การออกแบบและพัฒนาโปรแกรมประยุกต์ภูมิสารสนเทศออนไลน์สำหรับการวางแผนการเดินทางด้วยรถโดยสารประจำทาง

นายภาณุวัฒน์ อังคสุรักษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาระบบสารสนเทศปริภูมิทางวิศวกรรม ภาควิชาวิศวกรรมสำรวจ
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

DESIGN AND DEVELOPMENT OF ONLINE GIS APPLICATION FOR BUS TRIP PLANNING

Mr. Panuwat Angkasurak

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Spatial Information System in Engineering

Department of Survey Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2011

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การออกแบบและพัฒนาโปรแกรมประยุกต์ภูมิสารสนเทศออนไลน์สำหรับการวางแผนการเดินทางด้วยรถโดยสารประจำทาง
โดย	นายภาณุวัฒน์ อังคสุรักษ์
สาขาวิชา	ระบบสารสนเทศปริภูมิทางวิศวกรรม
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.สรรเพชญ ชื้อนิธิไพศาล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

…….. คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

…… ประธานกรรมการ
(ศาสตราจารย์ ดร.เฉลิมชนม์ สถิระพจน์)

…….. อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.สรรเพชญ ชื้อนิธิไพศาล)

…… กรรมการ
(อาจารย์ ดร.ธงทิศ ฉายากุล)

…… กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ (พิเศษ) ดร.ปริญญา ถนัดทาง)

ภาณุวัฒน์ อังคสุรักษ์ : การออกแบบและพัฒนาโปรแกรมประยุกต์ภูมิสารสนเทศออนไลน์สำหรับการวางแผนการเดินทางด้วยรถโดยสารประจำทาง. (DESIGN AND DEVELOPMENT OF ONLINE GIS APPLICATION FOR BUS TRIP PLANNING) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.สรรเพชญ ชื้อนิธิไพศาล, 100 หน้า.

งานวิจัยนี้เป็นการศึกษาและออกแบบแบบจำลองข้อมูลสำหรับจำลองพฤติกรรมการเดินทางด้วยรถโดยสารประจำทาง ภายในพื้นที่รับผิดชอบของเขตการเดินรถที่ 5 ของ ขสมก. ซึ่งเป็นพื้นที่ศึกษา โดยให้มีโครงสร้างข้อมูลของโครงข่ายการเดินทางที่เหมาะสมต่อการจัดเก็บ จัดการ และวิเคราะห์โครงข่ายด้วยระบบฐานข้อมูล Oracle Spatial

ในกระบวนการวิจัย ได้พัฒนาโปรแกรมประยุกต์ที่ทำงานบนเว็บเบราว์เซอร์ผ่านเครือข่ายอินเทอร์เน็ตประกอบด้วย ส่วนติดต่อกับผู้ใช้ ส่วนการวิเคราะห์โครงข่ายด้วยระบบฐานข้อมูล Oracle Spatial และส่วนของการรายงานผลลัพธ์จากการวิเคราะห์โครงข่ายทั้งในรูปของแผนที่และข้อมูลประกอบการเดินทาง ในส่วนของโปรแกรมการวิเคราะห์โครงข่าย ผู้วิจัยได้ปรับแต่งเพิ่มปัจจัยในการพิจารณา โดยให้พิจารณาถึงอัตราเร็วของรถโดยสารประจำทางที่แตกต่างกันไปตามช่วงเวลาของวัน

ผลการศึกษาพบว่า โปรแกรมประยุกต์ที่พัฒนาขึ้น สามารถนำมาใช้เป็นแนวทางในการวางแผนการเดินทางด้วยรถโดยสารประจำทางได้ โดยให้ผลลัพธ์ตรงตามเงื่อนไขของปัจจัยเรื่องอัตราเร็วของรถโดยสารที่แตกต่างกันในแต่ละช่วงของวัน และระบบฐานข้อมูล Oracle Spatial สามารถนำมาใช้จัดเก็บ จัดการ และวิเคราะห์ข้อมูลโครงข่ายการเดินทางด้วยรถโดยสารประจำทางได้อย่างเหมาะสม เนื่องจากมีการสนับสนุนแบบจำลองข้อมูลในลักษณะโครงข่าย และสามารถเอื้ออำนวยต่อการปรับแต่งเพิ่มปัจจัยต่างๆ ในการวิเคราะห์เชิงโครงข่ายได้

ภาควิชา วิศวกรรมสำรวจ ลายมือชื่อนิสิต .....
สาขาวิชา ระบบสารสนเทศปริภูมิทางวิศวกรรม. ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....
ปีการศึกษา 2554 .

5170425021 : MAJOR SPATIAL INFORMATION SYSTEM IN ENGINEERING
KEYWORDS :BUS TRANSPORT / SHORTEST TRAVELING TIME / ORACLE SPATIAL / GEOSERVER / OPENLAYERS

PANUWAT ANGKASURAK : DESIGN AND DEVELOPMENT OF ONLINE GIS APPLICATION FOR BUS TRIP PLANNING. ADVISOR : ASST.PROF. SANPHET CHUNITHIPAISAN, 96 pp.

The purpose of the research is to study and design data model of bus transportation behavior covering the bus operation area in zone 5 defined by Bangkok Mass Transit Authority (BMTA). The data structure of the model is appropriately designed to collect, manage, and analyze by Oracle Spatial: database management program.

The research methodologies of this study are designed to develop the application program that can operated via online web browser. The application consists of user interface section, network data analysis section (operating on Oracle Spatial program) and output section which is able to display both map and relevant data for the trip. For the network data analysis section, the research adjusts and increases the influenced factor for network analysis that is the variation of speed of the bus differentiated by the period of time in each day.

The results demonstrated that the developed application program could be utilized as the approach for bus trip planning. The outcomes corresponded to the condition regarding with the speed of the bus that is differentiated by the period of time in each day. Furthermore, Oracle Spatial database system could be operated for collecting, managing and analyzing for bus network transportation data effectively since Oracle supports network data model and is favorable for any adjustments of factors that affect the network analysis.

Department : Survey Engineering Student's Signature .....
Field of Study : Spatial Information System in Engineering Advisor's Signature .....
Academic Year : 2011

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ด้วยความช่วยเหลือและการสนับสนุนจากบุคคลหลายท่านโดยเฉพาะอย่างยิ่งข้าพเจ้าขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สรรเพชญ ชื้อนิธิไพศาล อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้ให้แนวความคิด คำแนะนำในการแก้ปัญหา และตรวจสอบวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงด้วยดี

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ประกอบด้วย ศาสตราจารย์ ดร.เฉลิมชนม์ สถิระพจน์ อาจารย์ ดร.ธงทิศ ฉายากุล และผู้ช่วยศาสตราจารย์ (พิเศษ) ดร.ปริญญา ถนัดทาง ที่ให้คำแนะนำในและตรวจสอบวิทยานิพนธ์ฉบับนี้จนสำเร็จโดยสมบูรณ์ รวมทั้งคณาจารย์ภาควิชาวิศวกรรมสำรวจ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ให้ความรู้แก่ข้าพเจ้า

ขอขอบพระคุณกระทรวงคมนาคมที่ให้ความอนุเคราะห์ข้อมูลที่ใช้ในงานวิจัย

ขอขอบพระคุณพันโท ดร.สรวิศ สุภเวชย์ สำหรับคำแนะนำอันมีค่า รวมถึงความช่วยเหลือในทุกๆ ด้าน

ขอขอบคุณคุณอรุณ บุรีรักษ์ ที่ช่วยตรวจทานและจัดรูปเล่มวิทยานิพนธ์จนเสร็จเรียบร้อย

ขอขอบคุณคุณนิวัฒน์ ตั้นตระกูล ที่ช่วยเรียบเรียงบทคัดย่อในส่วนของภาษาอังกฤษ

ขอขอบคุณคุณโชคอารีย์ ขาวพราย ผู้เป็นกำลังใจ

ท้ายสุดนี้ขอขอบพระคุณบิดา มารดา พี่สาวทุกๆ คน คุณพรรัตน์ เชยสมบัติ คุณลักษณา ศรีโกศัย คุณธัญญา จีระสมบูรณ์ยิ่ง คุณจันทริกา สนั่นเกียรติเจริญ คุณวารุนีย์ จี้ฟู ที่คอยให้การสนับสนุนและมอบความอบอุ่นเสมอมา

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง

บทคัดย่อภาษาอังกฤษ.....จ

กิตติกรรมประกาศ.....ฉ

สารบัญ.....ช

สารบัญตาราง.....ฎ

สารบัญภาพ.....ฎ

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 วัตถุประสงค์ของการวิจัย.....2

1.3 ขอบเขตของการวิจัย.....2

1.4 ข้อตกลงเบื้องต้น.....3

1.5 ข้อจำกัดของการวิจัย.....3

1.6 วิธีดำเนินการวิจัย.....3

1.7 ประโยชน์ที่คาดว่าจะได้รับ.....4

บทที่ 2 แนวคิดและทฤษฎีที่เกี่ยวข้อง.....5

2.1 สถานภาพของการให้บริการระบบรถโดยสารประจำทางในเขตกรุงเทพมหานครและปริมณฑล.....5

2.2 ทฤษฎี Graph.....9

2.3 ORACLE SPATIAL.....10

2.3.1 Oracle Network Data Model.....10

2.3.1.1 โครงสร้างข้อมูล (Network Tables).....11

2.3.1.2 การสร้างโครงข่าย (Defining Network).....12

2.3.2 Load-On-Demand Approach.....12

2.4 ทฤษฎีทางด้านการวิเคราะห์โครงข่าย (Network Analysis).....13

2.5 Web-based GIS.....15

2.6 Geoserver.....15

2.7 OpenLayers.....16

หน้า

2.8 AJAX (Asynchronous JavaScript and XML) 16
2.9 JSON (JavaScript Object Notation) 17
2.10 งานวิจัยที่เกี่ยวข้อง 18
2.10.1 ระบบสารสนเทศภูมิศาสตร์ผ่านเครือข่ายอินเทอร์เน็ต เพื่อเลือกเส้นทางที่ใช้เวลาน้อยที่สุดโดยระบบขนส่งมวลชนหลายรูปแบบ 18
2.10.2 การพัฒนาโปรแกรมประยุกต์บนอุปกรณ์คอมพิวเตอร์พกพา (PDA) เพื่อเลือกเส้นทางที่เหมาะสม ในระบบขนส่งมวลชนหลายรูปแบบ 20
บทที่ 3 การจัดเตรียมข้อมูลโครงข่ายสำหรับการเดินทางด้วยรถโดยสารประจำทาง 21
3.1 พฤติกรรมการเดินทางด้วยรถโดยสารประจำทาง 21
3.1.1 การเดินเท้า 21
3.1.2 รถโดยสารประจำทาง 21
3.2 การออกแบบแบบจำลองข้อมูล 22
3.2.1 ข้อมูลโครงข่ายรถโดยสารประจำทางที่ออกแบบแบบจำลองข้อมูลให้แทนด้วย link 22
3.2.2 ข้อมูลโครงข่ายรถโดยสารประจำทางที่ออกแบบให้แทนด้วย node 24
3.3 การออกแบบโครงสร้างการจัดเก็บข้อมูล 24
3.3.1 การออกแบบโครงสร้างข้อมูลของตารางหลักที่ใช้จัดเก็บข้อมูลโครงข่าย 24
3.3.2 การออกแบบรหัสประจำตัวให้กับข้อมูล link และ node 28
3.3.3 การออกแบบโครงสร้างตารางข้อมูลอื่นๆ ที่ต้องใช้ในงานวิจัย 28
3.4 การจัดเตรียมข้อมูลภูมิศาสตร์สำหรับระบบรถโดยสารประจำทาง 30
3.4.1 ข้อมูลที่เกี่ยวข้อง 30
3.4.2 การจัดทำและปรับปรุงข้อมูล 30
3.4.2.1 การจัดทำข้อมูลเชิงพื้นที่ 31
3.4.2.2 การจัดทำข้อมูลอรรถาธิบาย 34
3.5 การสร้างโครงข่ายรถโดยสารประจำทางด้วยระบบฐานข้อมูล Oracle Spatial 37
3.5.1 การนำเข้าข้อมูล Shapefile เพื่อจัดเก็บในระบบฐานข้อมูล 37
3.5.2 การสร้างโครงข่ายด้วยระบบฐานข้อมูล Oracle Spatial 39
3.5.3 การเพิ่มข้อมูลจากตารางของ Shapefile ไปยังตารางของโครงข่าย 40

3.5.4 การปรับปรุงข้อมูลในคอลัมน์ COST ของตารางที่จัดเก็บข้อมูล link ของโครงข่าย 40
3.6 การจัดทำตารางข้อมูลอื่นๆ ที่ต้องใช้ในงานวิจัย 42
บทที่ 4 การออกแบบและพัฒนาโปรแกรมประยุกต์ 43
4.1 การวิเคราะห์ความต้องการของระบบ 43
4.2 สถาปัตยกรรมของโปรแกรมประยุกต์ 43
4.3 การพัฒนาโปรแกรมประยุกต์ภูมิสารสนเทศออนไลน์สำหรับการวางแผนการเดินทางด้วยรถโดยสารประจำทาง 46
4.3.1 การพัฒนาโปรแกรมในส่วนของการแสดงผลแผนที่ของข้อมูลภูมิศาสตร์ 46
4.3.2 การพัฒนาโปรแกรมในส่วนของการกำหนดจุดเริ่มต้นการเดินทาง จุดสิ้นสุดการเดินทาง และเวลาที่เริ่มต้นออกเดินทาง 48
4.3.3 การพัฒนาโปรแกรมเพื่อค้นหาคำตอบจากวิเคราะห์โครงข่าย โดยใช้เวลาในการเดินทางน้อยที่สุด 51
4.3.4 การพัฒนาโปรแกรมในส่วนของการค้นหาหมายเลขสายรถที่ควรโดยสาร 54
4.3.5 การพัฒนาโปรแกรมในส่วนของการแสดงผลลัพธ์แผนที่ และรายงานผลเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุด 56
บทที่ 5 การทดสอบโปรแกรมประยุกต์ 58
5.1 การทดสอบการทำงานในส่วนของการแสดงผลแผนที่ 58
5.2 การทดสอบการทำงานในส่วนของการกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง 61
5.3 การทดสอบการทำงานในส่วนของวิเคราะห์โครงข่ายรถโดยสารประจำทางด้วยระบบฐานข้อมูล ORACLE SPATIAL 65
5.3.1 การทดสอบประสิทธิภาพของฟังก์ชั่นการวิเคราะห์โครงข่ายที่พัฒนาเพิ่มเติมขึ้น 65
5.3.1.1 กรณีกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทางให้อยู่ใกล้กัน 66
5.3.1.1.1 การวิเคราะห์โครงข่ายโดยใช้ค่า Cost จากตารางหลักของโครงข่าย 66
5.3.1.1.2 การวิเคราะห์โครงข่ายโดยใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) ร่วมกับข้อมูลในตารางเสริม (อัตราเร็ว) ของโครงข่าย 67

5.3.1.2 กรณีกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทางให้อยู่ไกลกัน..... 68
5.3.1.2.1 การวิเคราะห์โครงข่ายโดยใช้ค่า Cost จากตารางหลักของโครงข่าย..... 68
5.3.1.2.2 การวิเคราะห์โครงข่ายโดยใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) ร่วมกับข้อมูลในตารางเสริม (อัตราเร็ว) ของโครงข่าย..... 70
5.3.2 อภิปรายผลการทดสอบประสิทธิภาพของฟังก์ชั่นการวิเคราะห์โครงข่าย..... 71
5.3.3 การทดสอบการทำงานของส่วนการวิเคราะห์โครงข่ายรถโดยสารประจำทาง 72
5.3.3.1 การทดสอบฟังก์ชั่นของการวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุดและคำนึงถึงปัจจัยของเวลาในการเดินทางที่แตกต่างกันในแต่ละช่วงวัน..... 72
5.3.3.2 อภิปรายผลการทดสอบฟังก์ชั่นของการวิเคราะห์โครงข่ายที่พัฒนาขึ้นเพิ่มเติม..... 74
5.3.3.3 การทดสอบการทำงานของการวิเคราะห์โครงข่ายกรณีต้องมีการโดยสารรถประจำทางมากกว่า 1 สาย..... 76
5.3.3.4 อภิปรายผลการทดสอบการทำงานของการวิเคราะห์โครงข่ายกรณีต้องมีการโดยสารรถประจำทางมากกว่า 1 สาย..... 77
5.4 การทดสอบการทำงานในส่วนของการวิเคราะห์หาหมายเลขสายรถประจำทาง..... 78
บทที่ 6 สรุปผลการศึกษา..... 81
6.1 สรุปผลการวิจัย..... 81
6.2 อภิปรายผลการวิจัย..... 82
รายการอ้างอิง..... 85
ภาคผนวก..... 87
ภาคผนวก ก การนำเข้าข้อมูลภูมิศาสตร์ (SHAPEFILE) จัดเก็บในระบบฐานข้อมูล ORACLE..... 88
ภาคผนวก ข การเชื่อมต่อ GEOSERVER กับระบบฐานข้อมูล ORACLE SPATIAL..... 94
ประวัติผู้เขียนวิทยานิพนธ์..... 100

สารบัญตาราง

หน้า

ตารางที่ 2.1 ข้อมูลการเดินทางจากจุดต้นทาง-ปลายทาง ระหว่างเส้นทางรถโดยสารประจำทาง..... 5

ตารางที่ 2.2 ปริมาณผู้โดยสารที่เดินทางด้วยระบบขนส่งสาธารณะประเภทต่างๆ ที่ปี พ.ศ. 2551..... 6

ตารางที่ 2.3 ผลการวิเคราะห์ทางสถิติเกี่ยวกับลักษณะการเดินทางของกลุ่มตัวอย่าง... 7

ตารางที่ 2.4 ผลการวิเคราะห์ทางสถิติเกี่ยวกับทางเลือกในการเดินทางของกลุ่มตัวอย่าง..... 7

ตารางที่ 2.2 โครงสร้างข้อมูลของตาราง NODE..... 11

ตารางที่ 2.2 โครงสร้างข้อมูลของตาราง LINK..... 12

ตารางที่ 2.3 รหัสของ node ที่ใช้แทนป้าย สถานี หรือท่าเรือ สำหรับระบบขนส่งมวลชนแต่ละประเภท..... 19

ตารางที่ 3.1 โครงสร้างข้อมูลของตาราง BUS_LINKS..... 25

ตารางที่ 3.2 โครงสร้างข้อมูลของตาราง BUS_NODES..... 27

ตารางที่ 3.3 การออกแบบรหัสประจำตัวให้กับข้อมูล link และ node 28

ตารางที่ 3.4 ค่าอัตราเร็วของรูปแบบการเดินทางด้วยเท้าบนสะพานลอย, ทางม้าลายและทางเดินเท้า..... 29

ตารางที่ 3.5 ค่าอัตราเร็วของรูปแบบการเดินทางบนถนน 29

ตารางที่ 3.6 โครงสร้างข้อมูลของตาราง BUS..... 30

ตารางที่ 4.1 ตัวอย่างค่า index ของทุกๆ link จากตาราง LINKS_SPEED ที่แสดงถึงช่วงเวลาต่างๆ ของวัน..... 52

ตารางที่ 4.2 จำนวน link ที่เดินทางไปได้ของแต่ละหมายเลขสายรถ..... 55

ตารางที่ 5.1 แสดงเวลาที่ใช้ในการประมวลผลของส่วนวิเคราะห์โครงข่ายจากการเลือกใช้ค่า Cost จากตารางที่แตกต่างกัน..... 72

ตารางที่ 5.2 แสดงเวลาที่ใช้ในการเดินทางผ่าน link ที่ประกอบกันเป็นเส้นทางผลลัพธ์ 76

ตารางที่ 5.3 แสดงรหัสประจำตัวของ link ที่ประกอบกันเป็นเส้นทางผลลัพธ์ และหมายเลขสายรถที่ผ่าน..... 80

สารบัญภาพ

		หน้า
ภาพที่ 1.1	แสดงขอบเขตบริเวณพื้นที่ศึกษา……	2
ภาพที่ 2.1	รายได้เฉลี่ยของผู้ที่ใช้รถโดยสารประจำทางที่ได้จากการสำรวจจุดต้นทาง – ปลายทาง……..	8
ภาพที่ 2.2	ตัวอย่างของ Graph ในการประยุกต์ใช้ในการจำลองข้อมูลโครงข่ายถนน…….	10
ภาพที่ 2.3	Oracle Network Data Model……..	11
ภาพที่ 2.4	คำสั่งในการสร้างโครงข่าย……...	12
ภาพที่ 2.5	ตัวอย่างฟังก์ชั่นสำหรับระบุค่า Cost ในขั้นตอนวิธีของ Dijkstra…….	13
ภาพที่ 2.6	ผังแสดงขั้นตอนการทำงานของ Dijkstra's Algorithm……...	14
ภาพที่ 2.7	แสดงส่วนประกอบเบื้องต้นของระบบ Web-GIS……	15
ภาพที่ 2.8	โปรแกรมประยุกต์ภูมิสารสนเทศบนเว็บที่ใช้ OpenLayers ช่วยในการพัฒนา..	16
ภาพที่ 2.9	การทำงานของโปรแกรมประยุกต์บนเว็บแบบดั้งเดิม เปรียบเทียบกับการทำงานด้วยเทคนิค AJAX……	17
ภาพที่ 2.10	ตัวอย่างของข้อมูลแบบ JSON……	18
ภาพที่ 3.1	การออกแบบแบบจำลองข้อมูลเพื่ออธิบายพฤติกรรมการเดินทางและความสัมพันธ์เชิงพื้นที่ภายในโครงข่ายรถโดยสารประจำทาง……	23
ภาพที่ 3.2	การออกแบบความสัมพันธ์ของ link แบบทิศทางเดียว (Direct) และแบบไปและกลับ (Bi-Direct)……..	23
ภาพที่ 3.3	แสดงตัวแทนของจุดเชื่อมต่อการเดินทางในโครงข่ายการเดินทางด้วยรถโดยสารประจำทาง……	24
ภาพที่ 3.4	การสร้างเส้นทางเดินรถโดยสารเพียงเส้นเดียว ทำให้การเดินทางไม่สอดคล้องกับพฤติกรรมจริง……	32
ภาพที่ 3.5	การสร้างเส้นทางเดินรถแยกฝั่ง ทำให้การเดินทางในโครงข่ายสอดคล้องกับพฤติกรรมจริง……	32
ภาพที่ 3.6	ข้อมูลที่เป็นตัวแทนเส้นทางการเดินทางด้วยเท้ารูปแบบต่างๆ……	33
ภาพที่ 3.7	จุดเชื่อมต่อการเดินทางที่ออกแบบให้แทนด้วย node……	33
ภาพที่ 3.8	ตัวอย่างข้อมูลอรรถาธิบายของ link……	35
ภาพที่ 3.9	ตัวอย่างข้อมูลอรรถาธิบายของ node……	36

หน้า

ภาพที่ 3.10 ข้อมูล Shapefile ที่จัดทำขึ้น..... 36
ภาพที่ 3.11 ขั้นตอนในการสร้างโครงข่ายรถโดยสารประจำทางด้วยระบบฐานข้อมูล Oracle Spatial..... 37
ภาพที่ 3.12 ส่วนติดต่อกับผู้ใช้ของโปรแกรม Mapbuilder..... 38
ภาพที่ 3.13 ตัวอย่างข้อมูลของตาราง TRANSIT_LINK..... 38
ภาพที่ 3.14 ตัวอย่างข้อมูลของตาราง TRANSIT_NODE..... 39
ภาพที่ 3.15 ผลลัพธ์จากการสร้างโครงข่าย..... 39
ภาพที่ 3.16 คำสั่งในการเพิ่มข้อมูลไปยังตาราง BUS_LINKS..... 40
ภาพที่ 3.17 คำสั่งในการเพิ่มข้อมูลไปยังตาราง BUS_NODES..... 40
ภาพที่ 3.18 คำสั่งในการคำนวณความยาวของ link ในหน่วยเมตร..... 41
ภาพที่ 3.19 ตัวอย่างข้อมูลโครงข่ายที่จัดเก็บข้อมูล link ของตาราง BUS_LINKS..... 41
ภาพที่ 3.20 ตัวอย่างข้อมูลโครงข่ายที่จัดเก็บข้อมูล node ของตาราง BUS_NODES..... 41
ภาพที่ 3.21 ตัวอย่างข้อมูลของตาราง BUS..... 42
ภาพที่ 3.22 ตัวอย่างข้อมูลของตาราง LINKS_SPEED..... 42
ภาพที่ 4.1 สถาปัตยกรรมของโปรแกรมประยุกต์..... 45
ภาพที่ 4.2 แสดงขั้นตอนในการพัฒนาส่วนต่างๆ ของโปรแกรมประยุกต์สำหรับการวางแผนการเดินทางด้วยรถโดยสารประจำทาง..... 46
ภาพที่ 4.3 ตัวอย่างคำสั่งภาษา Javascript ในการร้องขอภาพแผนที่จากแม่ข่ายแผนที่ Geoserver และผู้ให้บริการภาพแผนที่จาก Google..... 47
ภาพที่ 4.4 แสดงหน้าจอส่วนของการแสดงผลแผนที่ และตำแหน่งของเครื่องมือที่ใช้ในการเลื่อน และย่อ/ขยายแผนที่..... 48
ภาพที่ 4.5 คำสั่งในการสืบค้นหาจุดที่ใกล้ที่สุดจากจุดที่ทราบค่าพิกัด..... 49
ภาพที่ 4.6 ตัวอย่างข้อมูลในรูปแบบของ JSON ที่ตอบกลับมาจาก Server..... 49
ภาพที่ 4.7 ขั้นตอนการทำงานของโปรแกรมการกำหนดจุดเริ่มต้น/สิ้นสุด ในการเดินทาง.. 50
ภาพที่ 4.8 ส่วนแสดงค่ารหัสประจำตัวของ node ที่เป็นจุดเริ่มต้น/สิ้นสุดการเดินทาง..... 50
ภาพที่ 4.9 ส่วนรับค่าจากผู้ใช้ (รหัสประจำตัวของจุดเริ่มต้น/สิ้นสุดการเดินทาง และเวลาในการเริ่มต้นเดินทาง)..... 51

หน้า

ภาพที่ 4.10	ผังการทำงานของฟังก์ชั่นที่พัฒนาเพิ่มเติมสำหรับการวิเคราะห์โครงข่ายด้วยขั้นตอนวิธีของ Dijkstra……	53
ภาพที่ 4.11	ผังการทำงานของโปรแกรมในการค้นหาหมายเลขสายรถประจำทางที่ต้องโดยสาร……..……...	54
ภาพที่ 4.12	ตัวอย่างของเส้นทางการเดินทางและหมายเลขสายรถที่ผ่าน……	55
ภาพที่ 4.13	ตัวอย่างของผลลัพธ์การวิเคราะห์โครงข่ายที่ตอบกลับมาจาก Server ด้วยรูปแบบข้อมูลแบบ JSON……	56
ภาพที่ 4.14	แสดงองค์ประกอบของหน้าจอตอบโต้กับผู้ใช้……	57
ภาพที่ 5.1	แสดงการทดสอบส่วนการแสดงแผนที่ของโปรแกรมประยุกต์ ขณะเริ่มต้นทำงาน……..	58
ภาพที่ 5.2	แสดงการทดสอบส่วนควบคุมการทำงานของแผนที่โดยการใช้ Control ของ OpenLayers……..……..	59
ภาพที่ 5.3	แสดงการทดสอบส่วนควบคุมการทำงานของแผนที่โดยการใช้เม้าส์……..	60
ภาพที่ 5.4	แสดงการทดสอบส่วนควบคุมการทำงานของแผนที่ในการเปิด/ปิดการแสดงผลชั้นข้อมูล……..……..	61
ภาพที่ 5.5	แสดงบริเวณที่ต้องการให้เป็นจุดเริ่มต้นการเดินทาง คือ พื้นที่ภายในมหาวิทยาลัยธรรมศาสตร์ ท่าพระจันทร์……..	62
ภาพที่ 5.6	ผลลัพธ์จากการกำหนดตำแหน่งเพื่อระบุจุดเริ่มต้นการเดินทาง……	62
ภาพที่ 5.7	เวลาที่ใช้ในการประมวลผลลัพธ์ของการกำหนดตำแหน่งที่เป็นจุดเริ่มต้นการเดินทาง……..……...	63
ภาพที่ 5.8	ผลลัพธ์ของการยกเลิกการกำหนดจุดเริ่มต้นการเดินทางโดยการคลิกที่ปุ่ม Reset……	63
ภาพที่ 5.9	แสดงบริเวณที่ต้องการให้เป็นจุดสิ้นสุดการเดินทางคือ ภายในจุฬาลงกรณ์มหาวิทยาลัย……	64
ภาพที่ 5.10	ผลลัพธ์จากการกำหนดตำแหน่งเพื่อระบุจุดสิ้นสุดการเดินทาง……..	64
ภาพที่ 5.11	เวลาที่ใช้ในการประมวลผลลัพธ์ของการกำหนดตำแหน่งที่เป็นจุดสิ้นสุดการเดินทาง……	65

หน้า

ภาพที่ 5.12 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่อยู่ใกล้กัน เพื่อทดสอบการวิเคราะห์โครงข่ายโดยใช้ค่า Cost จากตารางหลัก (ระยะทาง) ของโครงข่าย..... 66
ภาพที่ 5.13 ผลลัพธ์การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้ระยะทางสั้นที่สุดจากจุดที่อยู่ใกล้กัน 66
ภาพที่ 5.14 เวลาที่ใช้ในการประมวลผลจากการวิเคราะห์โครงข่ายที่ใช้ค่า Cost จากตารางหลักของโครงข่าย และจุดเริ่มต้น/สิ้นสุดการเดินทางอยู่ใกล้กัน.....……... 67
ภาพที่ 5.15 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่อยู่ใกล้กัน เพื่อทดสอบการวิเคราะห์โครงข่ายโดยใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) และตารางเสริม (อัตราเร็ว)..... 67
ภาพที่ 5.16 ผลลัพธ์การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาน้อยที่สุดจากจุดที่อยู่ใกล้กัน 68
ภาพที่ 5.17 เวลาที่ใช้ในการประมวลผลจากการวิเคราะห์โครงข่ายที่ใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) และตารางเสริม (อัตราเร็ว) และจุดเริ่มต้น/สิ้นสุดการเดินทางอยู่ใกล้กัน..... 68
ภาพที่ 5.18 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่อยู่ไกลกัน เพื่อทดสอบการวิเคราะห์โครงข่ายโดยใช้ค่า Cost จากตารางหลัก (ระยะทาง) ของโครงข่าย..... 69
ภาพที่ 5.19 ผลลัพธ์การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้ระยะทางสั้นที่สุดจากจุดที่อยู่ไกลกัน..... 69
ภาพที่ 5.20 เวลาที่ใช้ในการประมวลผลจากการวิเคราะห์โครงข่ายที่ใช้ค่า Cost จากตารางหลักของโครงข่าย และจุดเริ่มต้น/สิ้นสุดการเดินทางอยู่ไกลกัน.....……... 70
ภาพที่ 5.21 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่อยู่ไกลกัน เพื่อทดสอบการวิเคราะห์โครงข่ายโดยใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) และตารางเสริม (อัตราเร็ว)..... 70
ภาพที่ 5.22 ผลลัพธ์การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาน้อยที่สุดจากจุดที่อยู่ไกลกัน.....……... 71
ภาพที่ 5.23 เวลาที่ใช้ในการประมวลผลจากการวิเคราะห์โครงข่ายที่ใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) และตารางเสริม (อัตราเร็ว) และจุดเริ่มต้น/สิ้นสุดการเดินทางอยู่ไกลกัน..... 71

หน้า

ภาพที่ 5.24 แสดงจุดเริ่มต้นและจุดสิ้นสุดการเดินทางในการทดสอบฟังก์ชั่นในการวิเคราะห์โครงข่ายที่พัฒนาเพิ่มขึ้น..... 73
ภาพที่ 5.25 ผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อทดสอบฟังก์ชั่นที่พัฒนาขึ้น โดยเวลาเริ่มต้นเดินทางคือ 7.00 น..... 73
ภาพที่ 5.26 ผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อทดสอบฟังก์ชั่นที่พัฒนาขึ้น โดยเวลาเริ่มต้นเดินทางคือ 11.00 น..... 74
ภาพที่ 5.27 แสดงภาพขยายของผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อทดสอบฟังก์ชั่นที่พัฒนาขึ้น โดยเวลาเริ่มต้นเดินทางคือ 7.00 น..... 75
ภาพที่ 5.28 แสดงผลลัพธ์ของการวิเคราะห์โครงข่ายกรณีที่ต้องมีการโดยสารรถประจำทางมากกว่า 1 สาย..... 77
ภาพที่ 5.29 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางในการทดสอบการวิเคราะห์หาหมายเลขสายรถประจำทางที่ต้องโดยสาร..... 78
ภาพที่ 5.30 ผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อทดสอบฟังก์ชั่นการหาหมายเลขสายรถที่ต้องโดยสาร..... 79
ภาพที่ 5.31 ภาพขยายของตำแหน่งที่ต้องมีการเปลี่ยนสายรถในการเดินทาง..... 79

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การเดินทางถือเป็นกิจวัตรประจำวันที่สำคัญของมนุษย์ ในแต่ละวันประชาชนมากกว่า 1 ล้านคน (องค์การขนส่งมวลชนกรุงเทพ, 2555) สัญจรไปมาตามท้องถนนของกรุงเทพมหานครโดยอาศัยรถประจำทางขององค์การขนส่งมวลชนกรุงเทพ (ขสมก.) ที่ให้บริการเดินรถจำนวน 214 เส้นทาง การเลือกเส้นทางการเดินทางต้องเลือกตามสายทางการเดินรถที่มีอยู่ แหล่งข้อมูลที่ใช้ประกอบการตัดสินใจเลือกเส้นทาง เช่น บริการโทรศัพท์สอบถามสายรถจากศูนย์บริการสอบถามเส้นทางของขสมก. แต่ก็ยังมีข้อจำกัดบางประการ เช่น จะทราบเพียงว่าต้องขึ้นรถประจำทางสายใด ลงหรือต่อรถที่จุดใด ไม่เห็นเส้นทางการเดินทางทั้งหมด หรือหากเป็นช่วงเวลาเร่งด่วนถนนบางเส้นก็จะมีการจราจรที่ติดขัดทำให้ต้องใช้เวลาในการเดินทางผ่านถนนเส้นนั้นนานขึ้น ซึ่งหากเรามีข้อมูลสภาพการจราจรในแต่ละเส้นทางและช่วงเวลา ก็จะสามารถให้บริการข้อมูลเพื่อใช้ในการวางแผนการเดินทางโดยรถโดยสารประจำทางเพื่อหลีกเลี่ยงปัญหาการจราจรที่ติดขัดได้ดียิ่งขึ้น

ที่ผ่านมามีหลายหน่วยงานที่ทำการศึกษาและพัฒนาโครงการต่างๆ หลายโครงการเพื่อปรับปรุงและพัฒนาการให้บริการรถโดยสารประจำทางให้มีประสิทธิภาพสูงขึ้น เช่น โครงการพัฒนาระบบรถโดยสารประจำทางในเขตกรุงเทพมหานครและปริมณฑล ของสำนักงานนโยบายและแผนการขนส่งและจราจร พ.ศ.2551 ที่มีวัตถุประสงค์เพื่อปรับปรุงและพัฒนาโครงข่ายเส้นทางรถโดยสารประจำทาง และดำเนินโครงการนำร่องเพื่อทดลองปฏิบัติจริงที่เขตการเดินรถที่ 5 ของขสมก. ซึ่งได้มีการจัดทำระบบฐานข้อมูลภูมิสารสนเทศภูมิศาสตร์ของเส้นทางรถโดยสารและข้อมูลอื่นๆ เช่น ตารางเวลาเดินรถ ข้อมูลการเดินรถ ฯลฯ ครบในระดับหนึ่งแล้ว การมีระบบแนะนำเส้นทางและสายรถประจำทาง จะช่วยอำนวยความสะดวกให้กับผู้ที่ต้องการสืบค้นหาสายรถประจำทางในการเดินทาง ไม่ว่าจะเป็นนักเรียน คนทำงาน หรือคนที่มาจากต่างจังหวัด

ด้วยเหตุนี้ จึงเป็นที่มาของแนวความคิดที่จะศึกษาออกแบบแบบจำลองฐานข้อมูลโครงข่ายรถประจำทาง และพัฒนาโปรแกรมประยุกต์บนเว็บที่ช่วยในการตัดสินใจเลือกเส้นทางการเดินทางด้วยรถโดยสารประจำทาง โดยพิจารณาจากเวลาในการเดินทางน้อยที่สุดโดยคำนึงถึงเงื่อนไขของระยะเวลาในการเดินทางที่แตกต่างกันในแต่ละช่วงวัน เพื่อเป็นแหล่งข้อมูลหนึ่งที่สนับสนุนการตัดสินใจวางแผนการเดินทางให้กับผู้ใช้บริการรถโดยสารประจำทาง โดยใช้ระบบการจัดการฐานข้อมูล Oracle Spatial ทำหน้าที่ในการบริหารจัดเก็บ จัดการและวิเคราะห์ข้อมูลโครงข่ายรถโดยสารประจำทาง

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อศึกษาและออกแบบแบบจำลองและโครงสร้างฐานข้อมูล เพื่อสนับสนุนการบริหารจัดการโครงข่ายเส้นทางเดินรถประจำทางด้วยระบบฐานข้อมูล Oracle Spatial

1.2.2 เพื่อพัฒนาโปรแกรมประยุกต์ GIS บนเว็บสำหรับค้นหาและแนะนำเส้นทางการเดินทางด้วยรถโดยสารประจำทาง

1.3 ขอบเขตของการวิจัย

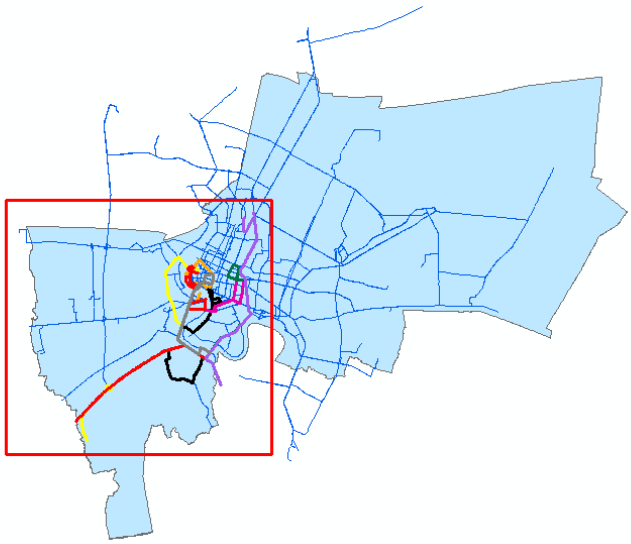
1.3.1 ขอบเขตเนื้อหาที่ศึกษา

1.3.1.1 ศึกษาและรวบรวมข้อมูลการเดินทางด้วยรถโดยสารประจำทาง ในที่นี้หมายถึงรถโดยสารประจำทางขนาดใหญ่ที่ให้บริการโดยขสมก. ไม่รวมรถโดยสารขนาดเล็ก (minibus) และรถตู้

1.3.1.2 การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่เหมาะสมใช้เงื่อนไขในการพิจารณาคือหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุด โดยอาศัยข้อมูลอัตราเร็วในการเดินทางของรถโดยสารประจำทางที่แตกต่างกันตามช่วงเวลาของวัน

1.3.2 ขอบเขตพื้นที่ศึกษา

ครอบคลุมพื้นที่ให้บริการของเขตการเดินรถที่ 5 ขององค์การขนส่งมวลขนกรุงเทพ ประกอบด้วยรถโดยสารประจำทาง จำนวน 13 เส้นทาง คือ สาย 20, 82, 138, 15, 21, 37, 75, 68, 76, 105, 111, 140 และ 141 มีจำนวนรถประจำการ 418 คัน (สำนักงานนโยบายและแผนการขนส่งและจราจร, 2552) ขอบเขตพื้นที่ศึกษาแสดงในภาพที่ 1.1



ภาพที่ 1.1 แสดงขอบเขตบริเวณพื้นที่ศึกษา

1.3.3 ข้อมูลที่ใช้ในงานวิจัย

ข้อมูลเส้นทางการเดินรถของรถโดยสารประจำทาง ได้จากการสืบค้นจากแหล่งข้อมูลที่เผยแพร่ต่อสาธารณะ คือ เว็บไซต์ของ ขสมก. ส่วนตำแหน่งป้ายรถโดยสารประจำทางอ้างอิงจากข้อมูลตำแหน่งป้ายรถโดยสารประจำทางของกระทรวงคมนาคม ข้อมูลดังกล่าวใช้ในการเป็นตำแหน่งอ้างอิงสำหรับจัดทำข้อมูลภูมิศาสตร์ของการเดินทางด้วยรถโดยสารประจำทางขึ้นใหม่เพื่อใช้ในงานวิจัย

1.4 ข้อตกลงเบื้องต้น

1.4.1 อัตราเร็วในการเดินทางของรถโดยสารประจำทางในช่วงเวลาต่างๆ ของวัน ผู้วิจัยได้สมมุติขึ้นให้อยู่ในช่วง 8-20 กิโลเมตรต่อชั่วโมง

1.4.2 งานวิจัยนี้กำหนดให้ระยะเวลาในการรอรถโดยสารเท่ากับ 15 นาที ทุกช่วงเวลา และรถโดยสารประจำทางทุกสายให้บริการตลอด 24 ชั่วโมง โดยจอดรับส่งผู้โดยสารทุกป้าย

1.5 ข้อจำกัดของการวิจัย

1.5.1 ตำแหน่งป้ายรถโดยสารประจำทาง, สะพานลอย, ทางม้าลาย, ทางเดินเท้าริมถนน และเส้นทางเดินรถโดยสารประจำทาง ผู้วิจัยได้จัดทำข้อมูลดังกล่าวขึ้นมาเอง เพื่อให้สอดคล้องกับความต้องการของงานวิจัยโดยอาศัยข้อมูลเส้นทางเดินรถโดยสารประจำทางของ ขสมก. ข้อมูลตำแหน่งป้ายรถโดยสารประจำทางของกระทรวงคมนาคม ร่วมกับภาพถ่ายดาวเทียมของ Google เป็นฐานอ้างอิง ไม่ได้ลงสนามสำรวจข้อมูลของตำแหน่งที่แท้จริง จึงไม่ควรนำไปใช้ในการอ้างอิงเพราะอาจมีความคลาดเคลื่อนเชิงตำแหน่ง

1.5.2 ปัจจัยที่ใช้ในการพิจารณาเลือกเส้นทางการเดินทางที่เหมาะสมมีหลายปัจจัย เช่น เวลาในการเดินทาง (Time Cost), ระยะทาง (Distance Cost) และค่าใช้จ่ายที่เป็นตัวเงิน (Toll Cost) ในงานวิจัยนี้เลือกใช้ปัจจัยด้านเวลาในการเดินทาง โดยอ้างอิงจากระยะทางผ่านอัตราเร็วของรถโดยสารที่สมมุติขึ้น

1.6 วิธีดำเนินการวิจัย

1.6.1 ศึกษาและรวบรวมแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

1.6.1.1 ทฤษฎีของ Graph

1.6.1.2 Dijkstra’s algorithm

1.6.1.3 Web Based GIS และเทคโนโลยีที่เกี่ยวข้อง

1.6.1.4 Oracle Spatial

1.6.1.5 งานวิจัยที่เกี่ยวข้องกับการประยุกต์ใช้ระบบสารสนเทศภูมิศาสตร์ ในการแนะนำเส้นทางการเดินทางด้วยระบบขนส่งมวลชน

1.6.2 กำหนดขอบเขตและวัตถุประสงค์ของงานวิจัย

1.6.3 รวบรวมข้อมูลจากหน่วยงานต่างๆ ที่เผยแพร่ข้อมูลด้านการคมนาคม เช่น ตำแหน่งป้ายรถโดยสารประจำทางจากกระทรวงคมนาคม, เส้นทางเดินรถโดยสารประจำทางจากเว็บไซต์ของ ขสมก.

1.6.4 ศึกษาและออกแบบแบบจำลองข้อมูล และโครงสร้างข้อมูลโครงข่ายการเดินทางด้วยรถโดยสารประจำทางให้เหมาะสมในการจัดเก็บและจัดการด้วยระบบฐานข้อมูล Oracle Spatial

1.6.5 จัดทำข้อมูลภูมิศาสตร์ของการเดินทางด้วยรถโดยสารประจำทาง โดยใช้โปรแกรม Quantum GIS (QGIS) จัดเก็บในรูปแบบของ Shapefile พร้อมทั้งจัดทำข้อมูลอรรถาธิบาย (Attribute) ให้สอดคล้องกับแบบจำลองและโครงสร้างข้อมูลที่ออกแบบไว้

1.6.6 นำเข้าข้อมูลภูมิศาสตร์ที่จัดทำขึ้นจัดเก็บในระบบฐานข้อมูลโดยให้มีโครงสร้างข้อมูลที่ถูกต้องและเหมาะสมกับ Oracle Network Data Model เพื่อให้สามารถใช้ฟังก์ชั่นในการวิเคราะห์โครงข่ายของ Oracle Spatial และจัดทำตารางข้อมูลที่ต้องใช้ในงานวิจัยเพิ่มเติม คือ ตารางจัดเก็บข้อมูลหมายเลขสายรถ และตารางจัดเก็บอัตราเร็วของรถโดยสารประจำทางที่เดินทางผ่านแต่ละช่วงถนน

1.6.7 พัฒนาโปรแกรมประยุกต์ที่ใช้ในการวิเคราะห์โครงข่ายเพื่อหาเส้นทางการเดินทางด้วยรถโดยสารประจำทางที่ใช้เวลาในการเดินทางน้อยที่สุด

1.6.7.1 พัฒนาในส่วนของการวิเคราะห์โครงข่ายบนเครื่องแม่ข่าย (Server)

1.6.7.2 พัฒนาในส่วนติดต่อกับผู้ใช้ (Client)

1.6.8 ทดสอบและปรับปรุงการทำงานของโปรแกรมประยุกต์

1.6.9 สรุปผลการวิจัย และจัดทำรายงาน เรียบเรียงวิทยานิพนธ์

1.7 ประโยชน์ที่คาดว่าจะได้รับ

สามารถออกแบบแบบจำลองข้อมูลเพื่อจำลองพฤติกรรมการเดินทางด้วยรถโดยสารประจำทาง ให้มีโครงสร้างข้อมูลที่เหมาะสมต่อการจัดเก็บ จัดการ และวิเคราะห์โครงข่ายด้วยระบบฐานข้อมูล Oracle Spatial และสามารถพัฒนาโปรแกรมประยุกต์ภูมิสารสนเทศออนไลน์เพื่อค้นหาเส้นทางการเดินทางที่เหมาะสมจากการโดยสารรถประจำทางได้

บทที่ 2
แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 สถานภาพของการให้บริการระบบรถโดยสารประจำทางในเขตกรุงเทพมหานครและปริมณฑล

สำนักงานนโยบายและแผนการขนส่งและจราจรได้ทำการศึกษาเพื่อปรับปรุงโครงข่ายเส้นทางเดินรถโดยสารประจำทางในเขตกรุงเทพมหานครและปริมณฑล และจัดทำโครงการนำร่อง (Pilot Project) เพื่อทดลองปฏิบัติงานจริงกับเส้นทางที่มีการปรับปรุงใหม่ โดยนำเอาเทคโนโลยีขนส่งอัจฉริยะ (Intelligent Transport Systems: ITS) ที่เหมาะสมมาประยุกต์ใช้ ในการศึกษาได้มีการศึกษาทบทวนผลการศึกษาและการดำเนินงานที่ผ่านมา รวมถึงมีการสำรวจและวิเคราะห์ข้อมูลด้านการจราจรและขนส่ง (สำนักงานนโยบายและแผนการขนส่งและจราจร, 2552) สรุปประเด็นสำคัญได้ดังนี้

- รถโดยสารประจำทางในเขตกรุงเทพมหานครและปริมณฑลที่ให้บริการโดยขสมก. ให้บริการเดินรถจำนวน 214 เส้นทาง มีระยะทางให้บริการรวมประมาณ 6,495 กิโลเมตร รถโดยสารที่ให้บริการประกอบด้วย รถธรรมดา รถปรับอากาศ และรถยูโร โดยเป็นรถโดยสารประจำทางของขสมก. จำนวน 3,509 คัน และรถร่วมบริการจำนวน 4,016 คัน รวมเป็น 7,525 คัน (องค์การขนส่งมวลชนกรุงเทพ, 2554)
- คู่ต้นทางและปลายทางที่มีการสุ่มเลือกทำการศึกษา มีระยะทางเดินทางยาวเฉลี่ยประมาณ 34.35 กิโลเมตร ใช้เวลาในการเดินทางเฉลี่ย 175.55 นาทีต่อเที่ยวการเดินทาง และมีจำนวนการต่อรถเฉลี่ย 1.73 ครั้งต่อเที่ยวการเดินทาง ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 ข้อมูลการเดินทางจากจุดต้นทาง-ปลายทาง ระหว่างเส้นทางรถโดยสารประจำทาง

ข้อมูลการเดินทาง	เส้นทางที่ให้บริการในปัจจุบัน (214 เส้นทาง)
	เฉลี่ย
จำนวนสายรถเมล์ที่ใช้	1.73
ระยะทางการเดินทาง (กม./เที่ยว)	34.35
ระยะเวลาการรอคอยรถโดยสาร (นาที/เที่ยว)	43.80
ระยะเวลาที่อยู่บนรถโดยสาร (นาที/เที่ยว)	131.75
ระยะเวลาการเดินทางรวม (นาที/เที่ยว)	175.55

- ปริมาณผู้โดยสารที่ใช้บริการรถโดยสารประจำทางหมวด 1 ที่เป็นรถโดยสารของขสมก. มีจำนวนประมาณ 1,694,466 คน-เที่ยวต่อวัน และรถร่วมบริการอีกประมาณ 1,649,749 คน-เที่ยวต่อวัน กรณีที่มีการปรับปรุงเส้นทางรถโดยสารประจำทางใหม่ (155 เส้นทาง) ปริมาณผู้โดยสารที่ใช้บริการรถโดยสารของขสมก. และรถร่วมบริการจะเพิ่มขึ้นเป็น 5,489,163 คน-เที่ยวต่อวัน ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 ปริมาณผู้โดยสารที่เดินทางด้วยระบบขนส่งสาธารณะประเภทต่างๆ ที่ปี พ.ศ. 2551

ประเภทของระบบขนส่งสาธารณะ	**ปริมาณผู้โดยสาร (คน-เที่ยว/วัน)**	
	214 เส้นทาง	**155 เส้นทาง**
รถ ขสมก. (หมวด 1)	1,694,466	5,489,163
รถร่วมบริการ (หมวด 1)	1,649,749	
รถในซอย	491,484	397,267
รถเมโทรบัส	305,723	207,911
รถโดยสารหมวด 3 และ 4	1,803,753	1,169,728
เรือโดยสาร	291,936	219,454
รถไฟชานเมือง	60,384	60,361
รถตู้มวลชน	1,164,276	623,524
รถไฟฟ้าสายสีเขียว	618,329	653,409
รถไฟฟ้าสายสีน้ำเงิน	162,755	182,992
รถไฟฟ้าสายสีแดง	-	-
รถไฟฟ้าสายสีม่วง	-	-
รวมทุกประเภท	8,242,855	9,003,809

หมายเหตุ : * เรือโดยสารทุกประเภท

** รถไฟฟ้าเฉลิมพระเกียรติ 6 รอบ พระชนมพรรษา สาย 1 (สุขุมวิท), สาย 2 (สีลม)

*** รถไฟฟ้าสายเฉลิมรัชมงคล (ช่วงบางซื่อ – หัวลำโพง)

- ลักษณะการเดินทางของผู้ที่ใช้รถโดยสารประจำทางพบว่า กลุ่มตัวอย่างใช้เวลาในการเดินทางรวมโดยเฉลี่ยประมาณ 43 – 50 นาที เวลาที่ใช้ในการรอรถโดยสารเฉลี่ยประมาณ 15 นาที ค่าโดยสารรวมของผู้ที่โดยสารรถประจำทางปรับอากาศประมาณ 23 บาท ส่วนรถไม่ปรับอากาศจะมีค่าโดยสารรวมประมาณ 12 บาท ดังแสดงในตารางที่ 2.3

ตารางที่ 2.3 ผลการวิเคราะห์ทางสถิติเกี่ยวกับลักษณะการเดินทางของกลุ่มตัวอย่าง

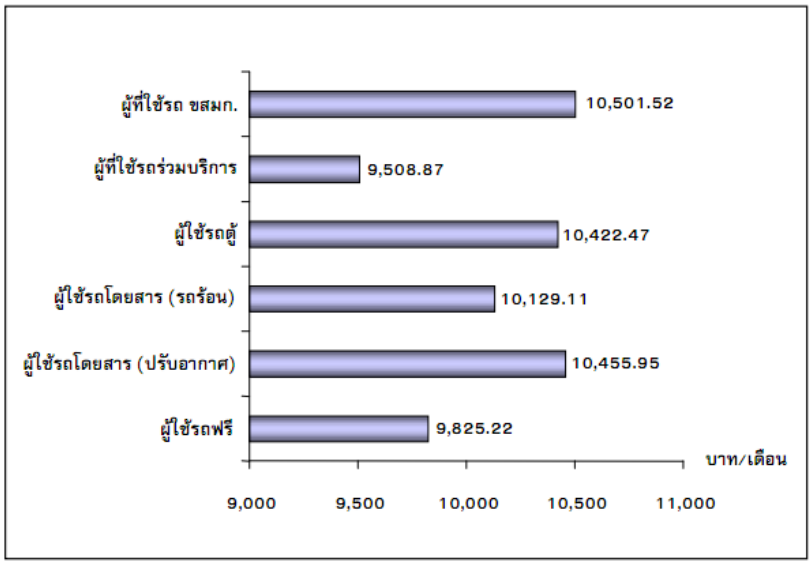
ตัวแปร	ค่าเฉลี่ย (ถ่วงน้ำหนัก)	ค่าเบี่ยงเบนมาตรฐาน	ค่าสูงสุด	ค่าต่ำสุด	ช่วงความเชื่อมั่นที่ 95%
ผู้ใช้รถโดยสารประจำทาง (ไม่เสียค่าใช้จ่าย)					
เวลาที่ใช้ในการเดินทางรวม (นาที)	43.64	24.07	240	5	41.87 – 45.42
เวลาที่ใช้ในการรอรถรวม (นาที)	14.76	7.79	70	0	14.19 – 15.34
ค่าโดยสารรวม (บาท)	3.66	7.10	10	0	3.14 – 4.19
จำนวนครั้งในการเปลี่ยน/ต่อรถ	1.35	0.50	3	1	1.31 – 1.38
ผู้ใช้รถโดยสารประจำทาง (ปรับอากาศ)					
เวลาที่ใช้ในการเดินทางรวม (นาที)	49.34	22.69	200	5	48.50 – 50.18
เวลาที่ใช้ในการรอรถรวม (นาที)	15.56	8.67	60	0	15.24 – 15.87
ค่าโดยสารรวม (บาท)	22.29	10.23	200	0	21.91 – 22.67
จำนวนครั้งในการเปลี่ยน/ต่อรถ	1.21	0.43	4	1	1.19 – 1.22
ผู้ใช้รถโดยสารประจำทาง (ไม่ปรับอากาศ)					
เวลาที่ใช้ในการเดินทางรวม (นาที)	43.46	21.96	250	5	43.02 – 43.89
เวลาที่ใช้ในการรอรถรวม (นาที)	12.40	5.70	60	0	12.30 – 12.53
ค่าโดยสารรวม (บาท)	11.21	8.50	100	7	10.68 – 10.90
จำนวนครั้งในการเปลี่ยน/ต่อรถ	1.20	0.42	4	1	1.19 – 1.21
ผู้ใช้รถตู้ประจำทาง					
เวลาที่ใช้ในการเดินทางรวม (นาที)	48.39	22.46	240	15	47.16 – 49.63
เวลาที่ใช้ในการรอรถรวม (นาที)	12.28	7.57	60	0	11.86 – 12.69
ค่าโดยสารรวม (บาท)	27.89	13.12	240	15	27.16 – 28.63
จำนวนครั้งในการเปลี่ยน/ต่อรถ	1.32	0.50	4	1	1.29 – 1.35

- ผลการวิเคราะห์ทางสถิติเกี่ยวกับลักษณะการเดินทางพบว่า กลุ่มตัวอย่างซึ่งเป็นผู้ที่ใช้รถโดยสารประจำทางมักไม่มีทางเลือกในการเดินทางมากนัก ร้อยละ 83 – 88 สามารถเลือกใช้รถโดยสารประจำทางในการเดินทางได้เท่านั้น ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 ผลการวิเคราะห์ทางสถิติเกี่ยวกับทางเลือกในการเดินทางของกลุ่มตัวอย่าง

ตัวแปร	สัดส่วนตัวอย่าง (ร้อยละ)			
	ผู้ใช้รถโดยสารธรรมดา (รถเมล์ฟรี)	ผู้ใช้รถโดยสารปรับอากาศ	ผู้ใช้รถโดยสารธรรมดา	ผู้ใช้รถตู้ประจำทาง
ทางเลือกในการเดินทาง				
รถประจำทางเท่านั้น	83.05	84.76	88.31	44.61
มีทางเลือกอื่นๆ	16.95	15.24	11.68	55.31

- ระดับรายได้ของผู้ใช้บริการรถโดยสารประจำทางของขสมก. จะมีรายได้เฉลี่ยประมาณ 10,501 บาทต่อเดือน ดังแสดงในภาพที่ 2.1



ภาพที่ 2.1 รายได้เฉลี่ยของผู้ที่ใช้รถโดยสารประจำทางที่ได้จากการสำรวจจุดต้นทาง - ปลายทาง

- ความเร็วเฉลี่ย (Average Travel Speed) ของรถโดยสารประจำทางที่ให้บริการในแต่ละเส้นทาง จะสมมุติให้ต่ำกว่าความเร็วของรถทั่วไปประมาณร้อยละ 30 เมื่อนำความเร็วเฉลี่ยของรถทั่วไปบนโครงข่ายถนนมาหารด้วย 1.30 จะได้ค่าความเร็วของรถโดยสารประจำทาง

- มีการจัดทำโครงการนำร่อง (Pilot Project) ณ เขตการเดินรถที่ 5 ของขสมก. ที่มีจำนวนรถประจำการของขสมก. 418 คัน (คิดเป็น 5 เปอร์เซ็นต์ของรถประจำทางของขสมก. และรถร่วมบริการทั้งหมด) เพื่อปรับปรุงระดับการให้บริการโดยมีการนำเอาโปรแกรมประยุกต์และระบบขนส่งอัจฉริยะ (Intelligent Transport Systems) ได้แก่ โปรแกรมจัดทำตารางเดินรถ โปรแกรมจัดทำตารางการทำงานของพนักงาน โปรแกรมจัดทำตารางการซ่อมบำรุงรถโดยสาร ระบบติดตามรถโดยสารประจำทางด้วย GPS และระบบการให้ข้อมูลข่าวสารแก่ผู้ใช้บริการแบบ Real time เป็นต้นมาทดลองดำเนินงานจริง เส้นทางที่เลือกมาดำเนินการเป็นโครงการนำร่อง มีจำนวน 4 สายทาง ได้แก่

- สาย 20 : ป้อมพระจุลฯ – ท่าดินแดง
- สาย 21 : วัดคู่สร้าง – จุฬาลงกรณ์มหาวิทยาลัย
- สาย 76 : แสมดำ – ประตูน้ำ
- สาย 105 : ตลาดมหาชัยเมืองใหม่ – คลองสาน

2.2 ทฤษฎี Graph

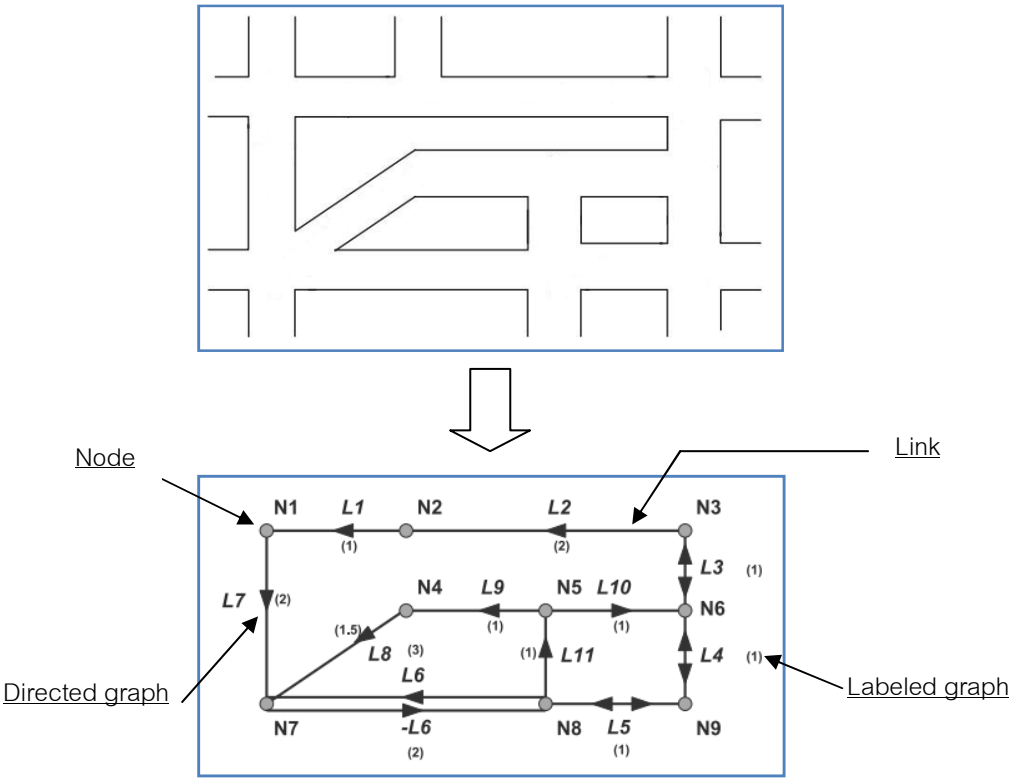
ทฤษฎีของ Graph เป็นสาขาวิชาหนึ่งของคณิตศาสตร์ ที่พูดถึงเรื่องของความสัมพันธ์หรือความเชื่อมต่อกันทางกายภาพของสมาชิกต่างๆ ภายในเซ็ตซึ่ง Graph มีคุณสมบัติดังต่อไปนี้

Graph (G) จะประกอบด้วยกลุ่มของจุดที่เรียกว่า nodes และกลุ่มของความสัมพันธ์ระหว่าง node ที่เรียกว่า links สามารถเขียนสัญลักษณ์แทนความสัมพันธ์ (e) ของความเชื่อมต่อกันทางกายภาพระหว่าง nodes 2 ตัว (x และ y) ที่ link นั้นเชื่อมโยงอยู่ได้ เช่น e = {x,y}

Graph สามารถนำมาออกแบบแบบจำลองข้อมูลเพื่อแสดงความสัมพันธ์เชิงตำแหน่ง และสามารถใช้ประโยชน์ของแบบจำลองข้อมูลของ Graph ได้มากขึ้นหากมีการเพิ่มส่วนขยายให้กับ Graph (Worboy, 2004) ดังตัวอย่างเช่น

- Directed graph คือ Graph ที่มีการระบุทิศทางของความสัมพันธ์ให้กับ link โดยส่วนมากจะใช้หัวลูกศรในการระบุทิศทาง
- Labeled graph คือ Graph ที่มีการระบุน้ำหนักของความสัมพันธ์ให้กับ link โดยมักระบุเป็นตัวเลขหรือตัวอักษรของค่าน้ำหนักในตำแหน่งที่ใกล้กับ link นั้นๆ

จากทฤษฎีของ Graph สามารถนำมาใช้จำลองข้อมูลโครงข่ายรถโดยสารประจำทางได้ โดยการใช้ Directed graph ในการจำลองทิศทางของการเคลื่อนที่ของรถโดยสารประจำทาง เช่น การเดินรถทางเดียว (one-way) และใช้ Labeled graph ในการแสดงค่าน้ำหนักของความสัมพันธ์ เช่น ระยะทางหรือระยะเวลาที่ใช้ในการเดินทางผ่านถนน ดังแสดงตัวอย่างของการประยุกต์ทฤษฎีของ Graph ดังแสดงในภาพที่ 2.2



ภาพที่ 2.2 ตัวอย่างของ Graph ในการประยุกต์ใช้ในการจำลองข้อมูลโครงข่ายถนน (Kothuri, et al.,2007)

2.3 Oracle Spatial

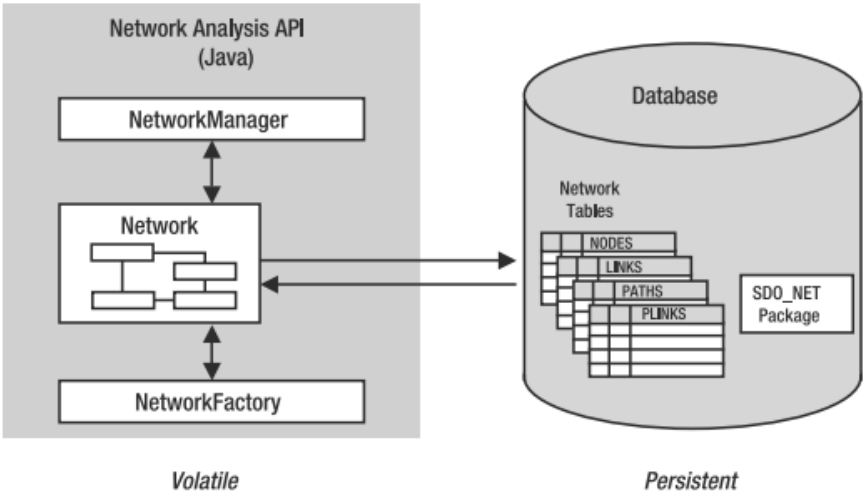
Oracle Database คือ ซอฟต์แวร์ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System:RDBMS) ที่ได้รับความนิยมและมีการใช้งานอย่างแพร่หลาย มีระบบที่ใช้ในการจัดเก็บและจัดการข้อมูลปริภูมิ ตั้งแต่เวอร์ชั่น 7.2 เรียกว่า Oracle MultiDimension จนถึงเวอร์ชั่น 8i จึงพัฒนาโครงสร้างข้อมูลที่ใช้จัดเก็บข้อมูลปริภูมิที่เรียกว่า SDO_GEOMETRY ซึ่งใช้มาจนถึงเวอร์ชั่นปัจจุบัน คือ เวอร์ชั่น 11g และยังมีรูปแบบการจัดเก็บข้อมูลและฟังก์ชั่นที่ใช้สำหรับวิเคราะห์ข้อมูลปริภูมิจำนวนข้างมาก เช่น Network Data Model, 3D Geometry, Spatial Web Service เป็นต้น

2.3.1 Oracle Network Data Model

Network data model เป็นแบบจำลองข้อมูลที่ใช้ในการจัดเก็บข้อมูลโครงข่ายปริภูมิของระบบฐานข้อมูล Oracle ประกอบด้วย 2 ส่วน คือ โครงสร้างข้อมูลของข้อมูลโครงข่าย (Network schema) และส่วนต่อประสานโปรแกรมประยุกต์ (Network APIs) ดังแสดงในภาพที่ 2.3

Network schema คือ รูปแบบโครงสร้างข้อมูลที่ใช้ในการจัดเก็บข้อมูลโครงข่ายลงในระบบฐานข้อมูล ซึ่งจำเป็นต้องมีอย่างน้อย 2 ตารางหลัก ได้แก่ ตาราง Link และ ตาราง Node

Network APIs ประกอบด้วย PL/SQL package สำหรับบริหารจัดการข้อมูลโครงข่ายในระบบฐานข้อมูล และ Java API สำหรับจัดการและวิเคราะห์โครงข่าย (Network Analysis)



ภาพที่ 2.3 Oracle Network Data Model (Kothuri, et al., 2007)

2.3.1.1 **โครงสร้างข้อมูล** (Network Tables)

การนิยามโครงข่ายจะใช้ตาราง 2 ตาราง คือตาราง node และตาราง link ซึ่งต้องมีโครงสร้างของตารางอย่างถูกต้อง และสามารถมีตาราง path และตาราง path link ซึ่งถือเป็นตารางเสริมสำหรับใช้ในการวิเคราะห์โครงข่ายด้วยก็ได้ โดยมีโครงสร้างข้อมูลของตาราง node และตาราง link ดังตารางที่ 2.5 และ 2.6

ตารางที่ 2.5 โครงสร้างข้อมูลของตาราง NODE

Column	**Data type**
NODE_ID*	NUMBER
geometry_column	SDO_GEOMETRY
cost_column	NUMBER
HIERARCHY_LEVEL	NUMBER
PARENT_NODE_ID	NUMBER
ACTIVE	CHAR(1)
NODE_NAME	VARCHAR2(32)
NODE_TYPE	VARCHAR2(24)

ตารางที่ 2.6 โครงสร้างข้อมูลของตาราง LINK

Column	**Data type**
LINK_ID*	NUMBER
START_NODE_ID*	NUMBER
END_NODE_ID*	NUMBER
geometry_column	SDO_GEOMETRY
cost_column	NUMBER
BIDIRECTED	CHAR(1)
PARENT_LINK_ID	NUMBER
ACTIVE	CHAR(1)
LINK_LEVEL	NUMBER
LINK_NAME	VARCHAR2
LINK_TYPE	VARCHAR2

2.3.1.2 **การสร้างโครงข่าย** (Defining Network)

การสร้างโครงข่ายจะใช้ฟังก์ชั่นในการบริหารจัดการโครงข่ายของ Oracle Spatial (SDO_NET package) โดยมีคำสั่งในการสร้างโครงข่ายดังแสดงในภาพที่ 2.4 คำสั่งดังกล่าวเป็นการสร้างตารางของโครงข่ายแบบอัตโนมัติโดยระบุชื่อของตารางที่เป็นส่วนประกอบของโครงข่าย ฟังก์ชั่นจะทำการสร้าง network metadata ให้เองจากชื่อตารางที่เราระบุ

```
BEGIN
 SDO_NET.CREATE_SDO_NETWORK (
   NETWORK              => 'TRANSIT_THESIS5',
   NO_OF_HIERARCHY_LEVELS   => 1,
   IS_DIRECTED          => TRUE,
   NODE_TABLE_NAME        => 'BUS_NODES',
   NODE_GEOM_COLUMN       => 'GEOM',
   NODE_COST_COLUMN       => NULL,
   LINK_TABLE_NAME        => 'BUS_LINKS',
   LINK_COST_COLUMN       => 'COST',
   LINK_GEOM_COLUMN       => 'GEOM',
   PATH_TABLE_NAME        => 'BUS_PATHS',
   PATH_GEOM_COLUMN       => 'GEOM',
   PATH_LINK_TABLE_NAME   => 'BUS_PLINKS'
 );
END;
/
```

ภาพที่ 2.4 คำสั่งในการสร้างโครงข่าย

2.3.2 Load-On-Demand Approach

ระบบฐานข้อมูล Oracle Spatial 11g มีวิธีการที่เรียกว่า Load On Demand ที่มีฟังก์ชั่นในการวิเคราะห์โครงข่ายที่หลากหลาย เช่น การหาเส้นทางที่มีค่าใช้จ่ายน้อยที่สุด (shortest

path), การหาตำแหน่งที่ใกล้กับจุดที่กำหนด (nearest neighbors) เป็นต้น และสามารถเพิ่มเงื่อนไขในการวิเคราะห์โครงข่ายได้ เช่น Multiple cost support (ORACLE, 2011) ที่กำหนดค่าน้ำหนักของความสัมพันธ์ (Cost) ให้กับ link ได้หลายค่า ทำให้ในการวิเคราะห์โครงข่ายสามารถเลือกค่า Cost มาใช้ในขั้นตอนวิธีของ Dijkstra ได้หลายค่า โดยการเขียนฟังก์ชั่นให้คืนค่า Cost เพื่อใช้ในการวิเคราะห์ โดยอาศัยคลาส LODAnalysisInfo ที่อยู่ในแพ็คเกจ oracle.spatial.network.lod ค่า Cost ที่ส่งคืนมาจากฟังก์ชั่นจะไปแทนที่ค่า Cost ตั้งต้นที่ระบุไว้ใน network metadata ตัวอย่างของฟังก์ชั่นดังแสดงในภาพที่ 2.5

```
public  class TravelTimeCalculator implements LinkCostCalculator
 {
  double SPEED_LIMIT_1 = 30;
  double SPEED_LIMIT_2 = 60;

  public double getLinkCost(LODAnalysisInfo analysisInfo)
  {
   LogicalLink link = analysisInfo.getNextLink();// find the link to be traversed
   int linkLevel= link.getLevel();
   double linkCost = link.getCost();// find the distance
   switch(linkLevel) // return the travel time as distance/speed limit
   {
    case 2:
     return linkCost/SPEED_LIMIT_2;
    default:
     return linkCost/SPEED_LIMIT_1;
   }
  }
  public int[] getUserDataCategories()
  {
   return null;
  }
 }
```

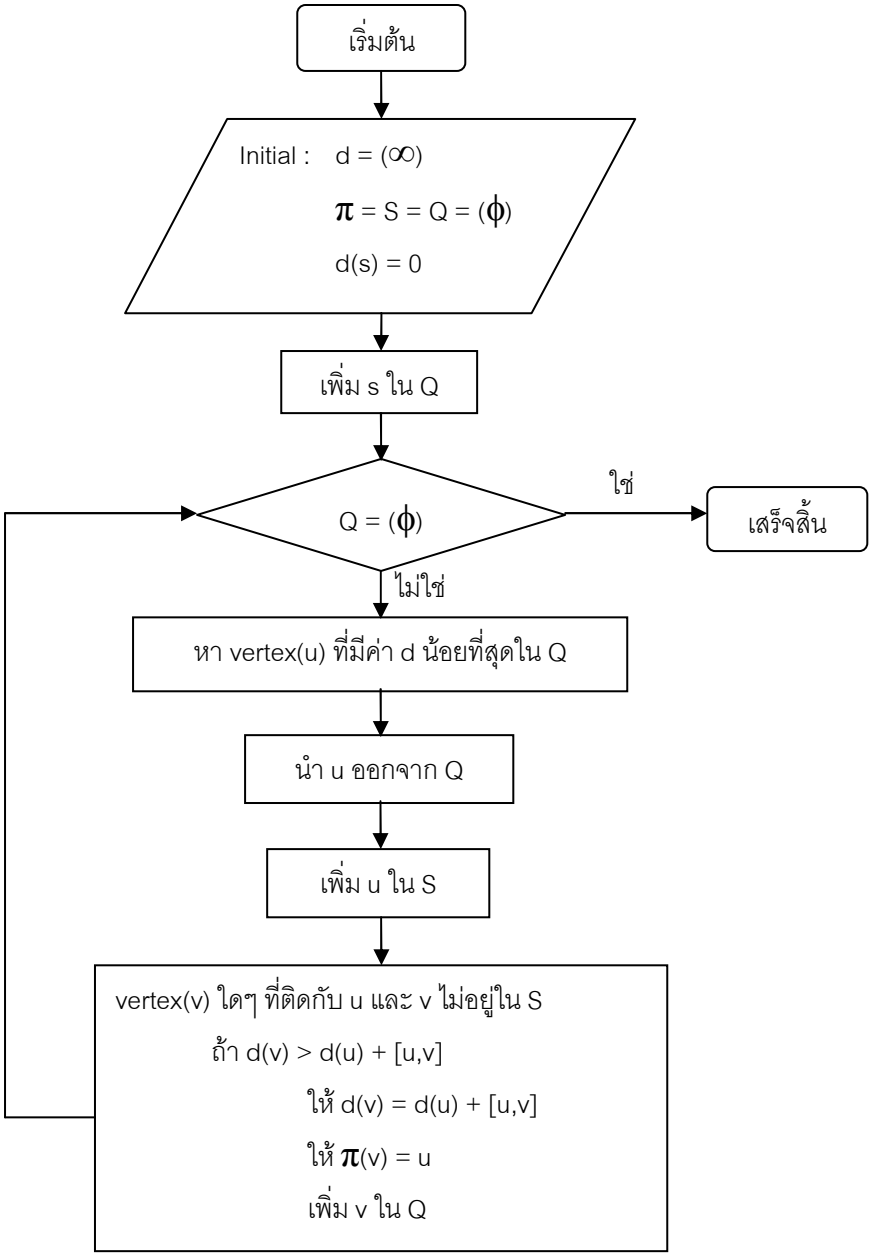
ภาพที่ 2.5 ตัวอย่างฟังก์ชั่นสำหรับระบุค่า Cost ในขั้นตอนวิธีของ Dijkstra (ORACLE, 2011)

2.4 **ทฤษฎีทางด้านการวิเคราะห์โครงข่าย** (Network Analysis)

เป็นขั้นตอนที่ใช้ในการคำนวณหาเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุด โดยใช้ขั้นตอนวิธีที่เรียกว่า Dijkstra's algorithm ที่มีประสิทธิภาพในการคำนวณหาเส้นทางระหว่าง node 2 จุด ที่มีค่าใช้จ่ายในการเดินทางน้อยที่สุด โดยข้อมูลโครงข่าย (Network) จะต้องมีการกำหนดทิศทางและน้ำหนักของความสัมพันธ์ต้องไม่น้อยกว่าศูนย์ (Chunithipaisan, 2009) ขั้นตอนวิธีของ Dijkstra สามารถเขียนเป็นผังการทำงานได้ดังภาพที่ 2.6

<u>กำหนดให้</u>

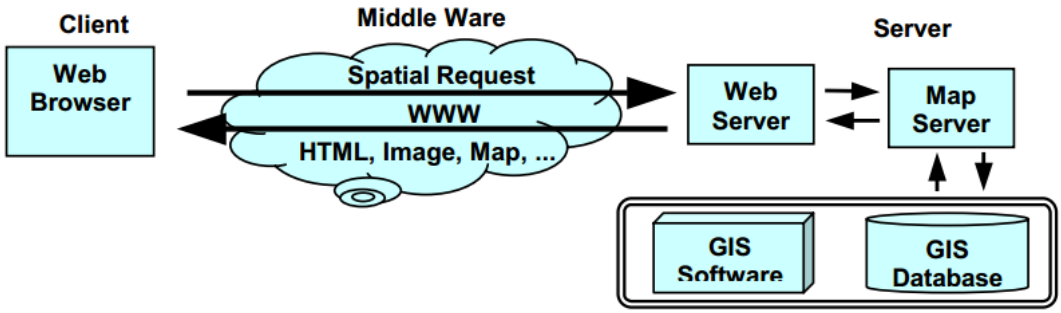
- d เก็บค่าใช้จ่ายที่น้อยที่สุดจากจุดเริ่มต้น (source) ไปยังแต่ละจุดที่พิจารณา (vertex)
- π เก็บค่าจุดที่เป็นต้นทาง (predecessor) ของจุดที่พิจารณาที่มีค่าใช้จ่ายน้อยที่สุด
- S เป็นเซ็ตของจุด (vertex) ที่มีการตัดสินใจแล้ว
- Q เป็นเซ็ตของจุด (vertex) ที่ยังไม่ได้ผ่านการตัดสินใจ
- s คือ จุดเริ่มต้นของการเดินทาง



ภาพที่ 2.6 ผังแสดงขั้นตอนการทำงานของ Dijkstra's Algorithm (ดัดแปลงจาก Waldura, 2007)

2.5 Web-based GIS

Web-GIS คือ โปรแกรมประยุกต์ภูมิสารสนเทศที่ทำงานบนอินเทอร์เน็ตผ่านโปรโตคอล WWW (World Wide Web) ที่มีจำนวนการใช้งานเพิ่มสูงขึ้นเนื่องจากมีการพัฒนาเรื่องของเทคโนโลยีด้านอินเทอร์เน็ตที่ก้าวหน้าและมีจำนวนผู้ใช้อินเทอร์เน็ตที่เพิ่มมากขึ้น โดยมีรูปแบบสถาปัตยกรรมแบบ 3-tier ดังแสดงในภาพที่ 2.7 กล่าวคือ ผู้ใช้จะทำงานที่เครื่องคอมพิวเตอร์ฝั่งลูกข่ายและส่งคำร้องขอไปยังโปรแกรมประยุกต์ที่อยู่บนฝั่งแม่ข่าย เครื่องแม่ข่ายจะทำการประมวลผลคำสั่งและส่งผลลัพธ์กลับมายังลูกข่าย



ภาพที่ 2.7 แสดงส่วนประกอบเบื้องต้นของระบบ Web-GIS (Alesheikh, et al., 2002)

2.6 Geoserver

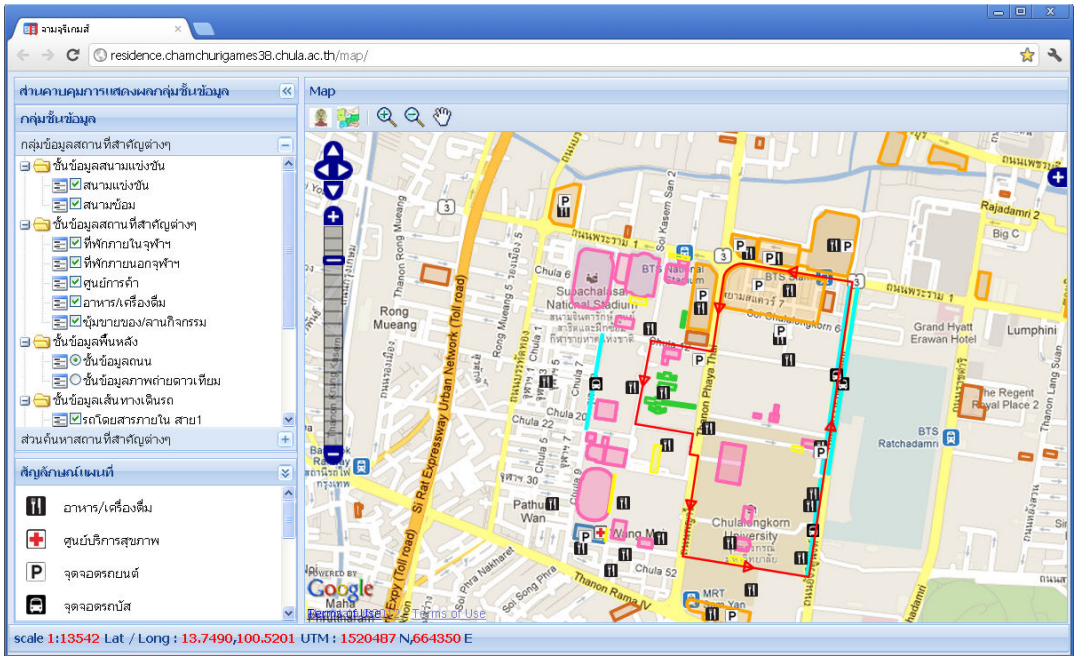
GeoServer เป็นชุดคำสั่งหรือซอฟต์แวร์แบบรหัสเปิดที่ทำงานอยู่บนเครื่องแม่ข่ายทำหน้าที่สำหรับให้บริการข้อมูลภูมิสารสนเทศผ่านเครือข่ายอินเทอร์เน็ต Geoserver พัฒนาด้วยภาษา Java จึงสามารถใช้งานได้ในทุก Platform ทั้งใน Windows, UNIX และ Linux เป็นต้น การกำหนดค่าการใช้งานต่างๆ สามารถกระทำได้ผ่านเว็บ ซึ่งทำให้สะดวกต่อการดูแลและปรับแต่งค่าการให้บริการ

Geoserver สนับสนุนการทำงานตามมาตรฐานต่างๆ ของ Open Geospatial Consortium (OGC) ในการให้บริการข้อมูลไม่ว่าจะเป็น Web Feature Service (WFS), Web Map Service (WMS), Web Coverage Service (WCS), Style Layer Descripter (SLD), Filter Encoding (FE), Geography Markup Language (GML) (Geoserver, 2012) เป็นต้น สามารถเชื่อมต่อกับแหล่งข้อมูลที่หลากหลายทั้งที่เป็น File-base เช่น Shapefile และที่เป็น Database เช่น Oracle, PostGIS, DB2, MySQL, SQL Server, ArcSDE รวมถึงสนับสนุนข้อมูลประเภทราสเตอร์ เช่น GeoTiff, MrSID, ECW, JPEG2000, Erdas Imagine เป็นต้น

2.7 OpenLayers

OpenLayers เป็นชุดคำสั่งรหัสเปิดที่เขียนด้วยภาษา JavaScript สำหรับช่วยพัฒนาโปรแกรมประยุกต์ด้านภูมิสารสนเทศบนเว็บ (OpenLayers, 2012) สนับสนุนการเชื่อมต่อกับระบบให้บริการข้อมูลภูมิสารสนเทศที่หลากหลาย เช่น WMS, WFS, Google, WorldWind, Yahoo, MultiMap, TileCache, MapGuide, ArcIMS เป็นต้น นอกจากนี้ OpenLayers ได้พัฒนาเครื่องมือควบคุมการแสดงผลแผนที่ต่างๆ จำนวนมาก เช่น การย่อ/ขยาย, การหาตำแหน่งจากตัวชี้ตำแหน่ง, เครื่องมือควบคุมการเปิด/ปิดการแสดงผล, เครื่องมือการวาดรูป เป็นต้น ตัวอย่างของโปรแกรมประยุกต์ภูมิสารสนเทศบนเว็บที่ใช้ OpenLayers ช่วยในการพัฒนาดังแสดงในภาพที่ 2.8

OpenLayers ถูกนำไปประยุกต์ใช้และต่อยอดในหลายๆ ซอฟต์แวร์และชุดคำสั่งประยุกต์รหัสเปิด เช่น เป็นเครื่องมือสำหรับแสดงผลข้อมูลใน GeoServer, เป็นเครื่องมือสำหรับต่อเชื่อมและแสดงข้อมูล Google ในซอฟต์แวร์ QGIS (Quantum GIS) เป็นต้น

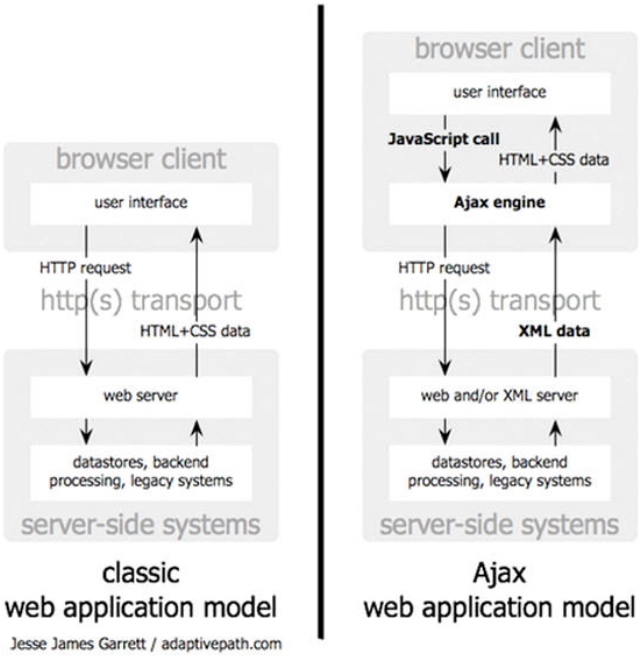


ภาพที่ 2.8 โปรแกรมประยุกต์ภูมิสารสนเทศบนเว็บที่ใช้ OpenLayers ช่วยในการพัฒนา (จุฬาลงกรณ์มหาวิทยาลัย, 2554)

2.8 AJAX (Asynchronous JavaScript and XML)

AJAX คือ เทคโนโลยีที่รวมเอาความสามารถของ JavaScript และ XML ไว้ด้วยกัน ช่วยให้เว็บไซต์มีรูปแบบการใช้งานและให้ความรู้สึกเหมือนกับเป็นซอฟแวร์ที่ติดตั้งไว้กับเครื่อง ต่างจากการทำงานของเว็บไซต์แบบดั้งเดิมที่ผู้ใช้ร้องขอข้อมูลจากแม่ข่าย (Server) แม่ข่ายก็จะทำการ

ประมวลผลข้อมูลจากการร้องขอและส่งผลลัพธ์กลับมาเป็นหน้า HTML ไปยังผู้ใช้ ซึ่งเป็นวิธีการแบบการร้องขอและการตอบรับ (Request and Response) ผู้ใช้ต้องรอระหว่างที่แม่ข่ายประมวลผล เป็นหลักการทำงานแบบ Synchronous หรือการติดต่อสื่อสารแบบต่อเนื่อง ดังแสดงในภาพที่ 2.9 (ซ้าย)



ภาพที่ 2.9 การทำงานของโปรแกรมประยุกต์บนเว็บแบบดั้งเดิม (ซ้าย) เปรียบเทียบกับการทำงานด้วยเทคนิค AJAX (ขวา) (Garrett, 2005)

เทคโนโลยี AJAX จะเป็นการทำงานแบบ Asynchronous หรือการติดต่อสื่อสารแบบไม่ต่อเนื่อง โดยจะทำงานในรูปแบบการรับส่งเฉพาะข้อมูล เมื่อแม่ข่ายรับคำร้องขอจากลูกข่ายและทำการประมวลผลแล้วจะส่งเฉพาะผลลัพธ์อาจอยู่ในรูปแบบ XML หรือ JSON มายังลูกข่าย เมื่อได้รับข้อมูลครบถ้วนแล้ว จะมีการเรียกใช้ JavaScript ที่ฝั่งลูกข่ายเพื่อนำข้อมูลไปใช้ต่อหรือแสดงผล ด้วยเทคนิคนี้ช่วยลดเวลาในการรับส่งข้อมูลจากแม่ข่ายและสามารถนำมาประยุกต์ใช้กับการพัฒนาโปรแกรมประยุกต์ภูมิสารสนเทศบนเว็บได้เป็นอย่างดี

2.9 JSON (JavaScript Object Notation)

JSON คือ รูปแบบของข้อมูลที่ใช้แลกเปลี่ยนระหว่าง browser กับ server โดยเป็นรูปแบบข้อความธรรมดา (plain text) ที่มนุษย์สามารถอ่านเข้าใจได้และโปรแกรมคอมพิวเตอร์สามารถสร้างและแปลงข้อมูลได้ง่าย (JSON, 2012) ปัจจุบัน JSON นิยมนำมาใช้เป็นรูปแบบข้อมูลสำหรับการพัฒนาโปรแกรมประยุกต์บนเว็บโดยเฉพาะที่พัฒนาด้วยเทคนิค AJAX เพราะมี

ความกระชับและเข้าใจง่ายกว่า XML การแปลงข้อมูลจากรูปแบบของ JSON ให้เป็น object ทำให้สามารถเรียกใช้งานข้อมูลผ่าน method ต่างๆ ได้ โดยมีรูปแบบโครงสร้างประโยค 2 รูปแบบ คือ

- แบบ object เป็นเซ็ตของข้อมูลที่มีรูปแบบ Key : Value pairs โดย object จะขึ้นต้นด้วยเครื่องหมาย { (left brace) และลงท้ายด้วยเครื่องหมาย } (right brace) ข้อมูลแต่ละข้อมูลจะแยกจากกันด้วยเครื่องหมาย , (comma)
- แบบ array เป็นเซ็ตของลำดับข้อมูล โดย array จะขึ้นต้นด้วยเครื่องหมาย [(left bracket) และลงท้ายด้วยเครื่องหมาย] (right bracket) ข้อมูลแต่ละข้อมูลจะแยกจากกันด้วยเครื่องหมาย , (comma)

ข้อมูลภายใน JSON สามารถเป็นได้ทั้งตัวอักษร, ตัวเลข, ข้อมูลที่เป็น object, ข้อมูลที่เป็น array, ข้อมูลทางตรรกะ, ข้อมูลไม่ทราบค่า (null) ตัวอย่างของข้อมูลแบบ JSON ดังแสดงในภาพที่ 2.10

```
{
   "firstName": "Panuwat",
   "lastName": "Angkasurak",
   "address": {
      "streetAddress": "payathai",
      "city": "bangkok",
      "postalCode": 10330
   },
   "mobile": [
      "0-2218-6651",
      "0-2218-3650"
   ]
}
```

ภาพที่ 2.10 ตัวอย่างของข้อมูลแบบ JSON

2.10 งานวิจัยที่เกี่ยวข้อง

2.10.1 ระบบสารสนเทศภูมิศาสตร์ผ่านเครือข่ายอินเทอร์เน็ต เพื่อเลือกเส้นทางที่ใช้เวลาน้อยที่สุดโดยระบบขนส่งมวลชนหลายรูปแบบ (มนต์ศักดิ์ โซ่เจริญธรรม, 2549)

งานวิจัยนี้กล่าวถึงการพัฒนาโปรแกรมประยุกต์บนเว็บที่ช่วยในการตัดสินใจเลือกเส้นทางในการเดินทางที่ใช้เวลาน้อยที่สุดจากการเดินทางด้วยระบบขนส่งมวลชนหลายรูปแบบผสมผสานกัน ประกอบด้วย รถโดยสารประจำทาง เรือข้ามฟาก เรือโดยสาร รถไฟฟ้าบีทีเอส และรถไฟฟ้ามหานคร (รถไฟฟ้าใต้ดิน) ในพื้นที่กรุงเทพมหานคร

โดยได้ทำการวิเคราะห์ ออกแบบ และนำเสนอแบบจำลองข้อมูลที่ใช้ในการจัดเก็บและจำลองพฤติกรรมของระบบขนส่งมวลชนรูปแบบต่างๆ และพัฒนาโปรแกรมประยุกต์บนเว็บใน

การค้นหาเส้นทางการเดินทางที่ใช้เวลาน้อยที่สุด จากการศึกษางานวิจัยนี้ สามารถสรุปประเด็นสำคัญๆ ได้ดังนี้

- โครงสร้างการจัดเก็บข้อมูลโครงข่ายของระบบขนส่งมวลชน แยกเป็นตารางที่ใช้จัดเก็บข้อมูลความสัมพันธ์เชิงโครงข่าย (Topology) เพื่อใช้ในการวิเคราะห์หาเส้นทางในการเดินทางที่ใช้เวลาน้อยที่สุด และตารางที่ใช้จัดเก็บข้อมูลเชิงเรขาคณิต (Geometry) สำหรับการแสดงผลแผนที่ทางฝั่งผู้ใช้

- โปรแกรมที่พัฒนาขึ้นจะดำเนินการตามขั้นตอนวิธีของ Dijkstra และได้ปรับแต่งเพิ่มปัจจัยต่างๆ เข้าไปในการพิจารณา คือ เวลาที่ระบบขนส่งมวลชนให้บริการ และอัตราเร็วของระบบขนส่งมวลชนที่วิ่งผ่านแต่ละ link

- การทำงานของโปรแกรมประยุกต์เป็นการทำงานแบบ Client - Server ผ่านเครือข่ายอินเทอร์เน็ต ข้อมูลโครงข่ายจัดเก็บในระบบฐานข้อมูล PostgreSQL และใช้ซอฟแวร์ Apache ทำหน้าที่เป็นเว็บเซอร์ฟเวอร์ ใช้โปรแกรม Map Server ทำหน้าที่เป็นแม่ข่ายแผนที่ ในส่วนของการควบคุมการทำงานของแผนที่ทางฝั่งผู้ใช้จะอาศัยโปรแกรม OpenLayers เป็นตัวควบคุม

- เงื่อนไขทางด้านเวลาที่ใช้ในการเดินทางด้วยรถโดยสารประจำทาง ได้ใช้เพียงค่าเดียวคือ คิดจากความเร็วเฉลี่ยของรถโดยสารประจำทางสำหรับการเดินทางผ่าน link ใดๆ และมีอัตราเร็วที่เท่ากันตลอดวัน

- มีการกำหนดรหัสประจำตัวเพื่อใช้ในการอ้างอิงและสืบค้นคุณสมบัติหรือชนิดให้กับ node ในโครงข่ายที่เป็นตัวแทนของป้าย สถานี หรือท่าเรือในระบบขนส่งมวลชน โดยรหัสประจำตัวจะประกอบด้วยตัวเลขหกหลัก ตัวเลขสามตัวแรกระบุถึงประเภทของระบบขนส่งมวลชน ส่วนตัวเลขสามตัวหลังจะใช้นับจำนวน node ดังแสดงในตารางที่ 2.7

ตารางที่ 2.7 รหัสของ node ที่ใช้แทนป้าย สถานี หรือท่าเรือ สำหรับระบบขนส่งมวลชนแต่ละประเภท

ประเภทระบบขนส่งมวลชน	ช่วงค่าของรหัสที่เป็นไปได้ของ node หัวและท้ายของ link	
	สามตัวแรก	สามตัวหลัง
การเดินเท้า	001 ถึง 999	001 ถึง 999
รถโดยสารประจำทาง	050 ถึง 777	001 ถึง 999
รถไฟฟ้าบีทีเอส	030 ถึง 040	001 ถึง 999
รถไฟฟ้ามหานคร	001 ถึง 020	001 ถึง 999
เรือข้ามฟาก	888	001 ถึง 999
เรือด่วนตามแม่น้ำเจ้าพระยา	999	001 ถึง 999

2.10.2 การพัฒนาโปรแกรมประยุกต์บนอุปกรณ์คอมพิวเตอร์พกพา (PDA) เพื่อเลือกเส้นทางที่เหมาะสม ในระบบขนส่งมวลชนหลายรูปแบบ (พลปรีชา ชิดบุรี, 2551)

งานวิจัยนี้กล่าวถึงการพัฒนาโปรแกรมประยุกต์บนอุปกรณ์คอมพิวเตอร์พกพา (PDA Phone) เพื่อแนะนำเส้นทางการเดินทางที่เหมาะสมโดยใช้เวลาในการเดินทางน้อยที่สุด จากการเดินทางด้วยระบบขนส่งมวลชนในกรุงเทพมหานครทั้งหมด 5 รูปแบบ ได้แก่ รถไฟฟ้าบีทีเอส รถไฟฟ้ามหานคร รถโดยสารประจำทาง เรือโดยสาร และเรือข้ามฟาก

โดยได้ทำการรวบรวมข้อมูลการเดินทางในระบบขนส่งมวลชนจากอัตราเร็วเฉลี่ยในการเดิน, อัตราเร็วเฉลี่ยของยานพาหนะที่ใช้โดยสาร และระยะเวลาที่ใช้ในการรอคอย จัดเก็บข้อมูลในรูปแบบของ GDF (Geographic Data Files) และจัดทำผลลัพธ์ของการวิเคราะห์โครงข่ายการเดินทางไว้ล่วงหน้า เพื่อใช้เป็นฐานข้อมูลในการค้นหาและแสดงคำตอบให้กับโปรแกรมประยุกต์ที่พัฒนาขึ้น จากการศึกษางานวิจัยนี้ สามารถสรุปประเด็นสำคัญๆ ได้ดังนี้

- การออกแบบแบบจำลองข้อมูลและการออกแบบโครงสร้างการจัดเก็บข้อมูลโครงข่าย แยกเป็นตารางที่ใช้จัดเก็บข้อมูลความสัมพันธ์เชิงโครงข่าย (Topology) และตารางที่ใช้จัดเก็บข้อมูลเชิงเรขาคณิต (Geometry)

- มีการจัดทำผลลัพธ์ของการวิเคราะห์โครงข่ายการเดินทางทั้งหมดไว้ล่วงหน้า โดยจัดทำข้อมูลภูมิศาสตร์ด้วยรูปแบบ GDF แล้วนำเข้าจัดเก็บในระบบฐานข้อมูลด้วยรูปแบบไฟล์ของ Microsoft SQL Server Compact Edition (.sdf Files) เพื่อให้เกิดประสิทธิภาพในการเรียกข้อมูลและใช้ขนาดเนื้อที่ในการจัดเก็บข้อมูลที่น้อยกว่ารูปแบบของ GDF ที่เป็น Text files

- เงื่อนไขทางด้านเวลาที่ใช้ในการเดินทางด้วยรถโดยสารประจำทาง ได้ใช้เพียงค่าเดียวคือ คิดจากความเร็วเฉลี่ยของรถโดยสารประจำทางสำหรับการเดินทางผ่าน link ใดๆ และมีอัตราเร็วที่เท่ากันตลอดวัน

- หมายเลขของสายการเดินรถโดยสารประจำทาง ไม่ได้แยกเส้นทางการเดินทางในขาไปและขากลับ

บทที่ 3
การจัดเตรียมข้อมูลโครงข่ายสำหรับการเดินทาง ด้วยรถโดยสารประจำทาง

ในบทนี้จะกล่าวถึงพฤติกรรมการเดินทางด้วยรถโดยสารประจำทาง เพื่อนำมาใช้ในการออกแบบแบบจำลองข้อมูลและโครงสร้างการจัดเก็บข้อมูล จากนั้นจึงจัดทำและจัดเก็บข้อมูลโครงข่ายในระบบสารสนเทศภูมิศาสตร์ด้วยโปรแกรม Quantum GIS ด้วยรูปแบบข้อมูลแบบ Shapefile ขั้นตอนสุดท้ายคือการนำเข้าข้อมูล Shapfile จัดเก็บในระบบฐานข้อมูล Oracle Spatial สำหรับการวิเคราะห์หาเส้นทางการเดินทางที่เหมาะสมต่อไป

3.1 พฤติกรรมการเดินทางด้วยรถโดยสารประจำทาง

การออกแบบแบบจำลองข้อมูลภูมิศาสตร์ (Data Model) คือ การจำลองพฤติกรรมของโลกจริง เพื่อจัดเก็บลงในระบบฐานข้อมูล จึงต้องมีการศึกษากิจกรรมต่างๆ ที่เกิดขึ้นในการเดินทางด้วยรถโดยสารประจำทาง งานวิจัยนี้อาศัยข้อมูลการศึกษาวิจัยที่ได้ศึกษาพฤติกรรมการเดินทางในระบบขนส่งมวลชนของกรุงเทพมหานคร (มนศักดิ์ โซ่เจริญธรรม, 2549) มีรายละเอียดดังนี้

3.1.1 การเดินเท้า

การเดินเท้า หมายถึงการเดินจากจุดใดๆ เพื่อไปยังจุดเริ่มต้นเพื่อเข้าสู่ระบบรถโดยสารประจำทาง การเดินเพื่อเปลี่ยนสายรถประจำทาง การเดินไปยังจุดหมายปลายทางโดยลัดเลาะไปตามทางเดินเท้า การเดินขึ้นสะพานลอย การเดินข้ามทางม้าลาย ข้ามแม่น้ำลำคลองหรือลอดอุโมงค์ อัตราเร็วที่ใช้ในการเดินเท้าขึ้นอยู่กับแต่ละบุคคลและสภาพแวดล้อมในการเดินเท้า ในงานวิจัยนี้กำหนดให้อัตราเร็วเฉลี่ยของการเดินเท่ากับ 5 กิโลเมตรต่อชั่วโมง และทางเดินเท้ารูปแบบต่างๆ สามารถเดินผ่านได้ตลอด 24 ชั่วโมง

3.1.2 รถโดยสารประจำทาง

รถโดยสารประจำทาง หมายถึง รถโดยสารประจำทางขนาดใหญ่ที่ให้บริการโดยองค์การขนส่งมวลชนกรุงเทพ ไม่รวมรถตู้และรถโดยสารขนาดเล็ก เส้นทางการเดินทางของรถโดยสารประจำทางจะมีเส้นทางที่แน่นอนตามแต่ละหมายเลขสายรถ แยกเป็นฝั่งสำหรับถนนที่มีการเดินทางสองทิศทาง ส่วนถนนที่มีการเดินรถทิศทางเดียวรถโดยสารประจำทางก็จะวิ่งทิศทางเดียว รถโดยสารประจำทางจะจอดรับส่งผู้โดยสารที่ป้ายจอดรถประจำทางที่อยู่ริมทางเดินเท้า เส้นทางที่

มีสายรถประจำทางผ่านมากกว่า 1 สาย ผู้เดินทางสามารถลงจากรถที่โดยสารมาที่ป้ายจอดรถประจำทางเพื่อเปลี่ยนสายรถได้ หรือสามารถเดินเท้าไปยังป้ายจอดรถประจำทางอื่นๆได้ การเปลี่ยนสายรถประจำทางสามารถเปลี่ยนที่ป้ายใดก็ได้ ที่มีการซ้อนทับของสายรถประจำทางตั้งแต่ 2 สายขึ้นไป

อัตราเร็วของรถโดยสารประจำทางในงานวิจัยนี้ ผู้วิจัยได้สมมติอัตราเร็วขึ้นเอง โดยในแต่ละช่วงของเส้นทาง (links) จะมีอัตราเร็วที่แตกต่างกันไปตามแต่ละช่วงเวลาของวัน โดยทุกๆ 15 นาที จะสมมติอัตราเร็วในการเดินทาง 1 ค่า ดังนั้นใน 1 วัน จะมีอัตราเร็วในการเดินทางทั้งหมด 96 ค่า และกำหนดให้ระยะเวลารอคอยในการรอขึ้นรถโดยสารประจำทางเท่ากับ 15 นาที

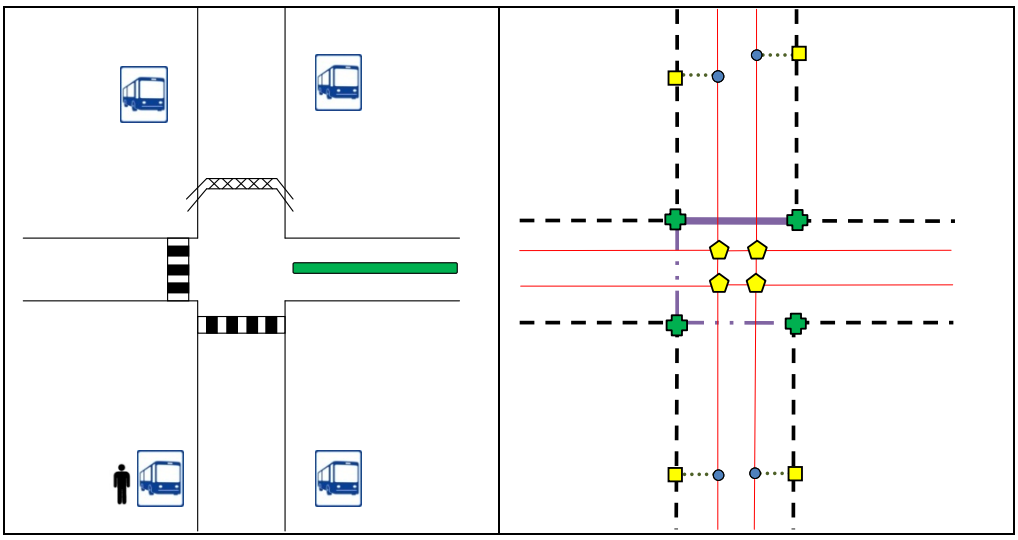
3.2 การออกแบบแบบจำลองข้อมูล

จากการศึกษาพฤติกรรมการเดินทางด้วยรถโดยสารประจำทาง ซึ่งประกอบด้วยการเดินบนทางเท้า การโดยสารรถประจำทาง การเดินข้ามทางม้าลาย การเดินข้ามสะพานลอย และการเดินจากป้ายรถประจำทางเพื่อขึ้นหรือลงรถโดยสาร ได้ทำการออกแบบแบบจำลองข้อมูลเพื่ออธิบายพฤติกรรมการเดินทางและความสัมพันธ์เชิงพื้นที่ระหว่างวัตถุ ให้อยู่ในรูปแบบของข้อมูลภูมิศาสตร์ และออกแบบความสัมพันธ์เชิงโครงข่าย (Topology) โดยอาศัยทฤษฎีของ Graph (ตามที่ได้กล่าวมาแล้วใน 2.2) ซึ่งจะจำลองพฤติกรรมการเดินทางจริงให้อยู่ในแบบจำลองข้อมูลที่ประกอบด้วย link และ node

3.2.1 ข้อมูลโครงข่ายรถโดยสารประจำทางที่ออกแบบแบบจำลองข้อมูลให้แทนด้วย link

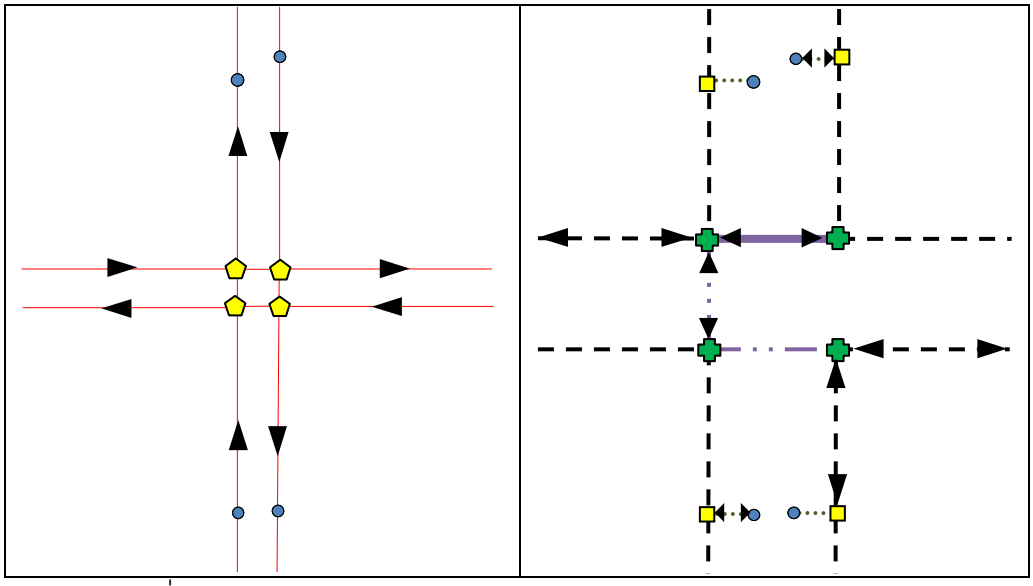
แบบจำลองข้อมูลที่ออกแบบให้ link เป็นตัวแทนของความสัมพันธ์หรือความเชื่อมโยงกันทางกายภาพของวัตถุที่อยู่ในโครงข่ายการเดินทางด้วยรถโดยสารประจำทางดังแสดงในภาพที่ 3.1 ได้แก่

- ถนน (——) เชื่อมระหว่าง แยกถนน, จุดจอดรถประจำทาง
- ทางเดินเท้า (- - - - -) เชื่อมระหว่าง ป้ายรถประจำทางกับป้ายรถประจำทาง, ป้ายรถประจำทางกับจุดข้ามทางม้าลาย, ป้ายรถประจำทางกับจุดข้ามสะพานลอย, จุดข้ามทางม้าลายกับจุดข้ามสะพานลอย
- ทางม้าลาย (— · · —) เชื่อมระหว่างจุดข้ามระหว่างสองฝั่งถนน
- สะพานลอย (▬▬▬) เชื่อมระหว่างจุดข้ามระหว่างสองฝั่งถนน
- เส้นเชื่อมทางเดินเท้าและถนน (.....) เชื่อมระหว่างป้ายรถประจำทางกับจุดจอดรถประจำทางบนถนน



ภาพที่ 3.1 การออกแบบแบบจำลองข้อมูลเพื่ออธิบายพฤติกรรมการเดินทาง
และความสัมพันธ์เชิงพื้นที่ภายในโครงข่ายรถโดยสารประจำทาง

ความสัมพันธ์ของ link ที่จำลองมาจากข้อมูลถนน ออกแบบให้มีทิศทางของความสัมพันธ์เป็นแบบทิศทางเดียว (Direct) คือ สามารถเดินทางไปตาม link ได้ในทิศทางเดียว ดังแสดงในภาพที่ 3.2 (ซ้าย) ส่วน link ที่จำลองมาจากข้อมูลทางเดินเท้า ทางม้าลาย สะพานลอย และเส้นเชื่อมทางเดินเท้ากับถนนออกแบบให้มีทิศทางของความสัมพันธ์แบบไปและกลับ (Bi-Direct) คือ สามารถเดินทางไปและเดินทางย้อนกลับได้ใน link เดียวกันดังแสดงในภาพที่ 3.2 (ขวา)

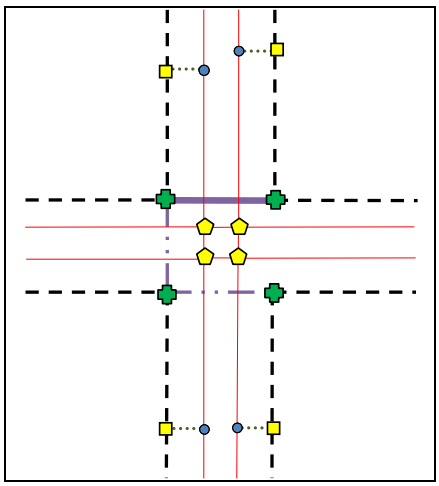


ภาพที่ 3.2 การออกแบบความสัมพันธ์ของ link แบบทิศทางเดียว (Direct)
และแบบไปและกลับ (Bi-Direct)

3.2.2 ข้อมูลโครงข่ายรถโดยสารประจำทางที่ออกแบบให้แทนด้วย node

แบบจำลองข้อมูลที่ออกแบบให้ node เป็นตัวแทนของวัตถุในโครงข่ายที่เป็นจุดเชื่อมต่อการเดินทาง ดังแสดงในภาพที่ 3.3 ได้แก่

- ตำแหน่งป้ายรถโดยสารประจำทาง (▫)
- จุดตัดของถนน (⬠)
- ตำแหน่งหยุดรับส่งผู้โดยสารของรถโดยสารประจำทาง (●)
- ตำแหน่งที่เป็นจุดข้ามทางม้าลายและสะพานลอย (✚)



ภาพที่ 3.3 แสดงตัวแทนของจุดเชื่อมต่อการเดินทางในโครงข่ายการเดินทางด้วยรถโดยสารประจำทาง

3.3 การออกแบบโครงสร้างการจัดเก็บข้อมูล

3.3.1 การออกแบบโครงสร้างข้อมูลของตารางหลักที่ใช้จัดเก็บข้อมูลโครงข่าย

เมื่อได้แบบจำลองของข้อมูลแล้ว จึงออกแบบโครงสร้างข้อมูลสำหรับจัดเก็บข้อมูลเชิงพื้นที่ (Spatial) และความสัมพันธ์เชิงโครงข่าย (Topology) ลงในระบบฐานข้อมูล Oracle Spatial ในรูปแบบของตาราง ในงานวิจัยจะใช้ความสามารถของระบบฐานข้อมูล Oracle ในการสร้างและวิเคราะห์โครงข่าย การนิยามโครงข่ายจำเป็นจะต้องมีตาราง 2 ตาราง คือ ตาราง link และ ตาราง node โดยต้องมีโครงสร้างข้อมูลที่ถูกต้องตรงกับ Oracle Network Schema (ตามที่ได้กล่าวมาแล้วใน 2.3.1.1)

ในงานวิจัยนี้ตาราง link ให้ชื่อว่า BUS_LINKS ใช้สำหรับจัดเก็บพฤติกรรมการเดินทางที่จำลองข้อมูลที่แทนด้วย link มีโครงสร้างข้อมูลดังแสดงในตารางที่ 3.1 ส่วนตาราง node ให้ชื่อว่า BUS_NODES ใช้สำหรับจัดเก็บข้อมูลโครงข่ายที่จำลองข้อมูลที่แทนด้วย node มีโครงสร้างข้อมูลดังแสดงในตารางที่ 3.2

ตารางที่ 3.1 โครงสร้างข้อมูลของตาราง BUS_LINKS

BUS_LINKS			
ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่างข้อมูล
LINK_ID	NUMBER	เลขจำนวนเต็มหกหลักแสดงรหัสประจำตัวที่เป็นเอกลักษณ์ของเส้นทางการเดินทาง	400220
START_NODE_ID	NUMBER	เลขจำนวนเต็มหกหลักแสดงรหัสประจำตัวที่เป็นเอกลักษณ์ของจุดเชื่อมต่อต้นทางของเส้นทางการเดินทาง	100697
END_NODE_ID	NUMBER	เลขจำนวนเต็มหกหลักแสดงรหัสประจำตัวที่เป็นเอกลักษณ์ของจุดเชื่อมต่อปลายทางของเส้นทางการเดินทาง	100700
LINK_NAME	VARCHAR2	ตัวอักษรแสดงชื่อของเส้นทาง	Praram2
LINK_TYPE	VARCHAR2	ตัวอักษรแสดงรูปแบบของการเดินทาง ซึ่งประกอบด้วย 1. BUS – รถโดยสารประจำทาง 2. FOOTPATH – ทางเดินเท้าริมถนน 3. CROSSWALK – ทางม้าลาย 4. OVERPASS – สะพานลอย 5. BUS_LINK – เส้นเชื่อมในการขึ้นลงรถโดยสาร	BUS
ACTIVE	CHAR (1)	ตัวอักษรสำหรับระบุว่า link นั้นๆ จะมองเห็นได้ในโครงข่ายหรือไม่ โดยมีค่าที่เป็นไปได้คือ Y = มองเห็น N = ไม่เห็น (link ที่ระบุให้มองไม่เห็นจะไม่ถูกนำมาพิจารณาในการวิเคราะห์โครงข่าย)	Y

ตารางที่ 3.1 โครงสร้างข้อมูลของตาราง BUS_LINKS (ต่อ)

BUS_LINKS			
ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่างข้อมูล
GEOM	SDO_GEOMETRY	วัตถุเรขาคณิตประเภทเส้นที่ใช้อธิบายรูปร่างของ link	MDSYS.SDO_GEOMETRY(2002,4326,null,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(100.570383391931,13.5938526586815,100.570393440269,13.5937751759468))
COST	NUMBER	เลขทศนิยม ระบุระยะทางของ link นั้นๆ ในหน่วยเมตร	604.4
BIDIRECTED	CHAR (1)	ตัวอักษรสำหรับระบุทิศทางการเดินทางว่า link นั้นๆ จะเดินทางแบบไปทางเดียว หรือแบบไปและกลับได้ โดยมีค่าที่เป็นไปได้คือ Y = Undirected N = Directed	N

ตารางที่ 3.2 โครงสร้างข้อมูลของตาราง BUS_NODES

BUS_NODES			
ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่างข้อมูล
NODE_ID	NUMBER	เลขจำนวนเต็มหกหลักแสดงรหัสประจำตัวที่เป็นเอกลักษณ์ของจุดเชื่อมต่อการเดินทางในโครงข่าย	600366
NODE_TYPE	VARCHAR2	ตัวอักษรแสดงรูปแบบของจุดเชื่อมต่อการเดินทาง ซึ่งประกอบด้วย 1. BUS_STATION – จุดที่รถโดยสารจอดรับส่งผู้โดยสาร 2. BUS_STOP – ป้ายรถโดยสารประจำทาง 3. SWITCH – จุดเชื่อมต่อระหว่างทางเดินเท้ากับสะพานลอยและทางม้าลาย	BUS_STOP
ACTIVE	CHAR (1)	ตัวอักษรสำหรับระบุว่า node นั้นๆ จะมองเห็นได้ในโครงข่ายหรือไม่ โดยมีค่าที่เป็นไปได้คือ Y = มองเห็น N = ไม่เห็น (node ที่ระบุให้มองไม่เห็นจะไม่ถูกนำมาพิจารณาในการวิเคราะห์โครงข่าย)	Y
GEOM	SDO_GEOMETRY	วัตถุเรขาคณิตประเภทจุดที่จัดเก็บค่าพิกัดของ node นั้นๆ	MDSYS.SDO_GEOMETRY(2001,4326,MDSYS.SDO_POINT_TYPE(100.388057571348,13.5911960899539,null),null,null)

3.3.2 การออกแบบรหัสประจำตัวให้กับข้อมูล link และ node

การระบุรหัสประจำตัว (ID) ให้กับ link และ node เพื่อให้สามารถอ้างอิงได้ถูกต้องตามชนิดของ link และ node นั้นๆ จึงออกแบบให้รหัสประจำตัวเป็นเลขจำนวนเต็มหกหลัก และสามารถจำแนกประเภทของ link หรือ node โดยใช้รหัสประจำตัวหลักแรกในการจำแนก โดยมีรายละเอียดของรหัสประจำตัวของข้อมูลโครงข่าย ดังแสดงในตารางที่ 3.3 เลขรหัสหลักแรกที่ใช้จำแนกชนิดของ link หรือ node ในตัวอย่างข้อมูลจะพิมพ์ด้วยตัวหนาและมีขีดเส้นใต้

ตารางที่ 3.3 การออกแบบรหัสประจำตัวให้กับข้อมูล link และ node

ประเภทของข้อมูลในโครงข่าย	ชนิดของข้อมูล	ช่วงค่ารหัสที่เป็นไปได้		ตัวอย่างข้อมูล
		เลขรหัส 3 ตัวแรก	เลขรหัส 3 ตัวหลัง	
แยกถนน, จุดรับส่งผู้โดยสารของรถประจำทาง	NODE	100 ถึง 199	001 ถึง 999	**1**01069
จุดข้ามสะพานลอย และทางม้าลาย	NODE	200 ถึง 299	001 ถึง 999	**2**00638
ป้ายรถโดยสารประจำทาง	NODE	600 ถึง 699	001 ถึง 999	**6**00660
สะพานลอย	LINK	300 ถึง 399	001 ถึง 999	**3**00022
ทางรถโดยสารประจำทาง	LINK	400 ถึง 499	001 ถึง 999	**4**01621
ทางม้าลาย	LINK	500 ถึง 599	001 ถึง 999	**5**00626
ทางเดินเท้า	LINK	700 ถึง 799	001 ถึง 999	**7**00355
เส้นเชื่อมป้ายจอดรถประจำทางกับจุดที่รถประจำทางจอด	LINK	900 ถึง 999	001 ถึง 999	**9**00783

3.3.3 การออกแบบโครงสร้างตารางข้อมูลอื่นๆ ที่ต้องใช้ในงานวิจัย

นอกจากตาราง link และ ตาราง node ได้ออกแบบแบบจำลองข้อมูลและโครงสร้างข้อมูลที่ต้องใช้ในงานวิจัยเพิ่มอีก 2 ตารางคือ ตาราง LINKS_SPEED สำหรับใช้ในการวิเคราะห์โครงข่ายและตาราง BUS สำหรับใช้ในการวิเคราะห์เพื่อหาหมายเลขสายรถที่ต้องโดยสาร

ตาราง LINK_SPEED จัดเก็บอัตราเร็ว (กิโลเมตรต่อชั่วโมง) ในการเดินทางผ่าน link นั้นๆ โดยทุกๆ 15 นาที จะมีค่าอัตราเร็ว 1 ค่า ดังนั้นในหนึ่งวันสำหรับแต่ละ link จะมีค่าอัตราเร็วทั้งหมด 96 ค่าโดย link ที่มีรูปแบบการเดินทางด้วยเท้า (สะพานลอย, ทางม้าลาย, ทางเดินเท้า) จะมีอัตราเร็วในการเดินทุกๆ ช่วงเวลาเท่ากัน คือ 5 กิโลเมตรต่อชั่วโมง (ตามที่ได้กล่าวมาแล้วใน

3.1.1) ดังแสดงตัวอย่างของข้อมูลในตารางที่ 3.4 ส่วน link ที่มีรูปแบบการเดินทางบนถนนจะมีอัตราเร็วที่แตกต่างกันไปในแต่ละช่วงวัน โดยเป็นอัตราเร็วที่ผู้วิจัยกำหนดเองเพื่อใช้ในการวิจัยดังแสดงตัวอย่างในตารางที่ 3.5 ตาราง LINKS_SPEED จัดเก็บแยกเป็นตารางต่างหากในระบบฐานข้อมูล และสามารถปรับแก้ได้แบบพลวัต เพื่อที่หากมีข้อมูลอัตราเร็วของรถโดยสารประจำทางที่เดินทางผ่าน link นั้นๆ จะได้สามารถแก้ไขให้มีความใกล้เคียงกับสภาพความเป็นจริงได้

ตารางที่ 3.4 ค่าอัตราเร็วของรูปแบบการเดินทางด้วยเท้าบนสะพานลอย, ทางม้าลาย และทางเดินเท้า

LINKS_SPEED			
ชื่อรายการ	**ชนิดข้อมูล**	**ตัวอย่างข้อมูล**	**ช่วงเวลา**
LINK_ID	NUMBER	700943	
SPEED_1	NUMBER	5 (กม./ชม.)	0.00 – 0.14
SPEED _2	NUMBER	5 (กม./ชม.)	0.15 – 0.29
SPEED _3	NUMBER	5 (กม./ชม.)	0.30 – 0.44
SPEED _4	NUMBER	5 (กม./ชม.)	0.45 – 0.59
SPEED _5	NUMBER	5 (กม./ชม.)	1.00 – 1.14
SPEED _6	NUMBER	5 (กม./ชม.)	1.15 – 1.29
…	…	…	…
SPEED _96	NUMBER	5 (กม./ชม.)	23.45 – 23.59

ตารางที่ 3.5 ตัวอย่างค่าอัตราเร็วของรูปแบบการเดินทางบนถนน

LINKS_SPEED			
ชื่อรายการ	**ชนิดข้อมูล**	**ตัวอย่างข้อมูล**	**ช่วงเวลา**
LINK_ID	NUMBER	400176	
SPEED_1	NUMBER	14 (กม./ชม.)	0.00 – 0.14
SPEED_2	NUMBER	13 (กม./ชม.)	0.15 – 0.29
SPEED_3	NUMBER	9 (กม./ชม.)	0.30 – 0.44
SPEED_4	NUMBER	12 (กม./ชม.)	0.45 – 0.59
SPEED_5	NUMBER	12 (กม./ชม.)	1.00 – 1.14
SPEED_6	NUMBER	11 (กม./ชม.)	1.15 – 1.29
…	…	…	…
SPEED_96	NUMBER	15 (กม./ชม.)	23.45 – 23.59

ตาราง BUS ออกแบบไว้สำหรับจัดเก็บหมายเลขสายรถประจำทางที่เดินทางผ่านแต่ละ link มีโครงสร้างข้อมูลดังแสดงในตารางที่ 3.6 ประกอบด้วยคอลัมน์ LINK_ID สำหรับจัดเก็บค่ารหัสประจำตัวของ link และคอลัมน์ BUS_NO สำหรับจัดเก็บหมายเลขสายรถประจำทางที่ผ่าน link นั้นๆ

ตารางที่ 3.6 โครงสร้างข้อมูลของตาราง BUS

BUS			
ชื่อรายการ	**ชนิดข้อมูล**	**คุณลักษณะ**	**ตัวอย่างข้อมูล**
LINK_ID	NUMBER	เลขจำนวนเต็มหกหลักแสดงรหัสประจำตัวที่เป็นเอกลักษณ์ของเส้นทางการเดินทาง	400005
BUS_NO	NUMBER	เลขจำนวนเต็มแสดงหมายเลขสายรถที่เดินทางผ่าน link นั้นๆ	15

3.4 การจัดเตรียมข้อมูลภูมิศาสตร์สำหรับระบบรถโดยสารประจำทาง

3.4.1 ข้อมูลที่เกี่ยวข้อง

ข้อมูลที่ใช้ผู้วิจัยได้จัดทำขึ้นโดยอ้างอิงตำแหน่งจากข้อมูลพื้นฐานด้านการคมนาคมของหน่วยงานต่างๆ ที่ได้จัดทำไว้บ้างแล้ว และอ้างอิงตำแหน่งจากแผนที่ภาพถ่ายดาวเทียมจากผู้ให้บริการข้อมูล คือ Google โดยมีข้อมูลที่ต้องใช้ในงานวิจัยดังต่อไปนี้

- เส้นทางถนน
- ตำแหน่งป้ายรถโดยสารประจำทาง
- จุดตัดของถนน
- เส้นทางเดินเท้าริมถนน
- ตำแหน่งสะพานลอย
- ตำแหน่งทางม้าลาย

3.4.2 การจัดทำและปรับปรุงข้อมูล

ข้อมูลโครงข่ายการเดินทางด้วยรถโดยสารประจำทางที่ต้องใช้ในงานวิจัย ผู้วิจัยได้จัดทำขึ้นเอง เพื่อให้เหมาะสมต่อการนำไปวิเคราะห์โครงข่าย โดยระบุถึงสิ่งที่เป็น link และ node ตามนิยามของทฤษฎีในการวิเคราะห์โครงข่าย และให้ตรงกับแบบจำลองข้อมูลที่ออกแบบไว้ใน 3.2.1 โดยใช้โปรแกรม Quantum GIS (QGIS) เวอร์ชั่น 1.6.0 ที่มีความสามารถในการจัดการข้อมูลปริภูมิ เช่น การแสดงผล สืบค้นข้อมูล การปรับแก้ข้อมูลปริภูมิ การแก้ไขข้อมูลอรรถาธิบาย และ

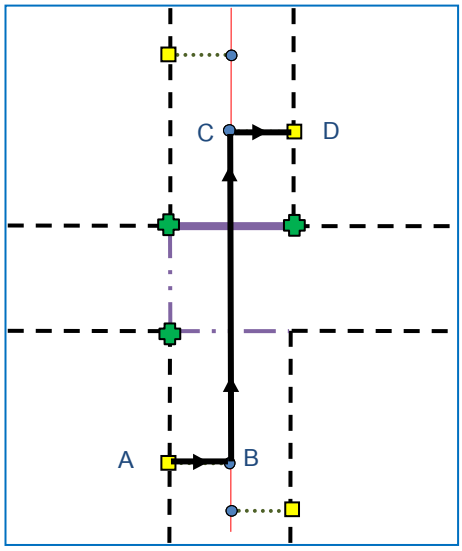
สนับสนุนการเชื่อมต่อกับฐานข้อมูลภายนอกและเรียกดูข้อมูลผ่านมาตรฐาน WMS/WFS โดยในงานวิจัยได้อาศัย Plugins เสริมของ QGIS ชื่อว่า OpenLayers Plugin เพื่อเรียกดูข้อมูลภาพแผนที่ดาวเทียมของ Google ผ่านมาตรฐาน WMS (Web Map Service) ในการอ้างอิงเพื่อสร้างและแก้ไขข้อมูลโครงข่ายที่ใช้ในงานวิจัย และจัดเก็บข้อมูลในรูปแบบของ Shapefile โดยมีขั้นตอนในการจัดทำข้อมูลดังต่อไปนี้

3.4.2.1 การจัดทำข้อมูลเชิงพื้นที่

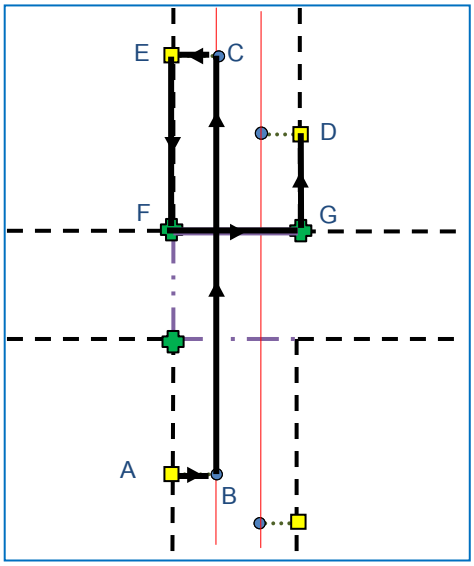
1. จัดทำข้อมูลที่เป็นตัวแทนเส้นทางการเดินรถโดยสารประจำทาง

ผู้วิจัยได้สร้าง (digitize) เส้นทางการเดินรถของรถโดยสารประจำทางขึ้นมาใหม่ โดยอ้างอิงตำแหน่งจากภาพถ่ายดาวเทียมจากผู้ให้บริการภาพแผนที่ คือ Google โดยสร้างเส้นทางเดินรถให้มีลักษณะแยกฝั่งกันสำหรับขาไปและขากลับ ของสายการเดินรถหมายเลขนั้นๆ และระบุให้มีทิศทางการเดินทางแบบไปทางเดียว (Directed) เพื่อให้สอดคล้องกับพฤติกรรมของรถโดยสารประจำทางดังที่กล่าวไว้ใน 3.1 เหตุที่ต้องสร้างแยกฝั่งเป็น 2 เส้น เพราะจะต้องมีการสร้างเส้นเชื่อมระหว่างเส้นทางเดินรถประจำทางและทางเดินเท้าเพื่อเป็นตัวแทนของความสัมพันธ์ในการเดินขึ้นและลงจากรถโดยสาร และเส้นเชื่อมนี้มีทิศทางการเดินทางแบบไปและกลับ (Bidirected) ดังที่ได้ออกแบบไว้ใน 3.2.1

หากเราสร้างเส้นทางเดินรถเพียงเส้นเดียวและกำหนดคุณสมบัติให้สามารถเดินทางไปและกลับ เมื่อสร้างเส้นเชื่อมการเดินขึ้น/ลงระหว่างทางเดินเท้ากับถนน จะเกิดปัญหาเรื่องของความสัมพันธ์ในการเดินทาง ไม่สอดคล้องกับพฤติกรรมในการเดินทางจริงๆ ดังแสดงในภาพที่ 3.4 กล่าวคือ จะทำให้ในการเดินทางผู้โดยสารสามารถขึ้นรถโดยสารจากฝั่งขาไป และข้ามไปลงยังอีกฝั่งของถนนได้ โดยไม่ต้องมีการข้ามทางม้าลายหรือสะพานลอย เส้นทางการเดินทางจะเป็นการเดินทางจาก A -->B -->C -->D ซึ่งไม่ตรงกับพฤติกรรมการเดินทางจริงที่ผู้โดยสารจะต้องลงจากรถ ณ ฝั่งที่รถกำลังเดินทางไป และการเดินข้ามไปยังอีกฝั่งของถนน จะต้องเดินข้ามสะพานลอยหรือทางม้าลายเท่านั้น จึงเป็นเหตุผลที่ต้องสร้างเส้นทางเดินรถแยกขาไปและขากลับ ซึ่งจะทำให้เส้นทางการเดินทางจะเดินทางจาก A --> B --> C --> E --> F --> G --> D ดังแสดงในภาพที่ 3.5



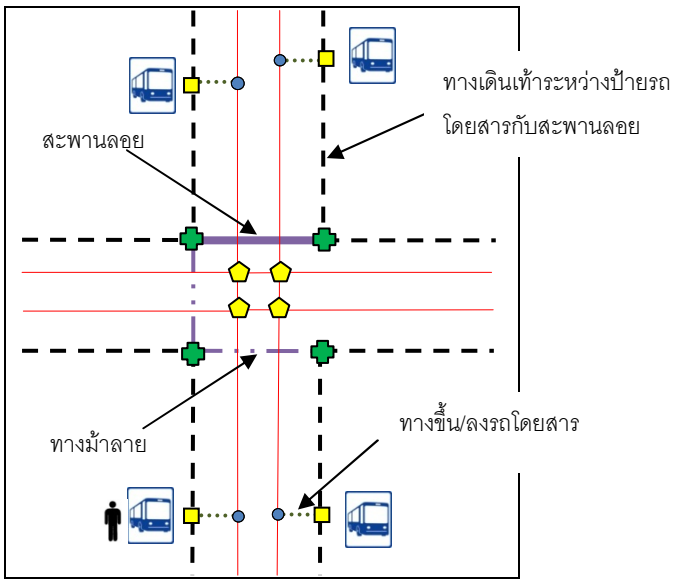
ภาพที่ 3.4 การสร้างเส้นทางเดินรถโดยสารเพียงเส้นเดียว ทำให้การเดินทางไม่สอดคล้องกับพฤติกรรมจริง



ภาพที่ 3.5 การสร้างเส้นทางเดินรถแยกฝั่ง ทำให้การเดินทางในโครงข่ายสอดคล้องกับพฤติกรรมจริง

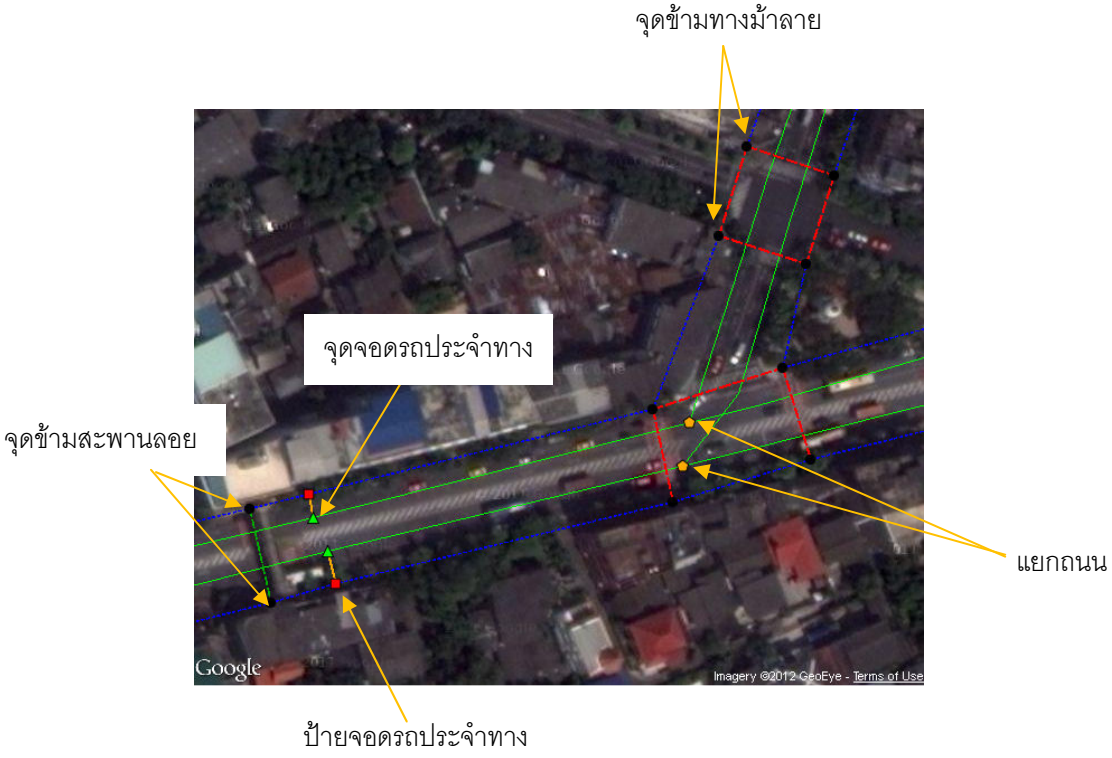
2. จัดทำข้อมูลที่เป็นตัวแทนเส้นทางเดินเท้า

เส้นทางเดินเท้าสร้าง (digitize) แยกเป็น 2 ฝั่งถนน โดยเป็นตัวแทนของทางเดินเท้าริมถนน, สะพานลอย, ทางม้าลาย และเส้นเชื่อมที่เป็นทางขึ้น/ลงรถโดยสารประจำทาง ดังแสดงในภาพที่ 3.6 และกำหนดทิศทางการเดินทางสามารถเดินทางได้ทั้งไปและกลับ (Bidirected)



ภาพที่ 3.6 ข้อมูลที่เป็นตัวแทนเส้นทางการเดินทางด้วยเท้ารูปแบบต่างๆ

3. จัดทำข้อมูลที่เป็นตัวแทนของตำแหน่งป้ายจอดรถประจำทาง, ตำแหน่งรับส่งผู้โดยสารของรถประจำทาง, จุดข้ามทางม้าลายและสะพานลอย จุดตัดของถนน ดังแสดงในภาพที่ 3.7



ภาพที่ 3.7 จุดเชื่อมต่อการเดินทางที่ออกแบบให้แทนด้วย node

3.4.2.2 การจัดทำข้อมูลอรรถาธิบาย

3.4.2.2.1 ข้อมูลอรรถาธิบายของ link (ดังแสดงตัวอย่างในภาพที่3.8) โดยมีข้อมูลที่ต้องระบุดังต่อไปนี้

1. เลขรหัสประจำตัวที่ระบุถึง link ต้องไม่ซ้ำกัน (unique) และเลขรหัสจะประกอบด้วยเลข 6 หลัก โดยเลขหลักแรกจะแตกต่างกันตามแต่ละประเภทของ link นั้นๆ ดังที่กล่าวไว้ในหัวข้อ 3.3.2

2. ระบุความสัมพันธ์เชิงโครงข่าย (Topology) โดยระบุเลขรหัสประจำตัวของ node เริ่มต้นของ link (start_node_id) และเลขรหัสประจำตัวของ node สิ้นสุด (end_node_id)

3. กำหนดทิศทางการเดินทางให้กับ link ว่าสามารถเดินทางได้ทิศทางเดียวหรือเดินทางได้ทั้งไปและกลับโดยระบุค่า Bidirected ให้เป็น Y (yes) สำหรับการเดินทางได้ทั้งไปและกลับ และระบุค่า Bidirected ให้เป็น N (No) สำหรับการเดินทางได้ทิศทางเดียว ในงานวิจัยนี้ link ที่เป็นตัวแทนของเส้นทางรถประจำทางเท่านั้นที่มีค่า Bidirected เป็น N (No)

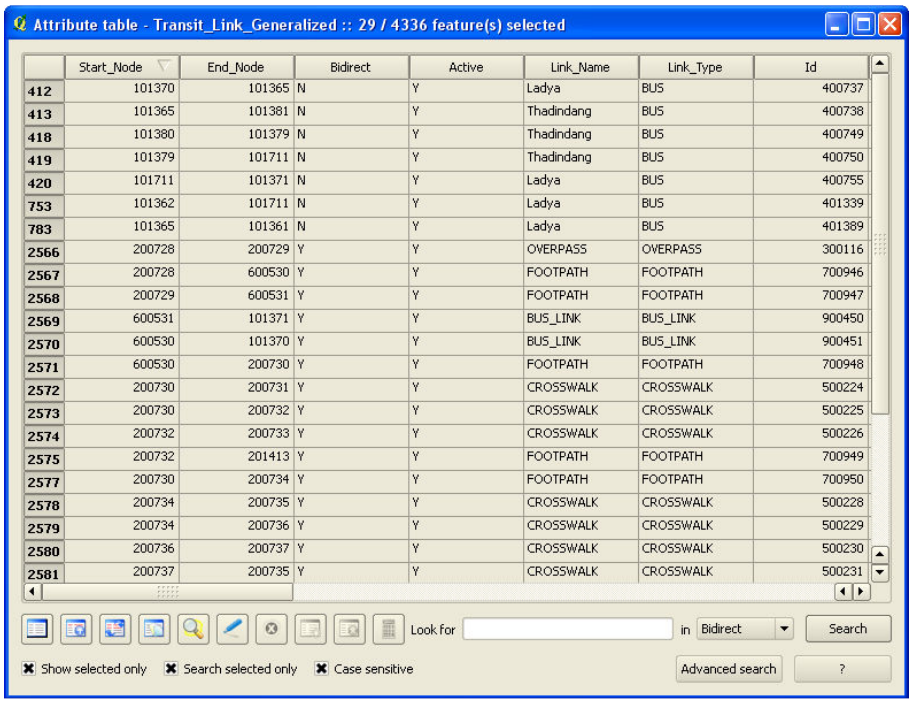
4. ระบุค่า LINK_TYPE โดยมีทั้งหมด 5 ประเภท คือ
   - BUS หมายถึง การเดินทางด้วยรถโดยสารประจำทาง
   - BUS_LINK หมายถึง การเดินทางขึ้นหรือลงรถโดยสารประจำทาง
   - OVERPASS หมายถึง การเดินทางข้ามสะพานลอย
   - CROSSWALK หมายถึง การเดินทางข้ามทางม้าลาย
   - FOOTPATH หมายถึง การเดินทางบนทางเดินเท้า

5. ระบุค่า LINK_NAME โดยเป็นชื่อของ link มีรายละเอียดดังนี้
   - ระบุชื่อของถนน สำหรับ เส้นทางเดินรถโดยสารประจำทาง
   - ระบุชื่อ CROSSWALK สำหรับ ทางม้าลาย
   - ระบุชื่อ OVERPASS สำหรับ สะพานลอย
   - ระบุชื่อ FOOTPATH สำหรับ ทางเดินเท้า
   - ระบุชื่อ BUS_LINK สำหรับ เส้นเชื่อมระหว่างทางเดินเท้ากับถนน

6. ระบุค่า COST ให้กับ link

โดยระบุเป็นระยะทางของ link นั้นๆ ในหน่วยเมตร การคำนวณระยะทางของแต่ละ link จะใช้ความสามารถของระบบฐานข้อมูล Oracle Spatial ในการคำนวณระยะทาง

ให้กับ link ภายหลังจากที่ได้จัดเก็บข้อมูลภูมิศาสตร์ลงในระบบฐานข้อมูลเรียบร้อยแล้ว โดยขอกล่าวถึงวิธีการในการคำนวณระยะทางของ link ใน 3.5.4



	Start_Node	End_Node	Bidirect	Active	Link_Name	Link_Type	Id
412	101370	101365	N	Y	Ladya	BUS	400737
413	101365	101381	N	Y	Thadindang	BUS	400738
418	101380	101379	N	Y	Thadindang	BUS	400749
419	101379	101711	N	Y	Thadindang	BUS	400750
420	101711	101371	N	Y	Ladya	BUS	400755
753	101362	101711	N	Y	Ladya	BUS	401339
783	101365	101361	N	Y	Ladya	BUS	401389
2566	200728	200729	Y	Y	OVERPASS	OVERPASS	300116
2567	200728	600530	Y	Y	FOOTPATH	FOOTPATH	700946
2568	200729	600531	Y	Y	FOOTPATH	FOOTPATH	700947
2569	600531	101371	Y	Y	BUS_LINK	BUS_LINK	900450
2570	600530	101370	Y	Y	BUS_LINK	BUS_LINK	900451
2571	600530	200730	Y	Y	FOOTPATH	FOOTPATH	700948
2572	200730	200731	Y	Y	CROSSWALK	CROSSWALK	500224
2573	200730	200732	Y	Y	CROSSWALK	CROSSWALK	500225
2574	200732	200733	Y	Y	CROSSWALK	CROSSWALK	500226
2575	200732	201413	Y	Y	FOOTPATH	FOOTPATH	700949
2577	200730	200734	Y	Y	FOOTPATH	FOOTPATH	700950
2578	200734	200735	Y	Y	CROSSWALK	CROSSWALK	500228
2579	200734	200736	Y	Y	CROSSWALK	CROSSWALK	500229
2580	200736	200737	Y	Y	CROSSWALK	CROSSWALK	500230
2581	200737	200735	Y	Y	CROSSWALK	CROSSWALK	500231

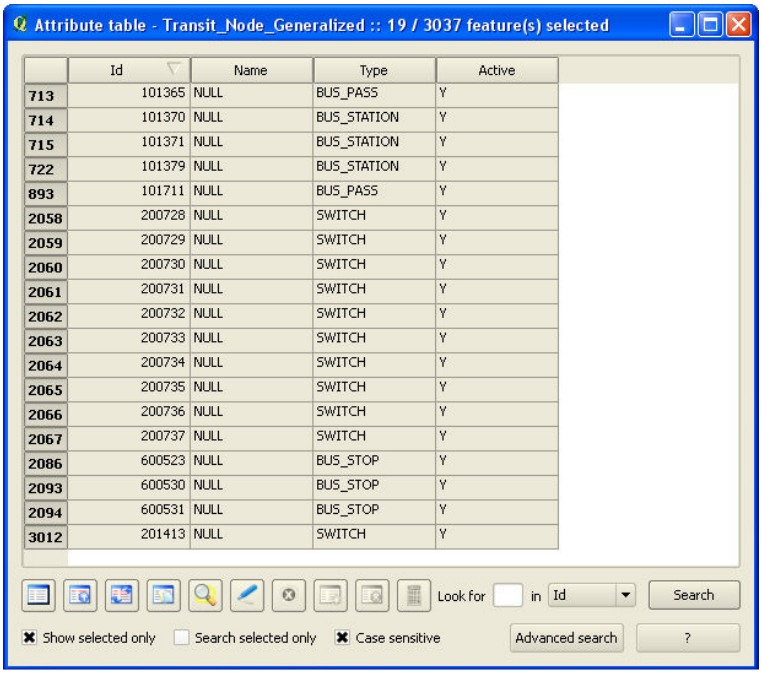
ภาพที่ 3.8 ตัวอย่างข้อมูลอรรถาธิบายของ link

3.4.2.2.2 ข้อมูลอรรถาธิบายของ node (ดังแสดงตัวอย่างในภาพที่ 3.9) มีข้อมูลที่ต้องระบุ ดังนี้

1. เลขรหัสประจำตัวที่ระบุถึง node ต้องไม่ซ้ำกัน (unique) และเลขรหัสประจำตัวจะประกอบด้วยเลข 6 หลัก โดยเลขหลักแรกจะแตกต่างกันตามแต่ละประเภทของ node นั้นๆ ตามที่ได้ออกแบบไว้ในหัวข้อ 3.3.2

2. ระบุค่า TYPE โดยมีทั้งหมด 4 ประเภท คือ

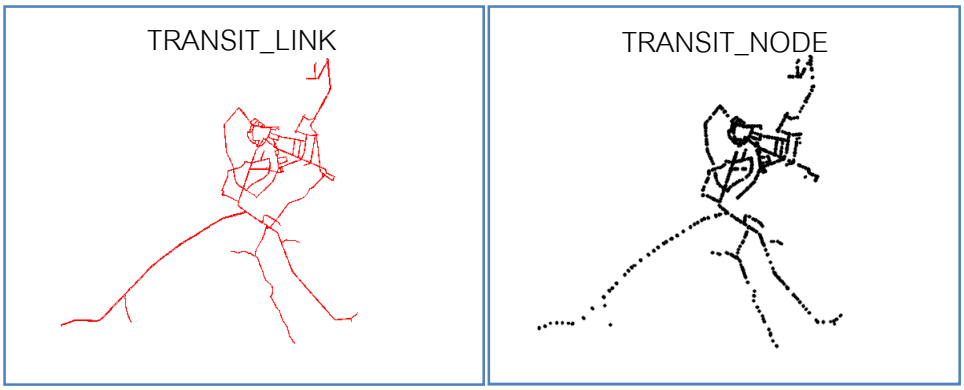
- BUS_STATION หมายถึง ตำแหน่งที่รถโดยสารจอดรับ/ส่ง ผู้โดยสาร
- BUS_PASS หมายถึง ตำแหน่งที่เป็นจุดตัดของถนน
- BUS_STOP หมายถึง ตำแหน่งป้ายจอดรถโดยสารประจำทาง
- SWITCH หมายถึง ตำแหน่งที่เป็นจุดข้ามทางม้าลายหรือสะพานลอย



ภาพที่ 3.9 ตัวอย่างข้อมูลอรรถาธิบายของ node

จากการจัดทำข้อมูลโครงข่ายสำหรับการเดินทางด้วยรถโดยสารประจำทาง จะได้ข้อมูลในรูปแบบของ Shapefile ทั้งหมด 2 ชั้นข้อมูล ดังแสดงในภาพที่ 3.10 คือ

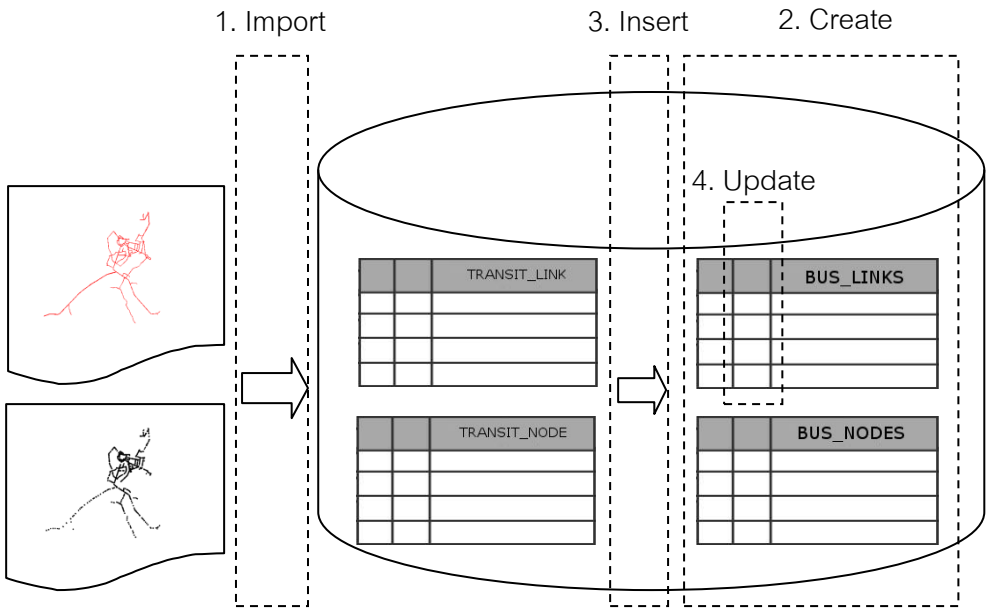
1. ข้อมูล Vector ประเภทจุด (Point) ของจุดเชื่อมต่อของเส้นทางการเดินทาง
2. ข้อมูล Vector ประเภทเส้น (Line) ของเส้นทางรถโดยสารประจำทาง



ภาพที่ 3.10 ข้อมูล Shapefile ที่จัดทำขึ้น

3.5 การสร้างโครงข่ายรถโดยสารประจำทางด้วยระบบฐานข้อมูล Oracle Spatial

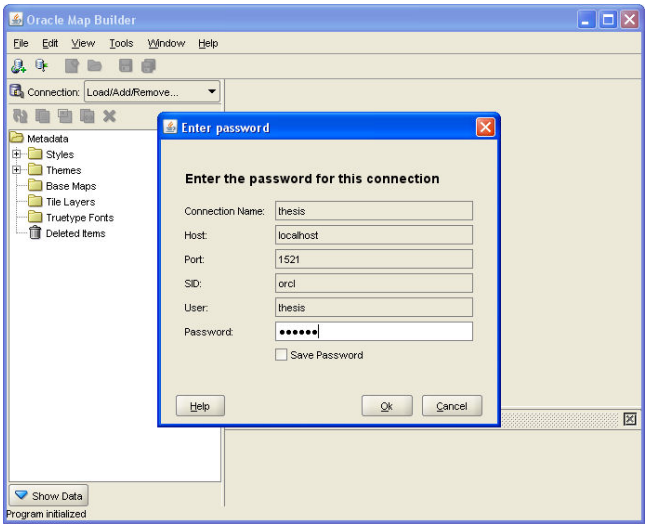
เมื่อจัดทำข้อมูลภูมิศาสตร์สำหรับรถโดยสารประจำทางแล้ว ขั้นตอนต่อไปคือการสร้างโครงข่ายรถโดยสารประจำทางด้วยระบบฐานข้อมูล Oracle โดยมีขั้นตอนคือ การนำเข้าข้อมูล (Import) Shapefile จัดเก็บในระบบฐานข้อมูล, การสร้างตาราง (Create) เพื่อจัดเก็บข้อมูลโครงข่าย, การเพิ่มข้อมูล (Insert) จากตารางของ Shapefile ไปยังตารางของโครงข่าย และการปรับปรุงข้อมูล (Update) ในคอลัมน์ COST ของตารางที่จัดเก็บข้อมูล link ของโครงข่าย ดังแสดงขั้นตอนในภาพที่ 3.11



ภาพที่ 3.11 ขั้นตอนในการสร้างโครงข่ายรถโดยสารประจำทาง
ด้วยระบบฐานข้อมูล Oracle Spatial

3.5.1 การนำเข้าข้อมูล Shapefile เพื่อจัดเก็บในระบบฐานข้อมูล

การนำเข้าข้อมูลเพื่อจัดเก็บลงในระบบฐานข้อมูล Oracle Spatial ใช้ซอฟต์แวร์ชื่อว่า Mapbuilder (NAVTEQ, 2010) ที่มีส่วนติดต่อกับผู้ใช้ดังแสดงในภาพที่ 3.12 รายละเอียดของการใช้งานซอฟต์แวร์ Mapbuilder เพื่อนำเข้าข้อมูลปริภูมิจัดเก็บในระบบฐานข้อมูล Oracle ขอกล่าวเพิ่มเติมในภาคผนวก ก.



ภาพที่ 3.12 ส่วนติดต่อกับผู้ใช้ของโปรแกรม Mapbuilder

ข้อมูล Shapefile ที่นำเข้า คือ ข้อมูล Vector ประเภท Line ตั้งชื่อตารางว่า TRANSIT_LINK และข้อมูล Vector ประเภท Point ตั้งชื่อตารางว่า TRANSIT_NODE ตัวอย่างของตารางข้อมูล Shapefile ที่จัดเก็บในระบบฐานข้อมูล Oracle แล้ว ดังแสดงในภาพที่ 3.13 และ 3.14

ID	START_NODE	END_NODE	LENGTH	BIDIRECT	GEOMETRY	LINK_TYPE	LINK_NAME	ACTIVE
400001	100037	100110	312.41	N	MDSYS.S...	BUS	Ratchadamri	Y
400002	100110	100115	294.22	N	MDSYS.S...	BUS	Ratchadamri	Y
400004	100115	100117	105.69	N	MDSYS.S...	BUS	Ratchadamri	Y
400005	100117	100119	226.34	N	MDSYS.S...	BUS	Ratchadamri	Y
400006	100119	100121	227.63	N	MDSYS.S...	BUS	Ratchadamri	Y
400007	100121	100123	157.3	N	MDSYS.S...	BUS	Ratchadamri	Y
400008	100123	100125	378.53	N	MDSYS.S...	BUS	Ratchadamri	Y
400009	100125	100127	395.97	N	MDSYS.S...	BUS	Ratchadamri	Y
400010	100127	101707	243.88	N	MDSYS.S...	BUS	Ratchadamri	Y
400013	101707	100175	94.82	N	MDSYS.S...	BUS	Silom	Y
400015	100175	100177	171.29	N	MDSYS.S...	BUS	Silom	Y
400016	100177	100179	108.74	N	MDSYS.S...	BUS	Silom	Y
400017	100179	100181	165.02	N	MDSYS.S...	BUS	Silom	Y
400018	100181	100187	660.79	N	MDSYS.S...	BUS	Silom	Y
400021	100187	100189	384.6	N	MDSYS.S...	BUS	Silom	Y
400022	100106	100095	148.46	N	MDSYS.S...	BUS	Phayathai	Y
400023	100095	100093	338.57	N	MDSYS.S...	BUS	Phayathai	Y
400024	100093	100090	210.9	N	MDSYS.S...	BUS	Phayathai	Y
400025	100090	100085	324.14	N	MDSYS.S...	BUS	Phayathai	Y
400028	100085	100083	296.5	N	MDSYS.S...	BUS	Phayathai	Y
400029	100083	100081	166.48	N	MDSYS.S...	BUS	Phayathai	Y
400030	100081	100044	74.9	N	MDSYS.S...	BUS	Phayathai	Y
400031	100043	100069	397.92	N	MDSYS.S...	BUS	Phetburi	Y
400032	100069	100080	74.28	N	MDSYS.S...	BUS	Phayathai	Y
400033	100080	100082	189.59	N	MDSYS.S...	BUS	Phayathai	Y

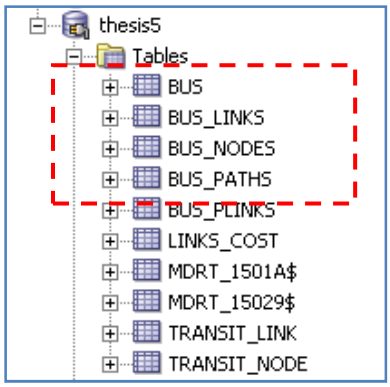
ภาพที่ 3.13 ตัวอย่างข้อมูลของตาราง TRANSIT_LINK

ID	TYPE	GEOMETRY	ACTIVE	NAME
100640	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100645	BUS_PASS	MDSYS.SDO_GEOMETRY...	Y	(null)
100646	BUS_PASS	MDSYS.SDO_GEOMETRY...	Y	(null)
100647	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100648	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100649	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100652	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100653	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100654	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100655	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100656	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100659	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100660	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100661	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100663	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100664	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100665	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100668	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)
100669	BUS_STATION	MDSYS.SDO_GEOMETRY...	Y	(null)

ภาพที่ 3.14 ตัวอย่างข้อมูลของตาราง TRANSIT_NODE

3.5.2 **การสร้างโครงข่ายด้วยระบบฐานข้อมูล** Oracle Spatial

การวิเคราะห์โครงข่ายโดยอาศัยความสามารถของระบบฐานข้อมูล Oracle Spatial โครงสร้างของข้อมูลที่ใช้จัดเก็บข้อมูลโครงข่ายที่ประกอบด้วย link และ node จะต้องมีโครงสร้างข้อมูลตรงตาม Oracle Network Schema ดังที่กล่าวไว้ใน 2.3.1.1 วิธีการที่จะระบุโครงสร้างข้อมูลโครงข่ายจะใช้ฟังก์ชั่นในการบริการจัดการโครงข่ายของระบบฐานข้อมูล Oracle (SDO_NET package) คำสั่งในการสร้างโครงข่ายดังแสดงใน 2.3.1.2 ผลลัพธ์จากการประมวลผลคำสั่งจะได้ตาราง 4 ตาราง คือ BUS_NODES, BUS_LINKS, BUS_PATHS และ BUS_PLINKS ดังแสดงในภาพที่ 3.15 โดยตารางที่จำเป็นต่อการวิเคราะห์โครงข่ายคือ ตาราง BUS_LINKS และ BUS_NODES ที่มีโครงสร้างข้อมูลดังที่กล่าวไว้แล้วใน 3.3.1



ภาพที่ 3.15 ผลลัพธ์จากการสร้างโครงข่าย

3.5.3 การเพิ่มข้อมูลจากตารางของ Shapefile ไปยังตารางของโครงข่าย

ตาราง BUS_LINKS และตาราง BUS_NODES จะเป็นตารางที่ใช้นิยามโครงข่ายซึ่งยังไม่มีข้อมูลใดๆ ขั้นตอนต่อไปคือการเพิ่มข้อมูล (Insert) จากตารางข้อมูล Shapefiles ที่นำเข้า (ตาราง TRANSIT_LINK และ TRANSIT_NODE) ไปยังตาราง BUS_ LINKS และ BUS_NODES โดยใช้คำสั่ง SQL ดังแสดงในภาพที่ 3.16 และ 3.17

```
INSERT INTO BUS_LINKS
 (link_id,link_name,  start_node_id, end_node_id, link_type, active, link_level,geom, cost, bidirected)
SELECT
i.ID, i.LINK_NAME, i.START_NODE, i.END_NODE, i.LINK_TYPE, i.ACTIVE, 1, i.GEOMETRY,1, i.BIDIRECT
FROM
TRANSIT_LINK  i
```

ภาพที่ 3.16 คำสั่งในการเพิ่มข้อมูลไปยังตาราง BUS_LINKS

```
INSERT INTO BUS_NODES (node_id, node_name, node_type, active, geom)
SELECT
 i.ID, i.NAME, i.TYPE, i.ACTIVE, i.GEOMETRY
FROM
TRANSIT_NODE  i
```

ภาพที่ 3.17 คำสั่งในการเพิ่มข้อมูลไปยังตาราง BUS_NODES

3.5.4 การปรับปรุงข้อมูลในคอลัมน์ COST ของตารางที่จัดเก็บข้อมูล link ของโครงข่าย

คำสั่งในการเพิ่มข้อมูลเข้าไปในตาราง BUS_LINKS ค่า COST ของทุกๆ link จะมีค่าเท่ากับ 1 ซึ่งในงานวิจัย คอลัมน์ COST นี้ต้องการให้จัดเก็บค่าของความยาวของ link ในหน่วยเมตร เพื่อจะนำไปใช้ในการวิเคราะห์โครงข่าย

คำสั่งในการปรับปรุง (Update) ค่า COST ของ link ให้มีค่าเท่ากับความยาวของ link นั้นๆ จะใช้ฟังก์ชั่นของระบบฐานข้อมูล Oracle Spatial (SDO_GEOM.SDO_LENGTH) โดยมีคำสั่งดังแสดงในภาพที่ 3.18

```
UPDATE BUS_LINKS A
Set A.COST = (SELECT SDO_GEOM.SDO_LENGTH(c.geom, m.diminfo)
FROM bus_links c, user_sdo_geom_metadata m
WHERE m.table_name = 'BUS_LINKS' AND m.column_name = 'GEOM' AND a.link_id= c.link_id)
```

ภาพที่ 3.18 คำสั่งในการคำนวณความยาวของ link ในหน่วยเมตร

เมื่อทำการเพิ่มข้อมูลในตาราง BUS_LINKS และ BUS_NODES และ update ค่า Cost ในตาราง BUS_LINKS แล้วจะได้ตารางที่จัดเก็บข้อมูลโครงข่าย ดังแสดงตัวอย่างในภาพที่ 3.19 และภาพที่ 3.20

LINK_ID	LINK_NAME	START_NODE_ID	END_NODE_ID	LINK_TYPE	COST	BIDIRECTED	GEOM	ACTIVE
400220	Praram2	100697	100700	BUS	604.41...	N	MDSYS.SDO_GEOMETRY...	Y
400223	Praram2	100691	100697	BUS	503.27...	N	MDSYS.SDO_GEOMETRY...	Y
400226	Praram2	100687	100691	BUS	653.83...	N	MDSYS.SDO_GEOMETRY...	Y
400228	Praram2	100682	100687	BUS	880.98...	N	MDSYS.SDO_GEOMETRY...	Y
400231	Praram2	100674	100682	BUS	1061.4...	N	MDSYS.SDO_GEOMETRY...	Y
400235	Praram2	100670	100674	BUS	430.60...	N	MDSYS.SDO_GEOMETRY...	Y
400237	Praram2	100668	100670	BUS	467.62...	N	MDSYS.SDO_GEOMETRY...	Y
400238	Praram2	100664	100668	BUS	1112.7...	N	MDSYS.SDO_GEOMETRY...	Y
400240	Praram2	100661	100664	BUS	606.80...	N	MDSYS.SDO_GEOMETRY...	Y
400241	Praram2	100660	100661	BUS	353.80...	N	MDSYS.SDO_GEOMETRY...	Y
400242	Praram2	100647	100660	BUS	756.27...	N	MDSYS.SDO_GEOMETRY...	Y
400244	Praram2	100638	100647	BUS	1503.0...	N	MDSYS.SDO_GEOMETRY...	Y
400249	Praram2	100645	100640	BUS	1447.9...	N	MDSYS.SDO_GEOMETRY...	Y
400250	Praram2	100646	100645	BUS	7.3576...	N	MDSYS.SDO_GEOMETRY...	Y
400251	Praram2	100656	100646	BUS	32.506...	N	MDSYS.SDO_GEOMETRY...	Y
400252	Praram2	100659	100656	BUS	734.33...	N	MDSYS.SDO_GEOMETRY...	Y
400254	Praram2	100663	100659	BUS	384.48...	N	MDSYS.SDO_GEOMETRY...	Y
400256	Praram2	100665	100663	BUS	596.64...	N	MDSYS.SDO_GEOMETRY...	Y
400257	Praram2	100669	100665	BUS	1086.1...	N	MDSYS.SDO_GEOMETRY...	Y
400259	Praram2	100672	100669	BUS	494.52...	N	MDSYS.SDO_GEOMETRY...	Y

ภาพที่ 3.19 ตัวอย่างข้อมูลโครงข่ายที่จัดเก็บข้อมูล link ของตาราง BUS_LINKS

NODE_ID	NODE_TYPE	ACTIVE	GEOM	NODE_NAME	PARTITION_ID
101069	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101070	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101071	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101072	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101073	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101074	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101075	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101076	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101077	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101078	BUS_PASS	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101080	BUS_PASS	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101082	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101083	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101085	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101086	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101087	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101088	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)
101089	BUS_STATION	Y	MDSYS.SDO_GEOMETRY(...	(null)	(null)

ภาพที่ 3.20 ตัวอย่างข้อมูลโครงข่ายที่จัดเก็บข้อมูล node ของตาราง BUS_NODES

3.6 การจัดทำตารางข้อมูลอื่น ๆ ที่ต้องใช้ในงานวิจัย

ขั้นตอนต่อไปคือ การสร้างตารางเพิ่มอีก 2 ตาราง ได้แก่ ตาราง LINKS_SPEED เพื่อจัดเก็บอัตราเร็วของรถโดยสารประจำทางในการเดินทางผ่าน link นั้นๆ และตาราง BUS เพื่อจัดเก็บหมายเลขสายรถที่เดินทางผ่าน link นั้นๆ โดยแต่ละตารางมีโครงสร้างการจัดเก็บข้อมูลดังที่กล่าวไว้ใน 3.3.3 เมื่อเพิ่มข้อมูลลงไปในตาราง BUS และ LINKS_SPEED แล้วจะมีตัวอย่างของข้อมูลดังแสดงในภาพที่ 3.21 และ 3.22

LINK_ID	BUS_NO
400001	76
400002	76
400004	15
400004	76
400005	15
400005	76
400006	76
400006	15
400007	76
400007	15
400008	15
400008	76
400009	15
400009	76

ภาพที่ 3.21 ตัวอย่างข้อมูลของตาราง BUS

LINK_ID	SPEED_1	SPEED_2	SPEED_3	SPEED_4	SPEED_5	SPEED_6	SPEED_96	LINK_NAME	LINK_TYPE
701285	5	5	5	5	5	5	5	FOOTPATH	FOOTPATH
900643	5	5	5	5	5	5	5	BUS_LINK	BUS_LINK
701794	5	5	5	5	5	5	5	FOOTPATH	FOOTPATH
701619	5	5	5	5	5	5	5	FOOTPATH	FOOTPATH
500519	5	5	5	5	5	5	5	CROSSWALK	CROSSWALK
400617	10	8	18	16	19	11	17	Prasumen	BUS
400425	16	8	19	17	10	10	10	Vipavadee_R...	BUS
400282	15	8	15	14	9	8	18	Praram2	BUS
400857	13	18	17	19	18	16	14	Jareankroong	BUS
400863	14	13	15	17	14	14	9	Sipraya	BUS
400893	13	15	18	16	12	8	10	Jareankroong	BUS
400916	16	13	12	8	16	12	18	Krungthonburi	BUS
400931	9	13	14	14	17	14	9	Prachatippok	BUS
401052	12	13	18	12	9	9	16	Koongkasem	BUS

ภาพที่ 3.22 ตัวอย่างข้อมูลของตาราง LINKS_SPEED

บทที่ 4

การออกแบบและพัฒนาโปรแกรมประยุกต์

หลังจากออกแบบแบบจำลองข้อมูลและโครงสร้างการจัดเก็บข้อมูลแล้ว ในบทนี้จะกล่าวถึงขั้นตอนการออกแบบและพัฒนาโปรแกรมประยุกต์ในการค้นหาเส้นทางที่เหมาะสมจากการโดยสารรถประจำทาง มีรายละเอียดดังต่อไปนี้

4.1 การวิเคราะห์ความต้องการของระบบ

จากการศึกษาถึงพฤติกรรมการเดินทางของการเดินทางด้วยรถโดยสารประจำทาง (ตามที่กล่าวไว้ใน 3.1) ผู้วิจัยได้วิเคราะห์ความต้องการการใช้งานของผู้ใช้งานปลายทาง และอาศัยการวิเคราะห์จากวัตถุประสงค์และขอบเขตงานวิจัย สรุปประเด็นสำคัญ เพื่อเป็นแนวทางในออกแบบพัฒนาโปรแกรมประยุกต์ ดังนี้

1. การกำหนดจุดเริ่มต้น จุดสิ้นสุดการเดินทางของผู้ใช้งาน สามารถกำหนดได้อย่างอิสระผ่านการกำหนดตำแหน่งบนแผนที่จากหน้าจอคอมพิวเตอร์ ผู้ใช้งานต้องอาศัยการสังเกตและอ่านแผนที่เป็นพอสมควรในการกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง และสามารถเลือกเวลาเริ่มต้นการเดินทางได้จากค่าเวลาที่เตรียมไว้ให้

3. การแสดงผลลัพธ์ของเส้นทางที่แนะนำ จะแสดงทั้งในรูปแผนที่เส้นทางและข้อมูลประกอบการเดินทางอย่างคร่าวๆ ให้กับผู้ใช้งาน กล่าวคือ ระบบควรมีความสามารถแนะนำผู้ใช้งานได้ว่า ต้องมีการเปลี่ยนสายรถประจำทางที่จุดใด และต้องโดยสารรถหมายเลขอะไร

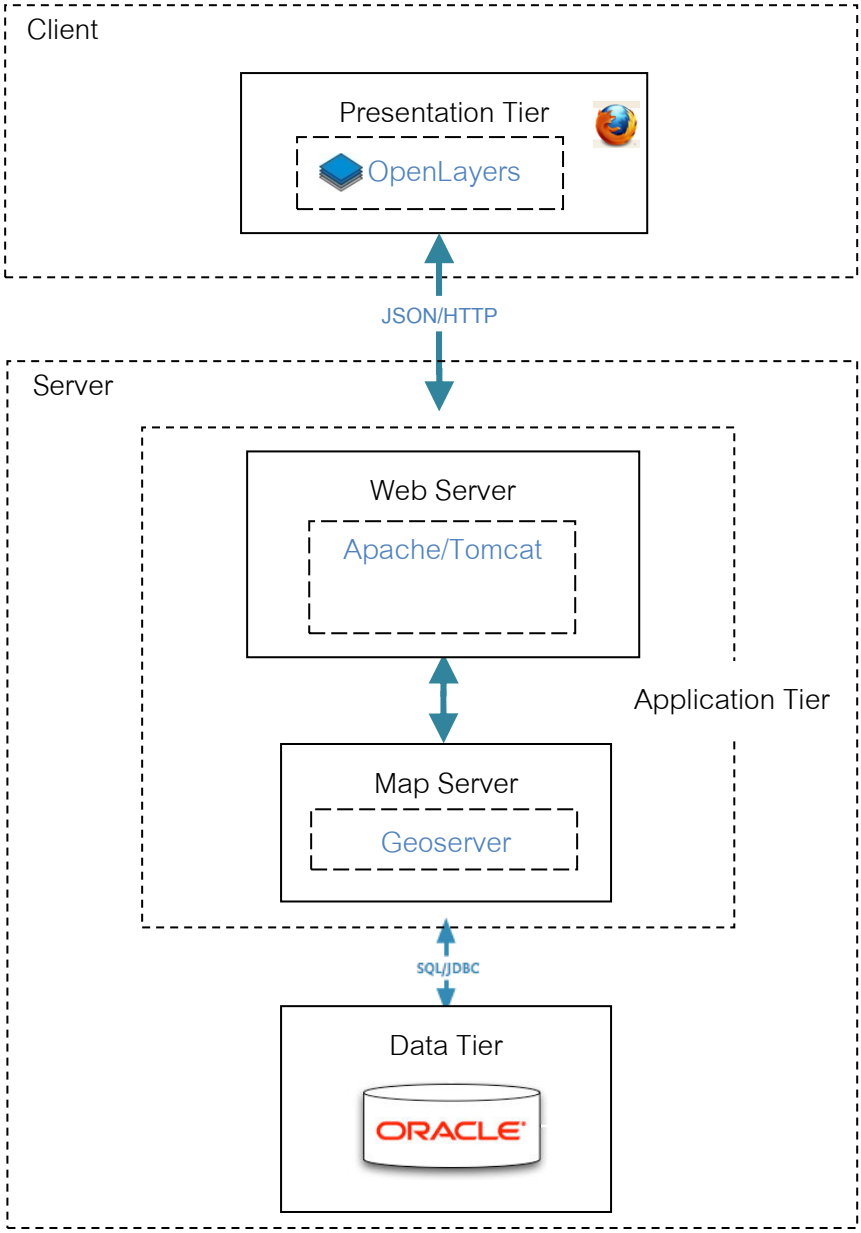
4. โปรแกรมและโครงสร้างข้อมูลต้องเอื้ออำนวยให้ผู้ดูแลระบบสามารถเปลี่ยนแปลงค่าอัตราเร็วเฉลี่ยของรถโดยสารประจำทางที่เดินทางผ่านช่วงถนนต่างๆ ได้

4.2 สถาปัตยกรรมของโปรแกรมประยุกต์

สถาปัตยกรรมของโปรแกรมประยุกต์แบ่งเป็น 3-tier ประกอบด้วย Data tier, Application Tier และ Presentation Tier ดังแสดงในภาพที่ 4.1 โดยแต่ละส่วนมีหน้าที่การทำงานดังต่อไปนี้

- Data Tier เป็นส่วนบริหารจัดเก็บ/จัดการฐานข้อมูล และประมวลผลฟังก์ชั่นการวิเคราะห์โครงข่าย โดยใช้ระบบฐานข้อมูล Oracle 11g จัดเก็บข้อมูลโครงข่ายที่จัดทำในรูปแบบของ Shapefile และจัดเก็บข้อมูลตารางอัตราเร็วของรถโดยสารประจำทางและตารางหมายเลขสายรถ

- Application Tier เป็นส่วนบริหารโปรแกรมประยุกต์ประกอบด้วย 2 ส่วน คือ Web Server และ Map Server โดย Web Server จะเป็นส่วนจัดการคำร้องขอจากผู้ใช้ส่งไปยัง Map Server และทำงานร่วมกับฟังก์ชั่นการวิเคราะห์โครงข่ายของระบบฐานข้อมูล Oracle Spatial โดยใช้เทคโนโลยีของ Servlet และ JSP ที่ทำงานภายใต้โปรแกรมสนับสนุนการทำงาน Servlet เรียกว่า Apache Tomcat ส่วน Map Server จะเป็นส่วนสร้างภาพแผนที่โดยใช้โปรแกรม Geoserver ทำหน้าที่เป็นแม่ข่ายแผนที่โดยเชื่อมโยงกับฐานข้อมูล Oracle เพื่อค้นคืนข้อมูลและสร้างรูปแผนที่
- Presentation Tier เป็นส่วนควบคุมการโต้ตอบกับผู้ใช้ โดยใช้ OpenLayers เป็นตัวควบคุมการทำงานของแผนที่ และจำลองข้อมูลเชิงพื้นที่ของผลลัพธ์จากการวิเคราะห์โครงข่ายของ Server ในรูปแบบของกราฟิก



ภาพที่ 4.1 สถาปัตยกรรมของโปรแกรมประยุกต์ (ดัดแปลงจาก Singh, et al., 2004)

4.3 การพัฒนาโปรแกรมประยุกต์ภูมิสารสนเทศออนไลน์สำหรับการวางแผนการเดินทางด้วยรถโดยสารประจำทาง

การพัฒนาโปรแกรมประยุกต์ในการค้นหาเส้นทางการเดินทางด้วยรถโดยสารประจำทาง แบ่งการพัฒนาโปรแกรมออกเป็นส่วนต่างๆ ตามฟังก์ชั่นการทำงานดังแสดงในภาพที่ 4.2

ภาพที่ 4.2 แสดงขั้นตอนในการพัฒนาส่วนต่างๆ ของโปรแกรมประยุกต์สำหรับการวางแผนการเดินทางด้วยรถโดยสารประจำทาง

4.3.1 การพัฒนาโปรแกรมในส่วนของการแสดงผลแผนที่ของข้อมูลภูมิศาสตร์

เป็นการพัฒนาโปรแกรมเพื่อนำข้อมูลเชิงเรขาคณิตของโครงข่ายการเดินทางด้วยรถโดยสารประจำทางที่จัดเก็บในระบบฐานข้อมูล Oracle Spatial มาแสดงผลแผนที่ผ่านเครือข่ายอินเทอร์เน็ตร่วมกับข้อมูลภาพแผนที่ดาวเทียมจาก Google ให้กับผู้ใช้งานปลายทาง (Client) โดยอาศัยซอฟต์แวร์ Geoserver ทำหน้าที่เป็นแม่ข่ายแผนที่ การกำหนดค่าการใช้งานต่างๆ รวมถึงการเชื่อมต่อแม่ข่ายแผนที่ Geoserver กับระบบฐานข้อมูล Oracle Spatial ขอกล่าวโดยละเอียดในภาคผนวก ข.

การติดต่อร้องขอภาพแผนที่ระหว่างส่วนติดต่อผู้ใช้กับ Geoserver จะใช้ OpenLayers ในการร้องขอ (request) ภาพแผนที่ว่าจะให้มีการแสดงผลอย่างไร และควบคุมการทำงานในระหว่างที่แผนที่มีการย่อ/ขยาย หรือเลื่อนแผนที่ ในงานวิจัยจะร้องขอภาพแผนที่ 2 ชั้นข้อมูล คือ เส้นทาง

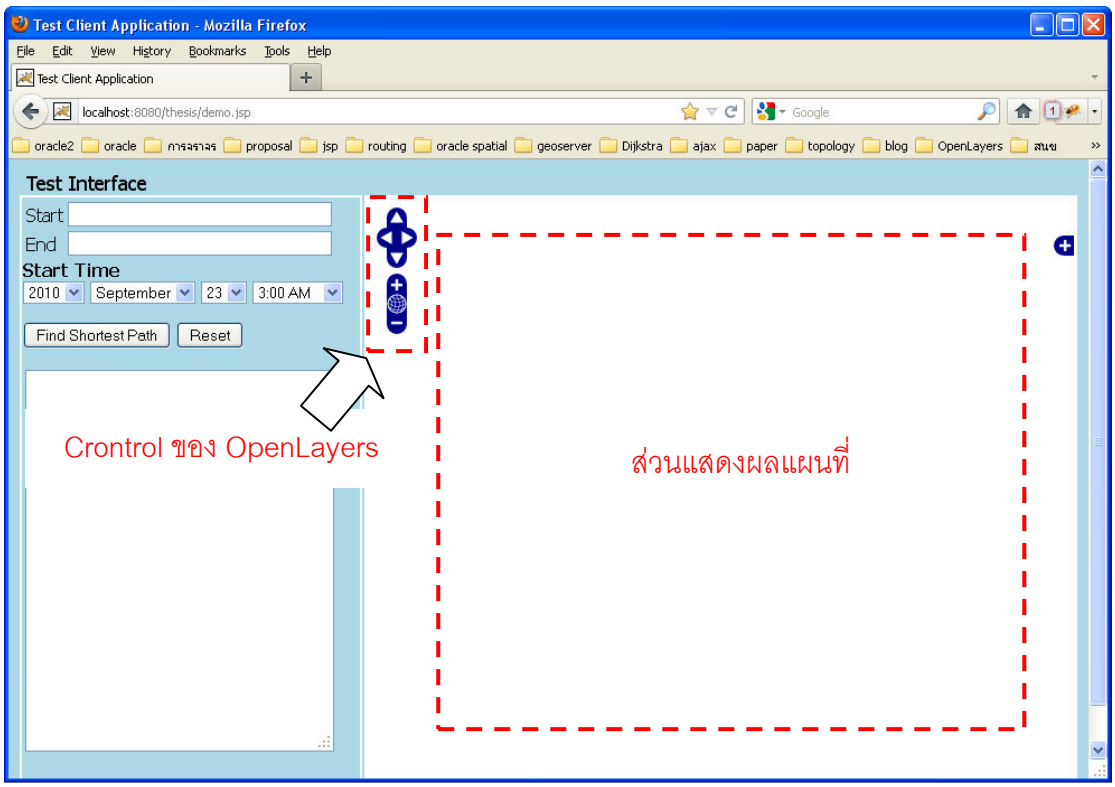
รถโดยสารประจำทางจากตาราง BUS_LINKS แสดงผลร่วมกับภาพแผนที่จากผู้ให้บริการข้อมูล Google เพื่อช่วยในการอ้างอิงเชิงตำแหน่งตัวอย่างของคำสั่งในการร้องขอภาพแผนที่ผ่านมาตรฐาน WMS (Web Map Service) ดังแสดงตัวอย่างคำสั่งในการร้องขอภาพแผนที่ในภาพที่ 4.3

```
var bus_links = new OpenLayers.Layer.WMS(
        'Bus Links',
        'http://127.0.0.1/geoserver/wms',
        {layers: 'Chula:BUS_LINKS', transparent:true},
        { }
    );
var gmap = new OpenLayers.Layer.Google(
        "Google Streets",
        {sphericalMercator: true, numZoomLevels:22},
       { }
    )
map.addLayers([bus_links, gmap]);
```

ภาพที่ 4.3 ตัวอย่างคำสั่งภาษา Javascript ในการร้องขอภาพแผนที่จากแม่ข่ายแผนที่ Geoserver และผู้ให้บริการภาพแผนที่จาก Google

ส่วนควบคุมการทำงานของแผนที่จะอาศัยความสามารถของ OpenLayers โดยออกแบบให้สามารถเลื่อนแผนที่ (Pan) และ ย่อ/ขยายแผนที่ (Zoom In/Zoom Out) ผู้ใช้งานสามารถทำได้ 2 วิธี คือ

- การใช้เม้าส์ (mouse) การเลื่อนแผนที่จะใช้วิธีคลิกเม้าส์ค้างไว้บนแผนที่และทำการลากไปยังบริเวณที่ต้องการ ส่วนการย่อหรือขยายแผนที่ สามารถทำได้โดยการดับเบิ้ลคลิก ณ ตำแหน่งที่ต้องการเพื่อขยายแผนที่ หรือหากเม้าส์มีปุ่มกลาง (Scroll) สามารถใช้ปุ่มกลางในการย่อ/ขยายแผนที่ได้
- การใช้เครื่องมือ (Controls) ของ OpenLayers ประกอบด้วยเครื่องมือในการเลื่อนแผนที่ และเครื่องมือในการย่อ/ขยายแผนที่ ดังแสดงในภาพที่ 4.4



ภาพที่ 4.4 แสดงหน้าจอส่วนของการแสดงผลแผนที่ และตำแหน่งของเครื่องมือที่ใช้ในการเลื่อนและย่อ/ขยายแผนที่

4.3.2 การพัฒนาโปรแกรมในส่วนของการกำหนดจุดเริ่มต้นการเดินทาง จุดสิ้นสุดการเดินทาง และเวลาที่เริ่มต้นออกเดินทาง

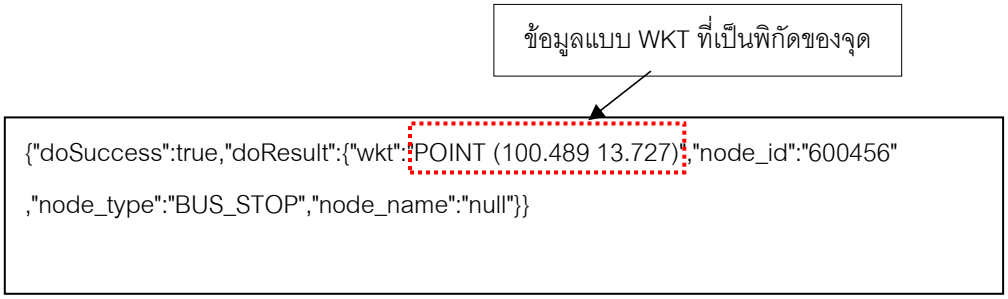
เป็นการพัฒนาโปรแกรม เพื่อให้ผู้ใช้สามารถกำหนดจุดเริ่มต้นการเดินทาง จุดสิ้นสุดการเดินทาง เพื่อส่งค่าไปยังโปรแกรมการประมวลผลเพื่อวิเคราะห์โครงข่าย ซึ่งได้พัฒนาโปรแกรมให้ผู้ใช้งานกำหนดตำแหน่งจุดเริ่มต้นและจุดสิ้นสุดการเดินทางได้จากการคลิกบนแผนที่จากหน้าจอ โดยโปรแกรมจะตรวจสอบค่าพิกัดที่ผู้ใช้ทำการกำหนดตำแหน่งบนแผนที่และส่งค่าพิกัดนั้นไปยังโปรแกรมที่อยู่บนฝั่ง Server เพื่อค้นหา node ที่เป็นจุดเชื่อมต่อการเดินทางของโครงข่ายที่ใกล้ที่สุดจากระบบฐานข้อมูล และนำไปใช้เป็นจุดเริ่มต้นและจุดสิ้นสุดของการเดินทาง

ฟังก์ชั่นที่ใช้ค้นหาจุดที่ใกล้ที่สุดกับตำแหน่งที่ผู้ใช้คลิก อาศัยฟังก์ชั่นของระบบฐานข้อมูล Oracle ในการคำนวณเชิงพื้นที่ (Spatial Operators) โดยใช้ Operator ที่ชื่อ SDO_NN และใช้ฟังก์ชั่น SDO_UTIL.TO_WKTGEOMETRY ในการแปลงค่าพิกัดของจุดเชื่อมต่อการเดินทางที่ค้นคืนได้ให้อยู่ในรูปแบบข้อมูลแบบ WKT (Well Known Text) ตัวอย่างของคำสั่งดังแสดงในภาพที่ 4.5

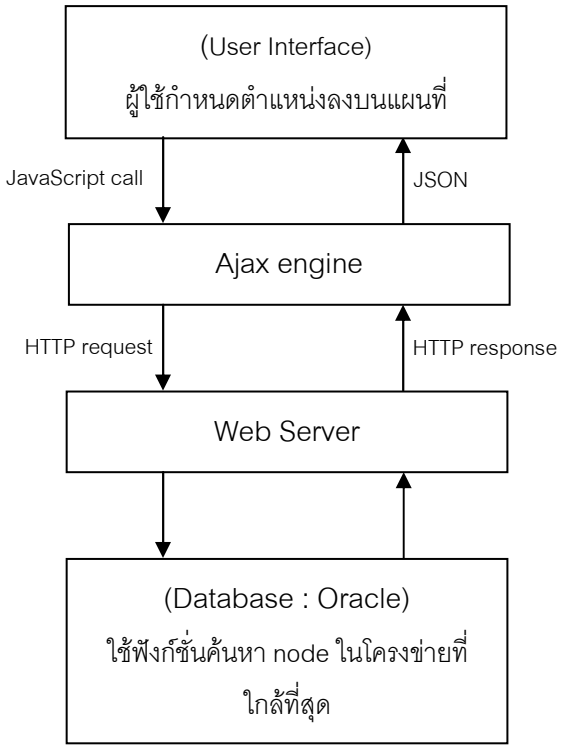
```
SELECT sdo_util.to_wktgeometry(n.geom) as wkt,n.node_id
FROM bus_nodes n
WHERE
SDO_NN(n.geom,
MDSYS.SDO_GEOMETRY(2001,4326,MDSYS.SDO_POINT_TYPE(100.497,13.740,null),null,null),
'sdo_num_res=1', 1 )='TRUE'
```

ภาพที่ 4.5 คำสั่งในการสืบค้นหาจุดที่ใกล้ที่สุดจากจุดที่ทราบค่าพิกัด

การรับส่งข้อมูลค่าพิกัดจากผู้ใช้ที่กำหนดตำแหน่งลงบนแผนที่ไปประมวลผลยังฝั่ง server อาศัยเทคโนโลยีของ AJAX รูปแบบของข้อมูลที่ใช้รับส่งระหว่างเครื่องแม่ข่ายและเครื่องลูกข่ายจะอยู่ในรูปแบบของ JSON (JavaScript Object Notation) ที่ออกแบบไว้ให้เหมาะสมกับงานวิจัย เนื่องจากภาษา Javascript สามารถแปลงข้อความที่ตอบกลับมาจาก Server ในรูปแบบของ JSON ให้เป็น Object ทำให้สามารถเรียกใช้ข้อมูลที่ตอบกลับมาผ่าน method ด้วยตัวแปรอ้างอิงได้ (สรวิศ สุภเวชย์, 2551) โดย object ที่เป็นข้อมูลรูปแบบ WKT จะถูกใช้ในการวาดตำแหน่งพิกัดของจุดให้อยู่ในรูปของกราฟิกด้วย OpenLayers เพื่อแสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดในการเดินทางบนแผนที่ ตัวอย่างของข้อมูลแบบ JSON ที่ตอบกลับมาจาก Server ดังแสดงในภาพที่ 4.6 และมีขั้นตอนการทำงานของโปรแกรมส่วนการกำหนดจุดเริ่มต้น/สิ้นสุดการเดินทางดังแสดงในภาพที่ 4.7

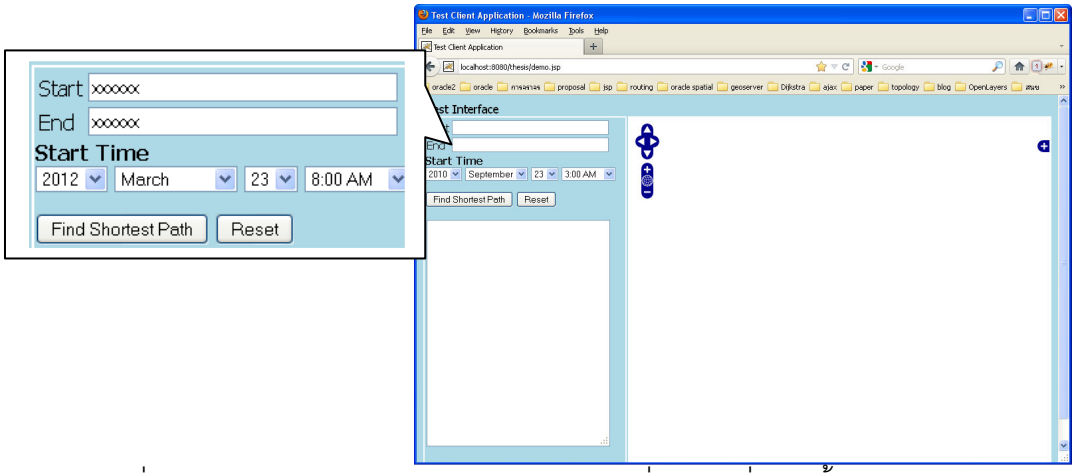


ภาพที่ 4.6 ตัวอย่างข้อมูลในรูปแบบของ JSON ที่ตอบกลับมาจาก Server



ภาพที่ 4.7 ขั้นตอนการทำงานของโปรแกรมการกำหนดจุดเริ่มต้น/สิ้นสุด ในการเดินทาง

ในส่วนของการแสดงผลรหัสประจำตัวของ node ภายในโครงข่ายที่อยู่ใกล้กับตำแหน่งที่ผู้ใช้คลิกบนแผนที่ ออกแบบให้มีฟอร์มรับค่าที่ได้จากการสืบค้นโดยจุดแรกที่ผู้ใช้กำหนดตำแหน่งจะเป็นจุดเริ่มต้นการเดินทาง (Start) และจุดที่สองที่ผู้ใช้กำหนดตำแหน่งจะเป็นจุดสิ้นสุดการเดินทาง (End) ดังแสดงในภาพที่ 4.8 ค่ารหัสประจำตัวของ node เริ่มต้นและสิ้นสุดการเดินทางจะนำไปใช้ในโปรแกรมค้นหาเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุดที่จะกล่าวถึงใน 4.3.3

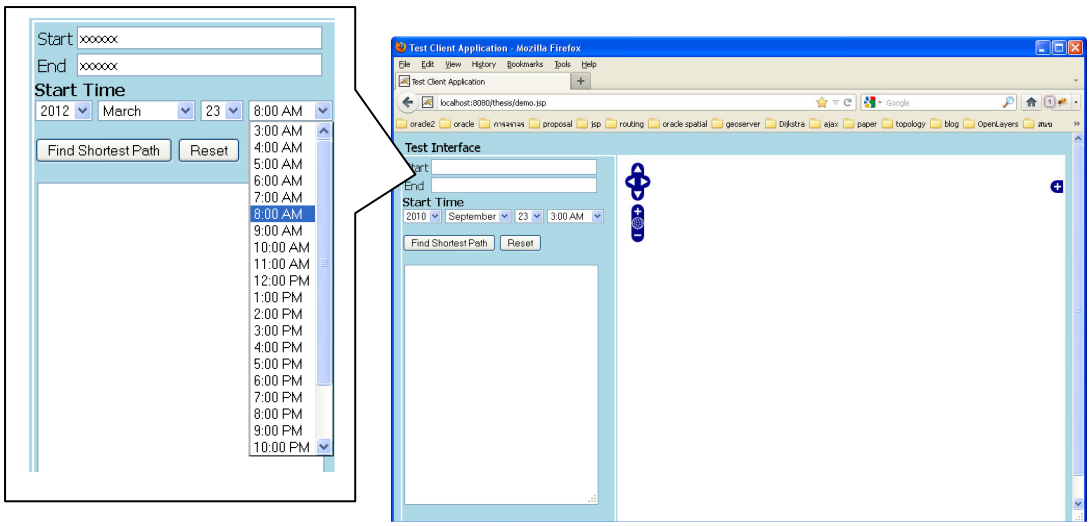


ภาพที่ 4.8 ส่วนแสดงค่ารหัสประจำตัวของ node ที่เป็นจุดเริ่มต้น/สิ้นสุดการเดินทาง

4.3.3 การพัฒนาโปรแกรมเพื่อค้นหาคำตอบจากวิเคราะห์โครงข่าย โดยใช้เวลาในการเดินทางน้อยที่สุด

เป็นการพัฒนาโปรแกรมเพื่อวิเคราะห์โครงข่ายหาเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุด โดยอาศัยความสามารถของฟังก์ชั่นการวิเคราะห์โครงข่ายของระบบฐานข้อมูล Oracle ด้วยขั้นตอนวิธีของ Dijkstra

โปรแกรมจะรับค่าที่ผู้ใช้เลือกจากโปรแกรมกำหนดจุดเริ่มต้น/สิ้นสุดการเดินทาง ได้แก่ เลขรหัสประจำตัวของจุดเริ่มต้น/จุดสิ้นสุดการเดินทาง และเวลาที่เริ่มต้นเดินทาง ดังแสดงในภาพที่ 4.9 เวลาที่เริ่มต้นเดินทางออกแบบให้เป็นตัวเลือกในหลักชั่วโมง เช่น 8.00, 9.00, 10.00 เป็นต้น และออกแบบปุ่มเพื่อส่งค่าที่ได้จากการเลือกของผู้ใช้ (จุดเริ่มต้น, จุดสิ้นสุด, เวลาเริ่มต้นเดินทาง) เพื่อส่งไปประมวลผลยัง Server ด้วยการคลิกที่ปุ่ม Find Shortest Path และหากต้องการล้างค่าที่เลือกจุดเริ่มต้น/สิ้นสุด สามารถทำได้โดยคลิกที่ปุ่ม Reset



ภาพที่ 4.9 ส่วนรับค่าจากผู้ใช้ (รหัสประจำตัวของจุดเริ่มต้น/สิ้นสุดการเดินทาง และเวลาในการเริ่มต้นเดินทาง)

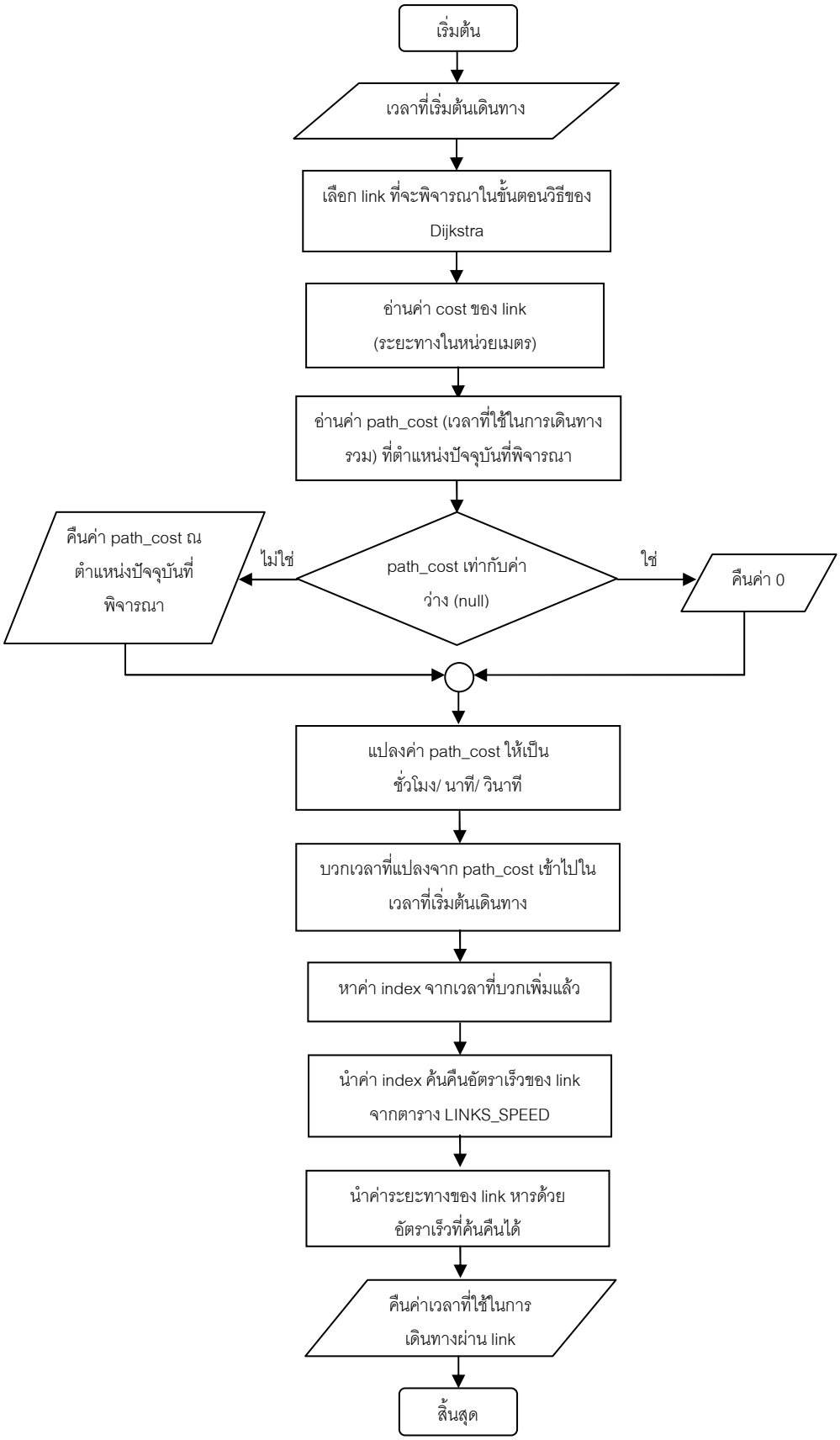
การวิเคราะห์โครงข่ายเพื่อหาเส้นทางการเดินทางด้วยรถโดยสารประจำทางที่ใช้เวลาในการเดินทางน้อยที่สุด ด้วยขั้นตอนวิธีของ Dijkstra หากไม่ได้มีการประยุกต์เพิ่มเติมในขั้นตอนการวิเคราะห์นี้ ค่าใช้จ่าย (Cost) ในการเดินทางผ่าน link ที่จะนำมาพิจารณาจะเป็นค่า Cost ตั้งต้นที่ได้ระบุไว้ใน Network Metadata (ระบุได้เพียงค่าเดียว) ซึ่งในงานวิจัยนี้ค่า Cost ตั้งต้นคือความยาวของ link ในหน่วยเมตร ผลลัพธ์จากการวิเคราะห์โครงข่ายก็จะได้เป็นเส้นทางการเดินทางที่ใช้ระยะทางในการเดินทางที่สั้นที่สุด

ในงานวิจัยต้องการวิเคราะห์โครงข่ายเพื่อหาเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุด และต้องคำนึงถึงอัตราเร็วในการเดินทางผ่าน link นั้นๆ ที่แตกต่างกันไปในแต่ละช่วงวัน ดังนั้น ในการวิเคราะห์โครงข่ายหากต้องการใช้ค่า Cost อื่นๆ ที่ไม่ได้กำหนดไว้ใน Network Metadata ต้องมีการเขียนฟังก์ชั่นประยุกต์ผ่านคลาสที่ชื่อว่า LODAnalysisInfo โดยใช้อินเทอเฟส LinkCostCalculator

ฟังก์ชั่นที่พัฒนาเพิ่มขึ้นในงานวิจัยนี้ ในขั้นตอนวิธีของ Dijkstra โปรแกรมจะทำการตรวจสอบเวลาเริ่มต้นในการเดินทางว่าเป็นเวลาเท่าใด การพิจารณาว่าจะเลือกเดินทางผ่าน link ใด ค่า Cost ที่จะนำมาพิจารณาจะเป็นระยะเวลาในการเดินทางผ่าน link นั้น โดยหาได้จากโปรแกรมจะส่งค่า index ไปสืบค้นในตาราง LINKS_SPEED ว่า link ที่กำลังพิจารณามีอัตราเร็วในการเดินทาง ณ เวลานั้นเป็นเท่าไร ค่า index นี้ได้มาจากการคำนวณเวลาที่รถโดยสารเดินทางไปถึง node ที่กำลังทำการวิเคราะห์ว่าเป็นเวลาเท่าไร เช่น รถโดยสารเดินทางมาถึง node ที่พิจารณาเวลา 0.00 น. พอดี ค่า index จะเท่ากับ 1 (ตัวอย่างค่า index แสดงในตารางที่ 4.1) โปรแกรมก็จะส่งค่า index นี้ไปสืบค้นจากตาราง LINKS_SPEED เพื่อนำค่าอัตราเร็วในการเดินทางมาหารกับระยะทางของ link นั้น เพื่อให้ได้เป็นเวลาในหน่วยวินาที และส่งคืนค่าเวลาในการเดินทางนี้เป็นค่า Cost ของ link ไว้ใช้ในการวิเคราะห์โครงข่ายตามขั้นตอนวิธีของ Dijkstra โดยฟังก์ชั่นที่ผู้วิจัยได้พัฒนาเพิ่มเติมมีขั้นตอนการทำงานดังแสดงในภาพที่ 4.10

ตารางที่ 4.1 ตัวอย่างค่า index ของทุกๆ link จากตาราง LINKS_SPEED ที่แสดงถึงช่วงเวลาต่างๆ ของวัน

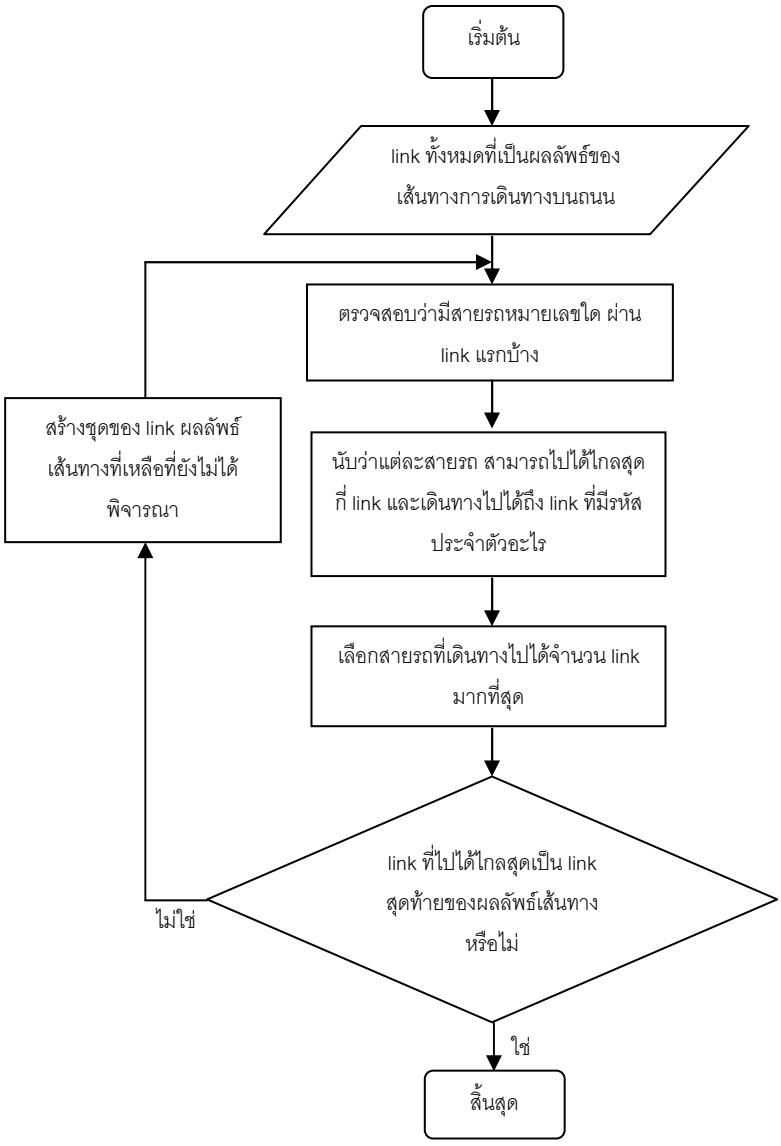
ชื่อรายการ	index	**ช่วงเวลา**
SPEED_1	1	0.00 – 0.14
SPEED _2	2	0.15 – 0.29
SPEED _3	3	0.30 – 0.44
SPEED _4	4	0.45 – 0.59
SPEED _5	5	1.00 – 1.14
…	…	….
SPEED _96	96	23.45 – 23.59



ภาพที่ 4.10 ผังการทำงานของฟังก์ชั่นที่พัฒนาเพิ่มเติมสำหรับการวิเคราะห์โครงข่ายด้วยขั้นตอนวิธีของ Dijkstra (ดัดแปลงจาก ORACLE, 2011)

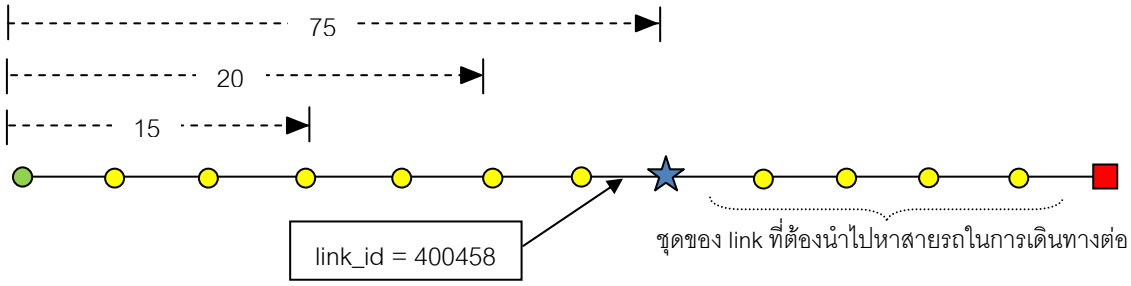
4.3.4 การพัฒนาโปรแกรมในส่วนของการค้นหาหมายเลขสายรถที่ควรโดยสาร

ขั้นตอนวิธีที่ใช้ในการหาสายรถ ได้พัฒนาฟังก์ชั่นในการตรวจสอบชุดของ link ผลลัพธ์ของเส้นทางการเดินทาง โดยจะพิจารณาเฉพาะ link ที่รูปแบบการเดินทางบนถนนเพราะจะมีข้อมูลของหมายเลขสายรถประจำทาง โดยอาศัยแนวคิดที่ว่า สายรถประจำทางสายใดเดินทางไปได้ไกลที่สุด ก็จะแนะนำสายรถนั้นโดยมีขั้นตอนการทำงานของโปรแกรมดังแสดงในภาพที่ 4.11



ภาพที่ 4.11 ผังการทำงานของโปรแกรมในการค้นหาหมายเลขสายรถประจำทางที่ต้องโดยสาร

ขั้นตอนของโปรแกรมจะทำการตรวจสอบ link แรกของผลลัพธ์เส้นทาง (บนถนน) ว่ามีสายรถประจำทางอะไรผ่านบ้าง จากนั้นจะนำสายรถที่สืบค้นได้ ไปหาว่าแต่ละสายรถเดินทางไปได้มากที่สุดกี่ link และเดินทางไปถึงที่ link รหัสประจำตัวอะไร เมื่อตรวจสอบครบทุกหมายเลขสายรถแล้ว สายรถที่จะแนะนำก็คือ สายรถที่เดินทางไปได้จำนวนมาก link ที่สุด เมื่อได้สายรถสายแรกแล้วก็จะทำการตรวจสอบว่า link สุดท้ายที่สายรถนั้นเดินทางไปได้ว่าเป็น link สุดท้ายของผลลัพธ์เส้นทางหรือไม่ ถ้าไม่ใช่ก็แสดงว่าต้องมีการโดยสารรถประจำทางต่อไป โปรแกรมก็จะส่งชุดของ link ที่ยังเหลือการเดินทางต่อ ไปให้ฟังก์ชั่นเพื่อหาสายรถที่ต้องโดยสารต่อโดยอาศัยหลักการเดียวกันกับตอนแนะนำสายรถสายแรก ดังตัวอย่างในภาพที่ 4.12 link แรกของเส้นทางผลลัพธ์มีสายรถที่ผ่านคือ สาย 15, 20 และ 75 โปรแกรมก็จะทำการตรวจสอบว่าแต่ละสายรถเดินทางไปได้จำนวนกี่ link และเดินทางไปได้ถึง link ใด ดังแสดงในตารางที่ 4.2 จะพบว่าสายรถหมายเลข 75 เดินทางไปได้ไกลสุด ก็จะแนะนำให้โดยสารรถหมายเลข 75 จากนั้นโปรแกรมก็จะส่งค่ารหัสของ link สุดท้ายที่สายรถ 75 ไปถึงคือ link ที่มีรหัสประจำตัวเป็น 400458 ไปตรวจสอบว่าเป็น link สุดท้ายของผลลัพธ์เส้นทางการเดินทางหรือไม่ ถ้าไม่ใช่ก็นำ link ที่อยู่ถัดจาก link สุดท้าย (รหัส 400458) ไปจนถึง link สุดท้ายของเส้นทางไปตรวจสอบหาสายรถที่ต้องโดยสารต่ออีกครั้งด้วยวิธีการเดิม



ภาพที่ 4.12 ตัวอย่างของเส้นทางการเดินทางและหมายเลขสายรถที่ผ่าน

ตารางที่ 4.2 จำนวน link ที่เดินทางไปได้ของแต่ละหมายเลขสายรถ

หมายเลขรถ	จำนวน link ที่ไปได้	รหัสของ link สุดท้าย
15	3	400257
20	5	400456
75	7	400458

4.3.5 การพัฒนาโปรแกรมในส่วนของการแสดงผลลัพธ์แผนที่ และรายงานผลเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุด

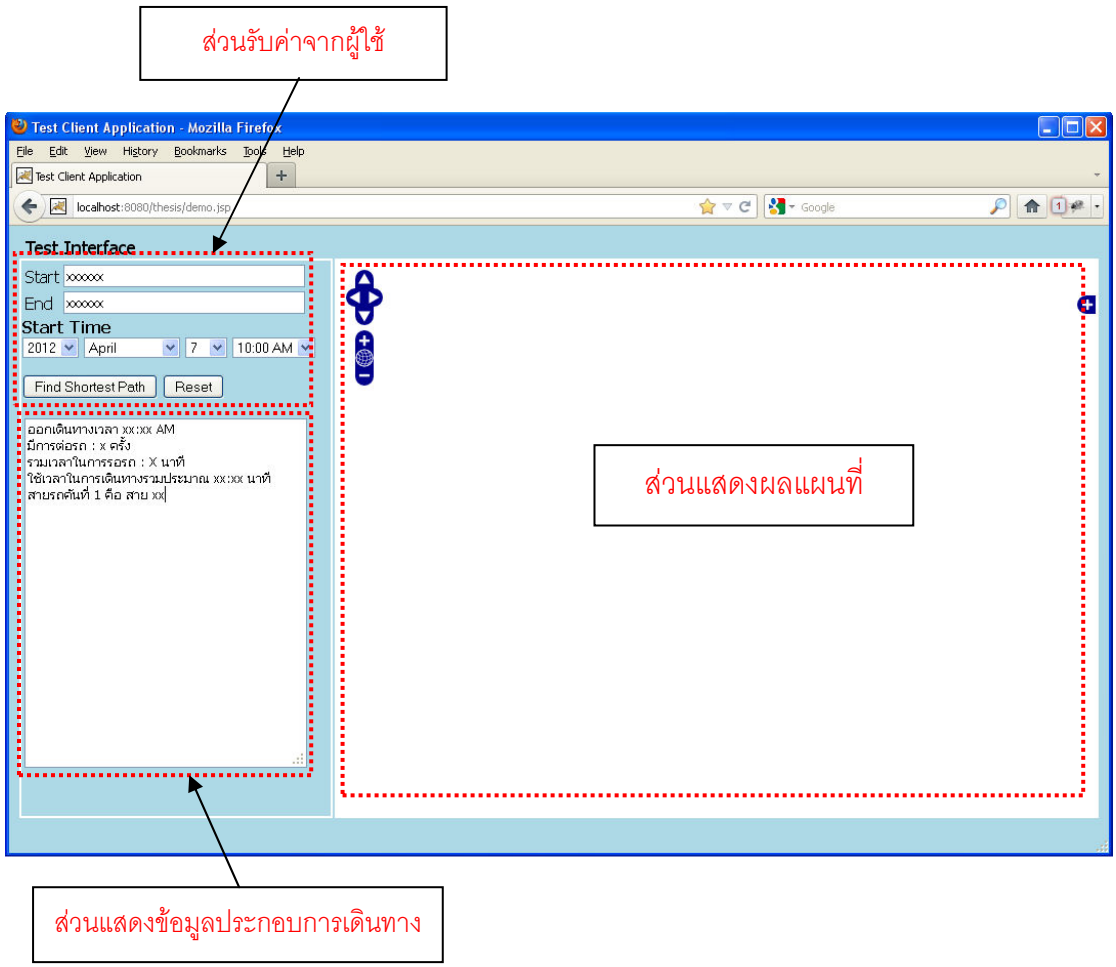
ผลลัพธ์จากการวิเคราะห์โครงข่ายที่ตอบกลับมาจาก Server จะอยู่ในรูปแบบของ JSON ดังแสดงตัวอย่างในภาพที่ 4.13 โปรแกรมจะทำการอ่านข้อความ JSON ให้เป็น Object ของข้อมูลแบบ WKT (Well Known Text) ที่เป็นมาตรฐานที่ใช้ในการกำหนดข้อมูลตำแหน่งพิกัดจุด เส้น และพื้นที่ในรูปแบบของข้อความ และใช้ OpenLayers ในการวาดข้อมูล WKT ให้เป็นรูปกราฟิก เพื่อแสดงผลลัพธ์แผนที่เส้นทางการเดินทางไปยังหน้าจอของฝั่งผู้ใช้

{"doSuccess":true,"doResult":{"wkt":"LINESTRING (100.41367051205 13.6451556997048,
100.413701934172 13.6450796701028, 100.414275353219 13.6454936899075,
100.415690144138 13.6465118109919, 100.420189072078 13.6498916290308,
100.420952737631 13.6504410401774, 100.424998825296 13.653469288132,
100.424951326004 13.6535401074166, 100.428563330092 13.6561803529946,
100.428659793109 13.6562792965343, 100.429945966672 13.6572010317257,
100.433354326612 13.6592267534683, 100.435567616951 13.660528622008,
100.435948109963 13.6599401783184, 100.436066009206 13.6600182903971,
100.439988838572 13.6624033000743, 100.440680156861 13.6628146857662,
100.440798056105 13.6628823820771, 100.445247412772 13.6655537671296,
100.445362632487 13.6656396883684, 100.450434979474 13.6687015884268,
100.450523403906 13.6687588684909)","cost":"3161.29964","travel":[{"busno":"111"
,"endnode":"400727"},{"busno":"82" ,"endnode":"400972"}]}}

ภาพที่ 4.13 ตัวอย่างของผลลัพธ์การวิเคราะห์โครงข่ายที่ตอบกลับมาจาก Server ด้วยรูปแบบข้อมูลแบบ JSON

การแสดงตำแหน่งของจุดที่ต้องมีการเปลี่ยนสายรถ โปรแกรมจะอ่านข้อมูลผลลัพธ์ (JSON) ที่ได้จากการวิเคราะห์โครงข่าย และส่งค่ารหัสประจำตัวของ node ที่เป็น end_node ของ link ที่ต้องมีการเปลี่ยนสายรถไปสืบค้นค่าพิกัดจากตาราง BUS_NODES และให้ OpenLayer ทำการวาดกราฟิกแบบจุดสำหรับแสดงตำแหน่งในแผนที่ของจุดที่ต้องมีการเปลี่ยนสายรถ

ส่วนที่เป็นข้อมูลประกอบการเดินทาง ออกแบบส่วนแสดงผลที่หน้าจอให้เป็นกรอบข้อความ โดยจะแสดงผลเวลาที่ผู้ใช้เริ่มต้นเดินทาง, เวลาที่ต้องใช้ในการเดินทางทั้งหมด, สายรถที่ต้องโดยสาร ดังแสดงในภาพที่ 4.14



ภาพที่ 4.14 แสดงองค์ประกอบของหน้าจอตอบโต้กับผู้ใช้งาน

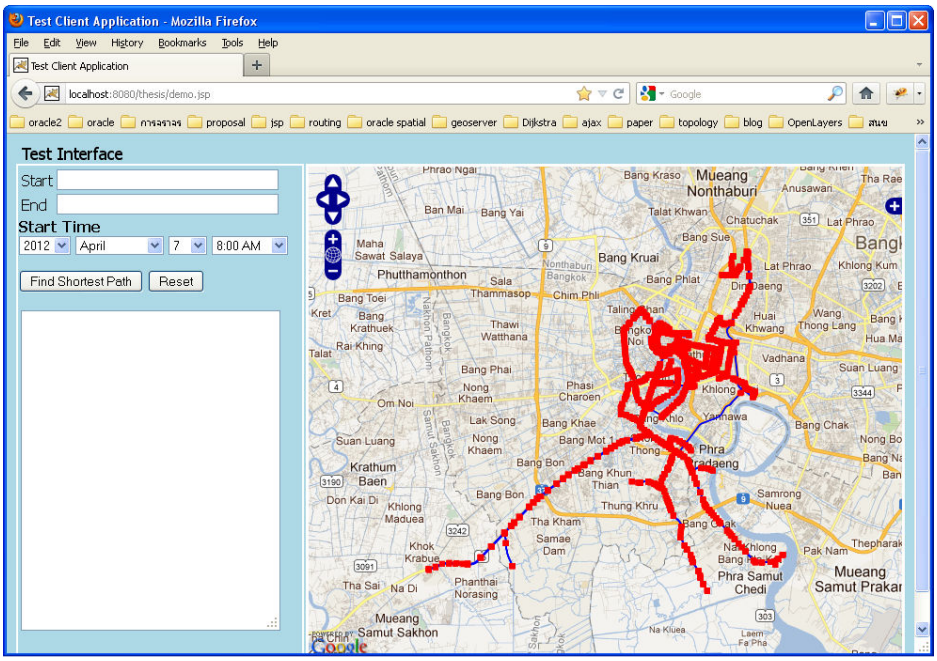
บทที่ 5

การทดสอบโปรแกรมประยุกต์

ในบทนี้จะกล่าวถึงการทดสอบโปรแกรมประยุกต์ที่พัฒนาขึ้น ประกอบด้วย การทดสอบการทำงานในส่วนการแสดงผลแผนที่, การทดสอบการทำงานในส่วนของการกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง, การทดสอบการทำงานในส่วนการวิเคราะห์เลือกเส้นทางที่เหมาะสมและการแสดงผลลัพธ์ และการทดสอบการวิเคราะห์หาหมายเลขสายรถโดยสาร มีรายละเอียดดังนี้

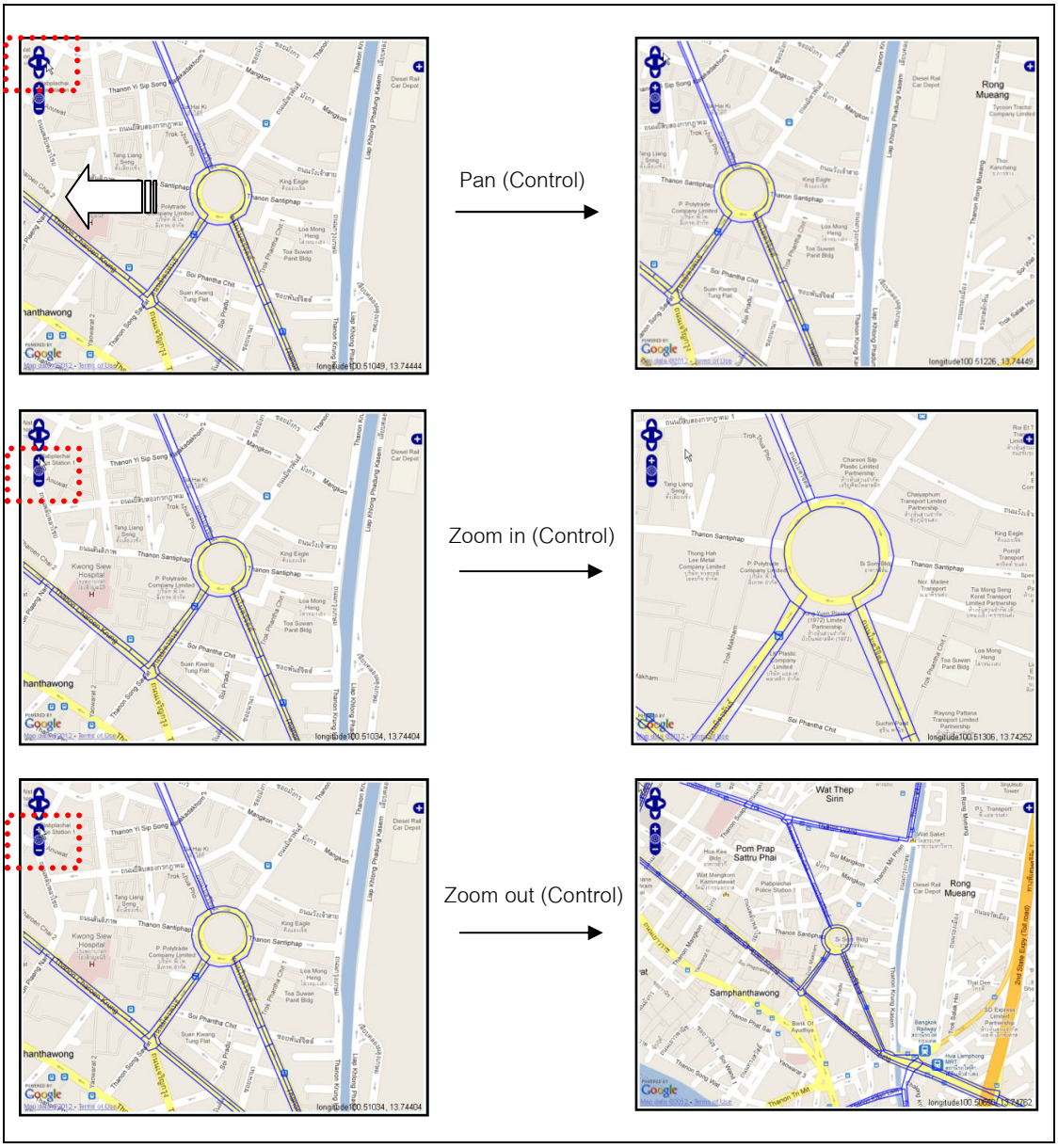
5.1 การทดสอบการทำงานในส่วนของการแสดงผลแผนที่

ส่วนแสดงแผนที่เป็นส่วนที่ติดต่อกับผู้ใช้ในการกำหนดตำแหน่งการเดินทาง และแสดงผลลัพธ์ของเส้นทางที่ได้จากการวิเคราะห์โครงข่าย การควบคุมการทำงานของแผนที่จะใช้ความสามารถของ OpenLayers โดยผู้วิจัยได้ทดสอบการแสดงผลแผนที่ของข้อมูลโครงข่ายการเดินทางด้วยรถโดยสารประจำทาง ซึ่งประกอบด้วย link และ node โดยแสดงผลร่วมกับภาพแผนที่จากผู้ให้บริการ Google ได้ผลดังแสดงในภาพที่ 5.1

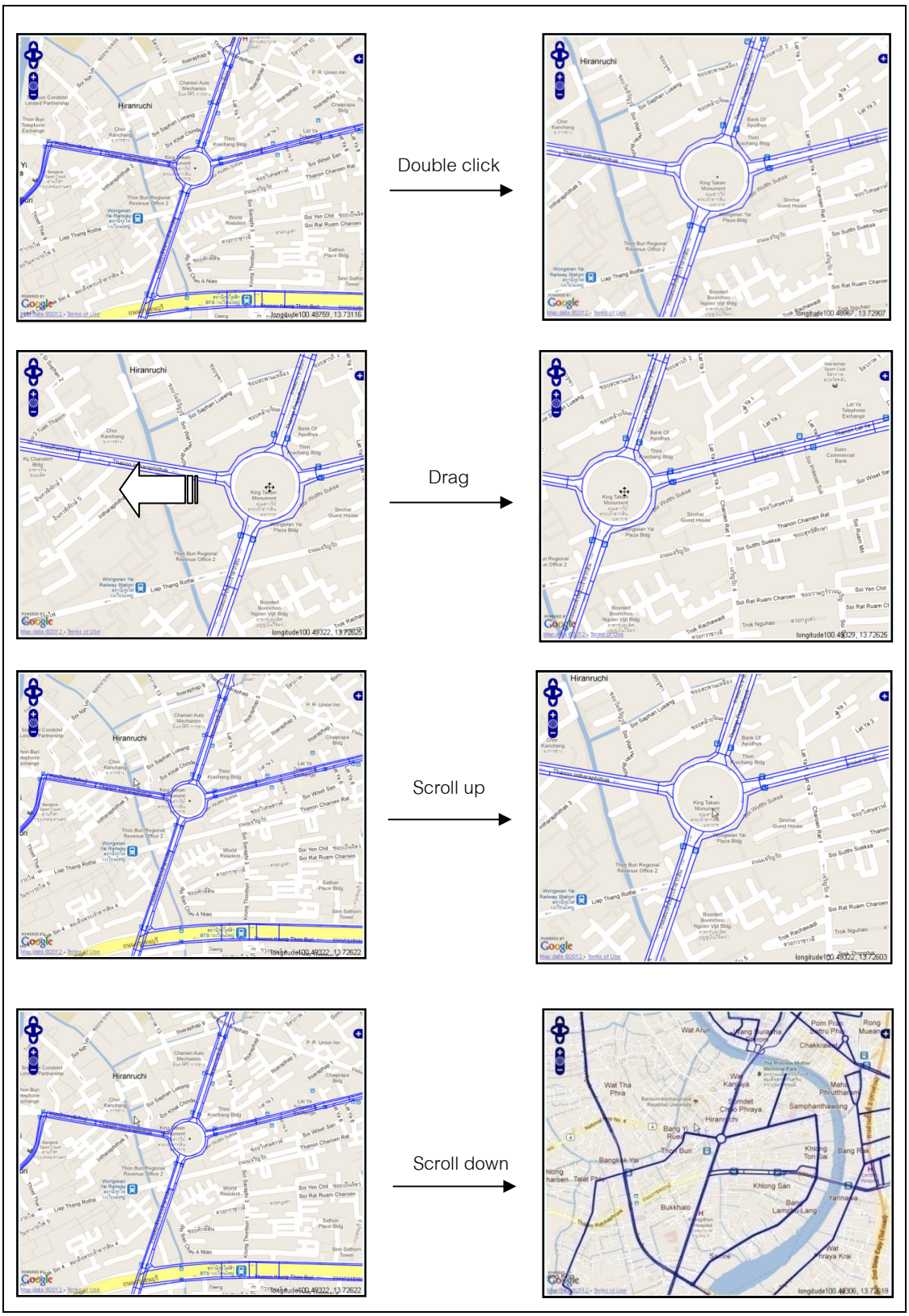


ภาพที่ 5.1 แสดงการทดสอบส่วนการแสดงแผนที่ของโปรแกรมประยุกต์ ขณะเริ่มต้นทำงาน

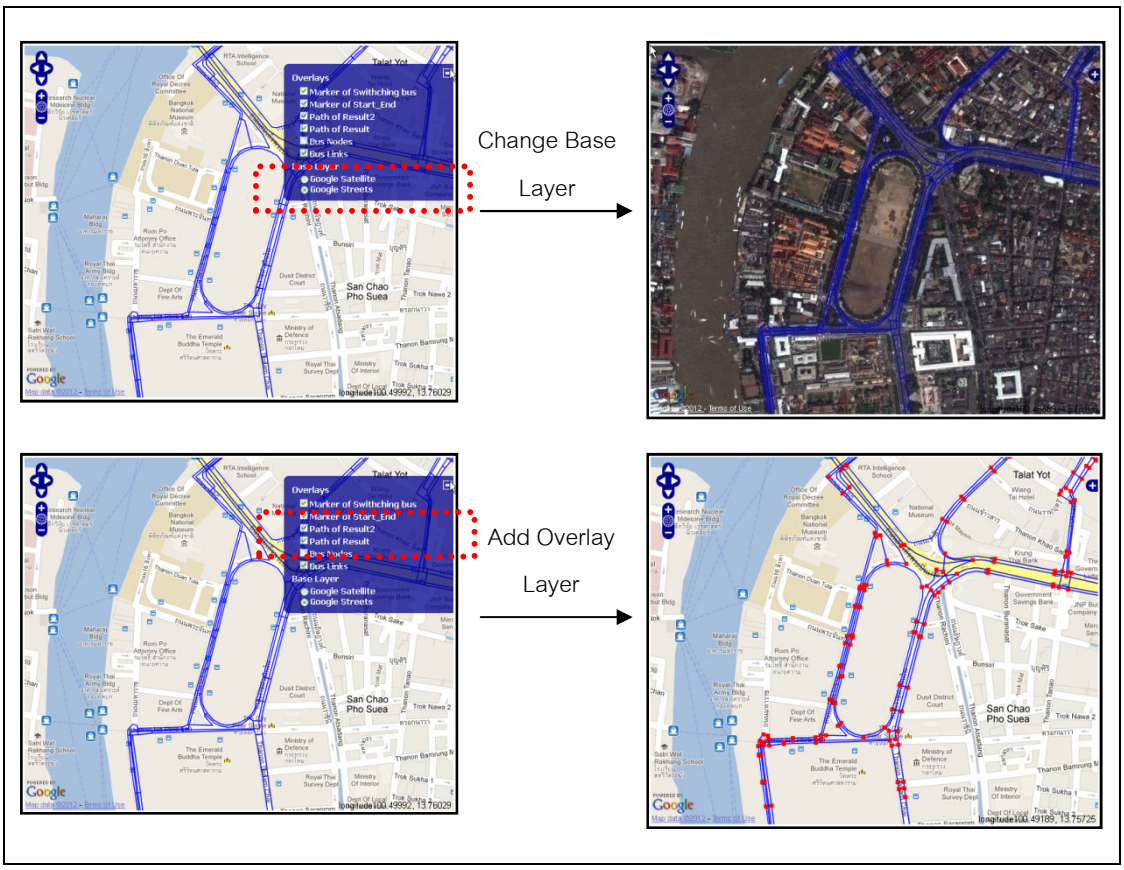
การทดสอบส่วนควบคุมการทำงานของแผนที่ ได้ทดสอบฟังก์ชั่นการทำงานร่วมกับแผนที่ ได้แก่ การเลื่อนแผนที่ (Pan), การย่อ/ขยายแผนที่ (Zoom in, Zoom out) และการเลือกเปิด/ปิดการแสดงผลของชั้นแผนที่ โดยแบ่งเป็นการทดสอบการทำงานจากการใช้เม้าส์ในการจัดการแผนที่และการใช้เครื่องมือ (Controls) ในการจัดการแผนที่ พบว่า สามารถควบคุมการทำงานของแผนที่ผ่านทางหน้าจอของผู้ใช้งานได้ตรงตามฟังก์ชั่นการทำงานที่ต้องการ ผลลัพธ์ของการควบคุมการทำงานของแผนที่ดังแสดงในภาพที่ 5.2 ภาพที่ 5.3 และภาพที่ 5.4



ภาพที่ 5.2 แสดงการทดสอบส่วนควบคุมการทำงานของแผนที่โดยการใช้ Control ของ OpenLayers



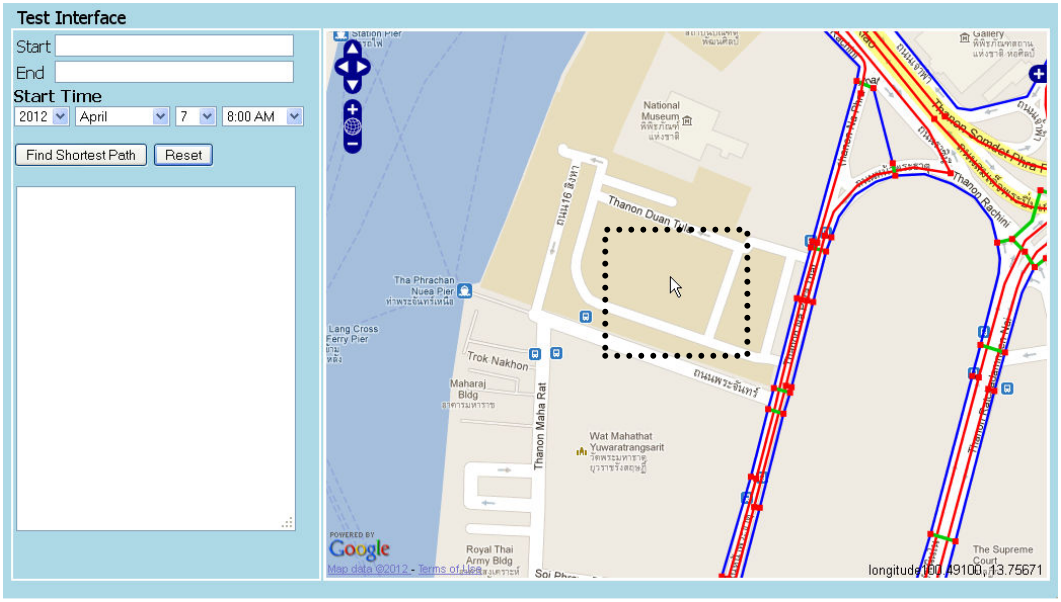
ภาพที่ 5.3 แสดงการทดสอบส่วนควบคุมการทำงานของแผนที่โดยการใช้เม้าส์



ภาพที่ 5.4 แสดงการทดสอบส่วนควบคุมการทำงานของแผนที่ในการเปิด/ปิดการแสดงผลชั้นข้อมูล

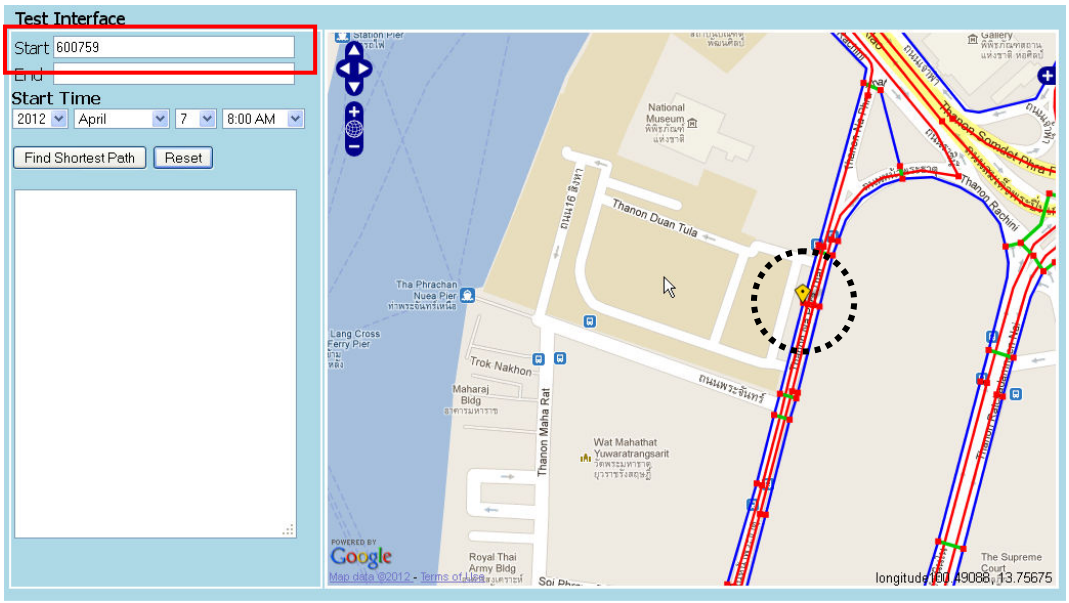
5.2 การทดสอบการทำงานในส่วนของการกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง

ทดสอบการทำงานของการกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง จากการคลิกกำหนดตำแหน่งลงบนแผนที่ โดยได้ทดลองสมมุติให้ผู้ใช้ต้องการเริ่มต้นเดินทางจากมหาวิทยาลัยธรรมศาสตร์ ท่าพระจันทร์ ได้ทำการใช้เครื่องมือในการจัดการแผนที่ขยายแผนที่ไปบริเวณที่ต้องการดังแสดงในภาพที่ 5.5



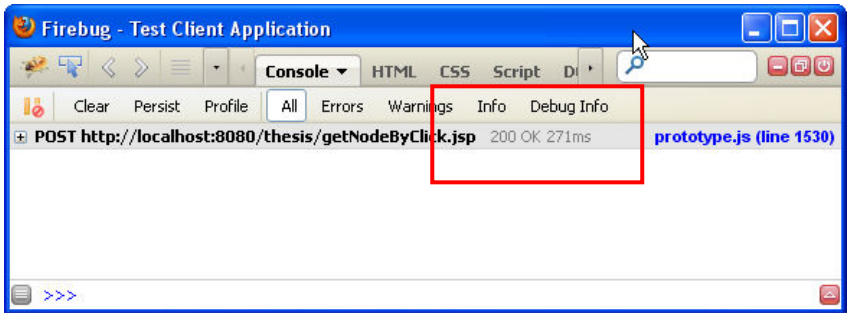
ภาพที่ 5.5 แสดงบริเวณที่ต้องการให้เป็นจุดเริ่มต้นการเดินทาง คือ พื้นที่ภายใน มหาวิทยาลัยธรรมศาสตร์ ท่าพระจันทร์

เมื่อคลิกลงบนแผนที่จะปรากฏเครื่องหมายแสดงตำแหน่ง (marker สีเหลือง) โดยจะเป็นตำแหน่งของ node ภายในโครงข่ายที่สืบค้นคืนได้และใกล้กับตำแหน่งที่ผู้ใช้เลือกมากที่สุด และแสดงหมายเลขรหัสประจำตัวของ node ที่เป็นจุดเริ่มต้นการเดินทางในฟอร์มรับข้อมูลในช่องของ Start ดังแสดงในภาพที่ 5.6



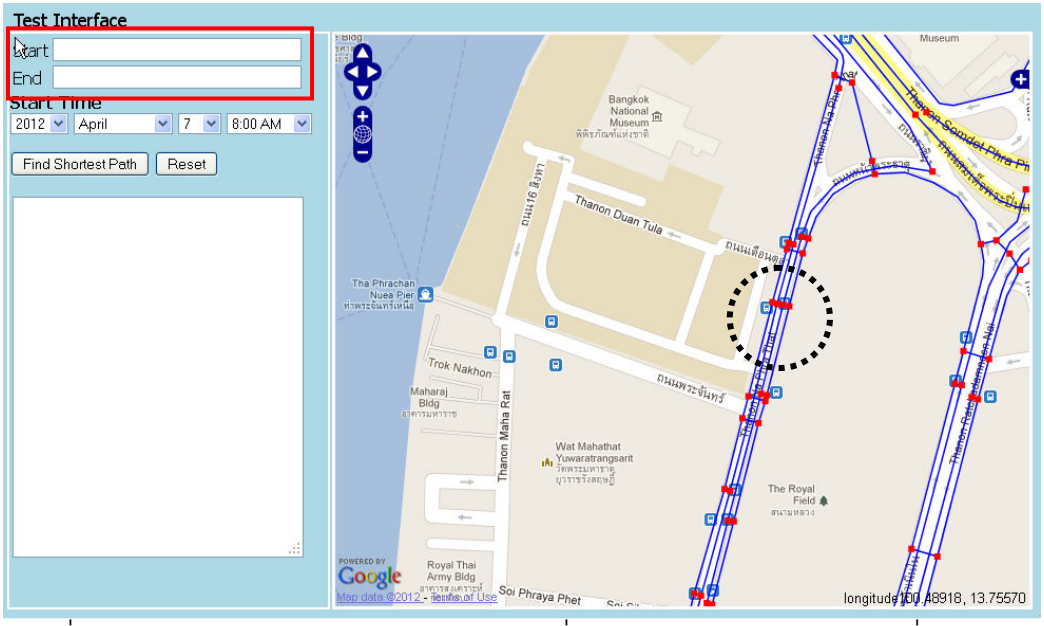
ภาพที่ 5.6 ผลลัพธ์จากการกำหนดตำแหน่งเพื่อระบุจุดเริ่มต้นการเดินทาง

ในงานวิจัยนี้โปรแกรมที่ใช้ทำหน้าที่เป็นเว็บเบราว์เซอร์ คือ Mozilla Firefox เวอร์ชั่น 11.0 ที่มี Extensions ที่ชื่อว่า Firebug ซึ่งเป็นส่วนสนับสนุนการพัฒนาโปรแกรมประยุกต์ ผู้วิจัยได้ใช้ Firebug ในการตรวจสอบเวลาที่ใช้ในการประมวลผลคำร้องขอจากผู้ใช้ โดยเวลาที่ใช้ในการส่งคำร้องขอไปยัง Server เพื่อสืบค้นหา node ในโครงข่ายที่ใกล้กับตำแหน่งที่ผู้ใช้คลิกลงบนแผนที่มากที่สุด และ Server ส่งผลลัพธ์กลับมายังผู้ใช้เพื่อแสดงผลลัพธ์แผนที่ของจุดที่จะเป็นจุดเริ่มต้นการเดินทาง ใช้เวลาในการประมวลผลลัพธ์ทั้งสิ้น 271 มิลลิวินาที ดังแสดงในภาพที่ 5.7



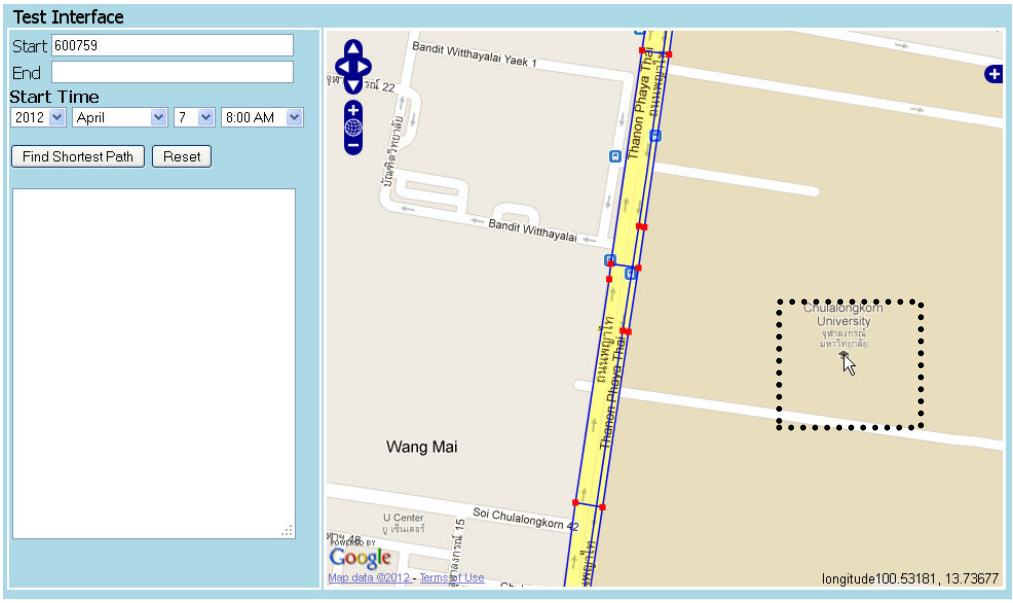
ภาพที่ 5.7 เวลาที่ใช้ในการประมวลผลลัพธ์ของการกำหนดตำแหน่งที่เป็นจุดเริ่มต้นการเดินทาง

การยกเลิกจุดเริ่มต้นหรือจุดสิ้นสุดการเดินทางที่ได้เลือกไว้ สามารถทำได้โดยการคลิกที่ปุ่ม Reset เมื่อทดสอบการยกเลิกจุดเริ่มต้นการเดินทางที่เลือกไว้ พบว่า เครื่องหมายที่แสดงตำแหน่งของจุดเริ่มต้นการเดินทางและหมายเลขรหัสประจำตัวของจุดนั้นที่อยู่ในฟอร์มรับค่าในช่อง Start จะถูกล้างออกไป ดังแสดงในภาพที่ 5.8

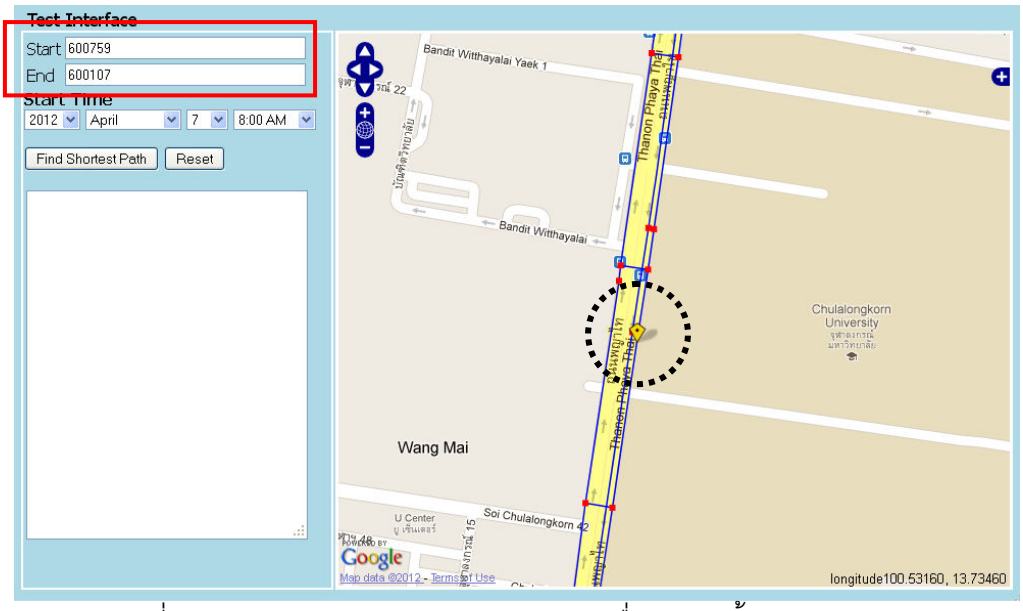


ภาพที่ 5.8 ผลลัพธ์ของการยกเลิกการกำหนดจุดเริ่มต้นการเดินทางโดยการคลิกที่ปุ่ม Reset

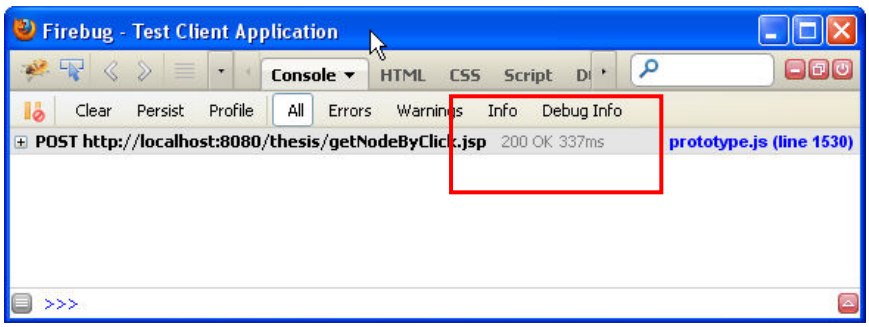
ทดลองกำหนดจุดที่เป็นจุดสิ้นสุดการเดินทาง โดยสมมุติเป็นบริเวณภายในพื้นที่ของจุฬาลงกรณ์มหาวิทยาลัยดังแสดงในภาพที่ 5.9 เมื่อทำการคลิกกำหนดตำแหน่งจะปรากฏเครื่องหมาย (marker) แสดงตำแหน่งที่เป็นจุดสิ้นสุดการเดินทางภายในโครงข่าย และมีเลขรหัสประจำตัวของจุดสิ้นสุดการเดินทางแสดงในฟอร์มรับค่าที่ช่อง End ดังแสดงในภาพที่ 5.10 และใช้เวลาในการประมวลผลคำร้องขอเท่ากับ 337 มิลลิวินาที ดังแสดงในภาพที่ 5.11



ภาพที่ 5.9 แสดงบริเวณที่ต้องการให้เป็นจุดสิ้นสุดการเดินทางคือ ภายในจุฬาลงกรณ์มหาวิทยาลัย



ภาพที่ 5.10 ผลลัพธ์จากการกำหนดตำแหน่งเพื่อระบุจุดสิ้นสุดการเดินทาง



ภาพที่ 5.11 เวลาที่ใช้ในการประมวลผลลัพธ์ของการกำหนดตำแหน่งที่เป็นจุดสิ้นสุดการเดินทาง

5.3 การทดสอบการทำงานในส่วนของวิเคราะห์โครงข่ายรถโดยสารประจำทางด้วยระบบฐานข้อมูล Oracle Spatial

5.3.1 การทดสอบประสิทธิภาพของฟังก์ชั่นการวิเคราะห์โครงข่ายที่พัฒนาเพิ่มเติมขึ้น

ในงานวิจัยจะใช้ความสามารถของระบบฐานข้อมูล Oracle Spatial ในการวิเคราะห์โครงข่าย และได้มีการพัฒนาฟังก์ชั่นเพิ่มเติมให้มีการนำค่า Cost ของตารางหลัก (ระยะทางของ link) มาคำนวณร่วมกับอัตราเร็วของรถโดยสารจากตาราง LINKS_SPEED ให้ได้เป็นเวลาในหน่วยวินาที เพื่อใช้ในกระบวนการวิเคราะห์โครงข่ายตามขั้นตอนวิธีของ Dijkstra ดังนั้นเพื่อเป็นการเปรียบเทียบประสิทธิภาพ (เวลาที่ใช้ในการประมวลผล) ของขั้นตอนวิธีในการวิเคราะห์โครงข่ายแบบต่างๆ จึงได้ออกแบบการทดสอบการทำงานของฟังก์ชั่นการวิเคราะห์โครงข่ายเป็น 2 วิธี คือ

1. การวิเคราะห์หาเส้นทางการเดินทางที่ใช้ระยะทางในการเดินทางน้อยที่สุด โดยค่า Cost ที่ใช้ในการวิเคราะห์โครงข่ายด้วยขั้นตอนวิธีของ Dijkstra จะเป็นค่า Cost ที่ระบุไว้ในตารางหลักของโครงข่าย (ระยะทางของ link)

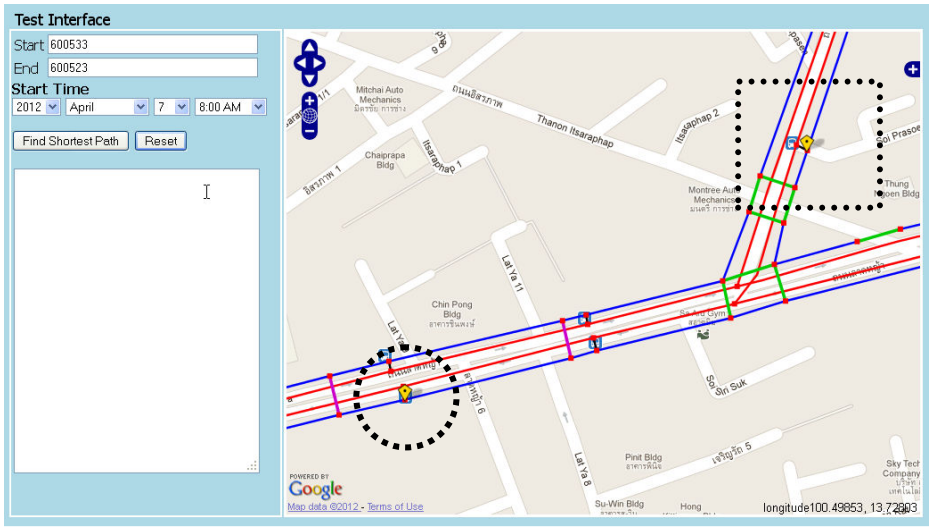
2. การวิเคราะห์หาเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุด โดยค่า Cost ที่ใช้ในการวิเคราะห์โครงข่ายด้วยขั้นตอนวิธีของ Dijkstra จะเป็นค่า Cost ของตารางหลัก (ระยะทาง) ที่คำนวณร่วมกับอัตราเร็วของรถโดยสารจากตาราง LINKS_SPEED ผ่านฟังก์ชั่นที่พัฒนาเพิ่มเติมขึ้น

การทดสอบทั้ง 2 วิธีดังกล่าว จะแยกเป็น 2 กรณีคือ กรณีที่หนึ่งจะกำหนดให้จุดเริ่มต้นและจุดสิ้นสุดการเดินทางอยู่ใกล้กัน ส่วนกรณีที่สองจะกำหนดให้จุดเริ่มต้นและจุดสิ้นสุดการเดินทางอยู่ไกลกัน โดยมีผลการทดสอบดังแสดงดังนี้

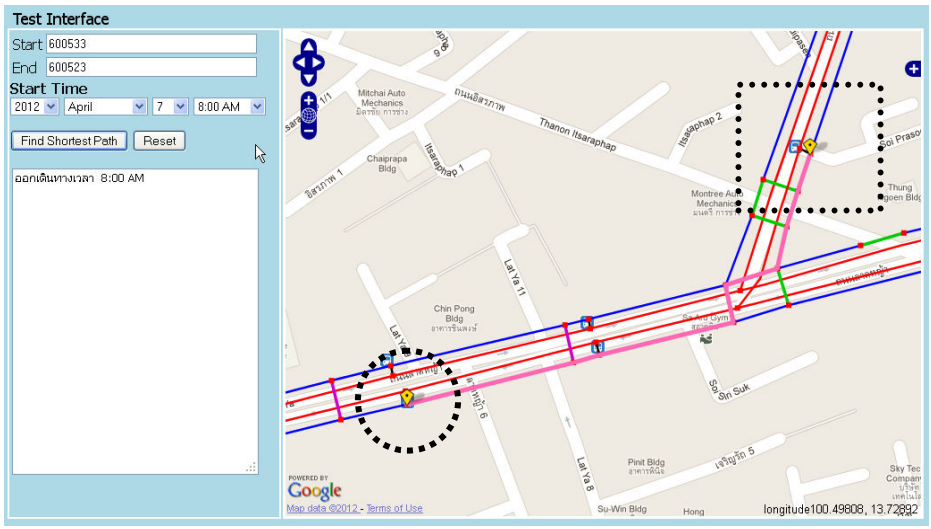
5.3.1.1 กรณีกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทางให้อยู่ใกล้กัน

5.3.1.1.1 การวิเคราะห์โครงข่ายโดยใช้ค่า Cost จากตารางหลักของโครงข่าย

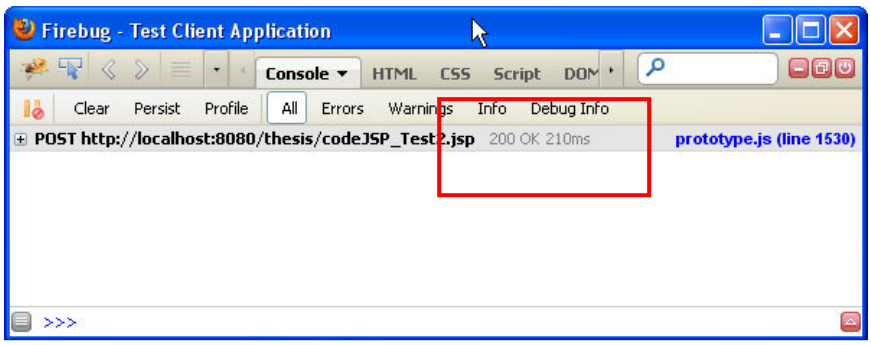
สมมุติให้จุดเริ่มต้นอยู่ที่ป้ายรถประจำทางหน้าปากซอยลาดหญ้า 6 จุดสิ้นสุดอยู่ที่ป้ายรถประจำทางหน้าปากซอยประเสริฐโชคดังแสดงในภาพที่ 5.12 จากนั้นทดลองวิเคราะห์โครงข่ายเพื่อหาเส้นทางการเดินทางที่ใช้ระยะทางน้อยที่สุดโดยการคลิกที่ปุ่ม Find Shortest Path โปรแกรมจะทำการประมวลผลได้เส้นทางผลลัพธ์ดังแสดงในภาพที่ 5.13 และใช้เวลาในการประมวลผลเท่ากับ 210 มิลลิวินาทีดังแสดงในภาพที่ 5.14



ภาพที่ 5.12 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่อยู่ใกล้กัน เพื่อทดสอบการวิเคราะห์โครงข่ายโดยใช้ค่า Cost จากตารางหลัก (ระยะทาง) ของโครงข่าย



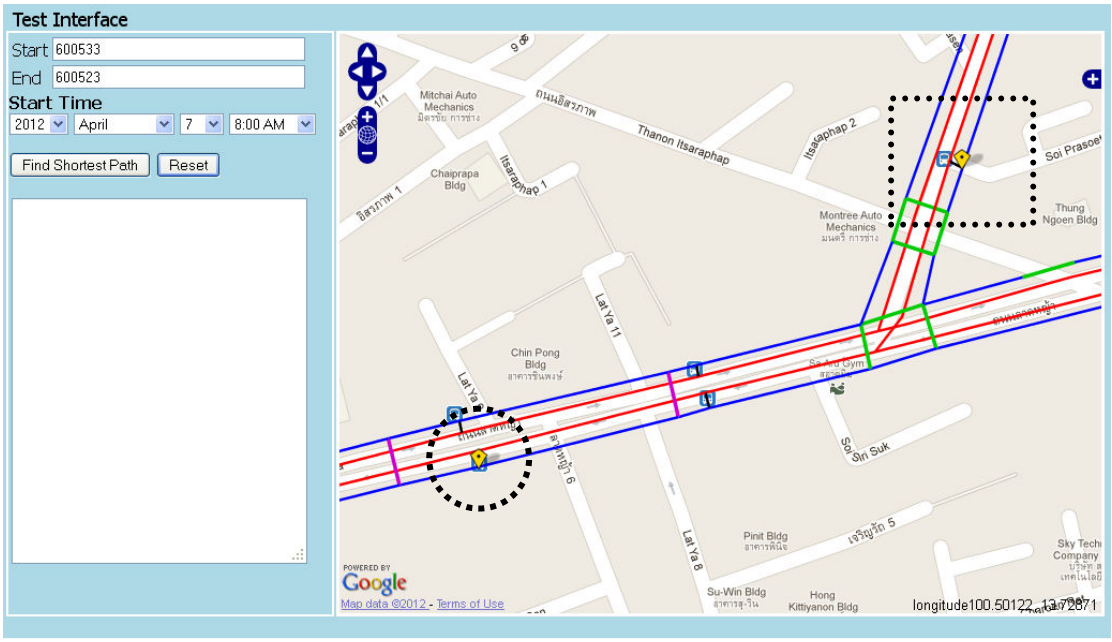
ภาพที่ 5.13 ผลลัพธ์การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้ระยะทางสั้นที่สุดจากจุดที่อยู่ใกล้กัน



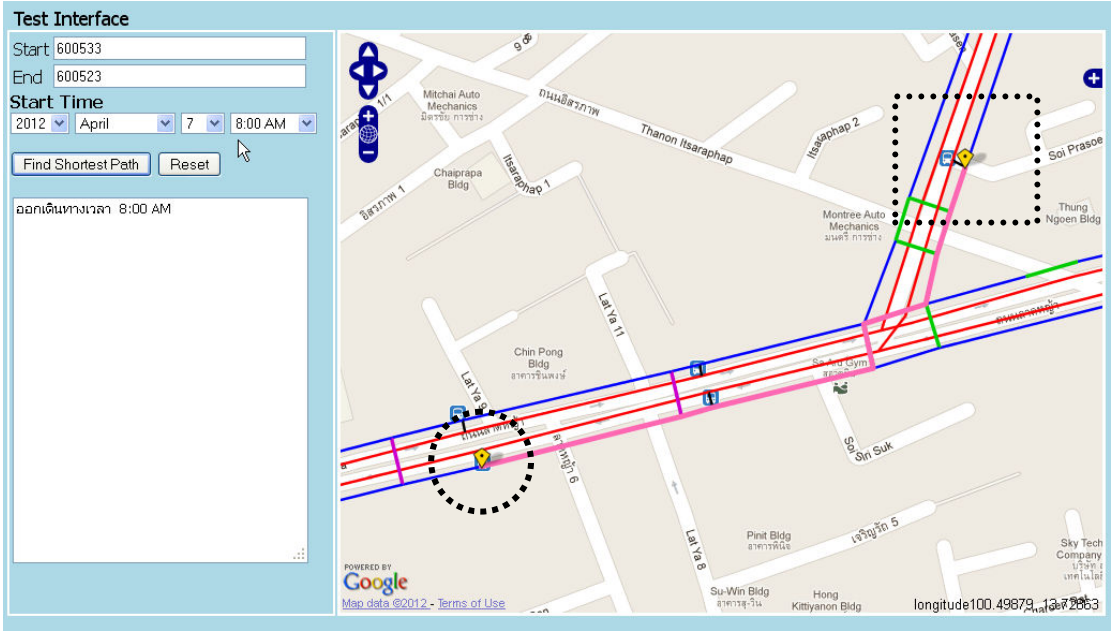
ภาพที่ 5.14 เวลาที่ใช้ในการประมวลผลจากการวิเคราะห์โครงข่ายที่ใช้ค่า Cost จากตารางหลักของโครงข่าย และจุดเริ่มต้น/สิ้นสุดการเดินทางอยู่ใกล้กัน

5.3.1.1.2 การวิเคราะห์โครงข่ายโดยใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) ร่วมกับข้อมูลในตารางเสริม (อัตราเร็ว) ของโครงข่าย

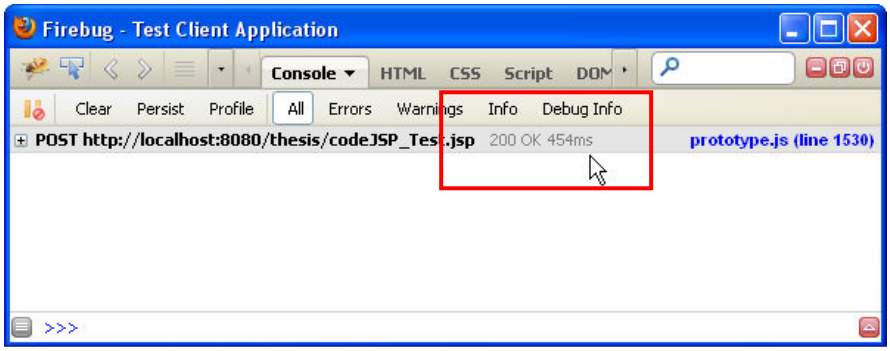
จุดเริ่มต้นและจุดสิ้นสุดการเดินทางกำหนดให้เป็นจุดเดียวกันกับการทดสอบในหัวข้อ 5.3.1.1.1 ดังแสดงในภาพที่ 5.15 จากนั้นทดลองวิเคราะห์โครงข่ายเพื่อหาเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุด โปรแกรมจะทำการประมวลผลได้เส้นทางผลลัพธ์ดังแสดงในภาพที่ 5.16 และใช้เวลาในการประมวลผลเท่ากับ 454 มิลลิวินาทีดังแสดงในภาพที่ 5.17



ภาพที่ 5.15 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่อยู่ใกล้กัน เพื่อทดสอบการวิเคราะห์โครงข่ายโดยใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) และตารางเสริม (อัตราเร็ว)



ภาพที่ 5.16 ผลลัพธ์การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาน้อยที่สุดจากจุดที่อยู่ใกล้กัน

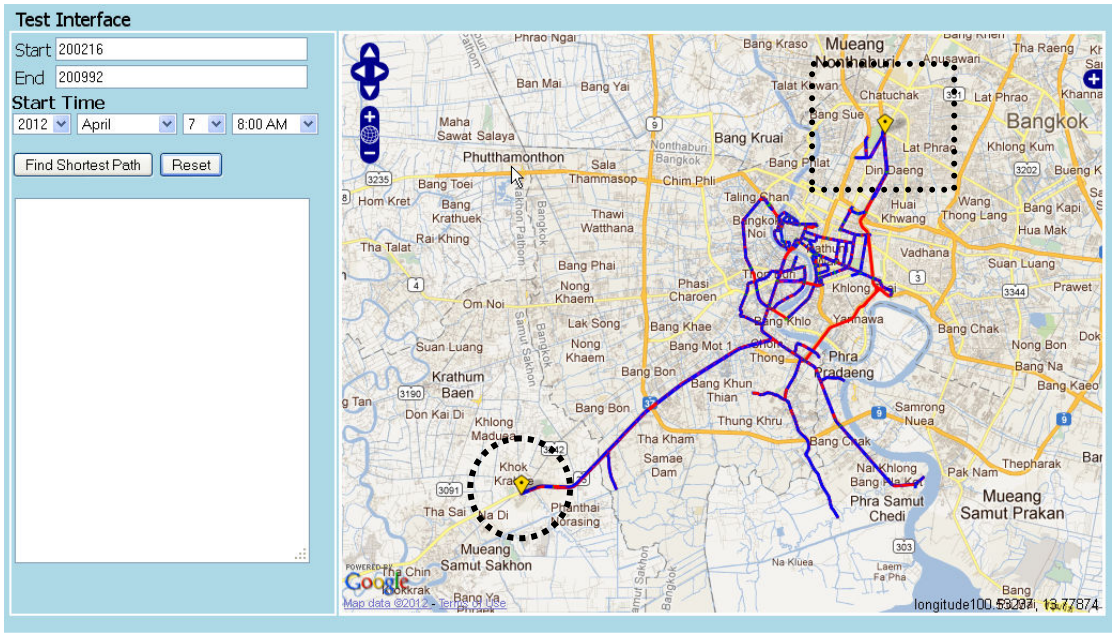


ภาพที่ 5.17 เวลาที่ใช้ในการประมวลผลจากการวิเคราะห์โครงข่ายที่ใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) และตารางเสริม (อัตราเร็ว) และจุดเริ่มต้น/สิ้นสุดการเดินทางอยู่ใกล้กัน

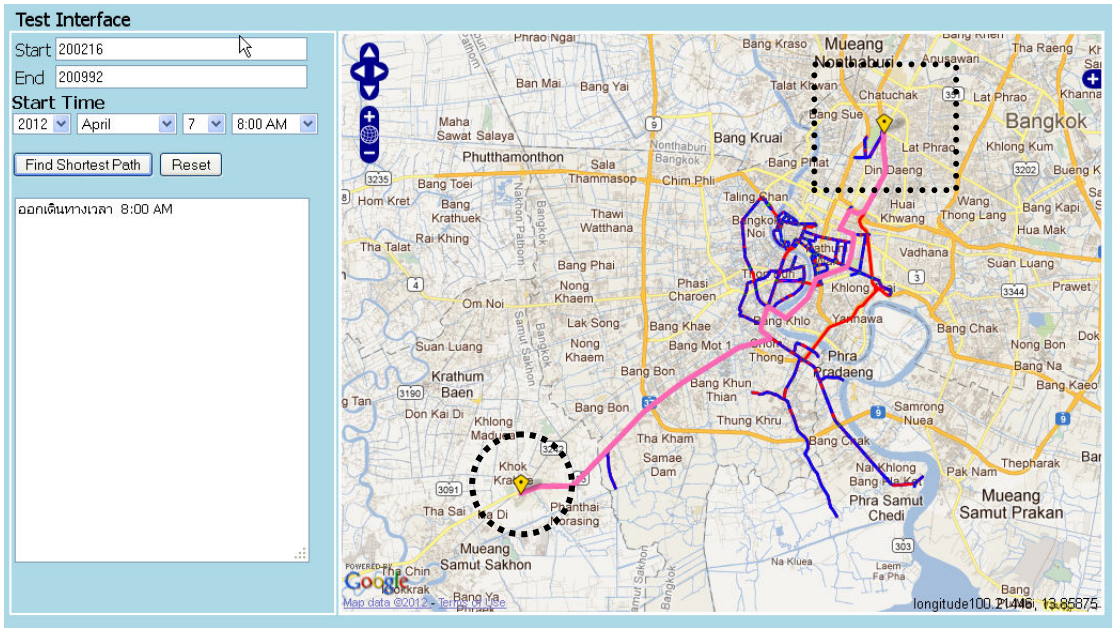
5.3.1.2 กรณีกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทางให้อยู่ไกลกัน

5.3.1.2.1 การวิเคราะห์โครงข่ายโดยใช้ค่า Cost จากตารางหลักของโครงข่าย

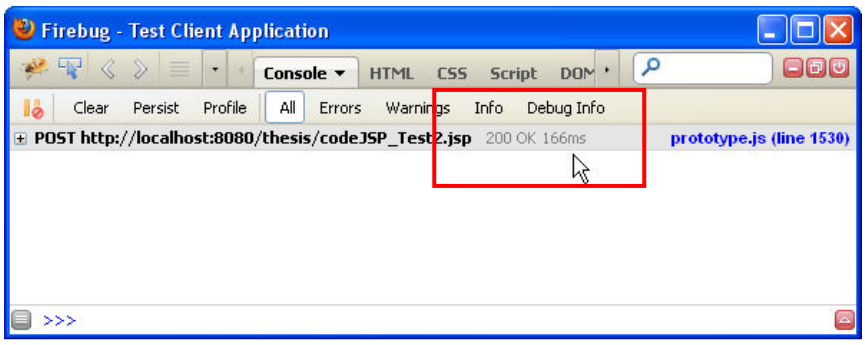
สมมุติให้จุดเริ่มต้นอยู่ที่จุดข้ามสะพานลอยบริเวณถนนพระราม 2 ตัดกับถนนเอกชัย และจุดสิ้นสุดอยู่ที่บริเวณสะพานลอยใกล้กับสถานีรถไฟฟ้าใต้ดินพหลโยธิน ดังแสดงในภาพที่ 5.18 จากนั้นทดลองวิเคราะห์โครงข่ายเพื่อหาเส้นทางการเดินทางที่ใช้ระยะทางน้อยที่สุดโดยการกดที่ปุ่ม Find Shortest Path โปรแกรมจะทำการประมวลผลได้เส้นทางผลลัพธ์ดังแสดงในภาพที่ 5.19 และใช้เวลาในการประมวลผลเท่ากับ 166 มิลลิวินาทีดังแสดงในภาพที่ 5.20



ภาพที่ 5.18 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่อยู่ไกลกัน เพื่อทดสอบการวิเคราะห์โครงข่ายโดยใช้ค่า Cost จากตารางหลัก (ระยะทาง) ของโครงข่าย



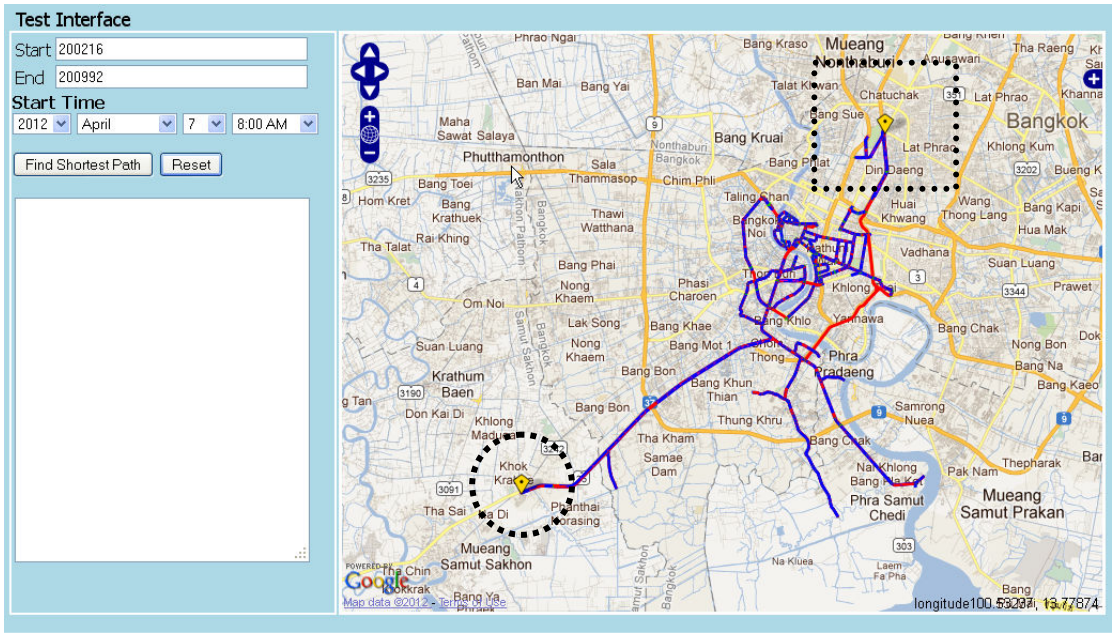
ภาพที่ 5.19 ผลลัพธ์การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้ระยะทางสั้นที่สุดจากจุดที่อยู่ไกลกัน



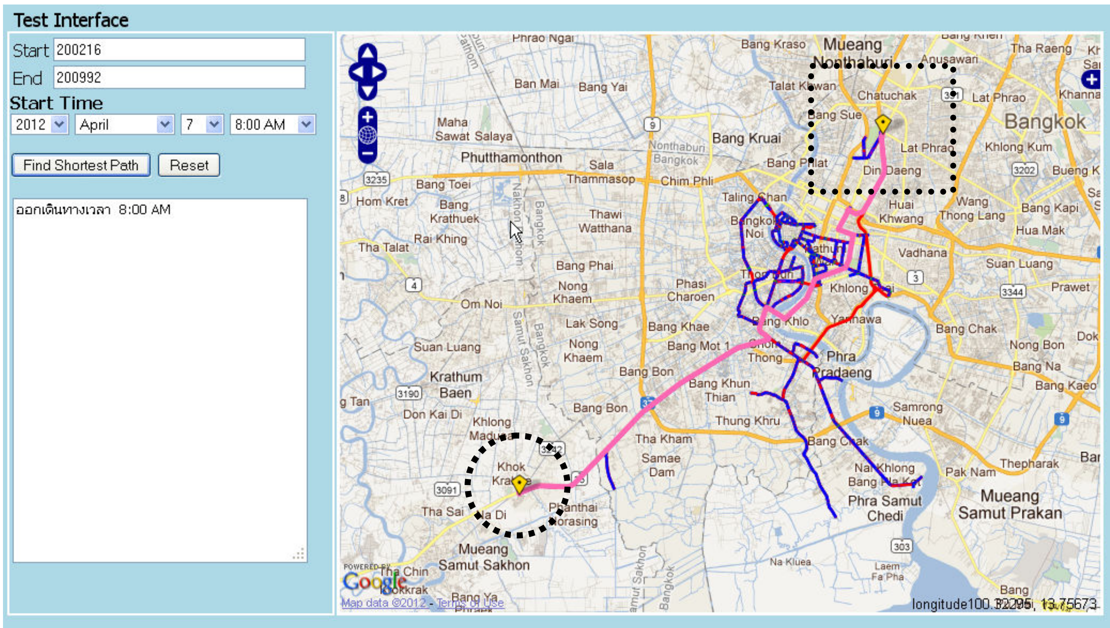
ภาพที่ 5.20 เวลาที่ใช้ในการประมวลผลจากการวิเคราะห์โครงข่ายที่ใช้ค่า Cost จากตารางหลักของโครงข่าย และจุดเริ่มต้น/สิ้นสุดการเดินทางอยู่ไกลกัน

5.3.1.2.2 การวิเคราะห์โครงข่ายโดยใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) ร่วมกับข้อมูลในตารางเสริม (อัตราเร็ว) ของโครงข่าย

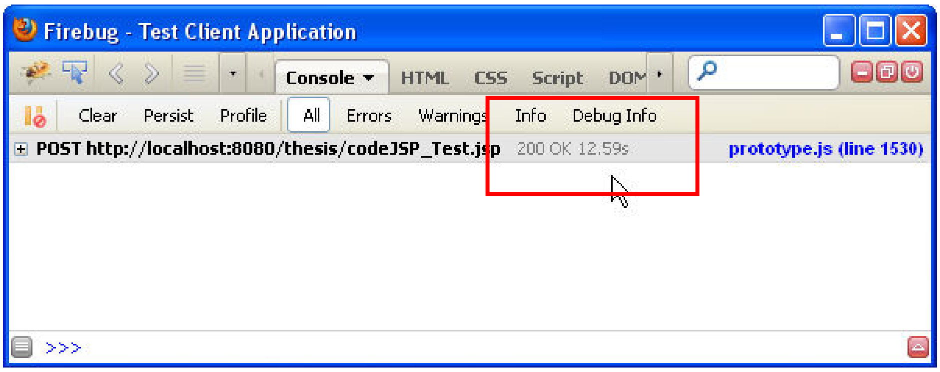
จุดเริ่มต้นและจุดสิ้นสุดการเดินทางกำหนดให้เป็นจุดเดียวกันกับการทดสอบในหัวข้อ 5.3.1.2.1 ดังแสดงในภาพที่ 5.21 จากนั้นทดลองวิเคราะห์โครงข่ายเพื่อหาเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุด โปรแกรมจะทำการประมวลผลได้เส้นทางผลลัพธ์ดังแสดงในภาพที่ 5.22 และใช้เวลาในการประมวลผลเท่ากับ 12.59 วินาทีดังแสดงในภาพที่ 5.23



ภาพที่ 5.21 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่อยู่ไกลกัน เพื่อทดสอบการวิเคราะห์โครงข่ายโดยใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) และตารางเสริม (อัตราเร็ว)



ภาพที่ 5.22 ผลลัพธ์การวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาน้อยที่สุดจากจุดที่อยู่ไกลกัน



ภาพที่ 5.23 เวลาที่ใช้ในการประมวลผลจากการวิเคราะห์โครงข่ายที่ใช้ค่า Cost ที่คำนวณได้จากตารางหลัก (ระยะทาง) และตารางเสริม (อัตราเร็ว) และจุดเริ่มต้น/สิ้นสุดการเดินทางอยู่ไกลกัน

5.3.2 อภิปรายผลการทดสอบประสิทธิภาพของฟังก์ชั่นการวิเคราะห์โครงข่าย

จากการทดสอบส่วนของโปรแกรมที่ใช้ในการวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุด โดยมีการเลือกใช้ค่า Cost ในขั้นตอนวิธีของ Dijkstra จากแหล่งที่แตกต่างกันพบว่า หากค่า Cost ที่นำมาใช้ในการวิเคราะห์เป็นค่า Cost จากตารางหลักของโครงข่ายที่ระบุไว้ใน Network Metadata ระยะเวลาที่ใช้ในการประมวลผลการวิเคราะห์โครงข่ายจะใช้เวลาน้อยมาก แม้ว่าจุดเริ่มต้นและจุดสิ้นสุดการเดินทางจะอยู่ไกลกันมากก็ตาม ดังแสดงเวลาในการประมวลผลในตารางที่ 5.1 แต่หากค่า Cost ที่นำมาใช้ในการวิเคราะห์ต้องมีการคำนวณร่วมกับข้อมูลจากตารางอื่นๆ ที่ไม่ใช่ตารางหลักของโครงข่าย เวลาที่ใช้ในการประมวลผลลัพธ์เส้นทางจะ

ใช้เวลาที่นานกว่าปกติมาก ซึ่งจากการจำลองสถานการณ์ในการทดสอบพบว่า ใช้เวลาในการประมวลผลนานถึง 12.59 วินาที ที่เป็นเช่นนี้เนื่องจากทุกๆ ครั้ง ระหว่างที่ขั้นตอนวิธีของ Dijkstra ทำการคัดเลือก link ที่จะใช้ในการเดินทาง ค่า Cost ที่ใช้ในการพิจารณาจะต้องใช้ฟังก์ชั่นที่พัฒนาเพิ่มขึ้นไปสืบค้นอัตราเร็วของรถโดยสารที่เดินทางผ่าน link ณ ขณะเวลานั้นๆ จากตารางภายนอกตารางหลักของโครงข่ายเสมอ ซึ่งต้องใช้เวลาในการสร้าง Statement เพื่อที่จะติดต่อกับฐานข้อมูลและสืบค้นหาค่าอัตราเร็ว นำมาหารกับระยะทางของ link เพื่อให้ได้เป็นเวลา (Cost) ที่ใช้ในการเดินทางผ่าน link

ตารางที่ 5.1 แสดงเวลาที่ใช้ในการประมวลผลของส่วนวิเคราะห์โครงข่ายจากการเลือกใช้ค่า Cost จากตารางที่แตกต่างกัน

ค่า Cost ที่นำมาใช้ในการวิเคราะห์โครงข่าย	**ระยะเวลาที่ใช้ในการประมวลผล**	
	จุดเริ่มต้น/สิ้นสุดอยู่ใกล้กัน	**จุดเริ่มต้น/สิ้นสุดอยู่ไกลกัน**
Cost จากตารางหลักของโครงข่าย	210 มิลลิวินาที	166 มิลลิวินาที
Cost จากตารางหลัก ที่คำนวณร่วมกับข้อมูลจากตารางเสริมภายนอก	454 มิลลิวินาที	12.59 วินาที

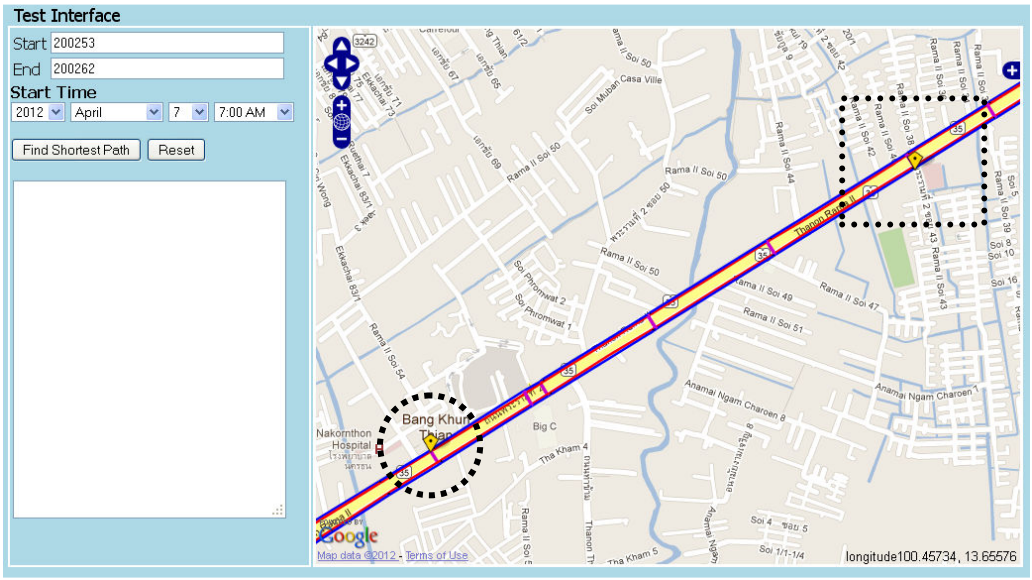
5.3.3 การทดสอบการทำงานของส่วนการวิเคราะห์โครงข่ายรถโดยสารประจำทาง

เป็นการทดสอบฟังก์ชั่นการวิเคราะห์โครงข่ายที่พัฒนาขึ้น โดยจะทดสอบการวิเคราะห์หาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุดโดยคำนึงถึงปัจจัยเวลาในการเดินทางที่แตกต่างกันไปในแต่ละช่วงของวัน และทดสอบการวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุด โดยมีการโดยสารรถประจำทางมากกว่า 1 สาย มีรายละเอียดดังนี้

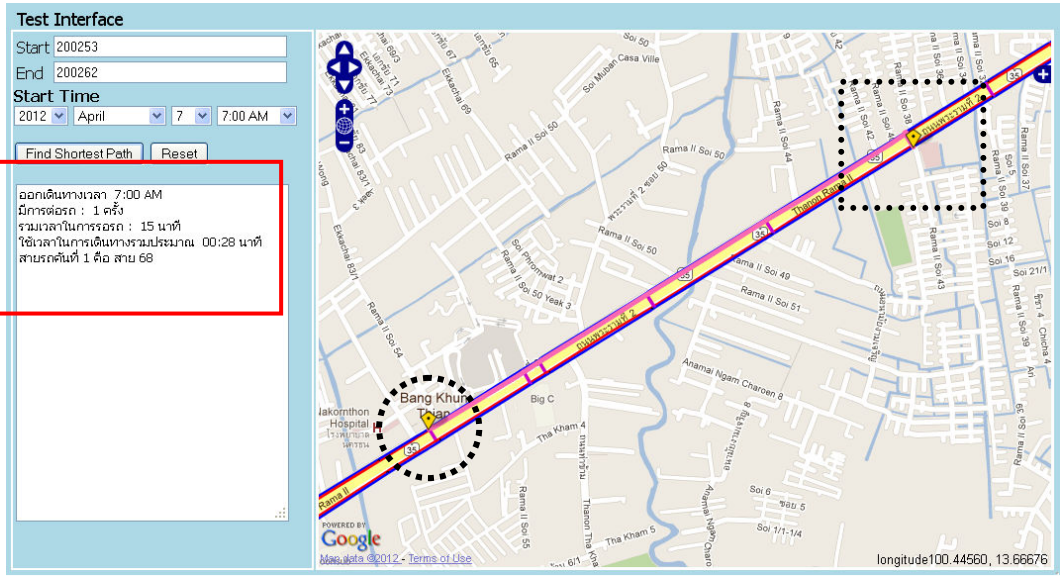
5.3.3.1 การทดสอบฟังก์ชั่นของการวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุดและคำนึงถึงปัจจัยของเวลาในการเดินทางที่แตกต่างกันในแต่ละช่วงวัน

เป็นการทดสอบฟังก์ชั่นที่พัฒนาขึ้นเพื่อใช้ในการปรับแต่งขั้นตอนวิธีของ Dijkstra ที่ใช้ในการวิเคราะห์โครงข่าย เพื่อให้พิจารณาปัจจัยของอัตราเร็วในการเดินทางที่แตกต่างกันไปตามช่วงเวลาของวัน การทดสอบจะกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทางที่เหมือนกัน แต่แตกต่างกันที่เวลาเริ่มต้นออกเดินทาง โดยสมมุติให้จุดเริ่มต้นและจุดสิ้นสุดการเดินทางอยู่บนถนนพระราม 2 ดังแสดงในภาพที่ 5.24 และกำหนดเวลาที่เริ่มต้นเดินทางเท่ากับ 7.00 น. (7.00 AM) จากนั้นทำการ

วิเคราะห์โครงข่ายจะได้ผลลัพธ์ดังแสดงในภาพที่ 5.25 ซึ่งคำตอบของการวิเคราะห์โครงข่ายได้ผลลัพธ์เป็นแผนที่เส้นทาง และเวลาที่ใช้ในการเดินทางคือ มีการโดยสารรถประจำทาง 1 สาย ใช้เวลาในการรอคอยรถโดยสาร 15 นาที เวลาที่ใช้ในการเดินทางเท่ากับ 13 นาที เมื่อรวมกับเวลาในการรอคอยรถโดยสารได้เวลาที่ต้องใช้ในการเดินทางทั้งหมดประมาณ 28 นาที

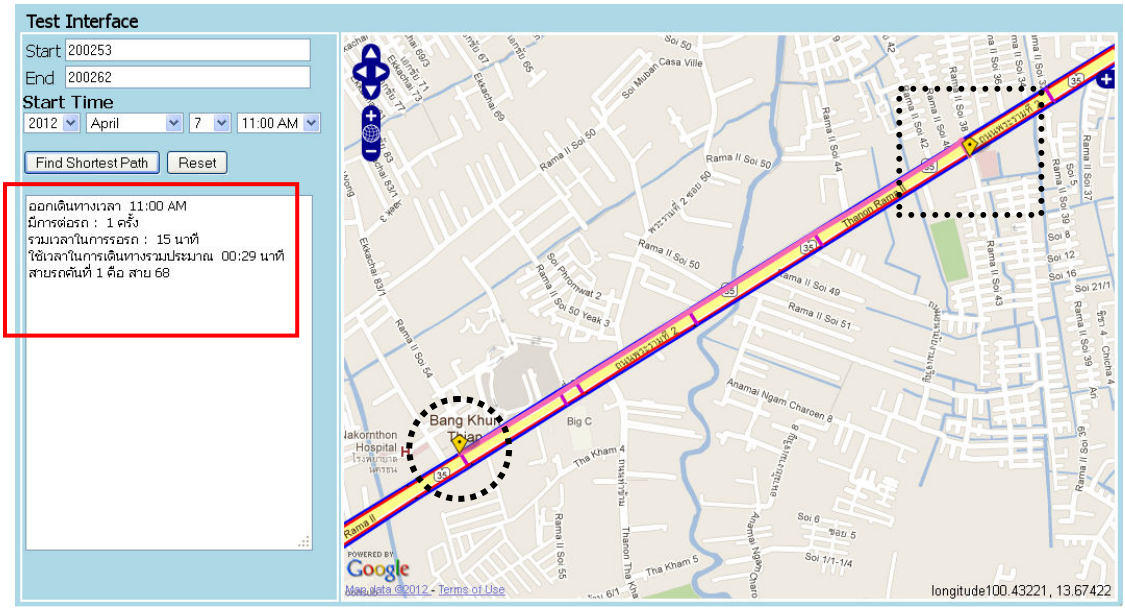


ภาพที่ 5.24 แสดงจุดเริ่มต้นและจุดสิ้นสุดการเดินทางในการทดสอบฟังก์ชั่นการวิเคราะห์โครงข่ายที่พัฒนาเพิ่มขึ้น



ภาพที่ 5.25 ผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อทดสอบฟังก์ชั่นที่พัฒนาขึ้น โดยเวลาเริ่มต้นเดินทางคือ 7.00 น.

ทดลองวิเคราะห์หาเส้นทางในการเดินทางใหม่อีกครั้งโดยกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทางณ ตำแหน่งเดิม แต่เปลี่ยนเวลาในการเริ่มต้นเดินทางเป็น 11.00 น. (11.00 AM) แล้วทำการวิเคราะห์หาเส้นทางการเดินทางได้ผลลัพธ์ดังแสดงในภาพที่ 5.26 ซึ่งจะได้ผลลัพธ์ของเวลาที่ใช้ในการเดินทางรวมประมาณ 29 นาที

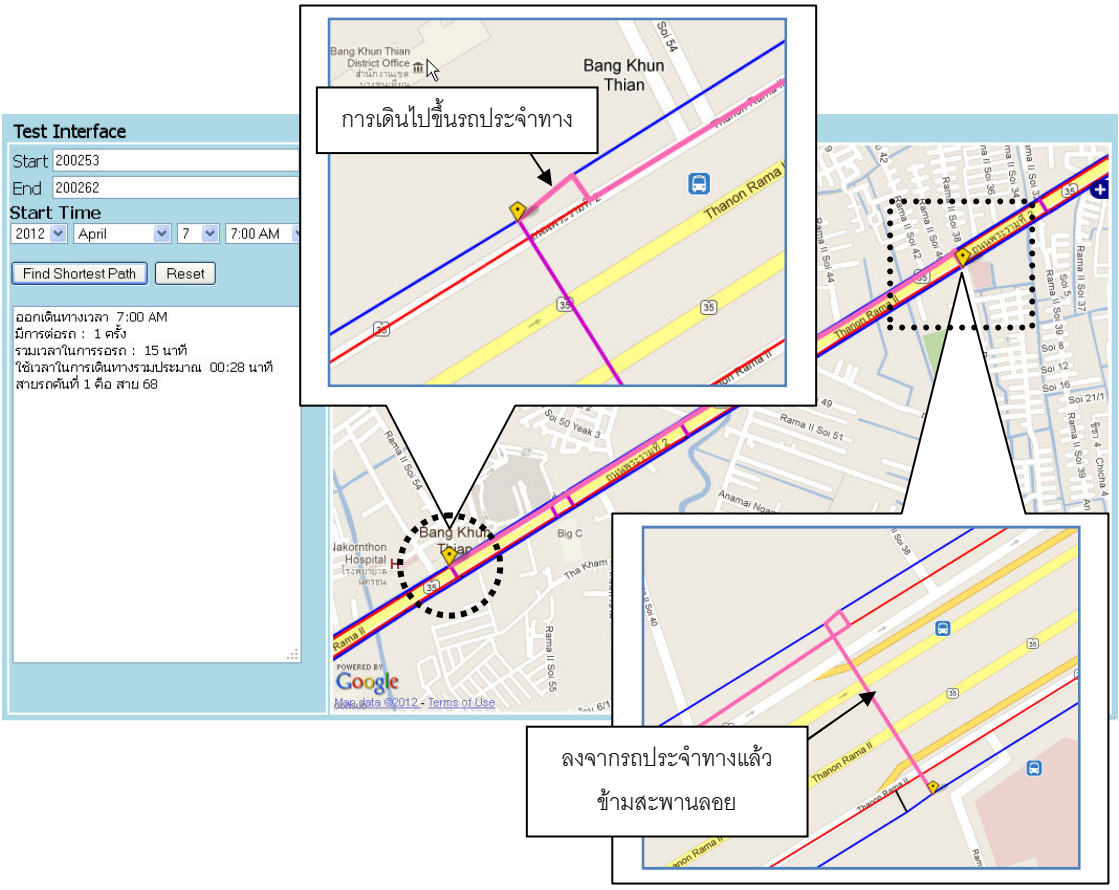


ภาพที่ 5.26 ผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อทดสอบฟังก์ชั่นที่พัฒนาขึ้น โดยเวลาเริ่มต้นเดินทางคือ 11.00 น.

5.3.3.2 อภิปรายผลการทดสอบฟังก์ชั่นของการวิเคราะห์โครงข่ายที่พัฒนาขึ้นเพิ่มเติม

การทดสอบฟังก์ชั่นการทำงานในหัวข้อ 5.3.3.1 จะเห็นว่า เมื่อมีการกำหนดเวลาในการเริ่มต้นการเดินทางที่แตกต่างกัน แม้ว่าจุดเริ่มต้นและจุดสิ้นสุดการเดินทางจะเหมือนกัน แต่ผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุด จะให้ผลลัพธ์ที่ไม่เหมือนกัน อาจแตกต่างกันทั้งในเรื่องของเส้นทางการเดินทางและเวลาที่ใช้ในการเดินทาง โดยจากสถานการณ์การเดินทางที่จำลองขึ้นพบว่า เมื่อเริ่มต้นออกเดินทางเมื่อเวลา 7.00 น. ผลลัพธ์ของเวลาในการเดินทางจะใช้เวลาประมาณ 28 นาที แต่เมื่อเปลี่ยนเวลาเริ่มต้นเดินทางเป็น 11.00 น. พบว่าผลลัพธ์ของเวลาที่ใช้ในการเดินทางจะใช้เวลาประมาณ 29 นาที

การตรวจสอบความถูกต้องของเวลาที่ใช้ในการเดินทางจากการประมวลผล ได้ทดลองตรวจสอบความถูกต้องของผลลัพธ์ที่ได้จากการวิเคราะห์โครงข่ายจากหัวข้อ 5.3.3.1 ที่เริ่มต้นออกเดินทางเมื่อเวลา 7.00 น. โดยได้ผลลัพธ์ของเส้นทางที่เริ่มต้นเดินทางจากจุดที่เป็นตำแหน่งข้ามสะพานลอย จากนั้นเป็นการเดินไปตามทางเดินเท้าและขึ้นรถโดยสารประจำทางที่ป้ายจอดรถโดยสาร จากนั้นเป็นการลงจากรถโดยสารเพื่อเดินขึ้นสะพานลอยข้ามไปยังจุดสิ้นสุดการเดินทางดังแสดงในภาพที่ 5.27 การตรวจสอบความถูกต้องของเวลาที่ใช้ในการเดินทาง จะใช้วิธีการอ่านค่าอัตราเร็วของรถโดยสาร ณ ขณะเวลานั้นๆ ในการเดินทางผ่าน link ที่ประกอบกันเป็นเส้นทางผลลัพธ์เพื่อนำมาคำนวณเป็นเวลาที่ใช้ในการเดินทางดังแสดงในตารางที่ 5.2 ได้เวลาในการเดินทางทั้งหมดเท่ากับ 13 นาที เมื่อรวมกับระยะเวลาในการรอคอยรถโดยสาร 1 ครั้ง อีก 15 นาที ดังนั้นเวลาในการเดินทางรวมทั้งหมดจะเท่ากับ 28 นาที ซึ่งตรงกับผลลัพธ์ของการคำนวณจากโปรแกรมประยุกต์



ภาพที่ 5.27 แสดงภาพขยายของผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อทดสอบฟังก์ชั่นที่พัฒนาขึ้น โดยเวลาเริ่มต้นเดินทางคือ 7.00 น.

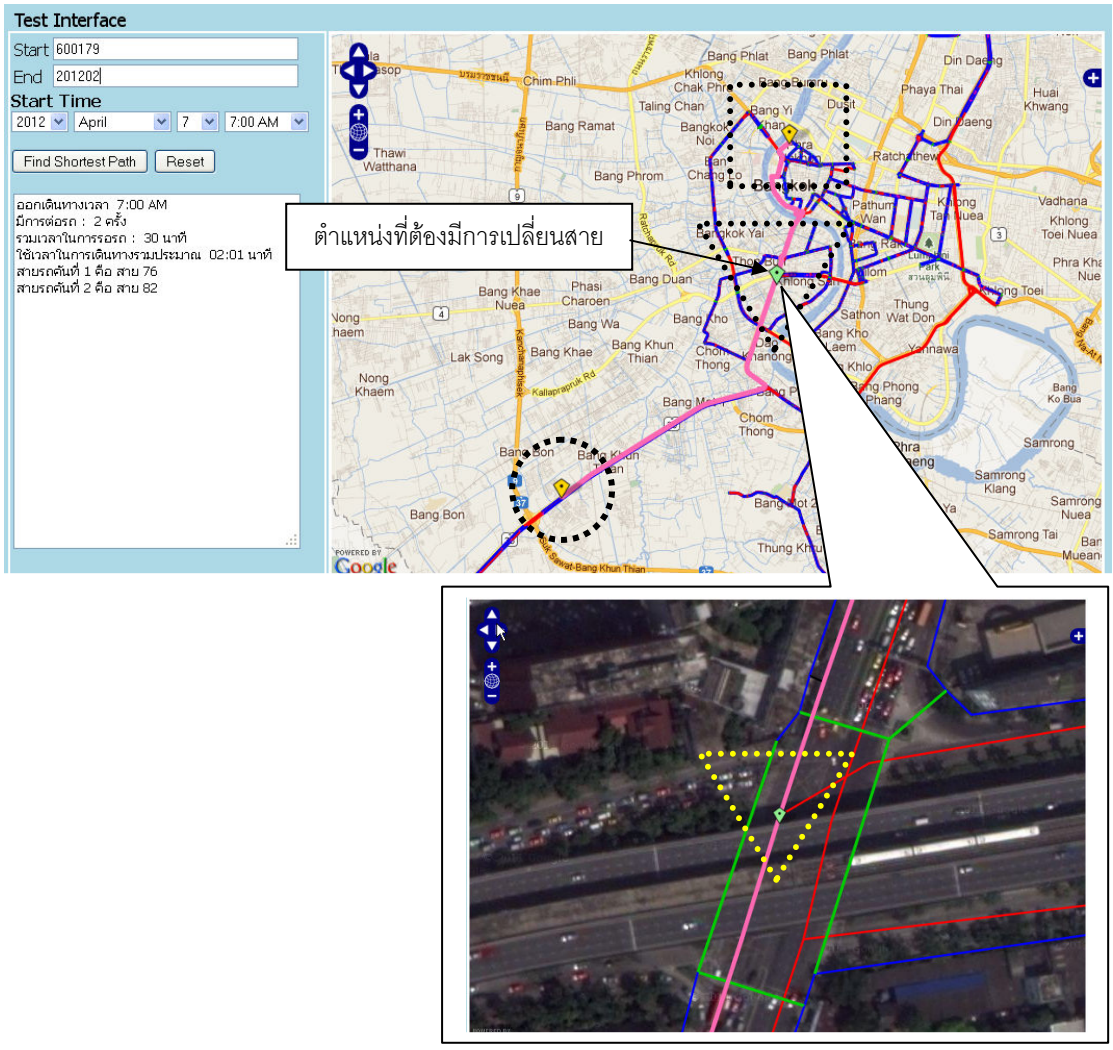
ตารางที่ 5.2 แสดงเวลาที่ใช้ในการเดินทางผ่าน link ที่ประกอบกันเป็นเส้นทางผลลัพธ์

LINK_ID	Index	**อัตราเร็ว กม./ชม.**	**ระยะทางของ link (เมตร)**	**เวลา (Cost) ที่ใช้ในการเดินทางผ่าน link (วินาที)**
700351	29	5	18.99	13.67
900151	29	5	8.17	5.88
400217	29	16	601.71	135.38
400215	29	18	588.19	117.63
400213	29	11	664.41	217.44
400210	29	11	757.61	247.94
900159	29	5	6.73	4.84
700334	29	5	7.37	5.3
300033	29	5	76.93	55.38
รวมเวลาเท่ากับ				**13 นาที**

* ค่า index = 29 หมายถึง อัตราเร็วของรถโดยสารในการเดินทางผ่าน link ในช่วงเวลา 7.15 – 7.29 น.

5.3.3.3 การทดสอบการทำงานของการวิเคราะห์โครงข่ายกรณีต้องมีการโดยสารรถประจำทางมากกว่า 1 สาย

ทดสอบการทำงานของฟังก์ชั่นการวิเคราะห์โครงข่ายที่พัฒนาเพิ่มเติมขึ้น ในกรณีที่เส้นทางผลลัพธ์ต้องมีการโดยสารรถประจำทางมากกว่า 1 สาย โดยสมมุติเริ่มเดินทางจากป้ายรถประจำทางทางบริเวณถนนพระราม 2 ซอย 60 ไปยังป้ายรถประจำทางบริเวณถนนพระอาทิตย์ และออกเดินทางเมื่อเวลา 7.00 น. เมื่อทำการวิเคราะห์โครงข่ายได้ผลลัพธ์ของแผนที่เส้นทาง และระยะเวลาที่ใช้ในการเดินทาง โดยมีการโดยสารรถประจำทาง 2 สายคือ สาย 76 และสาย 82 เวลาในการรอรถโดยสารเท่ากับ 30 นาที เมื่อรวมกับเวลาที่ใช้ในการเดินทางได้เป็นเวลาในการเดินทางรวมทั้งหมดประมาณ 2 ชั่วโมง 1 นาที ดังแสดงในภาพที่ 5.28 พบว่า โปรแกรมสามารถที่จะแสดงตำแหน่งของจุดที่ต้องมีการเปลี่ยนสายรถได้โดยแสดงเป็นเครื่องหมาย (marker สีเขียว)



ภาพที่ 5.28 แสดงผลลัพธ์ของการวิเคราะห์โครงข่ายกรณีที่ต้องมีการโดยสารรถประจำทางมากกว่า 1 สาย

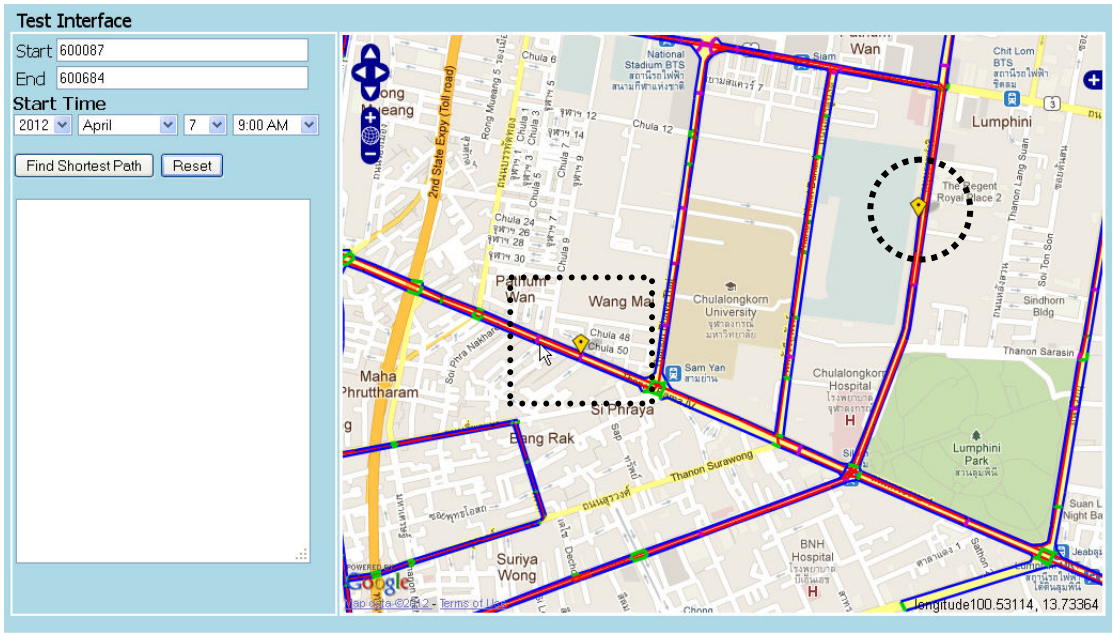
5.3.3.4 อภิปรายผลการทดสอบการทำงานของการวิเคราะห์โครงข่ายกรณีต้องมีการโดยสารรถประจำทางมากกว่า 1 สาย

จากการทดสอบการทำงานของโปรแกรมในการวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุดพบว่า โปรแกรมประยุกต์ที่พัฒนาขึ้นสามารถวิเคราะห์หาเส้นทางผลลัพธ์และแสดงผลแผนที่ได้ตรงตามเงื่อนไข และหากต้องมีการโดยสารรถประจำทางมากกว่า 1 สาย โปรแกรมสามารถแสดงตำแหน่งของจุดที่ต้องมีการเปลี่ยนสายรถโดยสารประจำทางได้ แต่ตำแหน่งที่ต้องมีการเปลี่ยนสายรถโดยสารที่แสดงบนแผนที่จะเป็นตำแหน่งของจุดตัดถนน ไม่ใช่ตำแหน่งของป้ายรถโดยสารประจำทางที่ต้องรอรถโดยสารสายต่อไป โปรแกรมเพียงแต่

แนะนำอย่างคร่าวๆ ว่าจุดตัดของถนนดังกล่าว เป็นตำแหน่งที่ต้องมีการเปลี่ยนรถโดยสารในการเดินทาง

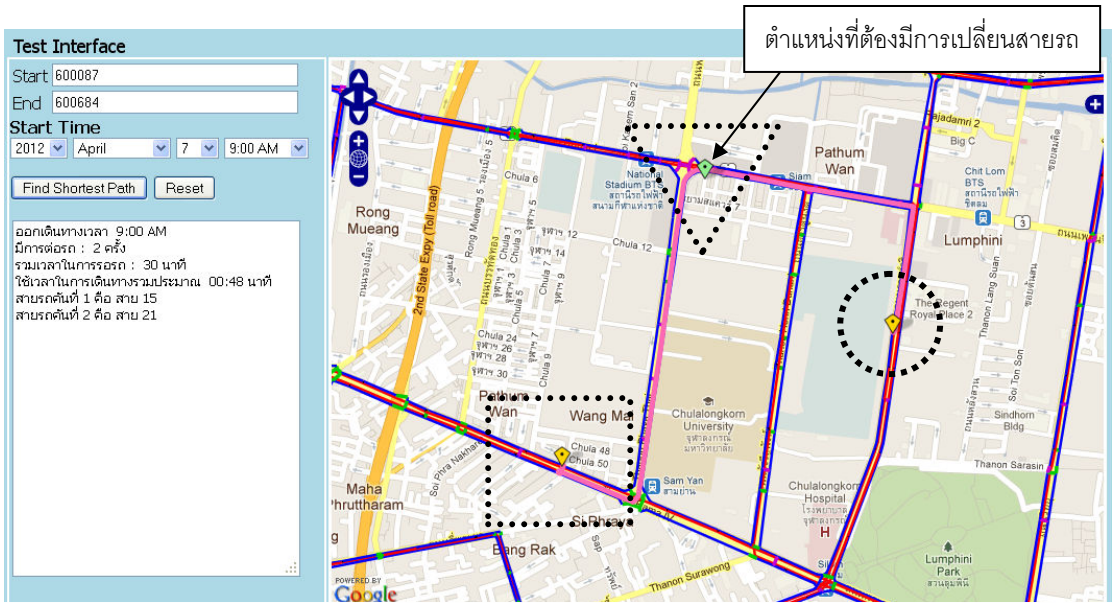
5.4 การทดสอบการทำงานในส่วนของการวิเคราะห์หาหมายเลขสายรถประจำทาง

ทดสอบการทำงานของฟังก์ชั่นที่ใช้ในการหาหมายเลขสายรถประจำทางที่ต้องโดยสาร โดยสมมุติให้เริ่มต้นการเดินทางจากป้ายรถโดยสารประจำทางบริเวณสถานีรถไฟฟ้าบีทีเอสราชดำริ และจุดสิ้นสุดการเดินทางอยู่ที่ป้ายรถโดยสารประจำทางหน้าปากซอยจุฬา 11 บนถนนพระราม 4 กำหนดเวลาในการเริ่มต้นออกเดินทางคือเวลา 9.00 น. (9.00 AM) ดังแสดงในภาพที่ 5.29

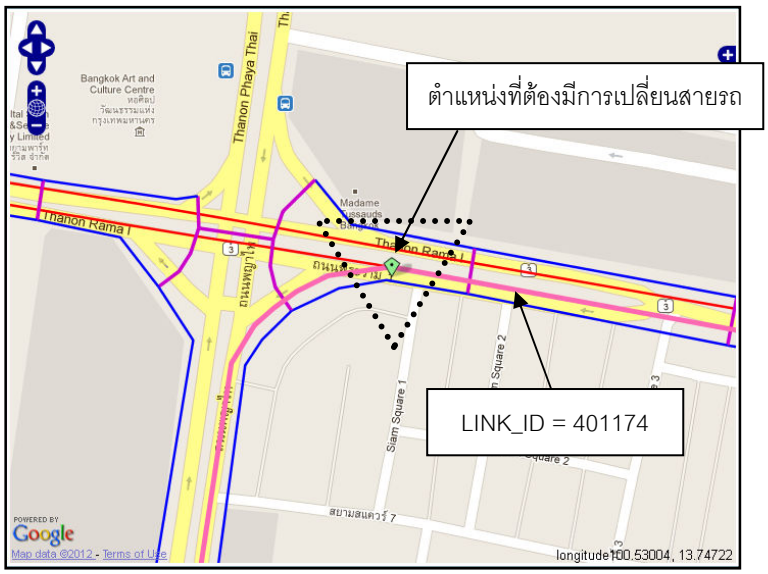


ภาพที่ 5.29 แสดงตำแหน่งของจุดเริ่มต้นและจุดสิ้นสุดการเดินทางในการทดสอบการวิเคราะห์หาหมายเลขสายรถประจำทางที่ต้องโดยสาร

จากนั้นทำการวิเคราะห์โครงข่ายเพื่อหาเส้นทางการเดินทางที่ใช้เวลาในการเดินทางน้อยที่สุดพบว่า เส้นทางผลลัพธ์ใช้เวลาในการเดินทางประมาณ 18 นาที และต้องมีการโดยสารรถประจำทาง 2 ครั้ง เท่ากับระยะเวลาในการรอคอยรถโดยสาร 30 นาที ดังนั้นใช้เวลาในการเดินทางรวมทั้งหมดประมาณ 48 นาที โดยสายรถที่แนะนำในการเดินทางสายแรกคือสาย 15 และต่อด้วยสายรถหมายเลข 21 ดังแสดงในภาพที่ 5.30 ตำแหน่งที่ต้องมีการเปลี่ยนสายรถ โปรแกรมจะทำการวาดเครื่องหมาย (marker สีเขียว) โดยเป็นตำแหน่งเดียวกันกับจุดตัดของแยกถนนดังแสดงในภาพที่ 5.31



ภาพที่ 5.30 ผลลัพธ์จากการวิเคราะห์โครงข่ายเพื่อทดสอบฟังก์ชั่นการหาหมายเลขสายรถที่ต้องโดยสาร



ภาพที่ 5.31 ภาพขยายของตำแหน่งที่ต้องมีการเปลี่ยนสายรถในการเดินทาง

การตรวจสอบความถูกต้องของสายรถประจำทางที่แนะนำในการเดินทางจากโปรแกรมประยุกต์ ใช้วิธีตรวจสอบจากข้อมูลในตารางที่ใช้จัดเก็บหมายเลขสายรถในระบบฐานข้อมูล (ตาราง BUS) โดยจากตารางที่ 5.3 จะแสดงหมายเลขรหัสประจำตัวของ link ที่เป็นเส้นทางผลลัพธ์จากการวิเคราะห์โครงข่าย และแสดงหมายเลขสายรถที่เดินทางผ่านแต่ละ link ซึ่งพบว่าสายรถหมายเลข 15 จะเป็นสายรถประจำทางที่แนะนำในการเดินทางช่วงแรก เนื่องจากเป็นสาย

รถประจำทางเดียวที่เดินทางผ่านจุดเริ่มต้น และสามารถโดยสารรถหมายเลข 15 มาได้ไกลจนถึง link ที่มีเลขรหัสประจำตัวเท่ากับ 401174 ก็จะต้องมีการเปลี่ยนสายรถในการเดินทาง โดยจะเป็นการเดินทางต่อด้วยสายรถหมายเลข 21 เพราะจะสามารถเดินทางไปได้ไกลจนถึงจุดสิ้นสุดการเดินทาง ในขณะที่รถโดยสารหมายเลข 141 จะเดินทางไปได้ไกลถึงแค่ link ที่มีเลขรหัสประจำตัวเท่ากับ 401183 เท่านั้น

ตารางที่ 5.3 แสดงรหัสประจำตัวของ link ที่ประกอบกันเป็นเส้นทางผลลัพธ์ และหมายเลขสายรถที่ผ่าน

LINK_ID	**หมายเลขสายรถที่ผ่าน** link		
	15	21	141
400048	/		
400047	/		
400046	/		
401224	/		
401225	/		
401173	/	/	/
401174	/	/	/
401175		/	/
401177		/	/
401179		/	/
401181		/	/
401182		/	/
401183		/	/
401185		/	
401186		/	

/ หมายถึง หมายเลขสายรถประจำทางนั้นเดินทางผ่าน

บทที่ 6

สรุปผลการศึกษา

6.1 สรุปผลการวิจัย

จากการดำเนินงานในการพัฒนาโปรแกรมประยุกต์บนเว็บเพื่อแนะนำเส้นทางการเดินทางที่เหมาะสมจากการเดินทางด้วยรถโดยสารประจำทาง ผลการทดสอบโปรแกรมได้ให้ผลลัพธ์ของเส้นทางการเดินทางที่แนะนำเป็นไปตามที่คาดหมาย ภายใต้เงื่อนไขที่กำหนดไว้ในงานวิจัย กล่าวคือ

1. เงื่อนไขทางด้านเวลาที่ใช้ในการโดยสารรถประจำทาง จะคำนวณจากข้อมูลอัตราเร็วที่ผู้วิจัยได้ออกแบบไว้ให้มีข้อมูลอัตราเร็วที่แตกต่างกัน 96 ค่า เพื่อเลือกใช้ในการวิเคราะห์โครงข่ายด้วยขั้นตอนวิธีของ Dijkstra

2. ระยะเวลาในการรอคอยรถประจำทางกำหนดให้เป็นค่าคงที่ คือ เท่ากับ 15 นาทีต่อการโดยสารรถประจำทาง 1 ครั้ง และสมมุติให้รถประจำทางทุกสายให้บริการตลอด 24 ชั่วโมง

ผลการทดสอบการใช้งานของโปรแกรมประยุกต์สามารถสรุปได้ดังนี้

1. รองรับการกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทางจากการกำหนดตำแหน่งลงบนแผนที่ โดยโปรแกรมจะทำการค้นหาจุดเชื่อมต่อการเดินทางในโครงข่ายที่ใกล้ที่สุดให้อัตโนมัติ

2. สามารถแนะนำเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุด โดยคำนึงถึงปัจจัยของเวลาในการเดินทางที่แตกต่างกันในแต่ละช่วงของวัน ขึ้นอยู่กับเวลาที่ผู้ใช้เริ่มต้นออกเดินทางว่าเป็นเวลาเท่าใด ซึ่งจะมีผลต่อการวิเคราะห์โครงข่าย และให้ผลลัพธ์ของเส้นทางที่แนะนำแตกต่างกัน

3. สามารถแสดงผลลัพธ์ของการวิเคราะห์โครงข่ายทั้งในรูปแบบของแผนที่เส้นทางการเดินทางและข้อมูลประกอบการเดินทางคือ สายรถประจำทางที่ต้องโดยสาร รวมถึงเวลาที่ต้องใช้ในการเดินทางโดยประมาณได้

โดยสรุปแล้วโปรแกรมประยุกต์ที่พัฒนาขึ้นสามารถนำมาช่วยในการตัดสินใจเลือกเส้นทางการเดินทางด้วยรถโดยสารประจำทาง โดยผลลัพธ์ของการแนะนำเส้นทางจะแตกต่างกันไปขึ้นอยู่กับเวลาที่เริ่มต้นออกเดินทาง เพราะจะเป็นปัจจัยที่นำมาใช้ในการเลือกอัตราเร็วของรถโดยสารประจำทางที่มีการเปลี่ยนแปลงตลอดวัน เพื่อใช้ในขั้นตอนของการวิเคราะห์โครงข่ายเพื่อหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุด

6.2 อภิปรายผลการวิจัย

ผลการศึกษาพบว่า โปรแกรมประยุกต์ที่พัฒนาขึ้น สามารถนำมาใช้เป็นแนวทางในการวางแผนการเดินทางด้วยรถโดยสารประจำทางได้ โดยให้ผลลัพธ์ตรงตามเงื่อนไขของปัจจัยเรื่องอัตราเร็วของรถโดยสารที่แตกต่างกันในแต่ละช่วงของวัน และระบบฐานข้อมูล Oracle Spatial มีความสามารถที่จะใช้ในการจัดเก็บ จัดการ และวิเคราะห์ข้อมูลโครงข่ายการเดินทางด้วยรถโดยสารประจำทางได้อย่างเหมาะสม เนื่องจากมีโครงสร้างข้อมูลที่ใช้ในการจัดเก็บข้อมูลปริภูมิ (SDO_GEOMETRY) และสนับสนุนแบบจำลองข้อมูลในลักษณะโครงข่าย (Oracle Network Data Model) อีกทั้งยังเอื้ออำนวยต่อการปรับแต่งเพิ่มปัจจัยต่างๆ ในการวิเคราะห์เชิงโครงข่ายได้

การพัฒนาโปรแกรมประยุกต์ในงานวิจัยนี้ มีประเด็นที่ควรคำนึงถึงในการพัฒนาต่อยอดเพื่อให้โปรแกรมมีความสามารถและมีประสิทธิภาพการทำงานที่ดีขึ้น ดังนี้

1. ในงานวิจัย ปัจจัยที่ใช้ในการพิจารณาเลือกเส้นทางการเดินทางจะใช้ปัจจัยเรื่องของเวลาในการเดินทาง (Time Cost) โดยอ้างอิงจากระยะทาง (Distance Cost) ผ่านอัตราเร็วที่สมมุติขึ้น ซึ่งในการเดินทางจริงยังมีปัจจัยอื่นๆ ที่มีผลต่อการตัดสินใจในการเลือกเส้นทางเดินทาง เช่น ค่าใช้จ่ายในการเดินทางที่เป็นตัวเงิน (Toll Cost) จึงควรมีการพัฒนาต่อยอดให้สามารถนำปัจจัยต่างๆ เหล่านี้มาพิจารณาเพิ่มเติมในการวิเคราะห์โครงข่ายให้ครบทุกปัจจัย จะช่วยให้โปรแกรมมีความยืดหยุ่นและตอบสนองความต้องการของผู้ใช้งานมากยิ่งขึ้น

2. การปรับแต่งขั้นตอนวิธีของ Dijkstra ที่ใช้ในการวิเคราะห์โครงข่ายการเดินทางด้วยรถโดยสารประจำทางเพื่อหาเส้นทางที่ใช้เวลาในการเดินทางน้อยที่สุด โดยให้พิจารณาปัจจัยเพิ่มเติม คือ อัตราเร็วของรถโดยสารประจำทางที่แตกต่างกันไปในแต่ละเวลาของวัน ซึ่งผู้วิจัยได้ออกแบบและจัดเก็บข้อมูลอัตราเร็วของรถโดยสารประจำทาง แยกเป็นตารางออกมาจากตารางหลักของโครงข่าย ทำให้ในขั้นตอนวิธีของ Dijkstra โปรแกรมจะต้องทำการเชื่อมต่อกับฐานข้อมูลและอ่านค่าอัตราเร็วจากตารางที่เก็บข้อมูลอัตราเร็วของรถประจำทางที่ออกแบบไว้ทุกๆ ครั้ง ส่งผลให้ใช้เวลาในการประมวลผลค่อนข้างนาน หากต้องการเพิ่มประสิทธิภาพให้ประมวลผลได้เร็วขึ้น ข้อมูลอัตราเร็วของรถโดยสารประจำทางนี้ ควรสร้างเป็น User data ด้วยรูปแบบข้อมูลแบบ BLOB (Binary Large Object) และจัดทำตาราง partition BLOB เพื่อนำเข้าข้อมูล โดยข้อมูล User data นี้จะถูกโหลดเก็บไว้ในหน่วยความจำก่อนการวิเคราะห์โครงข่าย ทำให้ไม่ต้องมีการเชื่อมต่อฐานข้อมูลทุกครั้ง และการประมวลผลลัพธ์จากการวิเคราะห์โครงข่ายจะมีประสิทธิภาพมากขึ้น

3. การกำหนดตำแหน่งจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง ผู้ใช้ต้องทำการระบุตำแหน่งจากแผนที่เท่านั้น ซึ่งต้องมีความคุ้นเคยกับพื้นที่ในระดับหนึ่ง การมีตัวเลือกให้กับผู้ใช้ในการระบุจุดเริ่มต้นหรือจุดสิ้นสุดการเดินทาง เช่น มีตัวเลือกหรือเปิดโอกาสให้ผู้ใช้สามารถระบุชื่อสถานที่สำคัญๆ หรือชื่อแยกของถนนสายหลัก เพื่อใช้เป็นจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง จะช่วยอำนวยความสะดวกมากขึ้นกรณีที่ผู้ใช้ไม่มีความคุ้นเคยกับพื้นที่ที่ต้องการเดินทางไป

4. ผลลัพธ์ของตำแหน่งที่ต้องมีการเปลี่ยนสายรถโดยสารระหว่างการเดินทางที่ได้จากโปรแกรมประยุกต์นี้ จะเป็นตำแหน่งของจุดตัดของถนน ไม่ได้เป็นตำแหน่งของป้ายรถโดยสารประจำทางที่ต้องใช้ในการรอรถโดยสารจริงๆ การพัฒนาโปรแกรมเพิ่มเติมให้สามารถระบุตำแหน่งป้ายจอดรถประจำทางที่ต้องมีการรอรถโดยสาร จะสามารถช่วยทำให้โปรแกรมประยุกต์ให้ผลลัพธ์การเดินทางที่ถูกต้องมากยิ่งขึ้น

5. ระยะเวลารอคอยรถโดยสารประจำทางในงานวิจัยนี้ ผู้วิจัยได้กำหนดขึ้นเองโดยประมาณให้การรอคอยรถโดยสารประจำทาง 1 สาย ต้องใช้เวลาเท่ากับ 15 นาที ซึ่งในความเป็นจริงระยะเวลาในการรอคอยรถโดยสารประจำทางจะมีความแตกต่างกันตามความถี่ของการปล่อยรถโดยสารและสภาพการจราจร ณ ขณะเวลานั้น การพัฒนาฟังก์ชั่นเพิ่มเติมสำหรับคำนวณระยะเวลารอคอยรถโดยสารประจำทางเพื่อให้มีค่าใกล้เคียงกับสภาพความเป็นจริง โดยอาจจัดเก็บเป็นตารางสถิติของระยะเวลารอคอยรถโดยสาร และให้โปรแกรมสืบค้นข้อมูลระยะเวลารอคอยรถโดยสารในช่วงเวลานั้นๆ จะช่วยให้โปรแกรมประยุกต์สามารถให้ผลลัพธ์ของการวิเคราะห์หาเส้นทางได้ใกล้เคียงกับสภาพความเป็นจริงมากยิ่งขึ้น

6. ตารางที่ใช้จัดเก็บข้อมูลอัตราเร็วของรถโดยสารประจำทางที่ผู้วิจัยได้ออกแบบเพิ่มเติมขึ้นเพื่อใช้ในขั้นตอนของการวิเคราะห์โครงข่าย เป็นเหมือนกับการจัดเก็บข้อมูลสถิติของอัตราเร็ว ณ ช่วงเวลาต่างๆ ซึ่งหากมีข้อมูลจากระบบตรวจจับสภาพการจราจรตามเวลาจริง หลักการทำงานและโครงสร้างการจัดเก็บข้อมูลในงานวิจัยนี้ สามารถนำมาดัดแปลงให้สามารถใช้ประโยชน์จากข้อมูลสภาพการจราจรตามเวลาจริงได้

7. ควรมีการพัฒนาโปรแกรมให้สามารถแก้ไขข้อมูลของโครงข่ายรถโดยสารประจำทาง รวมถึงข้อมูลที่มีผลต่อการคำนวณเส้นทาง เช่น อัตราเร็วของรถโดยสารประจำทางและอัตราเร็วในการเดินเท้า ให้สามารถกระทำผ่านหน้าจอแผนที่ของผู้ใช้ เพื่ออำนวยความสะดวกให้กับผู้ดูแลระบบ

8. ฟังก์ชั่นการวิเคราะห์โครงข่ายที่ได้มีการปรับแต่งเพิ่มเติม จะคำนึงถึงเวลาที่ผู้ใช้เริ่มต้นเดินทางเท่านั้น ควรมีการพัฒนาโปรแกรมให้สามารถพิจารณาถึงวันที่ผู้ใช้เดินทางได้ด้วย เช่น วัน

ทำงานจันทร์ถึงพฤหัสบดี, วันศุกร์ หรือวันหยุดเสาร์-อาทิตย์ เป็นต้น เนื่องจากวันในการเดินทางมีผลต่ออัตราเร็วของรถโดยสารประจำทางที่แตกต่างกันออกไป

9. ในขั้นตอนของการวิเคราะห์โครงข่ายจะวิเคราะห์หาเส้นทางการเดินทางที่ใช้เวลาน้อยที่สุดก่อน จากนั้นจึงนำผลลัพธ์เส้นทางมาค้นหาหมายเลขสายรถที่ต้องโดยสาร ซึ่งเป็นการทำงานแยกส่วนกัน ผลลัพธ์อาจมีจำนวนการต่อรถมากครั้ง ซึ่งมีผลต่อเวลาในการรอรถโดยสารรวมถึงค่าใช้จ่ายที่เพิ่มมากขึ้นด้วย เช่น โปรแกรมวิเคราะห์หาเส้นทางการเดินทางที่ใช้เวลาน้อยที่สุดได้เท่ากับ 1 ชั่วโมง แต่ส่วนค้นหาหมายเลขสายรถที่ต้องโดยสารให้คำตอบว่าต้องมีการต่อรถโดยสาร 3 ครั้ง จะทำให้ต้องใช้เวลารอรถโดยสารทั้ง 3 ครั้งเท่ากับ 45 นาที และทำให้เวลารวมที่ใช้ในการเดินทางเท่ากับ 1 ชั่วโมง 45 นาที ซึ่งอาจจะใช้เวลามากกว่าผลลัพธ์ในเส้นทางอื่นๆ จึงควรมีการพัฒนาโปรแกรมให้คำนึงถึงเรื่องของจำนวนครั้งในการโดยสารรถประจำทางในขั้นตอนการวิเคราะห์โครงข่ายด้วย เพื่อให้สอดคล้องกับสภาพความเป็นจริงมากยิ่งขึ้น

10. โปรแกรมประยุกต์ที่พัฒนาขึ้นยังไม่ได้เสนอผลลัพธ์ของเส้นทางการเดินทางที่อาจเป็นตัวเลือกอื่นๆ ทำให้ขาดความยืดหยุ่นให้กับผู้ใช้งาน จึงควรมีการพัฒนาโปรแกรมให้สามารถแนะนำเส้นทางการเดินทางที่เป็นทางเลือกอื่นๆ ได้ด้วย

11. ควรมีการนำข้อมูลสภาพการจราจรจริง มาเปรียบเทียบกับผลลัพธ์ที่ได้จากการประมวลผลของโปรแกรม เช่น ระยะเวลาที่ใช้ในการเดินทางรวมจากจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง เพื่อทดสอบความน่าเชื่อถือของโปรแกรม และใช้เป็นแนวทางในการปรับปรุงโปรแกรมประยุกต์

รายการอ้างอิง

ภาษาไทย

จุฬาลงกรณ์มหาวิทยาลัย. ระบบแผนที่: กีฬามหาวิทยาลัยแห่งประเทศไทย ครั้งที่ 38 [online]. 2554. แหล่งที่มา: http://residence.chamchurigames38.chula.ac.th/map/ [7 เมษายน 2555]

พลปรีชา ชิดบุรี. การพัฒนาโปรแกรมประยุกต์บนอุปกรณ์คอมพิวเตอร์พกพา (PDA) เพื่อเลือกเส้นทางที่เหมาะสม ในระบบขนส่งมวลชนหลายรูปแบบ. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาวิศวกรรมสำรวจ ภาควิชาวิศวกรรมสำรวจ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2551.

มนต์ศักดิ์ โซ่เจริญธรรม. ระบบสารสนเทศภูมิศาสตร์ผ่านเครือข่ายอินเทอร์เน็ต เพื่อเลือกเส้นทางที่ใช้เวลาน้อยที่สุด โดยระบบขนส่งมวลชนหลายรูปแบบ. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาวิศวกรรมสำรวจ ภาควิชาวิศวกรรมสำรวจ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2549.

สรวิศ สุภเวชย์ และสรรเพชญ ชื้อนิธิไพศาล. การพัฒนาโปรแกรมประยุกต์เพื่อบริหารจัดการข้อมูลภูมิสารสนเทศบนเวป. การประชุมวิชาการวิศวกรรมโยธาแห่งชาติ ครั้งที่ 13. 14 – 16 พฤษภาคม 2551 ณ โรงแรมจอมเทียน ปาล์ม บีช พัทยา

คมนาคม, กระทรวง. สำนักงานนโยบายและแผนการขนส่งและจราจร. รายงานฉบับสมบูรณ์ : โครงการการพัฒนาระบบรถโดยสารประจำทางในเขตกรุงเทพมหานครและปริมณฑล. กรุงเทพมหานคร : บริษัท แพลนโปร จำกัด, 2552.

องค์การขนส่งมวลชนกรุงเทพ. สถิติการเดินรถ [Portable Document Format]. 2554. แหล่งที่มา: http://www.bmta.co.th/doc/passenger_statistics.pdf [7 เมษายน 2555]

ภาษาอังกฤษ

Alesheikh, AA., Helali, H., and Behroz, HA. Web GIS: Technologies and Its Applications. [Portable Document Format]. 2002. Available from : http://www.isprs.org/proceedings/XXXIV/part4/pdfpapers/422.pdf [2012, May 5]

Chunithipaisan, S. Network Connectivity Model : The Development of Generic, Topology Aware Spatial Datasets and Models. U.S.A.: Lambert Academic Publishing, 2009.

Garrett, J. J. Ajax: A New Approach to Web Applications [Online]. 2005. Available from : http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications [2012, April 7]

Geoserver. Open source server [Online]. 2012. Available from : http://www.geoserver.org [2012, February 2]

JSON. Introducing JSON [Online]. 2012. Available from : http://www.json.org/ [2012, April 7]

Kothuri, R., Godfrind, A., and Beinat, E. Pro Oracle Spatial for Oracle database 11g. U.S.A. : Apress, 2007.

NAVTEQ. ORACLE MAP BUILDER HINTS AND TIPS [Portable Document Format]. 2010. Available from: http://download.oracle.com/otndocs/products/mapviewer/pdf/navteq_mapbuilder_handt.pdf [2012, April 7]

OpenLayers. Open source java script library [Online]. 2012. Available from : http://www.openlayers.org/ [2012, April 7]

Oracle. A Load-On-Demand Approach to Handling Large Network in the Oracle Spatial Network Data Model [Portable Document Format]. 2011. Available from : http://download.oracle.com/otndocs/products/spatial/pdf/ndmlod11gr2_2011.pdf [2012, April 7]

Singh, V., Singh, T., Langan, D., and Kumar, P. A Framwork for Internet GIS Based Computerized Visitor Information System for Theme Parks [Online]. 2004. Available from : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1398983&tag=1 [2012, April 7]

Waldura, R. Dijkstra's Shortest Path Algorithm in Java [Online]. 2007. Available from : http://www.renaud.waldura.com/doc/java/dijkstra/ [2012, April 7]

Worboys, M. F., and Duckham, M. GIS: A Computing Perspective, 2nd edition. U.S.A.: CRC Press, 2004.

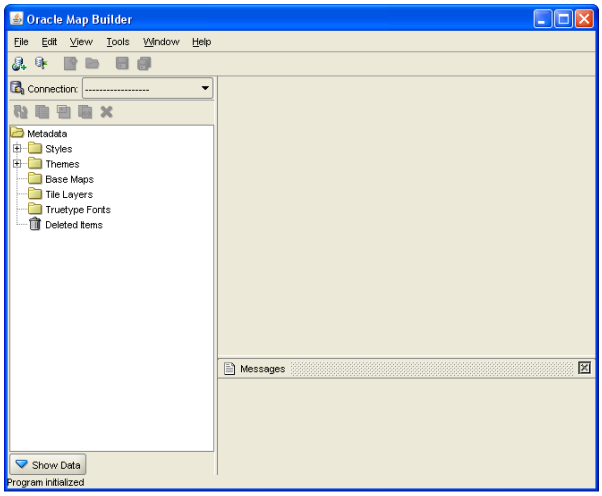
ภาคผนวก

ภาคผนวก ก
การนำเข้าข้อมูลภูมิศาสตร์ (Shapefile) จัดเก็บในระบบฐานข้อมูล Oracle

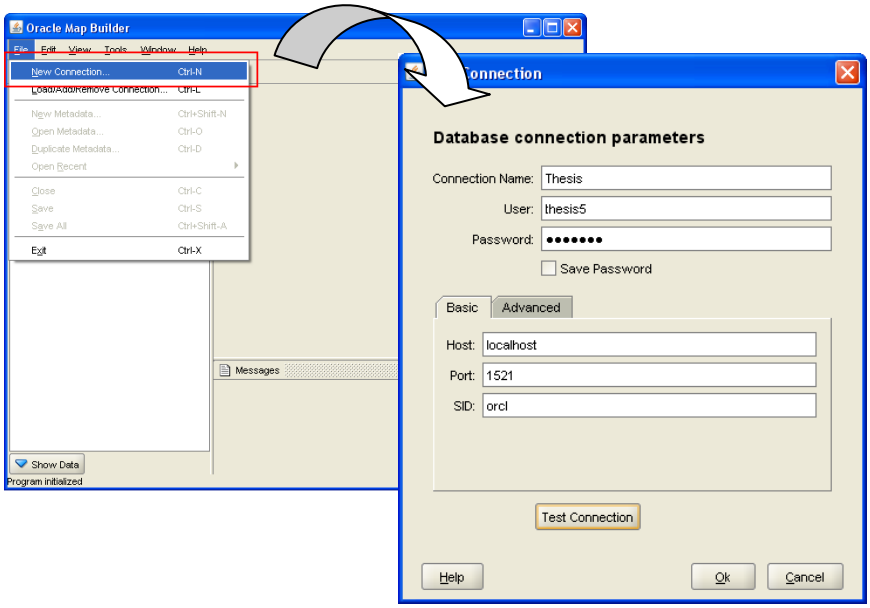
การนำเข้าข้อมูลภูมิศาสตร์ในรูปแบบของ Shapefile จัดเก็บในระบบฐานข้อมูล Oracle Spatial

การนำเข้าข้อมูลภูมิศาสตร์ในรูปแบบของ Shapefiles เพื่อจัดเก็บในระบบฐานข้อมูล Oracle Spatial จะอาศัยซอฟต์แวร์ที่ชื่อว่า Mapbuilder เวอร์ชั่น11g ps5 (11.1.1.6) ในการเชื่อมต่อกับระบบฐานข้อมูลและนำเข้าข้อมูล สามารถดาวน์โหลดได้จาก http://www.oracle.com/technetwork/middleware/mapviewer/downloads/index-100641.html

- เมื่อดาวน์โหลดมาแล้วจะได้ไฟล์ชื่อว่า mapbuilder.jar ให้ทำการเรียกใช้งานจะปรากฏหน้าต่างส่วนติดต่อกับผู้ใช้ดังแสดงในรูป



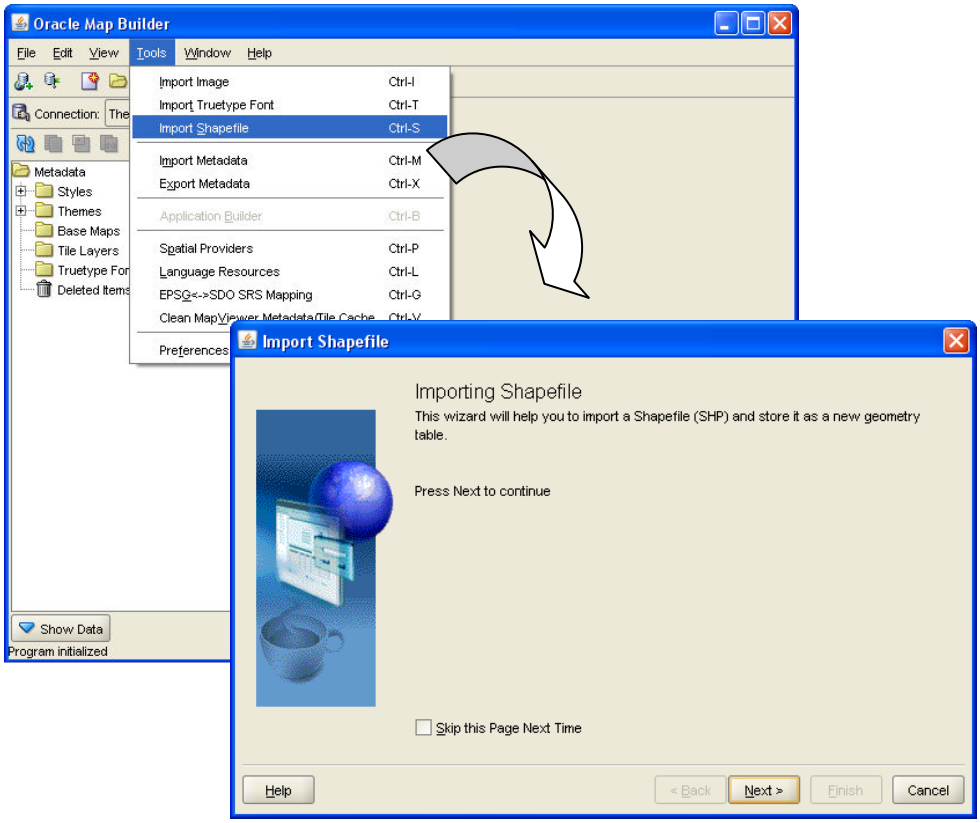
- ทำการเชื่อมต่อกับระบบฐานข้อมูล Oracle โดยเลือกที่คำสั่ง File --> New Connection จะปรากฏหน้าต่างสำหรับระบุค่าในการเชื่อมต่อหน้าต่างดังแสดงในรูป



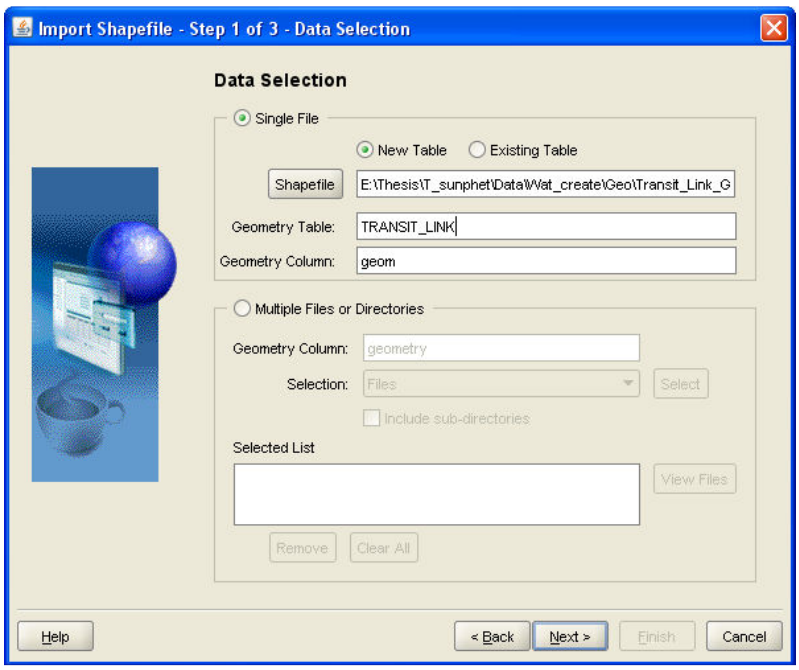
ค่าที่ต้องระบุในการเชื่อมต่อคือ

- Connection Name หมายถึง ชื่อของการเชื่อมต่อ
- User หมายถึง ชื่อของผู้ใช้ในระบบฐานข้อมูล
- Password หมายถึง รหัสผ่านของผู้ใช้งาน
- Host หมายถึง Host ในการเชื่อมต่อโดยระบุ IP address หรือ ระบุ localhost กรณีที่ใช้งานจากเครื่องเดียวกัน
- Port หมายถึง ช่องทางในการติดต่อระบบฐานข้อมูล ค่าตั้งต้นคือ 1521
- SID หมายถึง ชื่อเฉพาะของฐานข้อมูล ในงานวิจัยใช้ชื่อ orcl

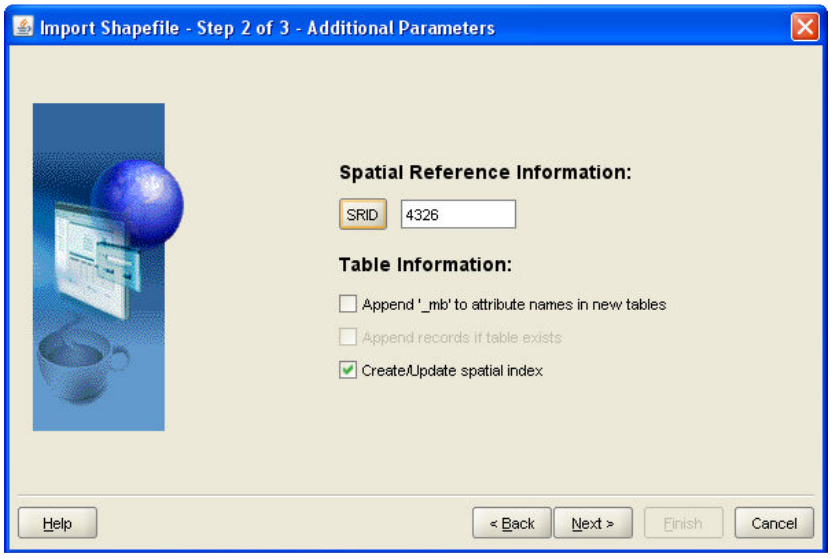
- เมื่อเชื่อมต่อกับระบบฐานข้อมูลแล้วให้ไปที่คำสั่ง Tools --> Import Shapefile จะปรากฏหน้าต่างในการเริ่มต้นการนำเข้าข้อมูลให้เลือกที่ Next ดังแสดงในรูป



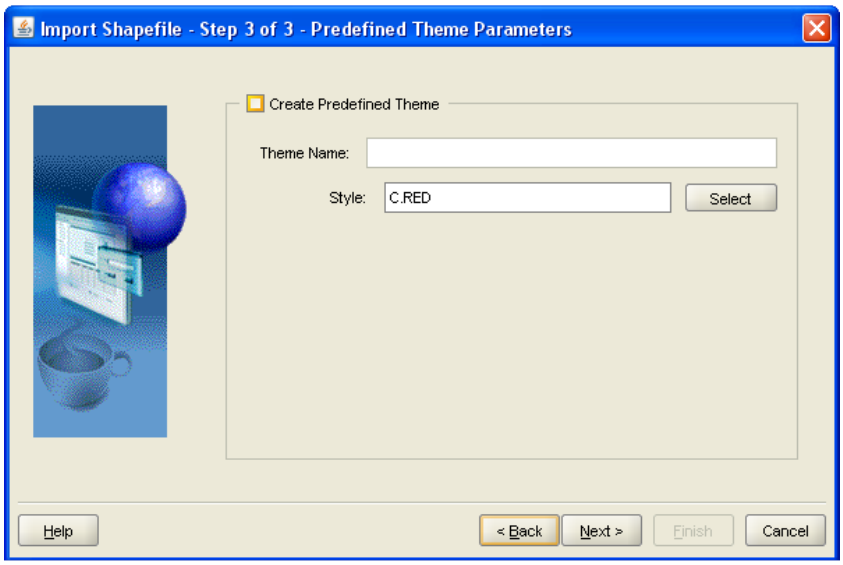
- จะปรากฏหน้าต่างในการเลือกชั้นข้อมูล Shapefile ที่ต้องการนำเข้า ให้เลือกข้อมูล Shapefile ที่ต้องการ จากนั้นตั้งชื่อตารางและชื่อคอลัมน์ที่ใช้ในการจัดเก็บข้อมูลเชิงตำแหน่ง และเลือกที่ Next



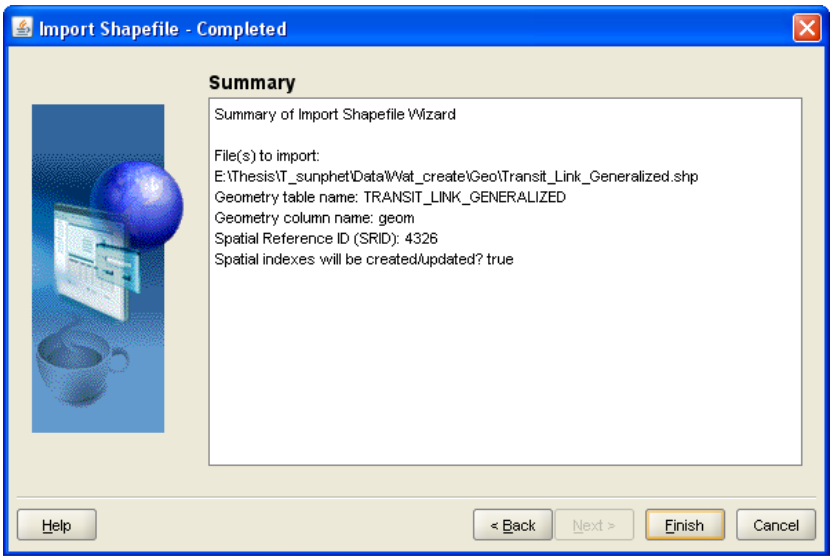
- จะปรากฏหน้าต่างเพื่อให้ระบุค่าของ SRID จากนั้นเลือกที่ Next



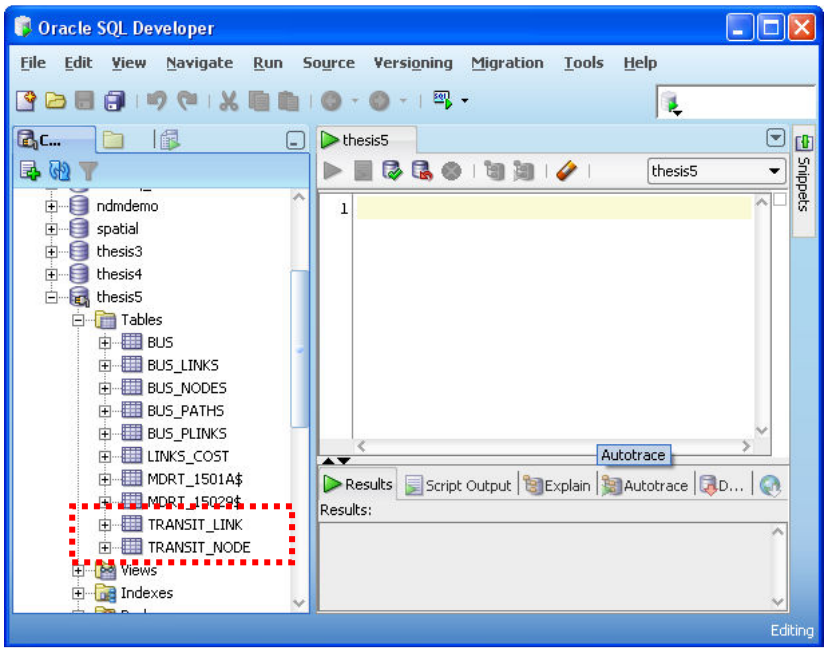
- จะปรากฏหน้าต่างเพื่อให้ระบุสีในการแสดงผลชั้นข้อมูลภูมิศาสตร์ที่นำเข้าโดยมีค่าเริ่มต้น คือ สีแดง เมื่อเลือกสีในการแสดงผลแล้วให้เลือกที่ Next



- จะปรากฏหน้าต่างสรุปของชั้นข้อมูลที่นำเข้า เพื่อตรวจสอบความถูกต้องของค่าต่างๆ ที่กำหนดอีกครั้ง จากนั้นให้คลิกที่ Finish ซอฟต์แวร์จะทำการนำเข้าข้อมูลจัดเก็บลงในระบบฐานข้อมูล



- ตรวจสอบดูผลลัพธ์ของการนำเข้าชั้นข้อมูลว่ามีตารางเพิ่มขึ้นมาหรือไม่ โดยตรวจสอบได้จากซอฟต์แวร์ Oracle SQL Developer หากนำเข้าเสร็จสมบูรณ์จะต้องมีตารางเพิ่มเข้ามาในระบบฐานข้อมูล

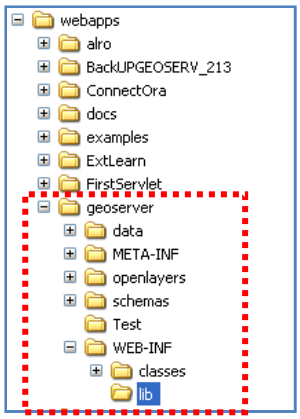


ภาคผนวก ข
การเชื่อมต่อ Geoserver กับระบบฐานข้อมูล Oracle Spatial

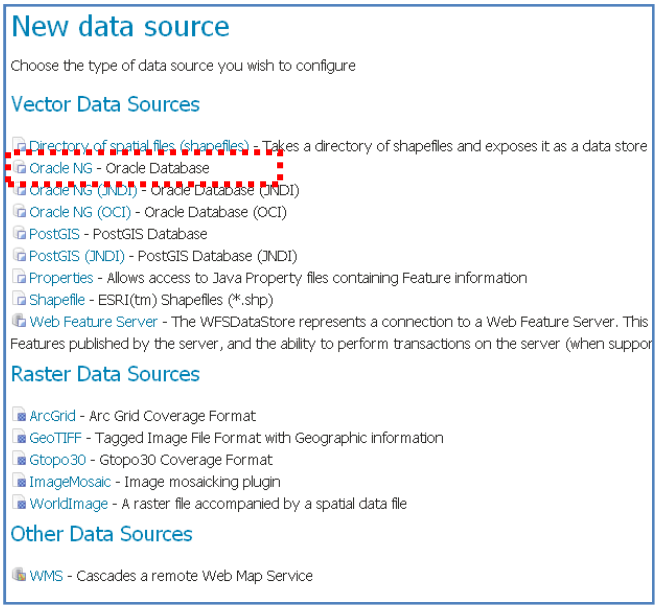
การเชื่อมต่อ Geoserver กับระบบฐานข้อมูล Oracle Spatial และการกำหนดค่าการให้บริการข้อมูลภูมิสารสนเทศผ่านเครือข่ายอินเทอร์เน็ต

การเชื่อมต่อชั้นข้อมูลปริภูมิในระบบฐานข้อมูล Oracle กับแม่ข่ายแผนที่ Geoserver เพื่อกำหนดค่าการให้บริการภาพแผนที่ มีขั้นตอนต่างๆ ดังต่อไปนี้

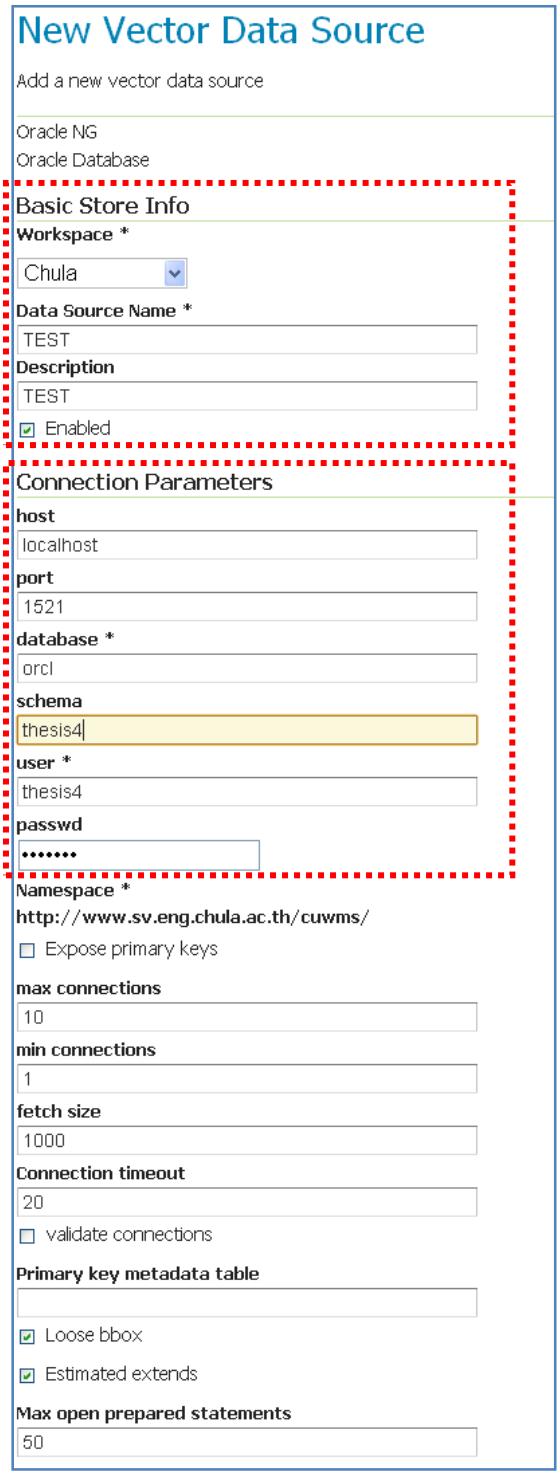
- ในงานวิจัยใช้ซอฟต์แวร์ Geoserver เวอร์ชั่น 2.11 ในการเชื่อมต่อกับระบบฐานข้อมูล Oracle Spatial จำเป็นต้องติดตั้ง Extensions เพื่อเพิ่ม Datastore ให้กับ Geoserver สามารถดาวน์โหลดได้จาก http://geoserver.org/display/GEOS/GeoServer+2.1.1 เมื่อดาวน์โหลดมาแล้วให้คัดลอกไฟล์ gt-jdbc-oracle-2.7.2.jar และ ojdbc14.jar ไปไว้ภายใต้ [container]/webapps/geoserver/WEB-INF/lib/ ดังแสดงในรูป



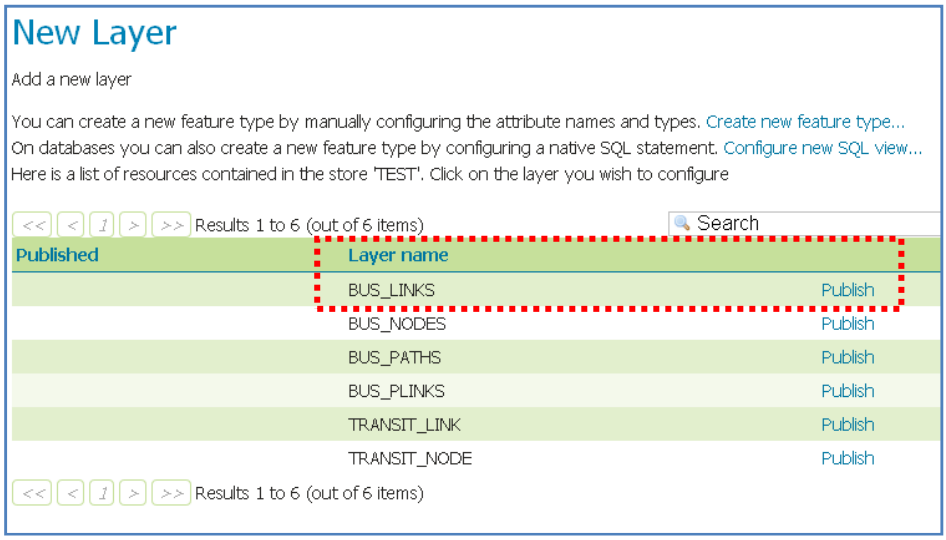
- ทำการ Restart Geoserver จากนั้นให้ไปที่ Vector Data Source จะปรากฏ Oracle NG-Oracle Database สำหรับเชื่อมต่อกับระบบฐานข้อมูล Oracle



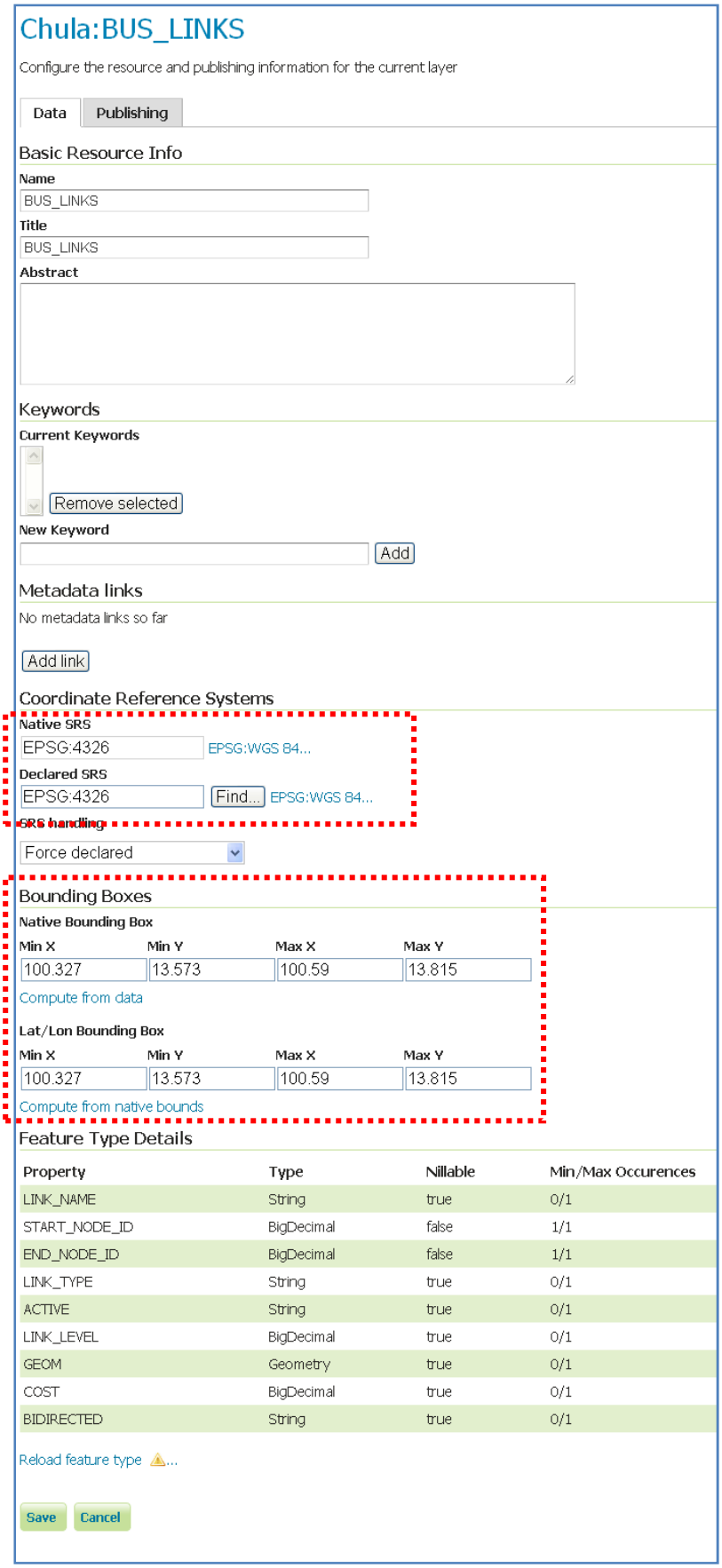
- เลือกไปที่ Oracle NG เพื่อกำหนดค่าการเชื่อมต่อ Geoserver เข้ากับระบบฐานข้อมูล Oracle Spatial จะปรากฏหน้าต่างในการกำหนดค่าต่างๆ ดังแสดงในรูป เมื่อระบุค่าต่างๆ เสร็จแล้วให้เลือกที่ Save



- จะปรากฏหน้าต่างให้เราเลือก Publish ชั้นข้อมูลที่ต้องการให้บริการภาพแผนที่ ในงานวิจัยนี้ได้แก่ชั้นข้อมูล BUS_LINKS และ BUS_NODES



- เมื่อเลือกที่ Publish จะปรากฏหน้าต่างในการกำหนดค่าต่างๆ ในการให้บริการภาพแผนที่ของชั้นข้อมูลที่เลือก ให้เราระบุค่าต่างๆ ให้ถูกต้อง ดังแสดงตัวอย่างในรูป



Chula:BUS_LINKS

Configure the resource and publishing information for the current layer

Data | Publishing

Basic Resource Info

Name: BUS_LINKS

Title: BUS_LINKS

Abstract:

Keywords

Current Keywords

Remove selected

New Keyword

Add

Metadata links

No metadata links so far

Add link

Coordinate Reference Systems

Native SRS: EPSG:4326 EPSG:WGS 84...

Declared SRS: EPSG:4326 Find... EPSG:WGS 84...

SRS handling: Force declared

Bounding Boxes

Native Bounding Box

Min X	Min Y	Max X	Max Y
100.327	13.573	100.59	13.815

Compute from data

Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
100.327	13.573	100.59	13.815

Compute from native bounds

Feature Type Details

Property	Type	Nillable	Min/Max Occurences
LINK_NAME	String	true	0/1
START_NODE_ID	BigDecimal	false	1/1
END_NODE_ID	BigDecimal	false	1/1
LINK_TYPE	String	true	0/1
ACTIVE	String	true	0/1
LINK_LEVEL	BigDecimal	true	0/1
GEOM	Geometry	true	0/1
COST	BigDecimal	true	0/1
BIDIRECTED	String	true	0/1

Reload feature type ...

Save Cancel

- ทดลองแสดงผลชั้นข้อมูลที่เชื่อมต่อจากระบบฐานข้อมูล Oracle โดยไปที่ Layer Preview และเลือกชั้นข้อมูลที่ได้ Publish ไว้ จะปรากฏหน้าต่างของภาพแผนที่ดังแสดงในรูป

ประวัติผู้เขียนวิทยานิพนธ์

ชื่อ: นายภาณุวัฒน์ อังคสุรักษ์

วันเดือนปีเกิด: 11 มีนาคม พ.ศ. 2525

คุณวุฒิทางการศึกษา:

พ.ศ. 2547	วิทยาศาสตรบัณฑิต ภาควิชาเทคโนโลยีชนบท คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์
พ.ศ. 2555	วิทยาศาสตรมหาบัณฑิต สาขาวิชาระบบสารสนเทศปริภูมิทางวิศวกรรม ภาควิชาวิศวกรรมสำรวจ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ประสบการณ์ทำงาน:

พ.ศ. 2553-2554	ผู้ช่วยสอนปฏิบัติการวิชา “หลักการสำรวจ” ภาควิชาวิศวกรรมสำรวจ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
พ.ศ. 2553-2555	ผู้ช่วยสอนปฏิบัติการวิชา “ระบบสารสนเทศทางภูมิศาสตร์เพื่อธุรกิจ” คณะเทคโนโลยีสารสนเทศและการสื่อสารมหาวิทยาลัยศิลปากร
พ.ศ. 2554-2555	ผู้ช่วยวิทยากรโครงการศึกษาและพัฒนาชุดคำสั่งประยุกต์ด้านการจัดการเชิงพื้นที่รหัสเปิด หลักสูตรชุดคำสั่งประยุกต์รหัสเปิดด้านการจัดทำระบบให้บริการข้อมูลเชิงพื้นที่โดยผ่านเครือข่ายอินเทอร์เน็ต กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร

รางวัลที่ได้รับ: บทความยอดเยี่ยมอันดับ 2 ในงานการประชุมวิชาการวิศวกรรมโยธาแห่งชาติ ครั้งที่ 17 ระหว่างวันที่ 9 – 11 พฤษภาคม พ.ศ. 2555 ณ โรงแรมเซ็นทาราโฮเทลแอนด์คอนเวนชั่นเซ็นเตอร์ อุดรธานี

ผลงานทางวิชาการ:บทความตีพิมพ์

1. **ภาณุวัฒน์ อังคสุรักษ์**, สรรเพชญ ชื้อนิธิไพศาล และสรวิศ ศุภเวชย์. 2555. การออกแบบและพัฒนาโปรแกรมประยุกต์ภูมิสารสนเทศออนไลน์สำหรับการวางแผนการเดินทางด้วยรถโดยสารประจำทาง. การประชุมวิชาการวิศวกรรมโยธาแห่งชาติครั้งที่ 17. โรงแรมเซ็นทาราโฮเทลแอนด์คอนเวนชั่นเซ็นเตอร์ จังหวัดอุดรธานี 9-11 พฤษภาคม 2555