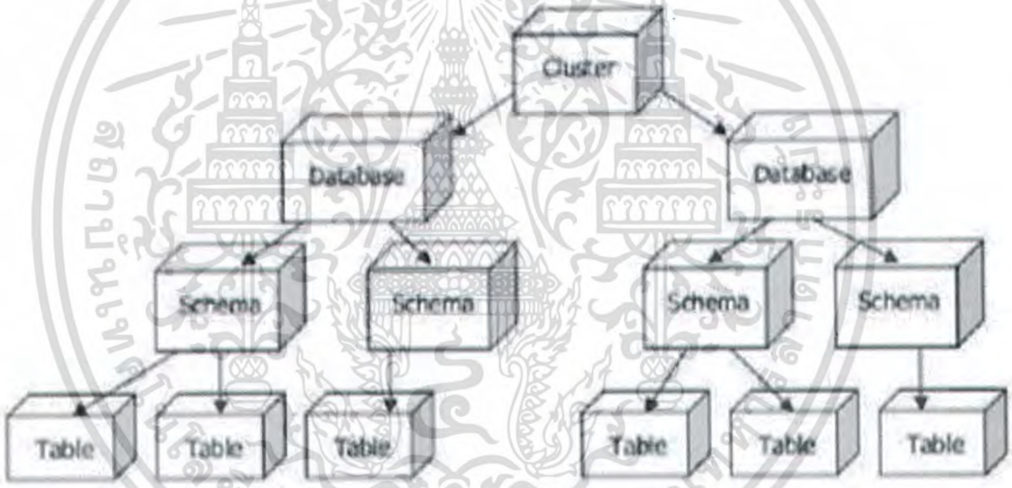
2.2.6 SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรมเพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) หมายถึงเราสามารถใช้คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อสั่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดยึดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่ง ซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

1. Select query ใช้สำหรับดึงข้อมูลที่ต้องการ
2. Update query ใช้สำหรับแก้ไขข้อมูล
3. Insert query ใช้สำหรับการเพิ่มข้อมูล
4. Delete query ใช้สำหรับลบข้อมูลออกไป [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ฐานข้อมูล (Database)

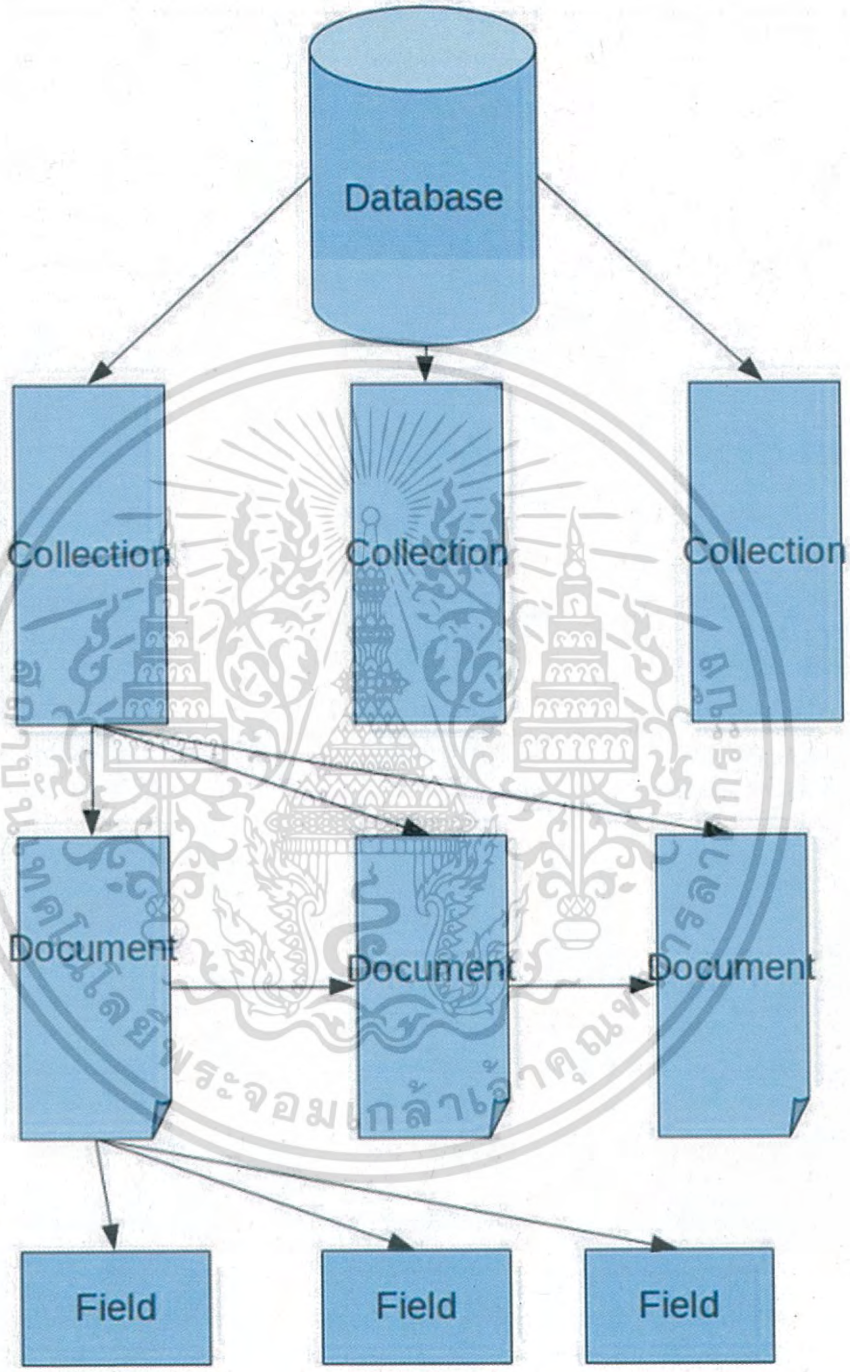
2.3.1 PostgreSQL (โพสต์เกรสคิวเอล) ชื่อเดิมคือ Postgres (โพสต์เกรส) เป็น DBMS เป็นระบบจัดการฐานข้อมูลแบบ object-relational database management system หรือ (ORDBMS) พัฒนาที่ University of California, Berkeley (มหาวิทยาลัย) เริ่มโครงการโดย Michael Stonebraker เมื่อปี พ.ศ. 2528 ในยุคแรกชื่อของระบบเรียกว่า post-Ingres เนื่องมาจาก จากเป็นระบบที่มีวิวัฒนาการมาจากระบบจัดการฐานข้อมูล Ingres นอกจากนี้ PostgreSQL ยังทำงานในหลายแพลทฟอร์มได้แก่ Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), และ Windows คุณสมบัติสำคัญของ PostgreSQL คือการมีคุณสมบัติ ACID ((Atomicity, Consistency, Isolation, Durability) ครบถ้วนโดยสนับสนุน foreign keys, joins, views, triggers, และ stored procedures (หลายภาษา) โดยมีชนิดข้อมูลใน SQL92 และ SQL99 ได้แก่ INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, และ TIMESTAMP ภาพโครงสร้างของ PostgreSQL ดังรูปที่ 2.6 [10]



รูปที่ 2.6 โครงสร้างของ PostgreSQL

2.3.2 Mongodb เป็น open-source document database โดยเป็นฐานข้อมูลแบบ NoSQL คือไม่มี relation (ความสัมพันธ์) ของตารางแบบ SQL ทั่วๆไป แต่จะเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) แทน การบันทึกข้อมูลทุกๆ record ใน Mongodb เรียกว่า Document ซึ่งจะเก็บค่าเป็น key และ value จะเห็นว่ามันก็คือ JSON และการเก็บข้อมูล document ใน Mongodb จะถูกเก็บไว้ใน Collections (เปรียบเทียบได้กับ Table ใน Relational Database ทั่วๆไป) แต่แตกต่างกันที่ Collection ไม่จำเป็นที่จะต้องมี schema เหมือนกันก็สามารถบันทึกข้อมูลได้ ภาพโครงสร้างของ MongoDB ดังรูปที่ 2.7 [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 โครงสร้างของ Mongodb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงาน

เนื่องจากมีข้อมูล gps อยู่ในฐานข้อมูล Mongodb ซึ่งจะต้องนำจุด gps ในฐานข้อมูลนี้ไป mapmatch เพื่อหาเส้นทางให้กับจุด gps แต่ละจุด คือ link_id แล้วนำ link_id นี้เพิ่มเข้าไปในตารางข้อมูลของ gps เพื่อนำ link_id นี้ไปหาความเร็ว ระยะเวลาในการเดินทาง และเส้นทางที่ดีที่สุดต่อไป ดังรูปที่ 3.1

รูปที่ 3.1 การ mapmatch

ซึ่งการ mapmatch นี้จะมีด้วยกันอยู่ 2 วิธี คือ แบบแรกจะเรียกฟังก์ชัน mapmatch จากฐานข้อมูล PostgreSQLกับข้อมูลเดิมที่มีอยู่แล้วในฐานข้อมูล Mongodb server 1 จะแบ่งไป mapmatch แต่ละ vid แต่เนื่องจากมีข้อมูล GPS เพิ่มเข้ามาใหม่ตลอดเวลาซึ่งมีทั้ง vid เดิม และ vid ใหม่ ซึ่งเป็นยากการหาจำนวนของแต่ละ vid จึงได้มีการย้ายตาราง link จากฐานข้อมูล PostgreSQLมาไว้ยัง Mongodb server 0 และเขียนฟังก์ชัน mapmatch ขึ้นมาใหม่จะได้ไม่ต้องเสียเวลาในการเชื่อมต่อฐานข้อมูล PostgreSQL อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การ mapmatch โดยเรียกฟังก์ชันจากฐานข้อมูล PostgreSQL

วิธีการนี้จะใช้กับข้อมูล gps ในฐานข้อมูล Mongodb server 1 ซึ่งสามารถเชื่อมต่อกันได้ทั้งหมด 6 servers ดังนี้

1) xxx.xxx.xx.26
2) xxx.xxx.xx.27
3) xxx.xxx.xx.28
4) xxx.xxx.xx.87
5) xxx.xxx.xx.88
6) xxx.xxx.xx.89

โดยมี server xxx.xxx.xx.27 เป็น server หลักที่เก็บฟังก์ชัน mapmatch ในฐานข้อมูล PostgreSQL จึงจะต้อง Export ข้อมูลนี้ ไปให้กับ servers ที่เหลือโดยใช้โปรแกรม FileZilla ในการถ่ายโอนไฟล์ให้กับแต่ละ servers และแต่ละ servers ก็จะต้องติดตั้ง PostgreSQL, Postgis แล้วสร้าง database "Traffic_Data" และ Import ข้อมูล mapmatch นี้ลงในฐานข้อมูล และทดสอบ mapmatch ในโปรแกรม putty ดังนี้

3.1.1 ติดตั้ง PostgreSQL และ Postgis และทดสอบ mapmatch

1) **ติดตั้ง Postgis** [12]
   sudo apt-get install postgresql postgresql-contrib postgis postgresql-9.3-postgis-2.1
2) **เชื่อมต่อ Postgres** [13]
   sudo su postgres
3) **เข้า Postgres**
   psql
4) **สร้าง database ชื่อ "Traffic_Data"**
   create database "Traffic_Data";
5) **เข้า database "Traffic_Data"**
   psql -d "Traffic_Data"
6) **สร้าง Extension**
   create extension postgis;
7) **Import ข้อมูล mapmatch ลงใน database "Traffic_Data"**
   psql -d "Traffic_Data" -f mapmatch.sql
8) **ทดสอบ mapmatch**
   SELECT gid FROM link.mapmatch (longitude,latitude);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ศึกษา Mongodb

1) **Create Database**

   use DATABASE_NAME

2) **Show Databases**

   show dbs

3) **Drop Database**

   db.dropDatabase()

4) **Create Collection** [14]

   db.createCollection(name, options)

5) **Drop Collection**

   db.collection.drop()

6) **Insert Document**

   db.COLLECTION_NAME.insert(document)

7) **Query Document**

   db.COLLECTION_NAME.find()

8) **DELECT Document**

   db.COLLECTION_NAME.remove(DELLECTION_CRITTERIA)

9) **Update Document**

   db.COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATE_DATA)

10) **Limit Records** [15]

    db.COLLECTION_NAME.find().limit(NUMBER)

11) **Skip**

    db.COLLECTION_NAME.find().limit(NUMBER).skip(NUMBER)

12) **Distinct** [16]

    db.COLLECTION_NAME.distinct(DISTINCT_OPERATION)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 Insert ข้อมูล และ update link_id

1) เชื่อมต่อ Mongodb และ show database ดังรูปที่ 3.2

```
mongos> show dbs
admin      0.016GB
config     0.047GB
gps      489.623GB
```

รูปที่ 3.2 Show database

2) Use gps และ show collections ดังรูปที่ 3.3

```
mongos> use gps
switched to db gps
mongos> show collections
gps
gpsbackup
link
```

รูปที่ 3.3 Show collections

3) Create collection mapmatch ดังรูปที่ 3.4

```
mongos> db.createCollection("mapmatch")
{
        "ok" : 1,
        "$gleStats" : {
                "lastOpTime" : Timestamp(1455593084, 1),
                "electionId" : ObjectId("56bcc218be1588c39fef65ab")
        }
}
```

รูปที่ 3.4 Create collection mapmatch

4) Show collection mapmatch ดังรูปที่ 3.5

```
mongos> show collections
gps
gpsbackup
link
mapmatch
```

รูปที่ 3.5 Show collection mapmatch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) Insert source_id, row_id, lat, long และ heading ดังรูปที่ 3.6

```
mongos> db.mapmatch.insert({
... "source_id":"601022",
... "row_id":1,
... "lat":14.075594,
... "long":100.601618,
... "heading":0
... })
WriteResult({ "nInserted" : 1 })
mongos>
```

รูปที่ 3.6 Insert ข้อมูล

6) เมื่อ insert ข้อมูล แล้วกลับมาดูในฐานข้อมูลจะพบว่ามี _id เพิ่มเข้ามา ดังรูปที่ 3.7

```
{
    "_id": ObjectId("566e6e354aa7e924aa11e95f"),
    "source_id": "601022",
    "row_id": 1,
    "lat": 14.075594,
    "long": 100.601618,
    "heading": 0
}
```

รูปที่ 3.7 ข้อมูลที่ Insert แล้ว

7) นำ _id ที่ได้นี้ไป update link_id ดังรูปที่ 3.8

```
mongos> db.mapmatch.update({"_id": ObjectId("566e6e354aa7e924aa11e95f")},
... {$set:{"link_id":60422}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
mongos>
```

รูปที่ 3.8 Update link_id

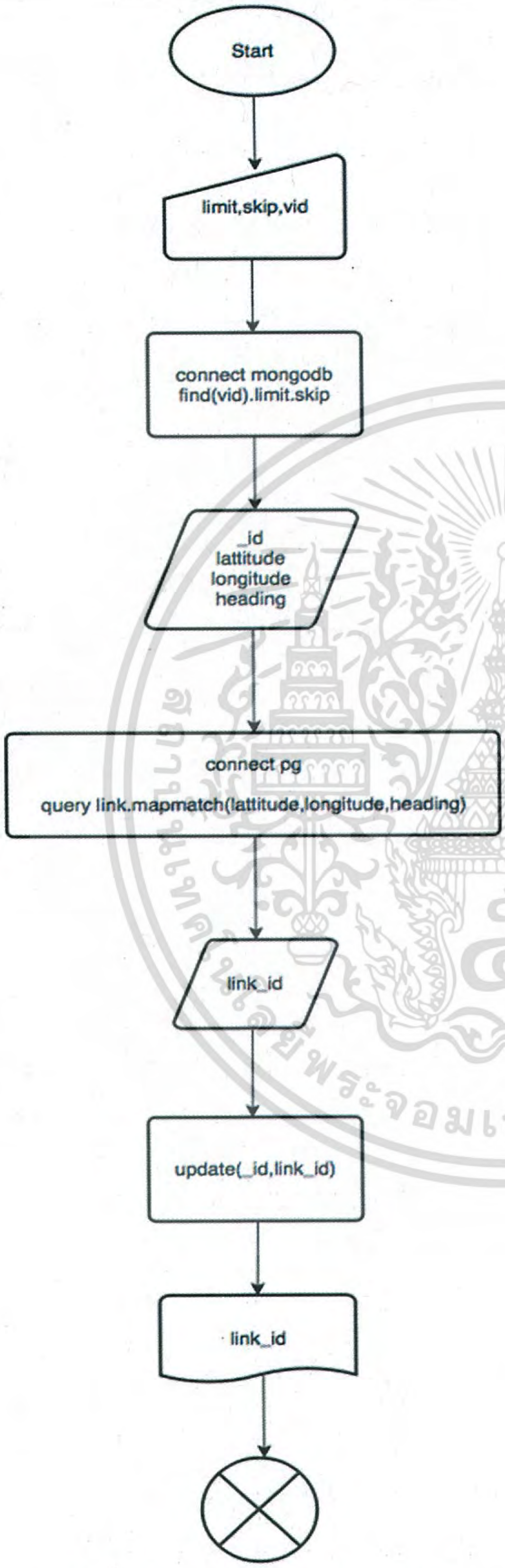
8) เมื่อ update link_id แล้วกลับมาดูในฐานข้อมูลจะพบว่ามี link _id เพิ่มเข้ามา ดังรูปที่ 3.9

```
{
    "_id": ObjectId("566e6e354aa7e924aa11e95f"),
    "source_id": "601022",
    "row_id": 1,
    "lat": 14.075594,
    "long": 100.601618,
    "heading": 0,
    "link_id": 60422
}
```

รูปที่ 3.9 link_id ที่เพิ่มเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 โปรแกรม mapmatch

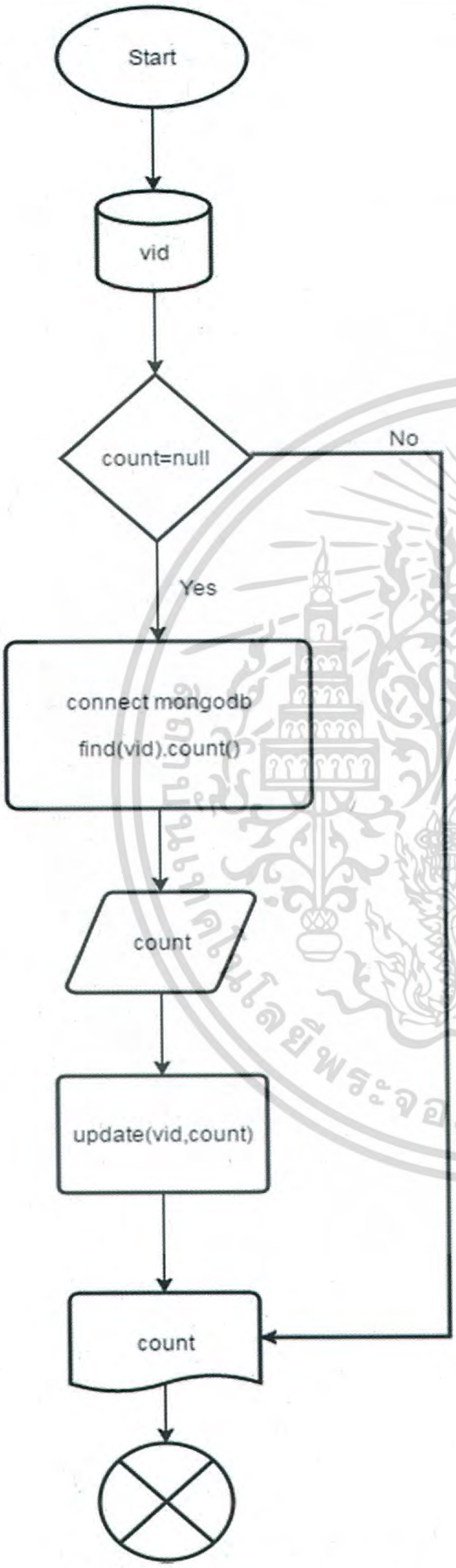


รูปที่ 3.10 Flowchart mapmatch

1) เริ่มต้นรับข้อมูล limit,skip,vid มา
2) เขียนฟังก์ชันติดต่อกับ Mongodb โดยนำเอา vid ไปค้นหาจำนวน ตั้งแต่ตัวที่ skip เป็นต้นไปจนครบ ตามจำนวนที่ limit
3) นำเอาข้อมูล lattitude, longitude และ heading ของข้อมูล gps แต่ละ _id
4) เขียนฟังก์ชันเชื่อมต่อกับ pgAdmin ไปค้นหาฟังก์ชัน mapmatch ในตาราง link นำค่า lattitude, longitude และ heading ไปใส่ในฟังก์ชัน
5) จะได้ค่า link_id ของข้อมูล gps ที่รับมา
6) เขียนฟังก์ชัน update link_id ไปเพิ่มในข้อมูล gps ของ_id ที่รับมา
7) แสดงค่าของ link_id ออกมา ดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 โปรแกรมหาจำนวนของแต่ละ vid



รูปที่ 3.11 Flowchart count

1) หา vid ทั้งหมดในฐานข้อมูล Mongodb และสร้างตารางเก็บข้อมูลนี้ไว้

2) เขียนโปรแกรมเชื่อมต่อตารางข้อมูลของ vid ตรวจดูว่ามีจำนวนของvidแต่ละตัวหรือไม่

- No คือถ้า count ไม่เท่ากับ null แสดงว่า vid นั้นมีจำนวนอยู่ในตารางข้อมูลอยู่แล้ว ก็ให้แสดงค่าจำนวนของ vid นั้นออกมาได้เลย

- Yes คือใช่ count=null แสดงว่ายังไม่มีจำนวนของ vid นั้นอยู่ในตาราง vid ก็ให้ดำเนินการดังต่อไปนี้

  - เขียนฟังก์ชันเชื่อมต่อ Mongodb และรับ vid นั้นมาหาจำนวนทั้งหมด

  - จะได้จำนวนทั้งหมดของ vid นั้นออกมา

  - นำจำนวนทั้งของ vid นั้นไป update ในตารางข้อมูลของ vid

  - แสดงค่าจำนวนของ vid นั้นออกมา ดังรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 โปรแกรมเรียกรัน mapmatch แบบ parallel

1) เริ่มต้นเรียกโปรแกรม getvid จะไปเชื่อมต่อฐานข้อมูลของ vid ที่เก็บจำนวนของแต่ละ vid

2) จะได้ข้อมูล vid และ count ของแต่ละ vid มาให้

3) นำจำนวนของ vid นั้นมาหารจำนวน server ที่มีอยู่คือ 6 server ก็จะได้จำนวน limit ที่จะส่งให้แต่ละ servers รันข้อมูลได้เท่ากันๆกัน

4) getvid จะส่งข้อมูล vid, limit, skip มายังโปรแกรม map ของแต่ละ server โดยค่าของ skip คือจำนวนที่จะส่งไปให้โปรแกรม mapmatch ตั้งแต่จำนวนนั้นเป็นต้นไป จนถึงค่าที่ limit ไว้ ดังนั้น server เริ่มต้นจะมีค่า skip=0 และserverถัดไปจะมีค่าเพิ่มขึ้นตามจำนวนที่ limitไปเรื่อยๆจนหมด เช่น limit=1000

   - server xxx.xxx.xx.26 จะมี skip=0
   - server xxx.xxx.xx.27 จะมี skip=1000
   - server xxx.xxx.xx.28 จะมี skip=2000
   - server xxx.xxx.xx.87 จะมี skip=3000
   - server xxx.xxx.xx.88 จะมี skip=4000
   - server xxx.xxx.xx.89 จะมี skip=5000

โปรแกรม map ของแต่ละ server จะรับค่า vid, limit, skip มาแบ่งเรียกโปรแกรม mapmatch อีกจำนวน 10 โปรแกรม จึงนำ limit ที่รับมามาหารด้วย 10 จะได้ limit ใหม่ที่มีค่าน้อยลงกว่าเดิม แล้วกำหนดค่า skip แรกให้เท่ากับค่า skip ที่แต่ละ server รับมา และskip ต่อไปจะมีค่าเท่ากับ skip ก่อนหน้าบวกกับจำนวนที่ limit ใหม่ไปเรื่อยๆจนครบ 10 ค่า ทุก server ก็จะรัน mapmatch ได้พร้อมกัน10โปรแกรมพร้อมๆกัน เช่น จาก limit=1000 นำมาหาร 10 จะได้ limit ใหม่เท่ากับ 100

- server xxx.xxx.xx.26 มี skip=0
  - mapmatch จะมี skip=0
  - mapmatch จะมี skip=100
  - mapmatch จะมี skip=200
  - mapmatch จะมี skip=300
  - mapmatch จะมี skip=400
  - mapmatch จะมี skip=500
  - mapmatch จะมี skip=600
  - mapmatch จะมี skip=700
  - mapmatch จะมี skip=800
  - mapmatch จะมี skip=900

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

- server xxx.xxx.xx.27 มี skip=1000
  - mapmatch จะมี skip=1000
  - mapmatch จะมี skip=1100
  - mapmatch จะมี skip=1200
  - mapmatch จะมี skip=1300
  - mapmatch จะมี skip=1400
  - mapmatch จะมี skip=1500
  - mapmatch จะมี skip=1600
  - mapmatch จะมี skip=1700
  - mapmatch จะมี skip=1800
  - mapmatch จะมี skip=1900
- server xxx.xxx.xx.28 มี skip=2000
  - mapmatch จะมี skip=2000
  - mapmatch จะมี skip=2100
  - mapmatch จะมี skip=2200
  - mapmatch จะมี skip=2300
  - mapmatch จะมี skip=2400
  - mapmatch จะมี skip=2500
  - mapmatch จะมี skip=2600
  - mapmatch จะมี skip=2700
  - mapmatch จะมี skip=2800
  - mapmatch จะมี skip=2900
- server xxx.xxx.xx.87 มี skip=3000
  - mapmatch จะมี skip=3000
  - mapmatch จะมี skip=3100
  - mapmatch จะมี skip=3200
  - mapmatch จะมี skip=3300
  - mapmatch จะมี skip=3400
  - mapmatch จะมี skip=3500
  - mapmatch จะมี skip=3600
  - mapmatch จะมี skip=3700
  - mapmatch จะมี skip=3800
  - mapmatch จะมี skip=3900

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และจะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
149191

- server xxx.xxx.xx.88 มี skip=4000
  - mapmatch จะมี skip=4000
  - mapmatch จะมี skip=4100
  - mapmatch จะมี skip=4200
  - mapmatch จะมี skip=4300
  - mapmatch จะมี skip=4400
  - mapmatch จะมี skip=4500
  - mapmatch จะมี skip=4600
  - mapmatch จะมี skip=4700
  - mapmatch จะมี skip=4800
  - mapmatch จะมี skip=4900
- server xxx.xxx.xx.89 มี skip=5000
  - mapmatch จะมี skip=5000
  - mapmatch จะมี skip=5100
  - mapmatch จะมี skip=5200
  - mapmatch จะมี skip=5300
  - mapmatch จะมี skip=5400
  - mapmatch จะมี skip=5500
  - mapmatch จะมี skip=5600
  - mapmatch จะมี skip=5700
  - mapmatch จะมี skip=5800
  - mapmatch จะมี skip=5900

5) เมื่อโปรแกรม mapmatch รับค่า limit, skip, vid ติดต่อกับ Mongodb โดยนำเอา vidไปค้นหาจำนวนตั้งแต่ตัวที่ skip เป็นต้นไปจนครบตามจำนวนที่ limit

6) นำเอาข้อมูล lattitude, longitude และ heading ของข้อมูล gps แต่ละ _id ออกมา

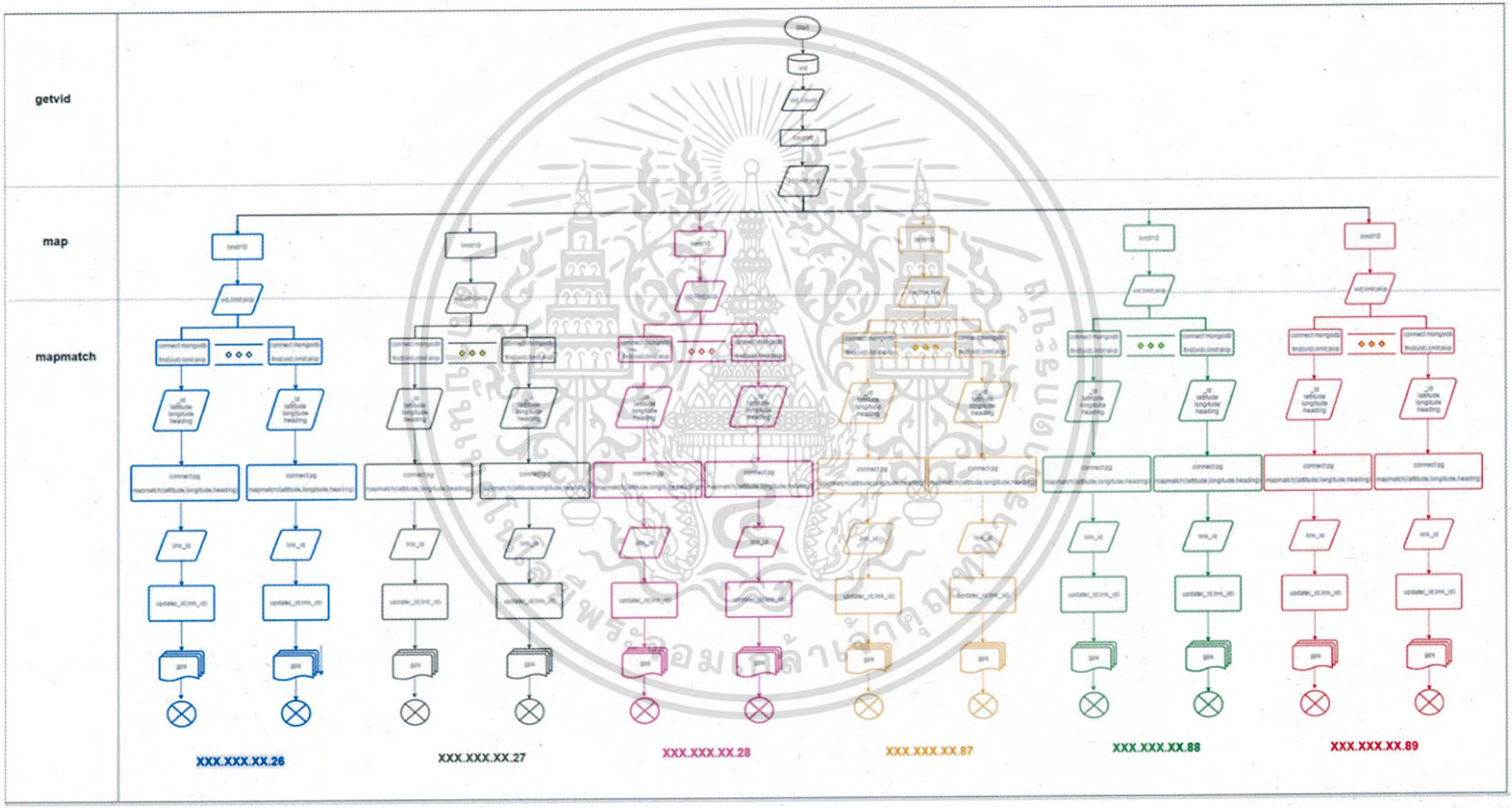
7) เชื่อมต่อกับ pgAdmin นำค่า lattitude, longitude, heading ไปค้นหา link_id ในฟังก์ชัน mapmatch

8) จะได้ค่า link_id ของข้อมูล gps ที่รับมา

9) update link_id ไปเพิ่มในข้อมูล gps ของ _id ที่รับมา

10) แสดงข้อมูลทั้งหมดในตาราง gps ของแต่ละ _id ออกมา ดังรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ภาพรวมการ mapmatch แบบ parallel

3.2 การ mapmatch โดยสร้างฟังก์ชันขึ้นมาใหม่ใน Mongodb

วิธีการนี้จะใช้กับข้อมูล GPS ในฐานข้อมูล Mongodb server 0 ซึ่งสามารถเชื่อมต่อกันได้ทั้งหมด 10 servers ดังนี้

1) xxx.xxx.xx.170
2) xxx.xxx.xx.171
3) xxx.xxx.xx.172
4) xxx.xxx.xx.173
5) xxx.xxx.xx.175
6) xxx.xxx.xx.176
7) xxx.xxx.xx.177
8) xxx.xxx.xx.178
9) xxx.xxx.xx.179
10) xxx.xxx.xx.180

โดยมี server xxx.xxx.xx.170 เป็น server หลัก ที่จะใช้เขียนโปรแกรม และสร้างฟังก์ชัน mapmatch ขึ้นมาใหม่ ก็จะต้องย้ายข้อมูลจากตาราง link ในฐานข้อมูล PostgreSQL server xxx.xxx.xx.27

3.2.1 Insert ข้อมูลตาราง link จากฐานข้อมูล PostgreSQL มาไว้ในฐานข้อมูล Mongodb

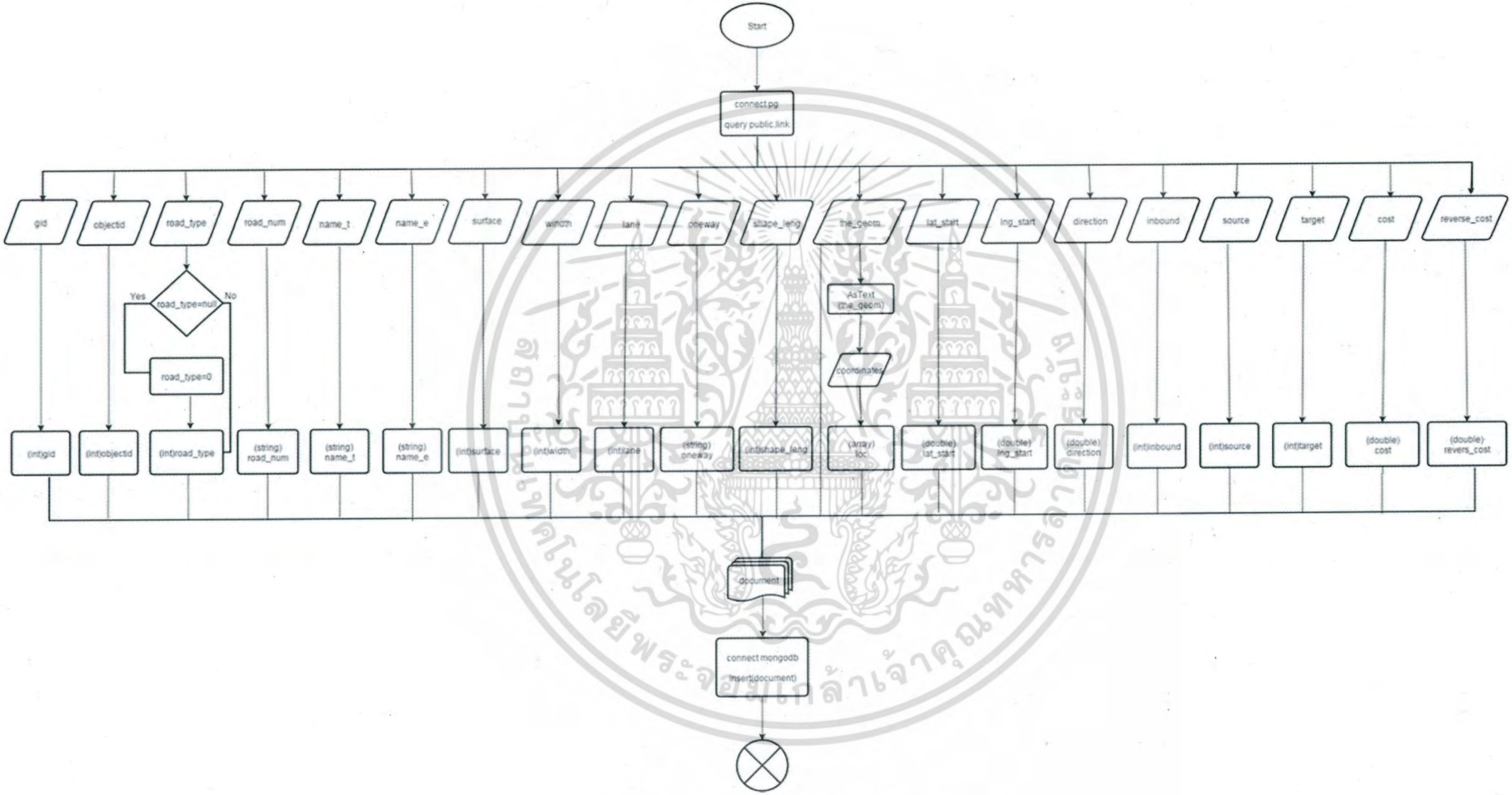
1) เริ่มต้นเชื่อมต่อ PostgreSQL
2) ค้นหาตาราง link รับข้อมูล
   - gid ไปเก็บไว้เป็น integer
   - objected ไปเก็บไว้เป็น integer
   - road_type มาตรวจสอบว่ามีข้อมูลหรือเป็น null ไหม
     - Yes คือ road_type=null หรือไม่มีข้อมูล จึงกำหนดให้ road_type=0 ก่อนจะไปเก็บไว้เป็น integer
     - No คือ road_type ไม่เท่ากับ null หรือมีข้อมูลอยู่แล้ว ก็นำข้อมูลนั้นไปเก็บไว้เป็น interger ได้เลย
   - road_num ไปเก็บไว้เป็น string
   - name_t ไปเก็บไว้เป็น string
   - name_e ไปเก็บไว้เป็น string
   - surface ไปเก็บไว้เป็น integer
   - windth ไปเก็บไว้เป็น integer
   - lane ไปเก็บไว้เป็น integer
   - oneway ไปเก็บไว้เป็น string

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- shape_leng ไปเก็บไว้เป็น integer
- the_geom ไปเก็บไว้เป็น string และจะต้องนำไปแยกเป็น latitude, longitude ของแต่ละจุดเก็บไว้เป็น array ของ loc [17]
- lat_start ไปเก็บไว้เป็น double
- lng_start ไปเก็บไว้เป็น double
- direction ไปเก็บไว้เป็น double
- inbound ไปเก็บไว้เป็น integer
- source ไปเก็บไว้เป็น integer
- target ไปเก็บไว้เป็น integer
- cost ไปเก็บไว้เป็น double
- reverse_cost ไปเก็บไว้เป็น double

3) รวมข้อมูลทั้งหมดนี้ไว้ใน document
4) เชื่อมต่อ Mongodb
5) Insert document เข้าไปที่ตาราง linkในฐานข้อมูล Mongodb ดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 Flowchart insert ข้อมูลตาราง link

3.2.2 โปรแกรมหา link_id จากตาราง link

1) เริ่มต้นรับข้อมูล id, latitude, longitude, heading และรัศมีมา

2) สร้าง polygon โดยนำ latitude, longitude ที่รับมาเป็นจุดศูนย์กลาง แล้วหาจุดรอบๆที่ห่างจากจุดศูนย์กลางตามรัศมีที่รับมา [18]

3) จะได้ coordinates คือจุดพิกัดรอบๆจุดนั้น

4) นำ coordinates มารวมไว้ใน document [19]

5) เชื่อมต่อตาราง link ใน mongodb ไปค้นหา gid ที่มีจุด lattitude, longitude ภายใน document

6) นำมาตรวจสอบดูว่าพบไหม หรือ curser=null

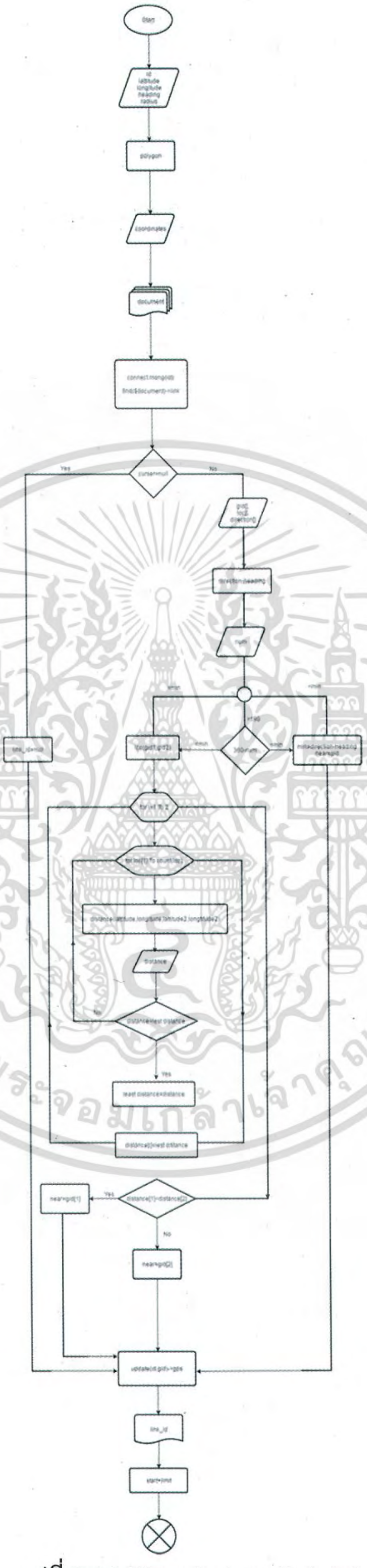
- Yes คือ curser=null จะไม่พบเส้นทางใดเลย link_id = null
- No คือ curser ไม่เท่ากับ null แสดงว่าพบ gid ก็นำค่านั้นออกมาพร้อมกับ loc คือ array ของจุดภายใน gid นั้น และนำ direction มาเปรียบเทียบกับ heading ก็จะได้ num จำนวนแรกที่ออกมาให้มีค่าเป็น min มาเปรียบ

- ถ้า num มีค่ามากกว่า 180 จะต้องนำจำนวนนั้นมาลบกับ 360 ก่อน จึงจะนำไปเปรียบเทียบว่ามีค่าน้อยกว่า min ไหม
- ถ้า num มีค่าน้อยกว่า min ก็จะกำหนดค่า min=num และ gid ของค่านั้นเป็นเส้นทางที่ใกล้ที่สุดแทน
- ถ้า num เท่ากับ min ก็จะต้องนำ loc ของทั้ง 2 gid นั้นมาแยกออกมาเป็น latitude, longitude ของแต่ละจุดมาเปรียบเทียบกับจุดที่รับมา และเก็บค่าที่น้อยที่สุดไว้เป็นระยะทางที่ของ gid นั้น แล้วนำระยะทางที่ได้ของ gid ทั้ง 2 มาเปรียบเทียบกันว่า distance มีค่าน้อยกว่า distance2 หรือไม่ [20]
    - Yes แสดงว่า ระยะทางของเส้นทางแรกใกล้กว่า ดังนั้น near=gid[1]
    - No แสดงว่า ระยะทางของเส้นที่ 2 มีค่าน้อยกว่าจึงใกล้กว่า ดังนั้น near=gid[2]

7) นำค่า gid ที่ได้ไป update เป็น link_id เพิ่มเข้าในตาราง gps ของ id ที่รับมา

8) แสดงค่าของ link_id ออกมา ดังรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



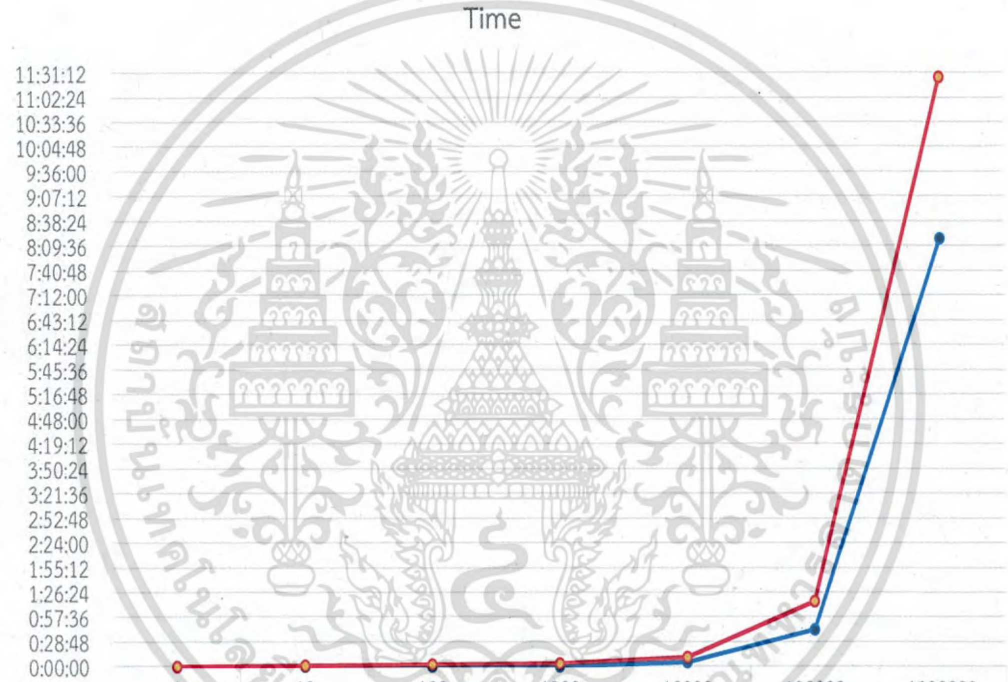
รูปที่ 3.14 Flowchat หา link_id

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลสหกิจศึกษาและการอภิปรายผล

จากการ mapmatch โดยเรียกฟังก์ชันจากฐานข้อมูล PostgreSQL กับข้อมูลเดิมที่มีอยู่แล้วในฐานข้อมูล mongodb server 1 ต่อ 1 เรคคอร์ดใช้เวลา 7 วินาที ซึ่งมากกว่าการ mapmatch โดยสร้างฟังก์ชันขึ้นมาใหม่ใน Mongodb server 0 ต่อ 1 เรคคอร์ดใช้เวลา 3 วินาที และเมื่อข้อมูลยิ่งเพิ่มมากขึ้นเรื่อยๆ เวลาก็จะเพิ่มมากขึ้นด้วย จะเห็นได้ชัดจากกราฟข้อมูล Time ข้างล่างนี้ เปรียบเทียบเวลาการ mapmatch ของข้อมูลตั้งแต่ 1 ถึง 1,000,000 เรคคอร์ด ของแต่ละ server



	1	10	100	1000	10000	100000	1000000
server 0	0:00:03	0:00:03	0:00:03	0:00:28	0:04:19	0:43:06	8:17:33
server 1	0:00:07	0:01:37	0:02:07	0:03:18	0:11:25	1:16:11	11:26:23

server 0 server 1

รูปที่ 4.1 กราฟเปรียบเทียบเวลาในการ mapmatch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลสหกิจศึกษาและข้อเสนอแนะ

5.1 สรุปผลสหกิจศึกษา

1) จุด gps ที่มีในฐานข้อมูล Mongodb server 1 มา mapmatch ลงบน link ได้ 86,569,388 จุด จากข้อมูลทั้งหมด 122,051,818 เรคคอร์ด และยังเหลืออยู่อีก 35,482,430 จุดที่ไม่พบ link_id ซึ่งเกิดจากจุด gps นั้นอยู่นอกเส้นทาง

2) พบ link_id ที่ mapmatch ได้ ทั้งหมด 268,760

3) ระบบสามารถทำงานได้รวดเร็วขึ้น โดยพร้อมกันอัตโนมัติ จากการ mapmatch โดยสร้างฟังก์ชันขึ้นมาใหม่ใน Mongodb server 0 สามารถหา link_id ได้รวดเร็วกว่าการ mapmatch ในฐานข้อมูล Mongodb server 1 ที่เรียกฟังก์ชันจากฐานข้อมูล PostgreSQL ทำให้เสียเวลาในการติดตั้งและเชื่อมต่อฐานข้อมูล PostgreSQL จึงควรใช้การ mapmatch ของโปรแกรมใน Mongodb server 0

5.2 ข้อเสนอแนะ

จากการ mapmatch gps ที่ไม่สามารถหา link_id ได้หมด เนื่องจากจุดนั้นอยู่นอกเส้นทาง หากต้องการที่จะหาเส้นทางของจุด gps ได้ทั้งหมดก่อนที่จะนำข้อมูลจุด GPS มา mapmatch จึงควรจะทำ data cleansing และจัดกลุ่มข้อมูลโดยแยกรถแต่ละคัน แบ่งช่วงเวลาในแต่ละวัน และตัดข้อมูลที่รถหยุดนิ่งหรือจอดอยู่นอกเส้นทางออกก่อน เพื่อให้สามารถทำการ mapmatch ได้รวดเร็วยิ่งขึ้น และพบเส้นทางมากขึ้นอีก หากมีข้อมูลจุด gps ที่ไม่พบเส้นทางในระหว่างที่พบจุดก่อนหน้าและจุดถัดไปเป็นเส้นทางเดียวกัน ก็จะถือว่า gps จุดนั้นมีเส้นทางเดียวกับจุดก่อนหน้าและจุดถัดไป

จากข้อมูล link_id ที่ได้จากการ mapmatch นี้สามารถนำไปใช้กับรถเมล์ได้โดยการเชื่อมต่อเส้นทางไปยังป้ายรถเมล์แต่ละจุด อาจเพิ่มเส้นทางเป็นทางน้ำหรือบริเวณที่ไม่ใช่ถนนให้สามารถนำไปใช้กับยานพาหนะอื่นๆได้ อย่างเช่น เรือ เป็นต้น และการหาเส้นทางที่มีรถนิยมวิ่งผ่าน หรือหาเส้นทางที่ดีที่สุดนี้ก็สามารถนำไปต่อยอดหาระยะเวลาในการเดินทาง ระยะเวลาที่รถจะไปถึงที่หมาย บอกเส้นทางที่มีรถติดในแต่ละช่วงเวลาให้ผู้ใช้สามารถหลีกเลี่ยงเส้นทางนั้นๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] สวทช.2015.**แนะนำ สวทช.** [online]. Available: http://www.nstda.or.th/aboutus-nstda

[2] NETEC.2015.**ประวัติเนคเทค.** [online]. Available: http://www.nectec.or.th

[3] Swiftlet.2015.**GOOGLE MAP API คืออะไร?** [online]. Available: https://swiftlet.co.th/google-api

[4] Admin.2009.**คู่มือเรียนรู้พื้นฐานการใช้โปรแกรม Quantum GIS (QGIS)**. [online]. Available: http://www.scitu.net/gcom/?p=1316.

[5] Krittidech Pomuang.2011. **การศึกษาการทำงานโปรแกรม PuTTy**. [online]. Available: http://www.msit.mut.ac.th/newweb/phpfile/show.php?Qid=6961

[6] Simon Tatham.2015. **WinSCP Portable**. [online]. Available: http://software.thaiware.com

[7] Mindphp.2012.**Filezilla คืออะไร วิธีการใช้งาน Filezilla - โปรแกรม FTP อัพโหลดไฟล์**. [online]. Available: http://www.mindphp.com/forums/viewtopic.php?t=13645.

[8] Memcache.2015. **PHP**. [online]. Available: http://www.thaiall.com

[9] Chaiyaporn.2015. **SQL คืออะไร.** [online]. Available: http://digitionary.com.

[10] Peekanung.2009. **PostgreSQL Manual (ภาษาไทย).** [online]. Available: http://www.scribd.com

[11] IMA8.2015. **MongoDB : สร้าง Database , เพิ่มข้อมูล , เรียกดูข้อมูล.** [online]. Available: http://ima8.me.

[12] Wiki.2015.**PostGIS/Installation**. [online]. Available: http://wiki.openstreetmap.org/wiki/PostGIS/Installation#Ubuntu_14.04_LTS.

[13] Prajuab Riabroy.2012. **ขั้นตอนการติดตั้ง PostGIS/Postgresql บน Ubuntu Server**. [online]. Available: http://priabroy.com/tag/postgis/.

[14] Tutorials Point.2015.**MongoDB - Create Collection**. [online]. Available: http://www.tutorialspoint.com/mongodb/mongodb_create_collection.htm.

[15] Benjamin. 2014. **Sqlite LIMIT / OFFSET query.** [online]. Available: http://stackoverflow.com/questions/3325515/sqlite-limit-offset-query.

[16] Mohsin.2012. **Select Distinct Count in MongoDB**. [online]. Available: https://dbamohsin.wordpress.com.

[17] PostGIS.2015.**ST_AsText**. [online]. Available: http://postgis.net

[18] Sarunyakan.2011. **วิธีการเบื้องต้นในการหาจุดในโพลิกอน**. [online]. Available: http://its.nectec.or.th/2011/08/29/find-point-in-polygon/

[19] Tobias Trelle. 2013. **MongoDB 2.4 Introduces Geospatial Indexing and Search for GeoJSON Geometries Point, LineString and Polygon**. [online]. Available: https://blog.codecentric.de.

[20] Shadow.2011.**Haversine formula**. [online]. Available: http://www.thaicreate.com/dotnet/forum/054719.html.

เอกสารนี้เป็นเอกสารเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

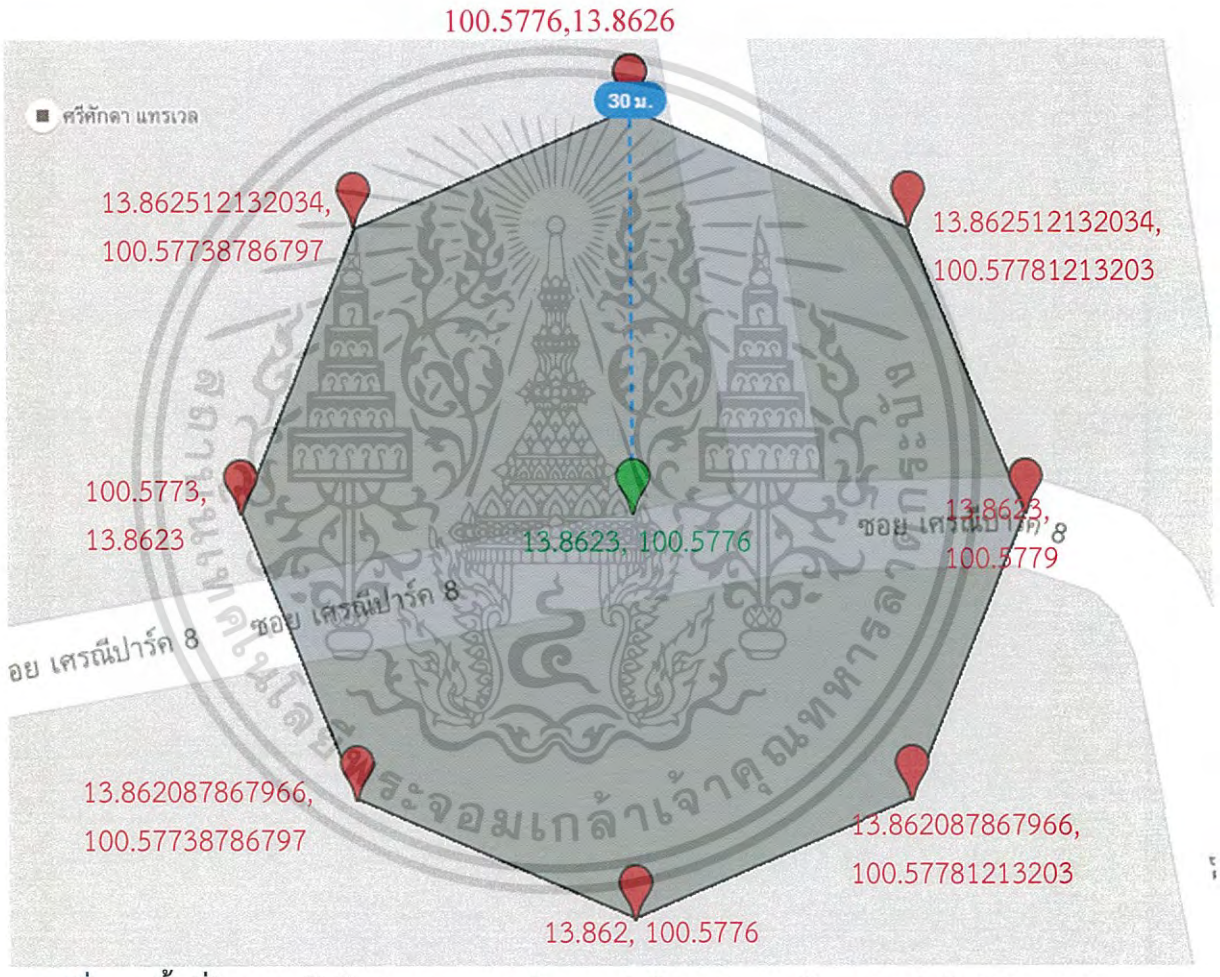
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การหา link_id เพื่อไป update ในฐานข้อมูล

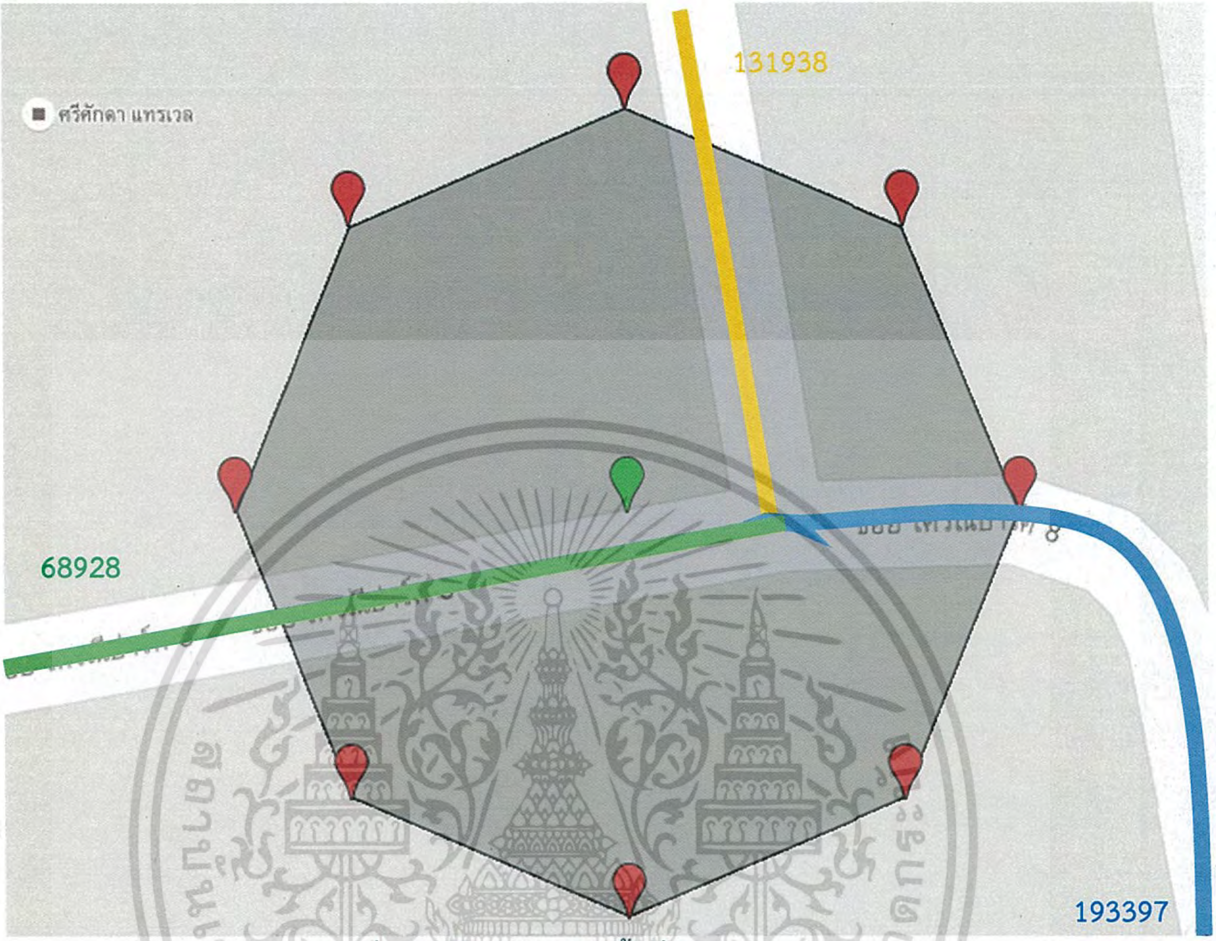
1) โปรแกรมจะรับค่า _id, latitude , longitude และ heading นั้นมา หาพื้นที่โดยรอบรัศมี 30 เมตร เช่น "_id": 56252fc8c08683d6558bd526 มี "latitude": 13.8623, "longitude": 100.5776 และ "heading": "144.0" จะได้พื้นที่ปิดดังรูปที่ ก.1



รูปที่ ก.1 พื้นที่โดยรอบรัศมี 30 เมตร ของ "latitude": 13.8623, "longitude": 100.5776

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) จะพบเส้นทางภายในรอบจุดดังรูปที่ ก.2



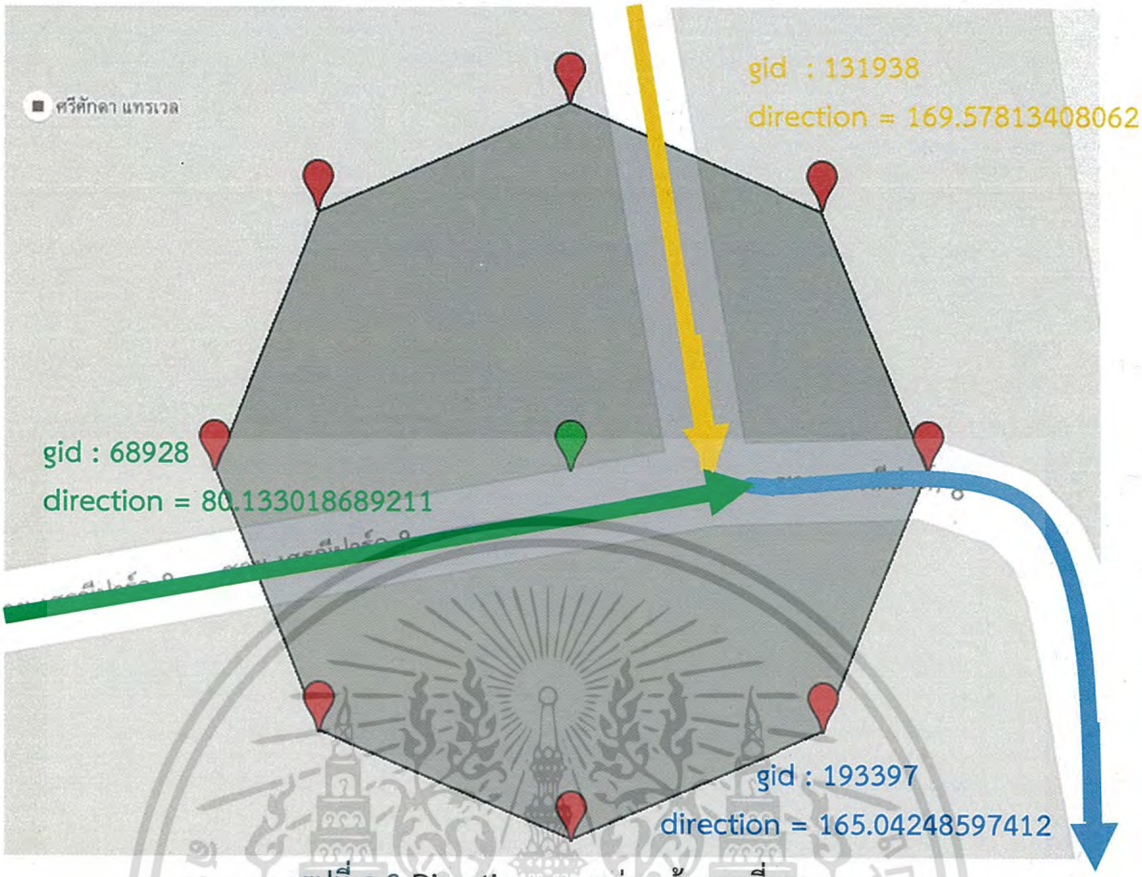
รูปที่ ก.2 เส้นทางภายในพื้นที่โดยรอบจุด

3) นำ direction ของแต่ละ gid มาเปรียบเทียบกับ heading ที่รับมาคือ 144.0

- gid : 68928
  direction = 80.133018689211
- gid : 131938
  direction = 169.57813408062
- gid : 193397
  direction = 165.04248597412

จะได้ gid : 68928 ที่เส้นทางใกล้เคียงกับจุดนี้มากที่สุด ดังรูปที่ ก.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 Direction ของแต่ละเส้นทางที่พบ

4) นำ gid ที่ได้ไป update เป็น link_id ในข้อมูลของ _id ที่รับมา จะได้ผลดังภาพที่ ก.4


```
{
  "_id": ObjectId("56252fc8c08683d6558bd526"),
  "source": "MasterTech",
  "vehicleID": "6010220024",
  "dataTimestamp": "2015-10-19 23:46:17",
  "dataDate": "2015-10-19",
  "latitude": 13.8623,
  "longitude": 100.5776,
  "speed": "0.0",
  "heading": "144.0",
  "engineStatus": "na",
  "fuelStatus": "-734.69",
  "linkID": NumberInt(0),
  "vehicleType": "0",
  "insertTimestamp": "2015-10-20 01:00:40",
  "time_second": "17",
  "time_min": "46",
  "time_hour": "23",
  "date_day": "19",
  "date_month": "10",
  "date_year": "2015",
  "date_dayweek": "Mon",
  "geo": {
    "loc": [
      100.5776,
      13.8623
    ]
  },
  "link_id": 68928
}
```

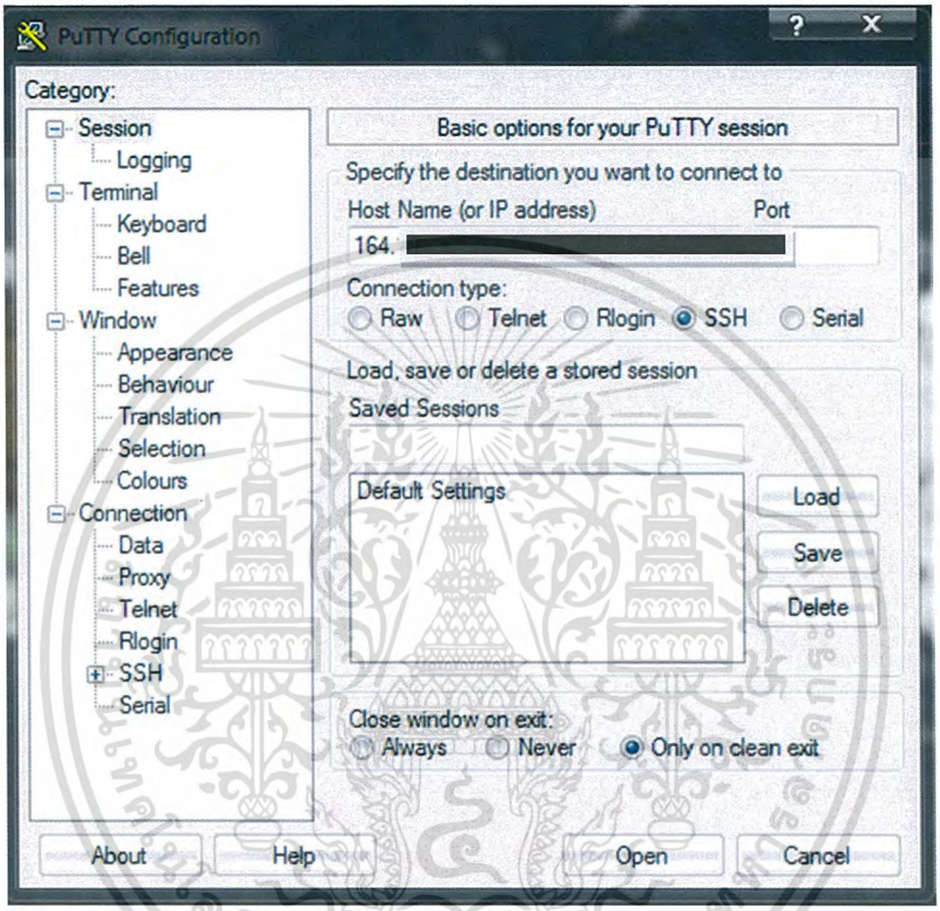

รูปที่ ก.4 ข้อมูลที่ Update link_id แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

การเรียกรันโปรแกรม mapmatch

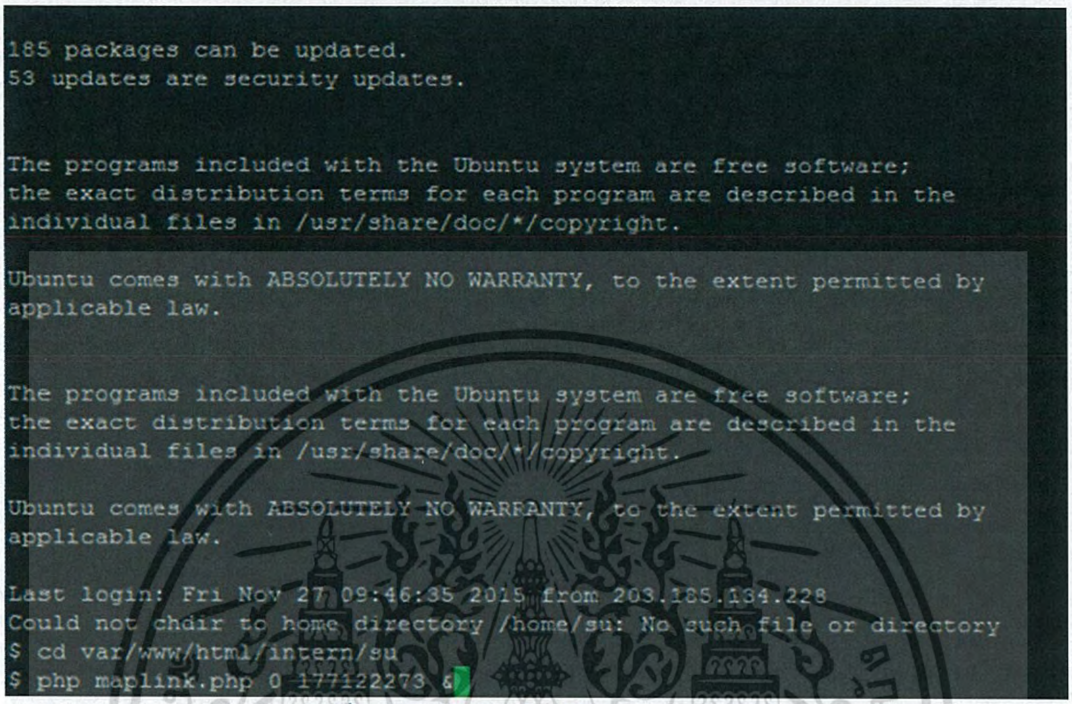
1) เปิดโปรแกรม putty และเข้า Host Name ดังรูปที่ ข.1



รูปที่ ข.1 เชื่อมต่อ server

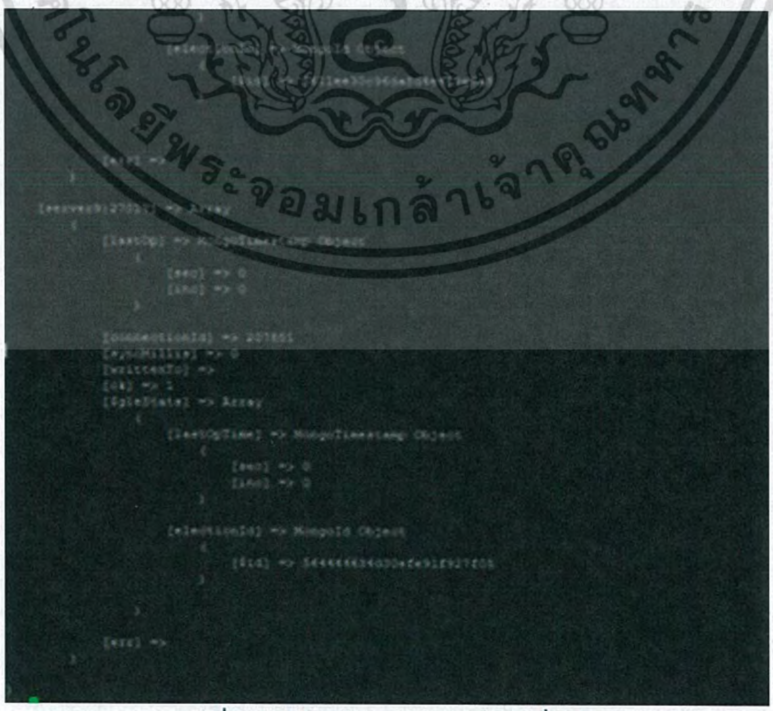
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) login เข้าสู่ directory และเรียกรันไฟล์ maplink.php ดังรูปที่ ข.2 จะเริ่มรันตั้งแต่ตัวที่ 0 เป็นต้นไป จนถึงตัวที่ 17122273 ในรอบที่จะรันต่อไปก็ให้เริ่มตั้งแต่ตัวที่ 17122273 ไปจนถึงตัวสุดท้ายในฐานข้อมูลที่เพิ่มเข้ามา



รูปที่ ข.2 การเรียกรันโปรแกรม maplink

3) โปรแกรม maplink ก็จะเริ่มรัน ดังรูปที่ ข.3



รูปที่ ข.3 โปรแกรม maplink เริ่มรัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้