ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

การพัฒนาระบบนำทางอัตโนมัติ

AUTOMATIC NAVIGATION SYSTEM DEVELOPMENT

โดย

ภาณุวัฒน์ เรืองรักษ์

PANUWAT RUANGRAK

อาจารย์ที่ปรึกษา

ผศ.ดร. ธนารัตน์ ชลิดาพงศ์

ส ภ
ว 4347
2550

H004895

เลขหมู่.....
เลขทะเบียน 04895
วัน,เดือน,ปี 6 พ.ย. 2551

b.11978794 X
i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC NAVIGATION SYSTEM DEVELOPMENT

PANUWAT RUANGRAK

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2008

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	การพัฒนาระบบนำทางอัตโนมัติ
นักศึกษา	นาย ภาณุวัฒน์ เรืองรักษ์
รหัสนักศึกษา	48066822
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	ผศ.ดร. ธนารัตน์ ชลิดาพงศ์

บทคัดย่อ

โครงการพัฒนาระบบงานฉบับนี้เสนอระบบนำทางอัตโนมัติที่สามารถนำเอาสภาพการจราจร ณ เวลาปัจจุบัน มาช่วยในการหาคำนวณหาเส้นทางที่เร็วที่สุดได้ ซึ่งต่างจากระบบนำทางอัตโนมัติในปัจจุบันที่มักจะหาได้เฉพาะเส้นทางที่สั้นที่สุด โครงการพัฒนาระบบงานฉบับนี้ได้กล่าวถึงการพัฒนาระบบนำทางอัตโนมัติ โดยประกอบด้วยการกำหนดความต้องการของระบบ การวิเคราะห์และออกแบบลักษณะการทำงานของระบบ วิธีการใช้ไดสตราอัลกอริทึมเพื่อหาเส้นทางสั้นที่สุดและเร็วที่สุด วิธีการการนำทางโดยการนำเสนอวิธีคำนวณหาข้อความนำทาง ณ ตำแหน่งต่างๆ และโครงสร้างข้อมูลที่ใช้ในระบบ และการพัฒนาระบบมีส่วนของไดสตราอัลกอริทึมเพื่อหาเส้นทางสั้นที่สุดและเร็วที่สุด ส่วนของการนำทางอัตโนมัติเป็นการคำนวณหาข้อความนำทางจากตำแหน่งปัจจุบันไปยังตำแหน่งเป้าหมาย รวมถึงการพัฒนาส่วนติดต่อระหว่างอุปกรณ์คอมพิวเตอร์แบบพกพากับ Traffic Monitoring Server เพื่อ Sync ข้อมูลการจราจรผ่าน FTP protocol โดยผลการพัฒนาคือ ระบบสามารถหาเส้นทางที่สั้นที่สุดและเร็วที่สุดได้อย่างถูกต้อง และสามารถคำนวณข้อความนำทาง ณ ตำแหน่งต่างๆ ได้ นอกจากนี้ระบบสามารถ Sync ข้อมูลกับ Traffic Monitoring Server ได้ แต่ปัญหาที่พบคือ ระบบสามารถทำงานได้เฉพาะในพิกัดทดสอบเท่านั้น ระบบไม่สามารถทำงานกับ latitude และ longitude ของระบบภูมิศาสตร์จริงๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Automatic Navigation System Development
Student	Mr. Panuwat Ruangrak
Student ID.	48066822
Degree	Master of Science
Programme	Information Science
Academic Year	2007
Advisor	Asst. Prof. Dr. Thanarat H. Chalidabhongse

ABSTRACT

This project proposes automatic navigation system which can use traffic date to find out the fastest route. It's differenced from current navigation system which can find out only the shortest route. This project describes the automatic navigation system development. The details are to define system requirement, analysis and design system characteristic, finding out shortest and fastest path by using Dijkstra algorithm, navigating methodology and computing the navigation message at each position and defining system data structure. The development phase are developing Dijkstra algorithm, developing automatic navigation in order to generate navigation message from current to target position, and developing connectivity between handheld device and traffic monitoring server in order to sync traffic data over FTP protocol. The result are the system can find out the correctly shortest and fastest route, and it can remind with correctly navigation message at variation position, and the system can sync traffic date with traffic monitoring server. But the problem is the system can process only test co-ordinate as handheld co-ordinate, and the system can not process the actual co-ordinate of real world geographic as latitude and longitude.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณ ผศ.ดร.ธนารัตน์ ชลิดาพงศ์ อาจารย์ที่ปรึกษาของโครงงานพัฒนาระบบงาน ที่กรุณาให้ความรู้และคำแนะนำอันเป็นประโยชน์อย่างมาก ต่อการพัฒนาโครงงานนี้ตลอดจนตรวจสอบแก้ไขจนกระทั่งโครงงานสำเร็จลุล่วง

ขอขอบคุณบุคคลากรของคณะเทคโนโลยีสารสนเทศทุกท่าน ที่ให้ความช่วยเหลือเรื่องเอกสารและให้ความอนุเคราะห์ตลอดมา

ขอขอบคุณเพื่อนๆ และพี่ๆ สาขาวิชาเทคโนโลยีสารสนเทศ ที่ได้ให้คำแนะนำช่วยเหลือผู้จัดทำ ในเรื่องแนวทางการเขียนและแก้ไขโปรแกรมให้สามารถทำงานได้ประสบผลสำเร็จ

ขอขอบคุณเพื่อนๆ และพี่ๆ ที่ร่วมงานกันในบริษัท ซี ซี ซอฟแวร์ จำกัด และ บริษัท โทเทิ่ล แอ็คเซ็ส คอมมูนิเคชั่น มหาชน จำกัด ที่เป็นกำลังใจ ให้คำปรึกษา และอำนวยความสะดวกตลอดระยะเวลาการศึกษา และการทำงานโครงการพัฒนาระบบ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา น้องๆทั้ง 3 คน และคุณช่อผกา ที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่อง ทำให้สามารถทำโครงงานสำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากโครงงานพัฒนาระบบงานฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

ภาณุวัฒน์ เรืองรักษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตและความต้องการของระบบ 1

1.4 ขั้นตอนการพัฒนาระบบ 2

1.5 เครื่องมือที่ใช้ในการพัฒนาระบบ 4

1.6 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 ระบบนำทางอัตโนมัติ 5

2.1 ระบบระบุตำแหน่งพื้นโลก 5

2.2 ระบบพิกัดในงานด้านแผนที่ 6

2.2.1 การจำแนกระบบพิกัด 7

2.2.1 ระบบพิกัดที่พบมากในงานแผนที่ 8

2.3 ไดสตราอัลกอริทึม 10

2.3.1 ขั้นตอนการทำงานของอัลกอริทึม 10

2.3.2 ตัวอย่างการหาระยะทางสั้นสุดโดยใช้ไดสตราอัลกอริทึม 11

2.4 วงจรการพัฒนาระบบ 13

2.5 Microsoft .NET Framework 14

บทที่ 3 การวิเคราะห์และออกแบบระบบ 17

3.1 กำหนดความต้องการของระบบ 17

3.2 ลักษณะการทำงานของระบบ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2.1 หลักการในการหาเส้นทางที่เหมาะสม..... 18

3.2.2 การนำทางอัตโนมัติ..... 19

3.3 Use Case Diagram..... 19

3.4 Sequence Diagram..... 22

3.5 Class Diagram..... 23

3.6 การออกแบบระบบฐานข้อมูล..... 25

3.7 อัลกอริทึมในการหาเส้นทาง..... 27

3.8 การคำนวณหาข้อความนำทาง..... 29

บทที่ 4 การพัฒนาระบบ..... 31

4.1 ขั้นตอนการพัฒนาระบบ..... 31

4.2 รายละเอียดการพัฒนาระบบ..... 31

4.2.1 การพัฒนาส่วนของการกำหนดรูปแบบข้อมูลของแผนที่และข้อมูลการจราจร และการนำเข้าสู่ระบบ..... 31

4.2.2 การพัฒนาส่วนของการแสดงผลแผนที่บนหน้าจอของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพา..... 32

4.2.3 การพัฒนาส่วนของการหาเส้นทางโดยไดสตราอัลกอริทึม..... 35

4.2.4 การพัฒนาส่วนของการแสดงเส้นทางที่สั้นที่สุดและเร็วที่สุดให้ผู้ใช้เลือกเพื่อกำหนดเส้นทางที่ต้องการเดินทาง..... 35

4.2.5 การพัฒนาส่วนของการนำทางอัตโนมัติ..... 38

4.2.6 การพัฒนาส่วนของการติดต่อกับ GPS เพื่อรับพิกัดของตำแหน่งปัจจุบัน..... 40

4.2.7 การพัฒนาส่วนของการติดต่อกับ Traffic Monitoring Server เพื่อรับข้อมูลการจราจร ณ เวลาปัจจุบัน..... 42

4.2.8 การพัฒนาส่วนของการค้นหาสถานที่ที่น่าสนใจ..... 44

4.2.9 การพัฒนาส่วนของการติดต่อกับผู้ใช้ 44

4.2.10 ทดสอบความถูกต้องของระบบ 46

บทที่ 5 บทสรุปและข้อเสนอแนะ..... 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

5.1 สรุปผลการพัฒนา..... 59
5.2 ปัญหาและอุปสรรค..... 59
5.3 ข้อเสนอแนะ..... 60

บรรณานุกรม..... 61

ภาคผนวก..... 62
ภาคผนวก ก. คู่มือติดตั้งระบบนำทางอัตโนมัติ..... 62
ภาคผนวก ก. คู่มือการใช้งานระบบนำทางอัตโนมัติ..... 65

ประวัติผู้เขียน..... 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

1.1 ตารางเวลาในการพัฒนาระบบ..... 3

2.1 ตารางสรุประบบพิกัดที่ใช้ในงานแผนที่..... 9

2.2 ตารางแสดงเส้นทางสั้นที่สุดจาก A ไปยังโหนดอื่นๆ.....11

2.3 ตารางแสดงเส้นทางสั้นที่สุดจาก A ไปยังโหนดอื่นๆเมื่อผ่านอัลกอริทึมไดสตรารอบที่ 1...12

2.4 ตารางแสดงเส้นทางสั้นที่สุดจาก A ไปยังโหนดอื่นๆเมื่อผ่านอัลกอริทึมไดสตรารอบที่ 2...12

2.5 ตารางแสดงเส้นทางสั้นที่สุดจาก A ไปยังโหนดอื่นๆเมื่อผ่านอัลกอริทึมไดสตรารอบที่ 6...13

3.1 รายละเอียดของตาราง WP.....26

3.2 รายละเอียดของตาราง WAY.....26

3.3 รายละเอียดของตาราง SHORTEST_PATH.....27

3.4 รายละเอียดของตาราง FASTEST_PATH.....27

3.5 ตารางแสดงข้อความนำทางของค่าแนวโน้ม X เพิ่มขึ้นและ Y ลดลง 30

3.6 ตารางแสดงข้อความนำทางทุก ๆ แนวโน้มของค่า X และ Y.....30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 กราฟตัวอย่างมี A เป็นโหนดเริ่มต้น..... 11

2.2 โครงสร้างของ Microsoft .NET Framework..... 15

3.1 แสดงการเชื่อมต่อของระบบนำทางอัตโนมัติ..... 18

3.2 Use Case Diagram ของระบบนำทางอัตโนมัติ.....20

3.3 Sequence Diagram ของ Use Case Navigate to the target way point..... 23

3.4 Class Diagram ของระบบนำทางอัตโนมัติ..... 24

3.5 รูปแสดง Flow Chart การทำงานของไดสตราอัลกอริทึม..... 28

3.6 รูปแสดงตัวอย่างการเคลื่อนที่จาก Current ผ่าน Intersection ไปยัง Target Point ต่าง ๆ 29

4.1 แสดงลักษณะของแกน X และแกน Y บนหน้าจอเครื่องคอมพิวเตอร์แบบพกพา..... 33

4.2 หน้าจอแสดงข้อมูลแผนที่บนเครื่องคอมพิวเตอร์แบบพกพา.....34

4.3 หน้าจอแสดงพิกัดของจุดที่สนใจและเวลาการเดินทางของแต่ละถนนบนแผนที่.....34

4.4 หน้าจอแสดงส่วนติดต่อกับผู้ใช้เพื่อรับ Input จุดเริ่มต้นและจุดสิ้นสุดที่ต้องการ..... 35

4.5 หน้าจอแสดงส่วนติดต่อกับผู้ใช้เพื่อแสดงผลการคำนวณเส้นทางแบบระยะทางสั้นที่สุด (Shortest Route).....37

4.6 หน้าจอแสดงส่วนติดต่อกับผู้ใช้เพื่อแสดงผลการคำนวณเส้นทางแบบใช้เวลาน้อยที่สุด (Fastest Route)..... 37

4.7 หน้าจอแสดงโหมดนำทางอัตโนมัติ..... 38

4.8 ตัวอย่างหน้าจอแสดงข้อความนำทางเมื่อเริ่มระบบนำทางอัตโนมัติ.....39

4.9 ตัวอย่างหน้าจอแสดงแสดงข้อความนำทางในโหมดทดสอบระบบนำทางอัตโนมัติ..... 40

4.10 หน้าจอแสดงการเลือก Com Port เพื่อเชื่อมต่อ GPS Receiver 41

4.11 หน้าจอแสดงเริ่มการเชื่อมต่อและรับข้อมูลจาก GPS Receiver 42

4.12 หน้าแสดงการกำหนด Server, User, Password และ Path สำหรับการ Sync ข้อมูลกับ Traffic Monitoring Server 43

4.13 หน้าแสดงเมนูส่วนเริ่มการติดต่อกับ Traffic Monitoring Server 43

4.14 ตัวอย่างหน้าจอการรับ input เพื่อค้นหาสถานที่ที่สนใจ..... 44

4.15 ตัวอย่างหน้าจอการแสดงตำแหน่งสถานที่จากการค้นหาบนแผนที่.....44

4.16 ตัวอย่างหน้าจอแสดงเมนูของระบบ..... 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.17 หน้าจอแสดงแผนที่ที่ได้จากการโหลดข้อมูลมาจาก file “wp.txt” และ “way.txt” โดยแสดงเป็นระยะทางระหว่างจุด.....47

4.18 หน้าจอแสดงแผนที่ที่ได้จากการโหลดข้อมูลมาจาก file “wp.txt” และ “way.txt” โดยแสดงเป็นเวลาในการเดินทางระหว่างจุด..... 48

4.19 หน้าจอแสดงแผนที่และพิกัด x,y ของจุดสนใจที่ได้จาก file “wp.txt”.....48

4.20 หน้าจอแสดงเส้นทางที่มีระยะทางสั้นที่สุดจาก Casaview ไป LPN 49

4.21 หน้าจอแสดงเส้นทางที่มีระยะทางสั้นที่สุดบนแผนที่จาก Casaview ไป LPN 50

4.22 หน้าจอแสดงเส้นทางที่ใช้เวลาน้อยที่สุดจาก Casaview ไป LPN 50

4.23 หน้าจอแสดงเส้นทางที่ใช้เวลาน้อยที่สุดบนแผนที่จาก Casaview ไป LPN 51

4.24 หน้าจอแสดงเส้นทางที่ผู้ใช้เลือก..... 52

4.25 หน้าจอแสดงผลการเริ่มต้นโหมดนำทางอัตโนมัติ..... 52

4.26 หน้าจอแสดงผลข้อความนำทาง ณ จุดที่ถัดจากจุดเริ่มต้น..... 53

4.27 หน้าจอแสดงผลข้อความนำทาง ณ จุดถัดไปจากจุดเริ่มต้น 2 จุด.....53

4.28 หน้าจอแสดงผลข้อความนำทางเมื่อถึงปลายทาง.....54

4.29 หน้าจอแสดงการเข้าสู่เมนูค้นหาสถานที่ที่สนใจ.....54

4.30 หน้าจอแสดงการเลือกประเภทสถานที่ที่สนใจ..... 55

4.31 หน้าจอแสดงผลสถานที่ตามประเภทที่ผู้ใช้เลือก.....55

4.32 หน้าจอแสดงเวลาของแต่ละถนนก่อนเปลี่ยนค่าที่ Traffic Monitoring Server 56

4.33 หน้าจอแสดงการตั้งค่าสำหรับการเชื่อมต่อกับ Traffic Monitoring Server 56

4.34 หน้าจอแสดงการเปิดให้มีการ Sync ข้อมูลกับ Traffic Monitoring Server 57

4.35 หน้าจอแสดงเวลาบนถนนแต่ละถนนหลังจากเปลี่ยนค่าเวลาที่ Traffic Monitoring Server แล้ว..... 58

ก-1 หน้าจอแสดงการติดตั้ง .NET Compact Framework 2.0 แบบติดตั้งผ่าน active sync ของเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ.....62

ก-2 หน้าจอแสดงแผนที่เมื่อเปิดโปรแกรมบนอุปกรณ์คอมพิวเตอร์แบบพกพา.....64

ข-1 หน้าจอแสดงการเลือกประเภทสถานที่ที่สนใจ.....65

ข-2 หน้าจอแสดงการกำหนดจุดเริ่มต้นและจุดปลายทาง..... 65

ข-3 หน้าจอแสดงเส้นทางที่มีระยะทางสั้นที่สุด..... 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

ข-4 หน้าจอแสดงเส้นทางที่ผู้ใช้เลือก..... 67

ข-5 หน้าจอแสดงการเริ่มต้นโหมดนำทางอัตโนมัติ..... 67

ข-6 หน้าจอแสดงข้อความนำทาง (Navigation Message)..... 68

ข-7 หน้าจอแสดงข้อความเมื่อถึงปลายทาง..... 68

ข-8 หน้าจอแสดงการเข้าสู่เมนูค้นหาสถานที่ที่สนใจ..... 69

ข-9 หน้าจอแสดงการเลือกประเภทสถานที่ที่สนใจ..... 69

ข-10 หน้าจอแสดงสถานที่ตามประเภทที่ผู้ใช้เลือก..... 70

ข-11 หน้าจอแสดงเมนูการเปิดให้แสดงพิกัด X,Y ของจุดบนแผนที่..... 70

ข-12 หน้าจอแสดงพิกัด x,y ของจุดบนแผนที่..... 71

ข-13 หน้าจอแสดงการเปิดการติดต่อกับ Traffic Monitoring Server 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

การเดินทางของผู้ใช้รถยนต์ในปัจจุบัน มักจะประสบกับปัญหาเรื่องการหลงทาง ไม่ทราบเส้นทาง หรือรถติดบนท้องถนนเป็นระยะเวลานาน ทำให้เกิดผลเสียกับผู้ขับขี่รถยนต์บนท้องถนน และอาจเกิดผลกระทบต่อธุรกิจได้ ระบบนำทางอัตโนมัติ (Automatic Navigation System) เป็นระบบที่ใช้การวิเคราะห์ระยะทางหรือสภาพการจราจรมากำหนดเส้นทางที่เหมาะสมในการเดินทางไว้ล่วงหน้า และใช้ตำแหน่งทางภูมิศาสตร์ของยานพาหนะจากอุปกรณ์ตัวรับ GPS (Global Positioning System Receiver) เพื่อบอกตำแหน่งที่อยู่บนแผนที่ และช่วยนำทางให้กับผู้ขับขี่เดินรถไปตามเส้นทางที่ได้กำหนดไว้ แต่ระบบนำทางอัตโนมัติในปัจจุบันมักจะเป็นแบบไม่ online ข้อมูลการจราจร คือไม่สามารถวิเคราะห์หาเส้นทางจากข้อมูลการจราจรแบบ real-time ณ เวลาปัจจุบันของถนนแต่ละเส้นได้ เช่น ไม่สามารถรู้ได้ว่าถนนเส้นไหนถูกปิดไม่ให้รถวิ่งผ่าน หรือ ไม่สามารถนำเอาเวลาที่ใช้ในการเดินทางของถนนแต่ละเส้นมาวิเคราะห์หาเส้นทางที่เร็วที่สุดได้ โครงงานฉบับนี้จึงทำขึ้นมาเพื่อปรับปรุงระบบนำทางอัตโนมัติให้สามารถใช้ข้อมูลการจราจรแบบ real-time มาช่วยในการวิเคราะห์หาเส้นทางเพื่อให้ผู้ใช้สามารถเลือกเส้นทางที่ตรงกับความต้องการมากที่สุดได้ โดยเอกสารฉบับนี้จะกล่าวถึงการพัฒนาระบบนำทางอัตโนมัติ โดยประกอบด้วยหัวข้อตามลำดับดังต่อไปนี้ บทนำ ระบบนำทางอัตโนมัติ เครื่องมือที่ใช้ในการพัฒนาระบบ การวิเคราะห์และออกแบบระบบ การพัฒนาระบบ และสุดท้ายเป็นบทสรุปและข้อเสนอแนะ

1.2 วัตถุประสงค์

เพื่อนำทางให้ผู้ขับขี่รถยนต์สามารถเดินทางไปยังจุดหมายได้ โดยใช้เส้นทางที่เหมาะสมที่สุด

1.3 ขอบเขตและความต้องการของระบบ

ระบบนำทางอัตโนมัติเป็นระบบที่ทำงานบนเครื่องคอมพิวเตอร์แบบพกพาซึ่งติดต่อกับ GPS เพื่อใช้ในการนำทาง โดยมีการรับข้อมูลการจราจร ณ เวลาปัจจุบันเพื่อช่วยในการหาเส้นทางที่เหมาะสมด้วย โดยมีขอบเขตและความต้องการของระบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สามารถบอกเส้นทางที่มีระยะทางสั้นที่สุดระหว่างจุดเริ่มต้นกับจุดที่ต้องการใดๆบนแผนที่ได้
2. สามารถบอกเส้นทางที่ใช้เวลาเดินทางสั้นที่สุดระหว่างจุดเริ่มต้นกับจุดที่ต้องการใดๆบนแผนที่ได้ โดยใช้ข้อมูลการจราจร ณ เวลาปัจจุบัน เพื่อหลีกเลี่ยงเส้นทางที่มีการจราจรแออัด หรือ เส้นทางที่ปิดไม่ให้รถผ่านได้
3. สามารถนำทางให้เดินทางไปตามเส้นทางที่กำหนดไว้ได้
4. สามารถบอกตำแหน่งปัจจุบันบนแผนที่ได้
5. สามารถคำนวณเส้นทางโดยใช้เวลาไม่เกิน 60 วินาที สำหรับ CPU OMAP850-195MHz บนเครื่องคอมพิวเตอร์แบบพกพา
6. สามารถค้นหาจุด Way Point หรือ สถานที่ที่ต้องการจะไปได้ เช่น ร้านอาหาร อู่ซ่อมรถ ธนาคาร เป็นต้น และสามารถให้ระบบนำทางไปยังสถานที่ที่ต้องการได้
7. เครื่องคอมพิวเตอร์แบบพกพาต้องสามารถเชื่อมต่ออินเตอร์เน็ทได้ เช่น ผ่าน GPRS หรือ Wireless LAN เพื่อรับข้อมูลการจราจร ณ เวลาปัจจุบัน มาใช้ในการหาเส้นทางที่ใช้เวลาเดินทางสั้นที่สุดได้
8. ข้อมูลการจราจรได้มาจาก Traffic Monitoring System ให้ถือว่าเป็นข้อมูลล่าสุดโดยระบบเพียงจำลองการติดต่อเพื่อให้ได้ข้อมูลมาเท่านั้น ระบบไม่ได้มีส่วนการทำงานที่เกี่ยวกับการแก้ไขหรือการได้มาซึ่งข้อมูลการจราจรบน Traffic Monitoring System

1.4 ขั้นตอนการพัฒนาระบบ

ในการพัฒนาระบบนำทางอัตโนมัตินี้ เป็นการพัฒนาแอพพลิเคชันในลักษณะของแอพพลิเคชันบนคอมพิวเตอร์แบบพกพา (Mobile Application) โดยมีขั้นตอนและแผนงานในการพัฒนาดังนี้

1. การวางแผนการพัฒนาระบบ (Planning Phase)
   - ศึกษาเกี่ยวกับอุปสรรคที่เกิดขึ้นในการเดินทางบนท้องถนนโดยการขับขี่ยานพาหนะ
   - ศึกษาความเป็นไปได้ในการนำระบบนำทางอัตโนมัติที่มีการนำคอมพิวเตอร์เข้ามาเกี่ยวข้อง ทั้งด้านฮาร์ดแวร์ และซอฟต์แวร์ เพื่อช่วยในการนำทางผู้ขับขี่ยานพาหนะบนท้องถนน
   - กำหนดแผนและตารางเวลาในการพัฒนาระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.1 ตารางเวลาในการพัฒนาระบบ

No.	Activity	Duration (Days)	Start Date	End Date
1	Planning	2	2007-11-15	2007-11-16
2	Analysis	5	2007-11-17	2007-11-21
3	Design	15	2007-11-22	2007-12-06
4	Development & Implementation	50	2007-12-07	2008-01-25
5	System Testing	7	2008-01-26	2008-02-01
6	Maintenance	7	2008-02-02	2008-02-08
7	Documentation	7	2008-02-09	2008-02-15
		93		

2. การวิเคราะห์ระบบ (Analysis Phase)
   - ค้นหาข้อมูลของระบบนำทางอัตโนมัติที่มีอยู่เดิม โดยจะหาข้อมูลจากเว็บไซต์ต่างๆ รวมถึง ผลิตภัณฑ์ในลักษณะนี้ที่ได้มีการพัฒนาและมีการใช้กันอยู่แล้ว
   - กำหนดขอบเขตของระบบที่จะทำโดยอาศัยข้อมูลที่ได้ทำการศึกษาได้มา
3. การออกแบบระบบ (Design Phase)
   - ทำการวิเคราะห์และออกแบบระบบงาน
   - ทำการออกแบบฐานข้อมูล
   - ทำการออกแบบโครงสร้างของโปรแกรมและฟังก์ชันการทำงานโดยจะรวมทั้งส่วนอินพุต เอาท์พุต และส่วนต่อประสานให้อยู่ภายในขอบเขตที่ตั้งไว้
4. การพัฒนาและติดตั้งระบบ (Development and Implementation Phase)
   - พัฒนาระบบและโปรแกรมตามที่ได้วิเคราะห์และออกแบบไว้
   - ทดสอบระบบในส่วนของฟังก์ชันการทำงานย่อยๆ
   - ติดตั้งระบบที่ได้พัฒนาลงบน Mobile Device
5. การทดสอบระบบ (Testing Phase)

   ทำการทดสอบระบบโดยรวมว่าสามารถใช้งานได้ตามที่วิเคราะห์วิเคราะห์และออกแบบไว้หรือไม่
6. การบำรุงรักษาระบบ (Maintenance Phase)

   เป็นขั้นตอนของการติดตามและประเมินผลการทำงานของระบบ ว่ามีจุดบกพร่องในส่วนใดบ้าง เพื่อทำการแก้ไข และปรับปรุงเพิ่มเติม
7. การทำเอกสารประกอบระบบ (Documentation)

   จัดทำเอกสารการพัฒนาระบบและคู่มือประกอบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

1.5 เครื่องมือที่ใช้ในการพัฒนาระบบ

a) เครื่องมือที่ใช้ในการพัฒนาโปรแกรม คือ Microsoft Visual Studio 2005 และภาษาที่ใช้คือ C# .NET

b) .NET Compact Framework 2.0

c) เครื่องคอมพิวเตอร์ที่นำมาใช้ในการพัฒนา คือ Laptop AMD Turion CPU 2.2 GHz, RAM 1 GB และระบบปฏิบัติการ Microsoft Windows XP Professional 2002

d) คอมพิวเตอร์แบบพกพาที่ใช้ในการทดสอบคือ Pocket PC Dopod 818 PRO ระบบปฏิบัติการ Microsoft Windows Mobile เวอร์ชัน 5.0 หน่วยประมวลผล (CPU) เป็น TI OMAP 850, 195 MHz และหน่วยความจำ 128MB Flash ROM - SDRAM 64MB การ์ดหน่วยความจำ MMC/SDIO memory slot

1.6 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากโครงการพัฒนาระบบงานนี้ มีดังนี้

1. ผู้ขับขี่รถบนถนนสามารถเดินทางไปยังสถานที่ที่ต้องการได้รวดเร็วขึ้น
2. ผู้ขับขี่สามารถหาเส้นทางได้ง่ายและสะดวกโดยไม่จำเป็นต้องใช้แผนที่กระดาษ ซึ่งยุ่งยากในการหาเส้นทางและอาจทำไปสู่การเกิดอุบัติเหตุได้ถ้าผู้ขับขี่ค้นหาแผนที่ในขณะขับรถ
3. ลดปัญหาการจราจรติดขัดได้โดยอาจเกิดจากการที่ผู้ขับขี่มีการเลือกเดินทางไปบนถนนสายรองที่มีการจรจรไม่หนาแน่น ทำให้ลดปัญหาการจราจรติดขัดบนถนนสายหลักได้
4. ช่วยให้ผู้ใช้งานสามารถค้นหาสถานที่สำคัญในบริเวณที่ใกล้ที่สุดได้ทันทีในเวลาที่จำเป็น เช่น ค้นหาโรงพยาบาลสัตว์ที่ใกล้ที่สุดเมื่อสัตว์เลี้ยงเกิดอุบัติเหตุ ค้นหาร้านอาหารที่ใกล้ที่สุด เป็นต้น
5. ช่วยให้ธุรกิจในพื้นที่ได้รับประโยชน์ เป็นการกระจายรายได้ โดยผู้ใช้เลือกใช้บริการในพื้นที่ที่ตนอยู่ใกล้ที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ระบบนำทางอัตโนมัติ

2.1 ระบบระบุตำแหน่งบนพื้นโลก

ระบบระบุตำแหน่งบนพื้นโลก GPS หรือ Global Positioning System เป็นระบบที่ช่วยระบุตำแหน่งหรือพิกัดทางภูมิศาสตร์บนพื้นโลกโดยดาวเทียมที่โคจรอยู่รอบโลกในตำแหน่งต่างๆ แต่เดิมระบบ GPS จะเปิดใช้สำหรับในด้านการทหารเท่านั้นแต่ต่อมาได้มีการเปิดให้กับคนทั่วไปใช้งานได้ จึงมีการประยุกต์ใช้ประโยชน์จาก GPS อย่างกว้างขวางในด้านต่างๆ ดังนี้

- ด้านการทหาร เช่น ติดตั้งลงไปในจรวดนำวิถี ใช้ในการตรวจจับตำแหน่งของนิวเคลียร์
- ด้านการนำทาง ใช้ประกอบกับแผนที่เพื่อนำทางยานพาหนะในทางบก ทางเรือ และทางอากาศ
- ด้านการสำรวจและการจัดทำแผนที่ ใช้เผื่อวางแผนในการสร้างถนน และการระบุตำแหน่งของสถานที่สำคัญต่างๆบนแผนที่
- นอกจากนี้ยังใช้เพื่อระบุเวลาที่ถูกต้อง ใช้ในธุรกิจโทรศัพท์เคลื่อนที่เพื่อวางแผนในการติดตั้งสถานีฐาน และใช้ในการระบุตำแหน่งของบุคคลหรือยานพาหนะเมื่อเกิดเหตุฉุกเฉิน เป็นต้น

ระบบ GPS จะทำงานได้ต้องประกอบด้วยส่วนสำคัญ 3 ส่วนดังนี้

1. ดาวเทียมและสถานีควบคุม
2. ตัวรับสัญญาณ หรือ GPS Receiver
3. โปรแกรมประยุกต์และแผนที่

โดยระบบ GPS ที่ใช้กันแพร่หลายในปัจจุบันใช้ระบบดาวเทียมที่ชื่อว่า NAVSTAR (Navigation Satellite Timing and Ranging GPS) ดูแลและบริหารงานโดยกระทรวงกลาโหมสหรัฐ (USDoD - United State Department of Defense) ซึ่งมีอยู่ทั้งหมด 28 ดวงและสงวนไว้ใช้ในด้านการทหารเท่านั้น 4 ดวง คนทั่วไปจึงใช้งานจริงได้ 24 ดวง ดาวเทียมทั้งหมดนี้มีสถานีควบคุมใหญ่อยู่ที่ฐานทัพอากาศ Schriever หรือ Falcon Air Force Base ที่เมืองโคโรลาโด สปริง พร้อมทั้งมีสถานีควบคุมย่อยอีก 5 แห่งกระจายอยู่ทั่วโลก

ดาวเทียมทั้ง 24 ดวงจะส่งสัญญาณคลื่นความถี่ไมโครเวฟซึ่งเป็นคลื่นความถี่สูงและสามารถทะลุผ่านชั้นบรรยากาศได้ลงมาที่พื้นโลกโดยคลื่นความถี่ที่ถูกส่งออกมาแบ่งเป็น 2 คลื่น คือ L1 (Primary Signal) และ L2 (Secondary Signal) โดย L1 ถูกเข้ารหัสแบบ C/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Coarse/Acquisition Code) รหัสนี้เปิดใช้โดยเสรี แต่ L2 ถูกเข้ารหัสแบบ P-Code ซึ่งถูกสงวนไว้สำหรับการทหารสหรัฐเท่านั้น ดังนั้นเครื่องรับ GPS ต้องสามารถรับคลื่นและแปลรหัส L1 ออกมาให้ได้ก็สามารถบอกพิกัดของตนเองได้ โดยข้อมูลที่ส่งมานั้นประกอบด้วย 2 ส่วนที่สำคัญ คือ

- ข้อมูลอัลมาแนค (Almanac Data) เป็นข้อมูลที่บอกตำแหน่งแบบหยาบๆของดาวเทียมแต่ละดวงทำให้มีประโยชน์ต่อเครื่องรับสัญญาณสามารถเลือกรับสัญญาณจากดาวเทียมดวงที่อยู่ใกล้ที่สุดได้
- ข้อมูลอีเฟเมอรีส (Ephemeris Data) เป็นข้อมูลล่าสุดที่ถูกส่งออกมาจากดาวเทียมแต่ละดวงเป็นข้อมูลที่สำคัญที่ใช้ในการคำนวณหาพิกัดของเครื่องรับ GPS ประกอบด้วย ข้อมูลวงโคจร สถานภาพของดาวเทียม วันเวลาที่เที่ยงตรง

เมื่อเครื่องรับสัญญาณ GPS สามารถถอดรหัสและคำนวณหาพิกัดมาได้แล้ว ก็ต้องมีโปรแกรมประยุกต์ที่นำเอาข้อมูลนี้ไปใช้ร่วมกับแผนที่ทำให้สามารถบอกพิกัดทางภูมิศาสตร์บนแผนที่ได้เกิดประโยชน์หลากหลาย โดยวิทยาการที่ว่านี้มักจะถูกเรียกว่า ระบบภูมิศาสตร์สารสนเทศ (GIS – Geographic Information Systems) ซึ่งเป็นระบบที่นำเอาข้อมูลความรู้ทางภูมิศาสตร์และข้อมูลจากดาวเทียมมาใช้ประโยชน์ร่วมกัน โดยโปรแกรมประยุกต์ที่นำเอาข้อมูลดังกล่าวไปใช้งาน เช่น โปรแกรมที่ใช้ในการนำทางบนท้องถนน มีการรับเอาพิกัดจากเครื่องรับ GPS แล้วนำมาระบุตำแหน่งบนแผนที่ทำให้เรารู้ว่าเราอยู่ ณ สถานที่ใดบนแผนที่ และจุดหมายปลายทางอยู่ที่ใด

แผนที่จะบอกตำแหน่งต่างๆบนพื้นโลกในรูปแบบสองมิติ โดยประกอบด้วยเส้นตรงแนวตั้งที่ลากผ่านขั้วโลกเหนือกับขั้วโลกใต้เรียกว่า เส้นรุ้ง หรือ เส้นละติจูด (Latitude) และเส้นแนวนอนเรียกว่าเส้นแวง หรือ เส้นลองจิจูด (Longitude) การบอกพิกัดคือ การบอกจุดที่เส้นรุ้งเส้นแวงมาตัดกันและบอกค่าเป็น Latitude และ Longitude ที่เท่าไร มีหน่วยเป็นองศา(Degree) ลิปดา(Minute) และ พิลิปดา(Second) โดย 60 พิลิปดาเท่ากับ 1 ลิปดา และ 60 ลิปดาเท่ากับ 1 องศา การอ้างพิกัดบนแผนที่จำเป็นต้องรู้ด้วยว่าแผนที่ได้ถูกจัดทำขึ้นโดยใช้การทำแผนที่รูปแบบใดหรือ อ้างอิงโดยใช้ Geodetic Datum รูปแบบใด ถ้าเครื่องรับ GPS ใช้การอ้างอิงพิกัดไม่ตรงกับรูปแบบของแผนที่ก็จำทำให้การบอกตำแหน่งไม่ถูกต้อง สำหรับแผนที่ที่ใช้กันในประเทศไทยใช้การอ้างอิงพิกัดแบบ India-Thailand

2.2 ระบบพิกัดในงานด้านแผนที่

แผนที่เป็นสื่อชนิดหนึ่งที่ทำหน้าที่ไม่ต่างจากสื่ออื่นๆคือเก็บและแสดงข้อมูลข่าวสารที่มีประโยชน์แก่ผู้ใช้ แผนที่ก็เหมือนสื่ออื่นๆที่ล้วนแต่มีรูปแบบในการเก็บและแสดงข้อมูลที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกลักษณ์เฉพาะของมัน เช่น ข้อความในเอกสารก็ต้องเป็นไปตามไวยากรณ์ของภาษาที่ใช้ แผนที่ก็เช่นกันต้องมีรูปแบบหรือไวยากรณ์ของมัน ไวยากรณ์ที่เป็นเอกลักษณ์อันหนึ่งของแผนที่ก็คือข้อมูลที่อยู่ในแผนที่นั้นเก็บตำแหน่งของสิ่งต่างๆบนโลกและแสดงข้อมูลนั้นตามตำแหน่งที่สามารถสื่อให้ผู้ใช้รู้ถึงตำแหน่งและความสัมพันธ์ทางตำแหน่งของสิ่งที่สนใจบนโลก ดังนั้นในทุกขั้นตอนของการทำแผนที่จึงล้วนเกี่ยวข้องกับการกำหนด,จัดการและแสดงตำแหน่งของสิ่งต่างๆที่อยู่บนโลก เริ่มตั้งแต่การสำรวจทั้งทางพื้นดินและทางอากาศ เพื่อจัดเก็บข้อมูลตำแหน่งของสิ่งต่างๆโดยอ้างอิงระบบพิกัดสามมิติ ไปจนถึงการแสดงข้อมูลตำแหน่งนั้นบนแผนที่ซึ่งเป็นระนาบสองมิติ ตำแหน่งของสิ่งต่างๆเหล่านั้นโดยทั่วไปแล้วแสดงในรูปของพิกัด ซึ่งเป็นตัวเลขหรือตัวอักษร หรือทั้งสองอย่างที่อ้างอิงอยู่กับจุดกำเนิดและแกนของระบบพิกัดเพื่อประโยชน์ในการกำหนดตำแหน่งของสิ่งที่ต้องการ ตำแหน่งที่เก็บและแสดงในแผนที่นั้นมีความเกี่ยวข้องและความสำคัญต่อมนุษย์เป็นอย่างมาก ทั้งในชีวิตประจำวัน การวางแผน และ การตัดสินใจในเรื่องต่างๆ ที่มีผลกระทบต่อมนุษย์และสิ่งต่างๆที่อยู่บนโลก ดังนั้นแผนที่จึงต้องแสดงตำแหน่งที่ถูกต้องตามความเป็นจริงของสิ่งที่ปรากฏบนโลก ด้วยเหตุนี้ผู้ที่ทำงานเกี่ยวกับแผนที่จึงจำเป็นต้องมีความรู้เรื่องระบบพิกัดที่ใช้ในงานด้านแผนที่เป็นอย่างดี เพื่อให้สามารถจัดการข้อมูลตำแหน่งได้อย่างถูกต้อง

2.2.1 การจำแนกระบบพิกัด

เนื่องจากการทำแผนที่มีขั้นตอนที่ค่อนข้างยุ่งยากและซับซ้อน แต่ละขั้นตอนอาจมีการใช้ระบบพิกัดที่ต่างกันออกไปตามเครื่องมือที่ใช้ในขั้นตอนนั้นๆ ระบบพิกัดที่ใช้ในงานแผนที่จึงมีความหลากหลาย หากจะจำแนกระบบพิกัดนั้นก็สามารถจำแนกได้หลายแบบ

(1) การจำแนกระบบพิกัดตามมิติ

ระบบพิกัดที่ใช้มากในงานแผนที่คือ ระบบพิกัดสองมิติ และ ระบบพิกัดสามมิติ ระบบพิกัดสองมิติใช้กำหนดตำแหน่งของจุดบนระนาบ เช่น การแสดงข้อมูลบนแผนที่ที่แสดงพิกัดหนึ่งในการกำหนดตำแหน่งในแนวตะวันออก-ตะวันตก และ อีกพิกัดหนึ่งในการกำหนดตำแหน่งในแนวเหนือ-ใต้ เช่น ตำแหน่งของหมู่บ้านหนึ่งอยู่ที่พิกัด (697700 ม.,1618600 ม.) ตัวอย่างของระบบพิกัดสองมิติได้แก่ ระบบพิกัด UTM (Universal Transverse Mercator) ที่ใช้ในแผนที่ภูมิประเทศของกรมแผนที่ทหาร ส่วนการกำหนดตำแหน่งบนโลกมักเป็นระบบพิกัดสามมิติ เช่น การกำหนดพิกัดโดยใช้ค่าลองจิจูด แลตติจูด และ ความสูงจากระดับทะเลปานกลาง ในการแสดงตำแหน่งของสิ่งต่างๆบนโลกที่อยู่ในระบบพิกัดสามมิติลงบนแผนที่ที่เป็นสองมิติ ทำผ่านกระบวนการฉายแผนที่ การวัดระยะบนแนวเส้นตรงนับได้ว่าเป็นระบบพิกัดหนึ่งมิติที่แสดงพิกัดในรูปของระยะที่วัดในแนวเส้นตรงนั้นจากจุดที่กำหนดเป็นจุดเริ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) การจำแนกระบบพิกัดตามลักษณะของแกนอ้างอิง

การจำแนกระบบพิกัดตามลักษณะของแกนอ้างอิงสามารถจำแนกระบบพิกัดออกได้เป็นระบบพิกัดฉาก (Rectangular หรือ Cartesian) และ ระบบพิกัดขั้ว (polar หรือ spherical) ระบบพิกัดฉากเป็นการกำหนดพิกัดโดยมีแกนที่ตั้งฉากซึ่งกันและกันเป็นแกนอ้างอิง การกำหนดพิกัดทำโดยระยะที่วัดเป็นแนวตั้งฉากกับแกนอ้างอิงแต่ละแกน ส่วนระบบพิกัดขั้วนั้นกำหนดตำแหน่งโดยใช้ระยะและมุม ระบบพิกัดขั้วเป็นการกำหนดจุดกำเนิดและแนวหรือระนาบอ้างอิง พิกัดจะแสดงในรูปของระยะที่วัดจากจุดกำเนิดที่กำหนดและมุมที่วัดจากแนวหรือระนาบอ้างอิง

(3) การจำแนกระบบพิกัดตามจุดกำเนิด

การจำแนกแบบนี้สามารถจำแนกระบบพิกัดออกเป็นระบบพิกัดที่มีจุดกำเนิดอยู่ที่จุดศูนย์กลางของโลกและระบบพิกัดที่มีจุดกำเนิดบนผิวโลก ตัวอย่างของระบบพิกัดที่มีจุดกำเนิดอยู่ที่จุดศูนย์กลางของโลกได้แก่ระบบที่ใช้กำหนดพิกัดบนโลกเช่นระบบพิกัดภูมิศาสตร์ ส่วนระบบพิกัดบนแผนที่โดยทั่วไปแล้วเป็นระบบพิกัดที่มีจุดกำเนิดบนผิวโลก

2.2.2 ระบบพิกัดที่พบมากในงานแผนที่

เราสามารถพูดถึงระบบพิกัดที่ใช้ในงานแผนที่เป็นกลุ่มๆได้โดยใช้การจำแนกระบบพิกัดแบบต่างๆตามที่อธิบายไปได้ดังนี้

(1) ระบบพิกัดแผนที่

- ระบบพิกัดฉากสองมิติที่มีจุดกำเนิดบนผิวโลก เช่น ระบบพิกัด UTM ที่ใช้ในแผนที่ภูมิประเทศของกรมแผนที่ทหาร ใช้แสดงตำแหน่งในทางราบของสิ่งต่างๆ ระบบพิกัด UTM มีจุดกำเนิดอยู่ที่จุดตัดระหว่างเส้นศูนย์สูตรกับเส้นเมริเดียนกึ่งกลางของโซน โดยกำหนดให้ที่จุดกำเนิดมีค่าพิกัดเป็น (E500000 m, N0 m) (E ย่อจาก Easting, N จาก Northing)
- ระบบพิกัดฉากสามมิติที่มีจุดกำเนิดบนผิวโลก บนแผนที่นั้นนอกจากจะแสดงตำแหน่งทางราบแล้วยังมีการแสดงข้อมูลความสูงของภูมิประเทศในรูปแบบต่างๆ เช่น เส้นชั้นความสูงและจุดความสูง ร่วมกับตำแหน่งทางราบ ข้อมูลความสูงนี้มักอ้างอิงจากระดับทะเลปานกลาง ในงานสำรวจมักมีการกำหนดพิกัดสำหรับพื้นที่งานขึ้นมาเพื่อความสะดวกในการคำนวณโดยใช้ระบบพิกัดแบบนี้

(2) ระบบพิกัดสามมิติที่มีจุดกำเนิดที่จุดศูนย์กลางของโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระบบพิกัดขั้วสามมิติที่มีจุดกำเนิดที่จุดศูนย์กลางของโลกที่ใช้ในงานแผนที่ได้แก่ระบบพิกัดภูมิศาสตร์ กำหนดพิกัดโดยใช้ค่าลองจิจูด แลตติจูด และ ความสูงจากพื้นผิวอ้างอิง ค่าลองจิจูดเป็นมุมระหว่างระนาบเมริเดียนหลักที่ผ่านกรีนิชกับระนาบเมริเดียนของจุดที่สนใจ แลตติจูดเป็นมุมระหว่างระนาบศูนย์สูตรกับเส้นในระนาบเมริเดียนที่ตั้งฉากกับพื้นผิวอ้างอิง ณ จุดที่สนใจ
- ระบบพิกัดฉากสามมิติที่มีจุดกำเนิดที่จุดศูนย์กลางของโลก เป็นระบบพิกัดที่มีสามแกน แกนแรกเกิดจากการตัดกันของระนาบศูนย์สูตรกับระนาบเมริเดียนหลัก แกนที่สองเกิดจากการตัดกันของระนาบศูนย์สูตรกับระนาบเมริเดียนที่มีค่าลองจิจูด 90 องศา และ แกนที่สามคือแกนหมุนของโลก ระบบพิกัดนี้มักใช้ในการคำนวณในขั้นตอนต่างๆโดยเฉพาะการแปลงค่าพิกัด และ การคำนวณที่เกี่ยวข้องกับตำแหน่งของดาวเทียม

(3) ระบบพิกัดขั้วที่มีจุดกำเนิดบนผิวโลก

- ระบบพิกัดขั้วสองมิติที่มีจุดกำเนิดบนผิวโลก เป็นระบบที่ใช้ในการเดินเรือหรือการนำร่องที่มีการกำหนดระยะและทิศทางในการเดินทาง โดยทิศทางมักเป็นมุมที่วัดตามเข็มนาฬิกาจากแนวของทิศเหนือ การสำรวจภาคพื้นดินเพื่อหาพิกัดทางราบโดยใช้กล้องสำรวจก็เป็นอีกงานหนึ่งที่ใช้ระบบพิกัดขั้วสองมิติ โดยข้อมูลที่จดในแบบสำรวจจะเป็นค่ามุมและระยะ
- ระบบพิกัดขั้วสามมิติที่มีจุดกำเนิดบนผิวโลก เป็นระบบพิกัดที่ใช้ในงานดาราศาสตร์ในการกำหนดตำแหน่งของดวงดาวจากจุดที่สนใจ โดยมีมุมเพิ่มอีกหนึ่งมุมคือมุมระหว่างระนาบที่สัมผัสกับผิวโลก ณ จุดที่สนใจกับเส้นที่ลากผ่านจุดนั้นไปยังดาว

ตารางต่อไปสรุประบบพิกัดที่ใช้ในงานแผนที่ โดยใช้การจำแนกและการแบ่งกลุ่มตามที่ได้อธิบายไป ระบบพิกัดแผนที่อยู่ในช่องที่มีพื้นสีเขียว ระบบพิกัดสามมิติที่มีจุดกำเนิดที่จุดศูนย์กลางของโลกอยู่ในช่องสีชมพู ส่วนระบบพิกัดขั้วที่มีจุดกำเนิดบนผิวโลกอยู่ในช่องสีฟ้า

ตารางที่ 2.1 ตารางสรุประบบพิกัดที่ใช้ในงานแผนที่

ระบบพิกัด	จุดกำเนิดอยู่ที่จุดศูนย์กลางของโลก		จุดกำเนิดบนผิวโลก	
	ระบบพิกัดฉาก	ระบบพิกัดขั้ว	ระบบพิกัดฉาก	ระบบพิกัดขั้ว
ระบบพิกัดสองมิติ	ไม่ค่อยใช้ในงานแผนที่		ระบบพิกัดแผนที่ (x,y หรือ E,N)	การเดินเรือ,การนำร่อง (s,Az)
ระบบพิกัดสามมิติ	ระบบพิกัดฉากสามมิติ (X,Y,Z)	ระบบพิกัดภูมิศาสตร์ (l,j,h)	ระบบพิกัดแผนที่+ความสูง (E,N,H หรือ enu)	งานดาราศาสตร์ (s,Az,El)

เอกสารนี้มีลิขสิทธิ์ ไม่อนุญาตให้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น

ระบบพิกัดในภาพดิจิตอลที่ใช้แถวและคอลัมน์นั้นถือได้ว่าเป็นระบบพิกัดฉากสองมิติแบบหนึ่ง ระบบพิกัดที่ได้กล่าวไปอาจไม่ครอบคลุมระบบพิกัดทั้งหมดที่ใช้ในงานแผนที่ แต่ก็คงเพียงพอที่จะทำให้เห็นถึงความหลากหลายของระบบพิกัดที่มักพบทั่วๆ ไป

2.3 ไดสตราอัลกอริทึม

ไดสตราอัลกอริทึม(Dijkstra Algorithm) เป็นอัลกอริทึมที่ใช้ในการหาระยะทางสั้นที่สุด (Shortest Path) ระหว่างโหนด (node) หรือ vertex บนกราฟ โดยมีจุดเริ่มต้นเพียงจุดเดียวและไม่มีค่าระยะทางหรือ weight ระหว่างโหนดเป็นลบ

2.3.1 ขั้นตอนการทำงานของอัลกอริทึม

ไดสตราอัลกอริทึมมีขั้นตอนการทำงานดังนี้

1. เริ่มต้นให้ทุกโหนดมีค่า d(v) = infinity และค่า previous(v) = undefined

โดยที่ d(v) คือ ค่าระยะทางทั้งหมดที่เดินทางจากจุดเริ่มต้นมาที่โหนด v

และ previous(v) คือ โหนดสุดท้ายที่ต้องผ่านเมื่อเดินทางจากจุดเริ่มต้นมาโหนด v

2. ให้ s เป็นโหนดเริ่มต้นและ d(s) = 0 , S = {s} (S เป็นเซตของโหนดที่ผ่านการพิจารณาแล้ว)
3. เลือกโหนดที่อยู่ติดกับ s ที่มีค่า weigh(s,v) น้อยที่สุดมาพิจารณา

โดยที่ weigh(s,v) คือ ระยะทางจากโหนด s ไปโหนด v

จะได้ว่า S = S U v , ให้โหนด v1 เป็นโหนดที่มีค่า weigh(s,v1) น้อยที่สุด

ดังนั้น S = {s,v1}และถ้าค่าของ d(s) + w(s,v1) < d(v1) ให้ d(v1) = d(s) + w(s,v1) และ previous(v1) = s

ซึ่งโหนดแรกที่เลือกมาจะได้ว่าเงื่อนไข d(s) + w(s,v1) < d(v1) เป็นจริงเสมอเพราะ d(v1) ยังเป็น infinity อยู่

4. เลือกโหนดที่เป็นขอบของเซตของโหนด S ที่มีค่า d(s') + w(s',v2) น้อยที่สุดมาพิจารณา

โดยที่ s' เป็นสมาชิกของ S และ v2 เป็นโหนดที่อยู่ติดกับ s' (Adjacent Node)

จะได้ว่า S = S U v2 และถ้าค่าของ d(s') + w(s',v2) < d(v2) ให้ d(v2) = d(s') + w(s',v2) และ previous(v2) = s'

5. ทำซ้ำข้อ 4 จนกระทั่งใน S มีครบทุกโหนด จะได้ว่า d(v) ของโหนดใดจะหมายถึงระยะทางทั้งหมดที่เดินทางจาก s ไปที่ v และ previous(v) หมายถึง โหนดสุดท้ายที่ต้องผ่านเมื่อการเดินทางจาก s ไป v
6. เริ่มต้นให้ทุกโหนดมีค่า

และถ้าสร้างเส้นทางย้อนกลับไปจากโหนด v โดย ใช้ previous node ไปเรื่อยจนกระทั่งถึงโหนด s จะได้เส้นทางระยะสั้นที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากไดสตราอัลกอริทึมข้างต้นพอจะสรุปขั้นตอนการทำงานได้ดังนี้

Step 0: เลือกโหนดเริ่มต้น

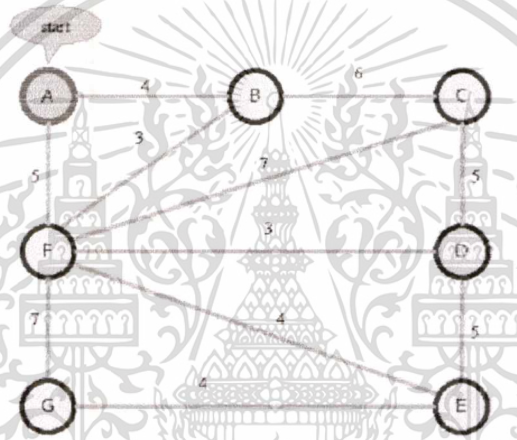
Step 1: คำนวณ cost ที่ใช้ในการเคลื่อนที่จากโหนดเริ่มต้นไปยังโหนดที่อยู่ติดกัน เก็บ cost ที่น้อยที่สุดที่ได้เดินทางถึงโหนดแต่โหนดไว้ และเก็บโหนดก่อนหน้าด้วย สุดท้ายคือเลือกโหนดที่มี cost น้อยที่สุดรวมเป็นโหนดเริ่มต้น

Step 2: ทำซ้ำ step 1 จนกระทั่งทุกโหนดถูกรวมไว้ด้วยกันเป็นโหนดเริ่มต้น

สุดท้ายจะได้เส้นทางสั้นที่สุดจากจุดเริ่มต้นที่เลือกไว้ในตอนแรกไปยังทุกโหนด

2.3.2 ตัวอย่างการหาระยะทางสั้นที่สุดโดยใช้ไดสตราอัลกอริทึม

การใช้อัลกอริทึมไดสตราในการหาเส้นทางสั้นสุดจากกราฟตัวอย่างดังต่อไปนี้



รูปที่ 2.1 กราฟตัวอย่างมี A เป็นโหนดเริ่มต้น

ตารางที่ 2.2 ตารางแสดงเส้นทางสั้นที่สุดจาก A ไปยังโหนดอื่นๆ

โหนด	d(v)	pred(v)
A	0	0
B	Infinity	0
C	Infinity	0
D	Infinity	0
E	Infinity	0
F	Infinity	0
G	Infinity	0

ขั้นตอนการทำงานมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เลือกโหนดเริ่มต้นในที่นี้เลือกจุด A และให้ทุกโหนดจะมีค่า d(v) = Infinity และ pred(v) = 0 ดังแสดงในรูปที่ 2.1 และ ตารางที่ 2.2
2. รอบที่ 1 คำนวณ cost ที่ใช้ในการเดินทางไปยังโหนดที่อยู่ติดกับ A จากรูปคือโหนด B และ F มี cost เป็น 4 กับ 5 ตามลำดับ และเลือกโหนดที่มี cost น้อยที่สุด นั้นคือโหนด B และเอาโหนด B มารวมเป็นโหนดเริ่มต้นและปรับปรุงตารางดังต่อไปนี้

ตารางที่ 2.3 แสดงเส้นทางสั้นที่สุดจากAไปยังโหนดอื่นๆเมื่อผ่านอัลกอริทึมไดสตรารอบที่ 1

โหนด	d(v)	pred(v)
A	0	0
B	4	A
C	Infinity	0
D	Infinity	0
E	Infinity	0
F	Infinity	0
G	Infinity	0

3. เริ่มรอบที่ 2 โดยคำนวณ cost ที่ใช้ในการเดินทางไปยังโหนดที่อยู่ติดกับ A,B จากรูปคือโหนด C และ F แต่มี 3 เส้นทางคือ BC มี cost = 4 + 6, BF มี cost = 4 + 3 และ AF มี cost = 0 + 5 จะเห็นว่า BC กับ BF ที่มี B เป็นโหนดเริ่มต้นต้องบวก 4 เข้าไปด้วยซึ่งเป็น cost ที่เดินทางผ่านมาแล้ว แต่ AF ไม่มี cost ที่เดินทางผ่านมาแล้วเพราะ A เป็นจุดเริ่มต้น และเลือกโหนดที่มี cost น้อยที่สุด นั้นคือโหนด F และเอาโหนด F มารวมเป็นโหนดเริ่มต้นและปรับปรุงตารางดังต่อไปนี้

ตารางที่ 2.4 แสดงเส้นทางสั้นที่สุดจากAไปยังโหนดอื่นๆเมื่อผ่านอัลกอริทึมไดสตรารอบที่ 2

โหนด	d(v)	pred(v)
A	0	0
B	4	A
C	Infinity	0
D	Infinity	0
E	Infinity	0
F	5	A
G	Infinity	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. และเมื่อครบ 6 รอบจะได้เส้นทางสั้นที่สุดไปยังทุกโหนดในกราฟดังต่อไปนี้

ตารางที่ 2.5 แสดงเส้นทางสั้นที่สุดจากAไปยังโหนดอื่นๆเมื่อผ่านอัลกอริทึมไดสตรารอบที่ 6

โหนด	d(v)	pred(v)
A	0	0
B	4	A
C	10	B
D	8	F
E	9	F
F	5	A
G	12	F

2.4 วงจรการพัฒนาระบบ (Shelly *et.al.* 2001)

วงจรการพัฒนาระบบ (System Development Life Cycle - SDLC) เป็นวงจรที่แสดงถึงกิจกรรมต่างๆ ในแต่ละขั้นตอน ตั้งแต่ริเริ่มจนกระทั่งสำเร็จ วงจรการพัฒนาระบบนี้จะทำให้เข้าใจถึงกินกรรมพื้นฐานและรายละเอียดต่างๆ ในการพัฒนาระบบ โดยมีอยู่ 7 ขั้นตอนด้วยกัน คือ

1. กำหนดปัญหา (Problem Definition) เป็นขั้นตอนของการกำหนดขอบเขตของปัญหา สาเหตุของปัญหาจากการดำเนินงานในปัจจุบัน ความเป็นไปได้กับการสร้างระบบใหม่ การกำหนดความต้องการของผู้ใช้งาน ทั้งนี้เป็นการทำเพื่อให้ได้ข้อกำหนดของระบบที่ชัดเจนขึ้นมา

2. วิเคราะห์ (Analysis) เป็นการวิเคราะห์ระบบการทำงานเดิม โดยอาศัยข้อกำหนดของระบบ ที่ได้จากขั้นตอนที่ 1 เพื่อสำหรับสร้างเป็นแบบจำลองเชิงแนวคิดขึ้นมา อันได้แก่ แผนภาพกระแสข้อมูล (Data Flow Diagram) คำอธิบายการประมวลผลข้อมูล (Process Description) และแบบจำลองข้อมูล (Data Model) ทำให้ทราบถึงรายละเอียดขั้นตอนการดำเนินงานในระบบว่าประกอบด้วยอะไรบ้าง มีความเกี่ยวข้องและสัมพันธ์กับสิ่งใด

3. ออกแบบ (Design) เป็นขั้นตอนของการนำผลลัพธ์ที่ได้จากการวิเคราะห์ในขั้นตอนที่ 2 มาพัฒนาเป็นแบบจำลองเชิงกายภาพให้สอดคล้องกัน โดยการออกแบบจะเริ่มจากส่วนของอุปกรณ์และเทคโนโลยีต่างๆ และโปรแกรมคอมพิวเตอร์ที่นำมาพัฒนา การออกแบบจำลองข้อมูล การออกแบบรายงาน และการออกแบบจอภาพในการติดต่อกับผู้ใช้งาน การจัดทำพจนานุกรมข้อมูล รวมไปถึงการต้นแบบของระบบด้วย

4. การพัฒนา (Development) เป็นขั้นตอนของการพัฒนาโปรแกรมจากที่ได้ทำการวิเคราะห์และออกแบบไว้ โดยต้องพิจารณาถึงความเหมาะสมกับเทคโนโลยีที่ใช้งานในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทดสอบ (Testing) เป็นขั้นตอนของการทดสอบระบบก่อนที่จะนำระบบที่ได้ไปใช้งานจริง ประกอบไปด้วยการตรวจสอบ 2 ส่วน ได้แก่ การตรวจสอบไวยากรณ์ของโปรแกรม และการตรวจสอบว่าระบบงานที่พัฒนาขึ้นมานั้นเป็นไปตามข้อกำหนดที่ได้ตั้งไว้หรือไม่

6. ติดตั้ง (Implement) เป็นขั้นตอนหลังจากที่ได้ทำการทดสอบระบบ ว่าสามารถทำงานได้จริงและตรงกับความต้องการของผู้ใช้ระบบ แล้วจึงดำเนินการติดตั้งระบบเพื่อใช้งานจริงต่อไป รวมไปถึงการจัดทำคู่มือใช้งานระบบอีกด้วย

7. บำรุงรักษา (Maintenance) เป็นขั้นตอนของการปรับปรุงแก้ไขระบบหลังจากที่ได้มีการติดตั้งและใช้งานแล้ว

2.5 Microsoft .NET Framework (Anderson and Francis. 2002 ; Shelly et.al. 2001)

.NET Framework คือ โครงสร้างของการพัฒนาโปรแกรมคอมพิวเตอร์ ที่ได้ออกแบบมาเพื่อให้ความสะดวกในการพัฒนาโปรแกรมสมัยใหม่ โดยสนับสนุนงานในระบบเครือข่าย อาทิ เช่น อินเทอร์เน็ต อินทราเน็ต และอุปกรณ์เคลื่อนที่ เป็นต้น มากขึ้น โดยมีแนวทางการพัฒนา 2 แนวทาง ได้แก่

1. การพัฒนาโปรแกรมในรูปแบบของเว็บเซอร์วิส จะเป็นหัวใจหลักในการพัฒนา และเรียกใช้งานโปรแกรมต่างๆ ในระบบอินเทอร์เน็ต เว็บเซอร์วิสจะช่วยให้การติดต่อสื่อสารระหว่างแอพลิเคชัน บนอินเทอร์เนตนั้นง่าย มีอิสระ และเป็นระบบมากยิ่งขึ้น

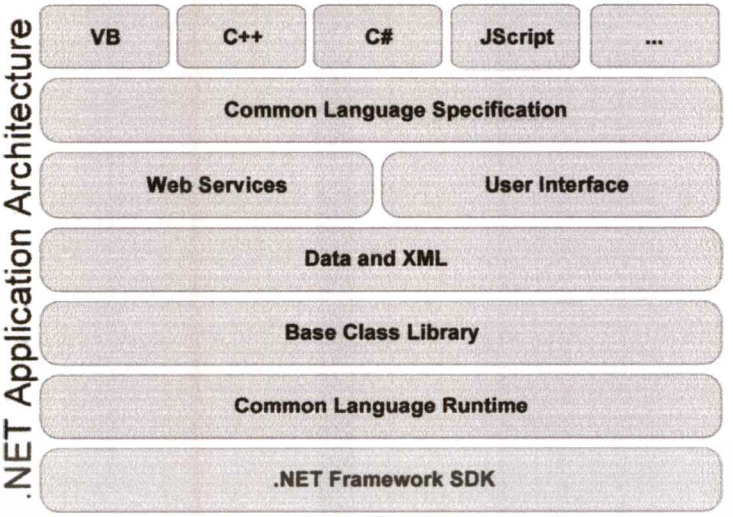
2. การพัฒนาโปรแกรมในระบบพีซี (คอมพิวเตอร์ตั้งโต๊ะ และโน้ตบุ๊ก) และอุปกรณ์เคลื่อนที่ที่สามารถเชื่อมต่อกับระบบอินเทอร์เน็ตได้ เช่น PDA และโทรศัพท์มือถือ เพราะจะมีบทบาท และประโยชน์มากขึ้น เมื่อสามารถติดต่อใช้งานโปรแกรมต่างๆ บนอินเทอร์เน็ตได้

Microsoft .NET Framework สามารถแบ่งส่วนประกอบต่างๆ ออกเป็นเลเยอร์ ได้ดังรูปที่ 2.2 (Jeffery Richter. 2000.)

Common Language Runtime Layer (CLR)

CLR ได้ถูกพัฒนาขึ้นมาเพื่อกำจัดความยุ่งยาก และข้อจำกัดต่างๆ ในการทำงานร่วมกันระหว่างโปรแกรมกับโปรแกรม หรือโปรแกรมกับเซิร์ฟเวอร์โดย CLR จะเป็นส่วนพื้นฐานที่คอยติดต่อกับ OS ทำหน้าที่เป็น Runtime Environment ให้กับโปรแกรมที่เขียนขึ้นสำหรับใช้ใน .NET CLR จะแบ่งส่วนของคอมไพเลอร์เป็น 2 ส่วน คือ ส่วนที่ใช้แปลรหัสต้นทาง (Source Code) ของโปรแกรมให้อยู่ในรูปของ Intermediate Language (IL) และอีกส่วนหนึ่งก็คือ การนำโค้ดที่อยู่ในรูปของ IL มาแปลอีกครั้งหนึ่งให้เป็น Native Code เพื่อใช้งานในทุกๆ ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 โครงสร้างของ Microsoft .NET Framework

Base Class Library

Base Class Library เป็น Class Library พื้นฐาน ที่โปรแกรมต่างๆ ที่ถูกพัฒนาขึ้นภายใต้ .NET Framework สามารถใช้งานร่วมกันได้ อาทิเช่น การติดต่อกับระบบฐานข้อมูล การติดต่อกับระบบ OS และอื่นๆ โดย Base Class Library นั้น เวลานำมาใช้งานจะถูกอ้างอิงในลักษณะของ Hierarchical Object Model ก็คือ ต้องระบุคลาสแม่ก่อน แล้วตามด้วยคลาสลูกที่ต้องการใช้งาน ไม่สามารถระบุคลาสแม่คลาสเดียวแล้ว จะใช้งานคลาสลูกได้หมด

ข้อมูลและ XML

ใน .NET Framework นั้นจะใช้ ADO.NET เป็นสื่อกลางระหว่างฐานข้อมูลกับโปรแกรม โดยมี XML เป็นมาตรฐานของข้อมูลที่ใช้สื่อสารกันนั่นเอง โดย ADO.NET นั้นจะมีโครงสร้างการติดต่อฐานข้อมูลแบบ Disconnected ก็คือ มีการปิดการเชื่อมต่อทุกครั้งหลังจากที่ใช้งานฐานข้อมูลเสร็จสิ้น โดยจะมี Dataset เป็นตัวที่รองรับข้อมูลที่ได้มาเก็บไว้ และสามารถนำไปใช้ได้เลย เปรียบได้เป็นเหมือน Recordset ของ ADO เวอร์ชันก่อน

ส่วนต่อประสานผู้ใช้

ส่วนต่อประสานผู้ใช้หรือ User Interface ภายใต้ .NET Framework นี้นั้นจะแบ่งออกเป็น 2 ชนิดได้แก่ Web Forms หรือ รูปแบบการแสดงผลบนเว็บไซต์ โดยเว็บแอพลิเคชันที่พัฒนาขึ้นนี้เราสามารถออกแบบและแก้ไขหน้าตาของเว็บเพจนั้นได้ง่ายและสะดวกขึ้น โดยที่เราไม่ต้องกังวลเรื่องโค้ดต่างๆ ในเว็บเพจอีกต่อไป เพราะ .NET Framework นั้น สามารถแยกส่วนที่เป็นโค้ดการทำงานต่างๆ ออกจากส่วนที่เป็น Layout ของเว็บเพจ ทำให้นักออกแบบเว็บสามารถออกแบบเว็บเพจได้ โดยไม่ต้องกังวลเกี่ยวกับโค้ดต่างๆ อีกต่อไป และคุณสมบัติที่สำคัญที่ถูกเพิ่มเข้ามาอีกอย่างหนึ่งซึ่ง ก็คือ มันสามารถจัดการการแสดงเว็บแอพลิเคชันให้ถูกต้องตามลักษณะของบราวเซอร์ที่ใช้ได้อย่างถูกต้องโดยอัตโนมัติ และอีกชนิดหนึ่งก็คือ Window Form หรือ รูปแบบการแสดงผลบนวินโดว์ก็ได้มีการพัฒนาเกี่ยวกับอินเทอร์เฟซต่างๆ ให้มีประสิทธิภาพมากขึ้น และ

เอกสารนี้เผยแพร่เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก้ไขจุดบกพร่องที่มีในรุ่นก่อนให้หมดไป อีกทั้งยังสนับสนุนการทำงานร่วมกับเว็บเซอร์วิสอีกด้วย

ภาษาที่ใช้ในการพัฒนา .NET Framework

ในสองเลเยอร์บนสุดจะเป็นส่วนของภาษาที่ใช้พัฒนาโปรแกรมบน .NET Framework และคุณสมบัติต่างๆ ของภาษาต่างๆ ที่มีร่วมกันภายใต้ .NET Framework นี้ โดยทางไมโครซอฟต์จะเห็นหนักไปที่ 3 – 4 ภาษาหลักๆ ได้แก่ VB.NET ที่พัฒนาต่อมาจาก VB แต่ถูกปรับปรุงให้เป็นไปตามแนวคิดของภาษาเชิงวัตถุ C# เป็นภาษาที่ถูกพัฒนาขึ้นมาใหม่ใน .NET Framework ซึ่งมีไวยากรณ์ต่างๆ ใกล้เคียงกับภาษา C++ หรือ Java แต่มีความง่ายในการพัฒนาโปรแกรมเหมือนกับ VB รวมไปถึง Visual C++ และ Javascript.NET นอกจากนี้ยังมีภาษาอื่นๆ ที่ไม่ใช่ของทางไมโครซอฟต์โดยตรง แต่เป็นของบริษัทอื่นๆ (Third Party) ที่มีการสร้างคอมไพเลอร์ขึ้นมาเอง ตามหลักการของ .NET Framework อาทิ เช่น Perl.NET และ Python.NET เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 กำหนดความต้องการของระบบ

ระบบนำทางอัตโนมัติเป็นระบบที่ทำงานบนเครื่องคอมพิวเตอร์แบบพกพาซึ่งติดต่อกับ GPS เพื่อใช้ในการนำทาง โดยมีการรับข้อมูลการจราจร ณ เวลาปัจจุบันเพื่อช่วยในการหาเส้นทางที่เหมาะสมด้วย โดยมีความต้องการของระบบดังนี้

1. สามารถบอกเส้นทางที่มีระยะทางสั้นที่สุดระหว่างจุดเริ่มต้นกับจุดที่ต้องการใดๆบนแผนที่ได้
2. สามารถบอกเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดระหว่างจุดเริ่มต้นกับจุดที่ต้องการใดๆ บนแผนที่ได้ โดยใช้ข้อมูลการจราจร ณ เวลาปัจจุบัน เพื่อหลีกเลี่ยงเส้นทางที่มีการจราจรแออัด หรือ เส้นทางที่ปิดไม่ให้รถผ่านได้
3. สามารถนำทางให้เดินทางไปตามเส้นทางที่กำหนดไว้ได้
4. สามารถบอกตำแหน่งปัจจุบันบนแผนที่ได้
5. สามารถคำนวณเส้นทางโดยใช้เวลาไม่เกิน 60 วินาที สำหรับ CPU OMAP850-195MHz บนเครื่องคอมพิวเตอร์แบบพกพา
6. สามารถค้นหาจุด Way Point หรือ สถานที่ที่ต้องการจะไปได้ เช่น ร้านอาหาร อู่ซ่อมรถ ธนาคาร เป็นต้น และสามารถให้ระบบนำทางไปยังสถานที่ที่ต้องการได้
7. เครื่องคอมพิวเตอร์แบบพกพาต้องสามารถเชื่อมต่ออินเตอร์เน็ทได้ เช่น ผ่าน GPRS หรือ Wireless LAN เพื่อรับข้อมูลการจราจร ณ เวลาปัจจุบัน มาใช้ในการหาเส้นทางที่ใช้เวลาเดินทางสั้นที่สุดได้
8. ข้อมูลการจราจรได้มาจาก Traffic Monitoring System ให้ถือว่าเป็นข้อมูลล่าสุดโดยระบบเพียงจำลองการติดต่อเพื่อให้ได้ข้อมูลมาเท่านั้น ระบบไม่ได้มีส่วนการทำงานที่เกี่ยวกับการแก้ไขหรือการได้มาซึ่งข้อมูลการจราจรบน Traffic Monitoring System

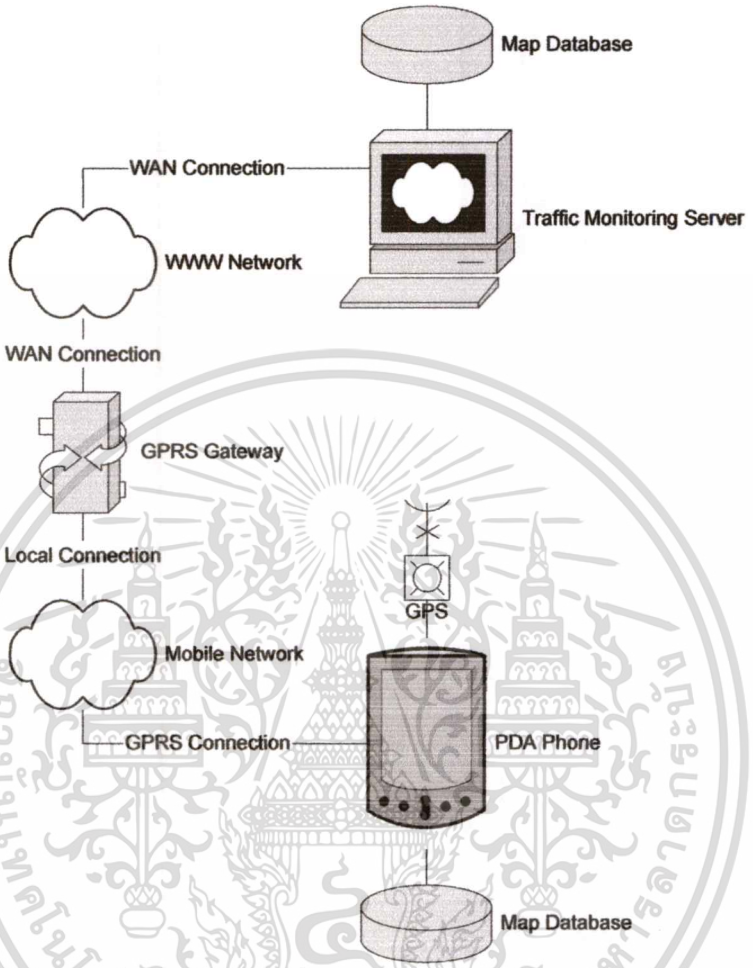
3.2 ลักษณะการทำงานของระบบ

ระบบนำทางอัตโนมัติจะมีลักษณะการทำงานคือ ผู้ใช้มีเครื่องคอมพิวเตอร์แบบพกพาที่สามารถเชื่อมต่อกับเครื่องรับสัญญาณจาก GPS ได้ และมีการเชื่อมต่ออินเตอร์เน็ทผ่านระบบ GPRS เพื่อรับข้อมูลแผนที่แบบ real-time จาก Server ได้ และนำมาเก็บไว้ในฐานข้อมูลแผนที่ซึ่งมีสถานที่ถนนรวมทั้งระยะทางและระยะเวลาในการเดินทางของแต่ละถนนไว้ด้วย และสุดท้ายมีโปรแกรมประยุกต์ทำหน้าที่คำนวณหาระยะทางสั้นที่สุดและเร็วที่สุด และรับพิกัดจากเครื่องรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

04895

GPS มาประมวลผลและระบุตำแหน่งบนแผนที่ เมื่อผู้ใช้ต้องการจะเดินทางไปยังจุดใดในแผนที่ก็สามารถนำทางไปยังเส้นทางที่ต้องการได้



รูปที่ 3.1 แสดงการเชื่อมต่อของระบบนำทางอัตโนมัติ

3.2.1 หลักการในการหาเส้นทางที่เหมาะสม

การหาเส้นทางที่เหมาะสมจะใช้เกณฑ์การตัดสินใจแบ่งเป็น 2 ส่วนดังนี้

1. การหาเส้นทางที่มีระยะทางสั้นที่สุด
2. การหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดโดยใช้สภาพการจราจร ณ เวลาปัจจุบันมาใช้ การคำนวณหาระยะทางสั้นที่สุดจะใช้ไดสตราอัลกอริทึมโดยให้โหนดแทนด้วยทางแยก และสถานที่ที่สนใจ ส่วนถนนจะแทนด้วยเส้นตรงจากทางแยกหนึ่งไปอีกทางแยกหนึ่ง ส่วนน้ำหนัก(Weight) แทนด้วยระยะทางจากทางแยกหนึ่งไปอีกทางแยกหนึ่ง หรือ แทนด้วนเวลาที่ใช้ในการเดินทางของแต่ละถนน

จากนั้นทำการเก็บข้อมลเหล่านี้ลงบนฐานข้อมูลซึ่งอ้างอิงไปยังแผนที่ และใช้ข้อมูลดังกล่าวในการหาเส้นทางที่สั้นที่สุด หรือ เร็วที่สุดโดยใช้อัลกอริทึม ไดสตรา เมื่อนำไปเทียบกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนที่จะได้เป็นเส้นทางจากจุดหรือทางแยกหนึ่งไปสู่อีกทางแยกหนึ่งต่อเนื่องกันไปเรื่อยๆ จนถึงปลายทาง

3.2.2 การนำทางอัตโนมัติ

เมื่อมีการหาเส้นทางที่เหมาะสมที่สุดโดยใช้ไดสตราอัลกอริทึมเรียบร้อยแล้ว ก็มาถึงการนำโดยระบบจะรับค่าพิกัดจากเครื่องรับ GPS มาเทียบกับเส้นทางบนแผนที่ และเมื่อถึงทางแยกก็จะแจ้งเตือนผู้ใช้ให้เลี้ยวขาวเลี้ยวซ้ายหรือตรงไป เมื่อผู้ใช้ออกนอกเส้นทางก็จะแจ้งเตือนให้เลี้ยวกลับเข้าเส้นทาง หรือถามผู้ใช้ว่าต้องการให้นำทางไปยังเส้นทางอื่นโดยเริ่มที่ตำแหน่งปัจจุบันหรือไม่

3.3 Use Case Diagram

Use Case เป็น diagram ที่บอกว่ามีใครเกี่ยวข้องกับระบบบ้างและใช้ระบบเพื่อวัตถุประสงค์อะไร

Use Case: Navigate to the target way point

Actors หลัก: User

เป้าหมาย: เพื่อให้ระบบนำทางไปยังปลายทาง

เงื่อนไขก่อน: ต้องใส่ปลายทางที่ต้องการก่อน

เงื่อนไขสิ้นสุดที่สำเร็จ: ถ้าตำแหน่งของ User อยู่ในตำแหน่งเดียวกับตำแหน่งปลายทาง

เงื่อนไขสิ้นสุดที่ล้มเหลว: ถ้าไม่ได้เลือกเส้นทางที่ต้องการการเดินทางไปยังปลายทาง

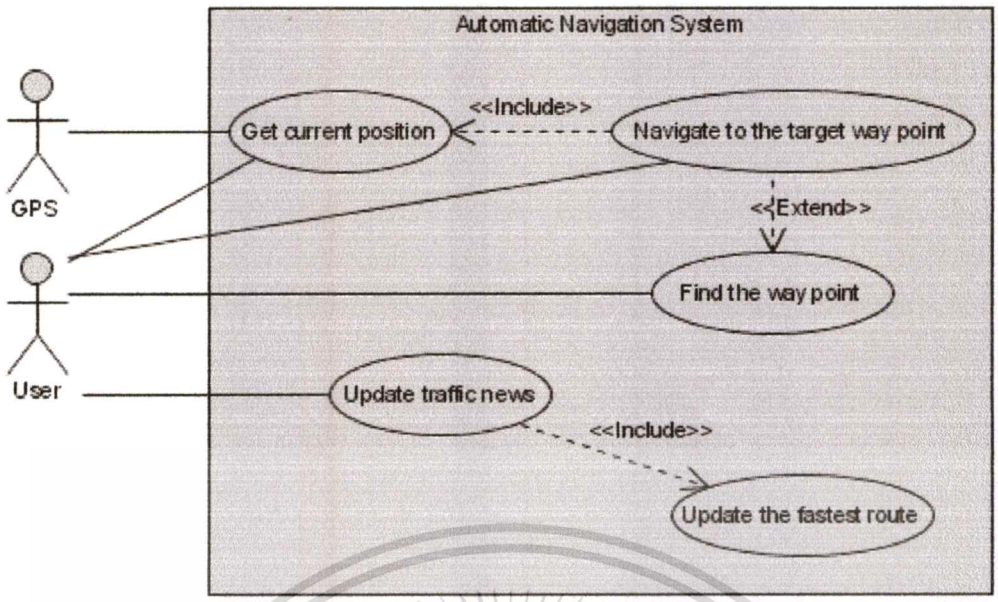
เหตุการณ์หลัก:

1. User ร้องขอเส้นทางสั้นที่สุดและเร็วที่สุดไปยังตำแหน่งปลายทาง
2. ระบบส่งเส้นทางที่สั้นที่สุดและเร็วที่สุดให้ User เลือก
3. User เลือกเส้นทางที่ต้องการ
4. ระบบรับตำแหน่งปัจจุบันของ User จาก GPS receiver
5. ระบบส่งข้อความเตือนให้ User เดินทางไปตามเส้นทาง
6. ระบบส่งข้อความบอก User เมื่อถึงจุดหมายแล้ว

ข้อเพิ่มเติม:

ถ้าไม่มีเส้นทางที่สามารถนำทางไปได้หรือผู้ใช้ไม่ได้เลือกเส้นทางที่ต้องการการเดินทางไปยังปลายทางจะมีข้อความเตือนขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 Use Case Diagram ของระบบนำทางอัตโนมัติ

Use Case: Get current position

Actors หลัก: User

เป้าหมาย: เพื่อระบุตำแหน่งปัจจุบันบนแผนที่

เงื่อนไขก่อน: GPS receiver อยู่ในสถานะพร้อมทำงานและสามารถบอกพิกัดได้

เงื่อนไขสิ้นสุดที่สำเร็จ: ถ้า GPS receiver อยู่ในสถานะพร้อมทำงานจะส่งพิกัดออกมาและระบบจะระบุตำแหน่งบนแผนที่ได้

เงื่อนไขสิ้นสุดที่ล้มเหลว: ถ้า GPS receiver ไม่อยู่ในสถานะพร้อมทำงาน จะมีข้อความเตือนขึ้นมา

เหตุการณ์หลัก:

1. User ร้องขอตำแหน่งปัจจุบัน
2. ระบบรับข้อมูลพิกัดมาจาก GPS receiver
3. ระบบนำพิกัดไปคำนวณและแปลงเป็นตำแหน่งบนแผนที่

ข้อเพิ่มเติม:

ถ้า GPS receiver ไม่อยู่ในสถานะพร้อมทำงาน จะมีข้อความเตือนขึ้นมา

Use Case: Find the way point

Actors หลัก: User

เป้าหมาย: เพื่อหาสถานที่หรือตำแหน่งที่ต้องการเดินทางไป

เงื่อนไขก่อน: ต้องมีชื่อ ประเภท พิกัด และตำแหน่งบนแผนที่ ของสถานที่ที่ต้องการอยู่ในฐานข้อมูลของระบบแล้ว

เงื่อนไขสิ้นสุดที่สำเร็จ: ถ้าสถานที่ที่ต้องการค้นหามีในระบบจะแสดงตำแหน่งบนแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไขสิ้นสุดที่ล้มเหลว: ถ้าไม่มีสถานที่ที่ต้องการอยู่ในระบบจะไม่สามารถแสดงตำแหน่งบนแผนที่ได้

เหตุการณ์หลัก:

1. Userใส่ชื่อสถานที่ หรือ ประเภทของสถานที่ ที่ต้องการค้นหา
2. ระบบทำกาค้นหาข้อมูลสถานที่ตามที่Userต้องการ
3. ระบบนำพิกัดไปคำนวณและแปลงเป็นตำแหน่งบนแผนที่
4. User เลือกให้ระบบนำทางไปยังสถานที่ที่ต้องการ
5. ระบบนำทาง User ไปยังสถานที่ที่ต้องการ (Use Case: Navigate to the target way point)

ข้อเพิ่มเติม:

ถ้าไม่มีสถานที่ที่ต้องการอยู่ในระบบจะแสดงข้อความเตือนขึ้นมา

Use Case: Update traffic news

Actors หลัก: User

เป้าหมาย: เพื่อปรับปรุงข่าวสารการจราจร

เงื่อนไขก่อน: เป็นข่าวการจราจรของถนนที่มีในระบบ

เงื่อนไขสิ้นสุดที่สำเร็จ: ถ้าเป็นข่าวการจราจรของถนนที่มีในระบบจะปรับปรุงเวลาที่ใช้ในการเดินทางผ่านถนนนั้น

เงื่อนไขสิ้นสุดที่ล้มเหลว: ถ้าไม่มีถนนในระบบจะไม่สามารถปรับปรุงข้อมูลได้

เหตุการณ์หลัก:

1. User ร้องขอค่าสภาพการจราจรล่าสุด
2. ระบบร้องขอไปยัง Traffic Monitoring Server
3. Server ส่งข้อมูลการจราจรกลับมา
4. ระบบปรับปรุงค่าเวลาการเดินทางของถนนตามสภาพการจราจรที่ได้รับจาก Server

ข้อเพิ่มเติม:

ระบบสามารถร้องขอข้อมูลได้โดยอัตโนมัติ

Use Case: Update the fastest route

Actors หลัก: User

เป้าหมาย: เพื่อปรับปรุงเส้นทางที่ใช้เวลาน้อยที่สุด

เงื่อนไขก่อน: -

เงื่อนไขสิ้นสุดที่สำเร็จ: -

เงื่อนไขสิ้นสุดที่ล้มเหลว: -

เหตุการณ์หลัก:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ระบบทำการคำนวณหาเส้นทางที่ใช้เวลาสั้นที่สุดในการเดินทางไปแต่ละสถานที่ใหม่โดยใช้อัลกอริทึมไดสตรา

ข้อเพิ่มเติม:

-

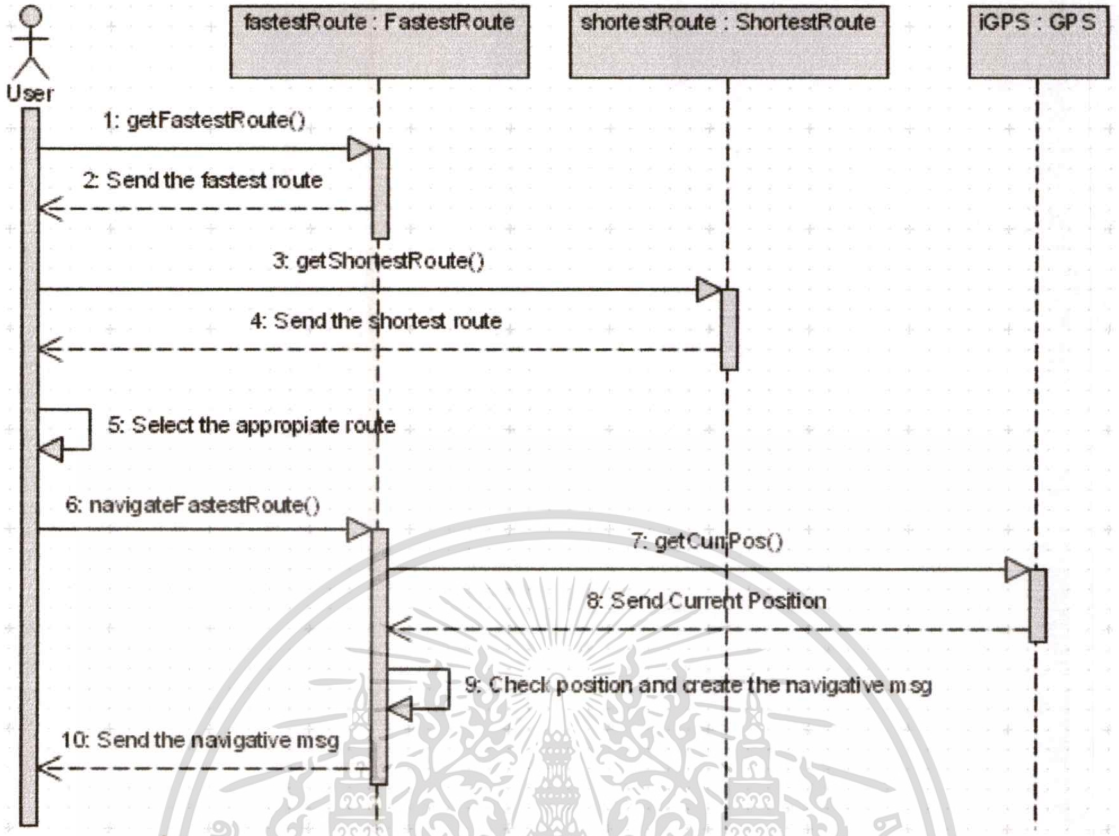
3.4 Sequence Diagram

Sequence Diagram จะแสดงให้เห็นว่าใน Use Case นั้น อ็อบเจกต์ แต่ละตัวจะติดต่อสื่อสารกันอย่างไรและมีขั้นตอนการทำงานอย่างไร โดย Sequence Diagram จะแบ่งเป็น 2 แกน แกนตั้งเป็นแกนเวลาของ อ็อบเจกต์ ที่ทำงานในช่วงหนึ่งๆ แกนนอนจะแสดงขั้นตอนการส่ง message กันระหว่าง อ็อบเจกต์ ซึ่ง Sequence Diagram จะเน้นแกนเวลาเป็นสำคัญโดยถ้าเวลาเปลี่ยนขั้นตอนก็จะเปลี่ยนด้วย ตัวอย่าง Sequence Diagram ของ Use Case Navigate to the target way point ดังรูปที่ 3.3

จาก Sequence Diagram ของ Use Case Navigate to the target way point สรุปเป็นขั้นตอนการทำงานได้ดังต่อไปนี้

1. User ต้องการทราบเส้นทางที่เร็วที่สุดจากตำแหน่งปัจจุบันไปยังตำแหน่งที่ต้องการ โดยเรียกผ่าน เมท็อด getFastestRoute ของ อ็อบเจกต์ fastestRoute
2. อ็อบเจกต์ fastestRoute จะได้เส้นทางที่เร็วที่สุดจากตำแหน่งปัจจุบันไปยังตำแหน่งที่ต้องการกลับมา
3. User ต้องการทราบเส้นทางที่สั้นที่สุดจากตำแหน่งปัจจุบันไปยังตำแหน่งใดๆ โดยเรียกผ่าน เมท็อดgetShortestRoute ของ อ็อบเจกต์ shortestRoute
4. อ็อบเจกต์ shortestRoute จะส่งเส้นทางที่สั้นที่สุดจากตำแหน่งปัจจุบันไปยังตำแหน่งที่ต้องการกลับมา
5. User เลือกเส้นทางที่สั้นที่สุดหรือเร็วที่สุด
6. User เลือกให้ระบบนำทางไปตามเส้นทางที่เลือกโดยเรียกผ่าน เมท็อด navigateFastestRoute ของ อ็อบเจกต์ fastestRoute หรือ ถ้าเลือกของเส้นทางที่สั้นที่สุดจะเป็น เมท็อด getShortestRoute ของ อ็อบเจกต์ shortestRoute
7. อ็อบเจกต์ fastestRoute/shortestRoute จะเรียก เมท็อด getCurrPos ของ อ็อบเจกต์ iGPS
8. อ็อบเจกต์ iGPS จะส่งตำแหน่งปัจจุบันของ User กลับมา
9. อ็อบเจกต์ fastestRoute/shortestRoute ตรวจสอบตำแหน่งปัจจุบันกับเส้นทาง หรือ เป็นตำแหน่งปลายทางแล้วหรือไม่
10. อ็อบเจกต์ fastestRoute/shortestRoute จะส่ง message กลับไปให้ User ให้เลี้ยวซ้าย เลี้ยวขวา หรือเป็นตำแหน่งตำแหน่งที่ต้องการแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 Sequence Diagram ของ Use Case Navigate to the target way point

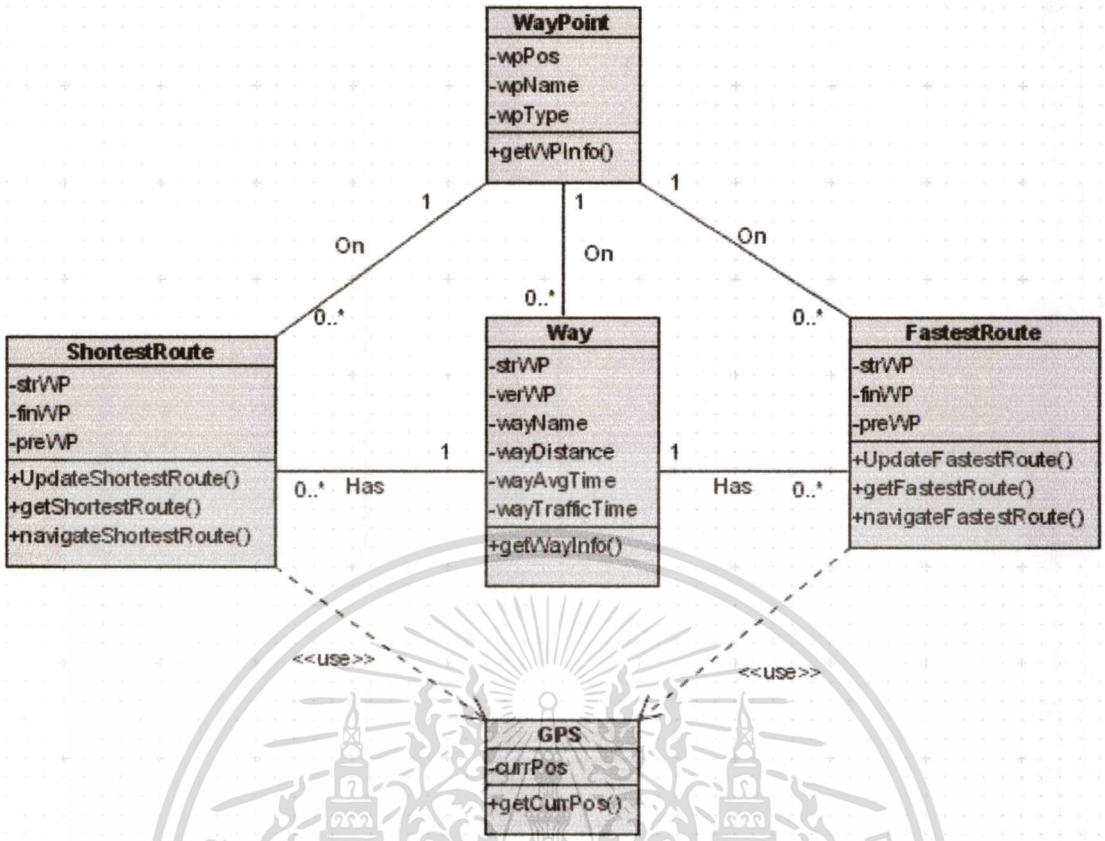
3.5 Class Diagram

Class Diagram จะแสดงถึงโครงสร้างของระบบอันประกอบไปด้วยคลาสต่างๆ และความสัมพันธ์ระหว่างคลาส ระบบนำทางอัตโนมัติมี class diagram ของระบบดังนี้

จากรูปที่ 3.4เป็น Class Diagram ของระบบนำทางอัตโนมัติ ซึ่งมีรายละเอียดดังนี้

1. คลาส WayPoint เป็นคลาสที่แสดงถึงข้อมูลของจุดที่แทนด้วยสถานที่ แยก หรือตำแหน่งต่างๆบนแผนที่ มี แอทริบิว และ เมท็อด ดังต่อไปนี้
   - wpPos คือ ตำแหน่งของสถานที่
   - wpName คือ ชื่อของสถานที่
   - wpType คือ ประเภทของสถานที่ เช่น แยกไฟแดง ร้านอาหาร
   + getWPInfo() แสดงรายละเอียดของสถานที่ต่างๆ
2. คลาส Way เป็นคลาสที่แสดงถึงข้อมูลของทางหรือถนนที่เชื่อมต่อระหว่างจุดบนทางหรือสถานที่ต่างๆบนแผนที่ มี แอทริบิว และ เมท็อด ดังต่อไปนี้
   - strWP คือ จุดเริ่มต้นของทาง
   - verWP คือ จุดสิ้นสุดของทาง
   - wayName คือ ชื่อทาง หรือ ชื่อถนน
   - wayAvgTime คือ เวลาเฉลี่ยที่ใช้เดินทางบนทางหรือถนนเส้นนี้เป็นค่าอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 Class Diagram ของระบบนำทางอัตโนมัติ

- wayTrafficTime คือ เวลาที่ใช้ในการเดินทางโดยนำค่าเฉลี่ยไปบวกเพิ่มขึ้นหรือลดลงตามสภาพการจราจร

+ getWPInfo() แสดงรายละเอียดของทางหรือถนนต่างๆ

3. คลาส ShortestRoute เป็นคลาสที่แสดงถึงข้อมูลของเส้นทางที่สั้นที่สุดในการเดินทางจากจุดใดๆไปยังทุกจุดบนแผนที่ โดยสร้างขึ้นตามไดสตราอัลกอริทึม ซึ่งมี แอทริบิว และเมท็อด ดังต่อไปนี้

- strWP คือ จุดเริ่มต้นของเส้นทาง
- finWP คือ จุดสิ้นสุดของเส้นทาง
- preWP คือ จุดก่อนถึงจุดสิ้นสุดของเส้นทาง

+ UpdateShortestRoute นำทางหรือถนนจาก คลาส way ทั้งหมดมาเข้า ไดสตราอัลกอริทึม โดยให้ cost เป็น wayDistance หรือระยะทางของถนนจะได้เส้นทางสั้นที่สุดและนำเส้นทางสั้นที่สุดของทุกจุดบนแผนที่เก็บไว้ใน คลาส ShortestRoute

+ getShortestRoute แสดงเส้นทางสั้นที่สุดจากจุดหนึ่งไปยังจุดหนึ่ง โดยส่ง parameter เป็น strWP และ finWP เข้าไปค้นหาเมื่อได้ค่ากลับมาตรวจสอบว่า preWP เป็นค่าเดียวกับ strWP หรือไม่ถ้าไม่ให้เรียก เมท็อดe getShortestRoute อีกครั้งหนึ่งโดยให้ finWP = preWP ทำการค้นหาอีกครั้งจนกระทั่งได้ว่า preWP = strWP จะได้เส้นทางที่สั้นที่สุดในการเดินทางออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+ navigateShortestRoute นำทางโดยใช้เส้นทางสั้นที่สุดจากจุดหนึ่งไปยังอีกจุดหนึ่งโดยจะนำตำแหน่งปัจจุบันซึ่งได้จากการเรียก เมท็อด getCurrPos จาก คลาส GPS ไปเทียบกับเส้นทางที่ว่าอยู่ ณ ตำแหน่งใดและต่อไปต้องเดินทางหรือเลี้ยวไปทางทิศใด

4. คลาส FastestRoute เป็น คลาส ที่แสดงถึงข้อมูลของเส้นทางที่เร็วที่สุดในการเดินทางจากจุดใดๆไปยังทุกจุดบนแผนที่ โดยสร้างขึ้นตาม ไดสตราอัลกอริทึม ซึ่งมี แอทริบิว และเมท็อด ดังต่อไปนี้

- strWP คือ จุดเริ่มต้นของเส้นทาง
- finWP คือ จุดสิ้นสุดของเส้นทาง
- preWP คือ จุดก่อนถึงจุดสิ้นสุดของเส้นทาง

+ UpdateFastestRoute นำทางหรือถนนจาก คลาส way ทั้งหมดมาเข้า ไดสตราอัลกอริทึม โดยให้ cost เป็น wayTrafficTime หรือเวลาที่ใช้ในการเดินทางบนถนน จะได้เส้นทางเร็วที่สุดและนำเส้นทางเร็วที่สุดของทุกจุดบนแผนที่เก็บไว้ใน คลาส FastestRoute

+ getFastestRoute แสดงเส้นทางเร็วที่สุดจากจุดหนึ่งไปยังจุดหนึ่ง โดยส่ง parameter เป็น strWP และ finWP เข้าไปค้นหาเมื่อได้ค่ากลับมาตรวจสอบว่า preWP เป็นค่าเดียวกับ strWP หรือไม่ถ้าไม่ให้เรียก เมท็อดe getFastestRoute อีกครั้งหนึ่งโดยให้ finWP = preWP ทำการค้นหาอีกครั้งจนกระทั่งได้ว่า preWP = strWP จะได้เส้นทางที่เร็วที่สุดในการเดินทางออกมา

+ navigateFastestRoute นำทางโดยใช้เส้นทางเร็วที่สุดจากจุดหนึ่งไปยังอีกจุดหนึ่งโดยจะนำตำแหน่งปัจจุบันซึ่งได้จากการเรียก เมท็อด getCurrPos จาก คลาส GPS ไปเทียบกับเส้นทางที่ว่าอยู่ ณ ตำแหน่งใดและต่อไปต้องเดินทางหรือเลี้ยวไปทางทิศใด

5. คลาส GPS เป็น คลาส ใช้เพื่อติดต่อกับ GPS Receiver เพื่อขอค่าตำแหน่งปัจจุบันมี แอทริบิว และ เมท็อด ดังต่อไปนี้

- currPos คือ ค่าของตำแหน่งปัจจุบันที่ได้จาก GPS Receiver

+ getCurrPos จะคืนค่าของตำแหน่งปัจจุบันที่ได้จาก GPS Receiver

3.6 การออกแบบระบบฐานข้อมูล

จาก Class Diagram นำมาออกแบบฐานข้อมูลของระบบซึ่งประกอบไปด้วย 4 ตาราง ดังนี้คือ

1. ตาราง WP (Way Point) เก็บพิกัดและรายละเอียดของจุดหรือสถานที่สำคัญต่างๆ เช่น สี่แยก อาคาร โรงพยาบาล เป็นต้น

2. ตาราง WAY เก็บรายละเอียดของทางหรือถนน โดยเกิดจาก Way Point 2 จุด เชื่อมต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ตาราง SHORTEST_PATH เก็บเส้นทางที่สั้นที่สุดจากจุดเริ่มต้นไปยังจุดเป้าหมายใดๆ เป็นข้อมูลที่ได้จากอัลกอริทึมไดสตราโดยใช้ระยะทางของแต่ละถนนในการคำนวณ

4. ตาราง FASTEST_PATH เก็บเส้นทางที่เร็วที่สุดจากจุดเริ่มต้นไปยังจุดเป้าหมายใดๆ เป็นข้อมูลที่ได้จากอัลกอริทึมไดสตราโดยใช้เวลาในการเดินทางของแต่ละถนนในการคำนวณ

ตารางทั้งหมดมีรายละเอียดของแต่ละตารางดังนี้

ตารางที่ 3.1 รายละเอียดของตาราง WP

คีย์	ลำดับ	ชื่อ Field	ประเภท	รายละเอียด
PK	1	WP_NO	int	หมายเลขของจุดที่สนใจ
	2	WP_POS_X	int	ค่าพิกัดของจุดในแนวแกน x บนอุปกรณ์คอมพิวเตอร์แบบพกพา
	3	WP_POS_Y	int	ค่าพิกัดของจุดในแนวแกน y บนอุปกรณ์คอมพิวเตอร์แบบพกพา
	4	WP_NAME	varchar(100)	ชื่อของสถานที่
	5	WP_TYPE	varchar(100)	ประเภทของสถานที่

ตารางที่ 3.2 รายละเอียดของตาราง WAY

คีย์	ลำดับ	ชื่อ Field	ประเภท	รายละเอียด
PK, FK	1	STR_WP_NO	int	หมายเลขของจุดเริ่มต้นทาง
PK, FK	2	VER_WP_NO	int	หมายเลขของจุดปลายทาง
	3	WAY_NAME	varchar(100)	ชื่อถนน
	4	WAY_DISTANCE	int	ระยะทางหน่วยเป็นเมตร
	5	WAY_TRAFFIC_TIME	int	เวลาในการเดินทาง ณ ปัจจุบัน หน่วยเป็นวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 รายละเอียดของตาราง SHORTEST_PATH

คีย์	ลำดับ	ชื่อ Field	ประเภท	รายละเอียด
PK, FK	1	STR_WP_NO	int	หมายเลขของจุดเริ่มต้นเส้นทาง
PK, FK	2	FIN_WP_NO	int	หมายเลขของจุดสิ้นสุดเส้นทาง
FK	3	PRE_WP_NO	int	หมายเลขของจุดที่ผ่านก่อนถึงจุดเป้าหมาย

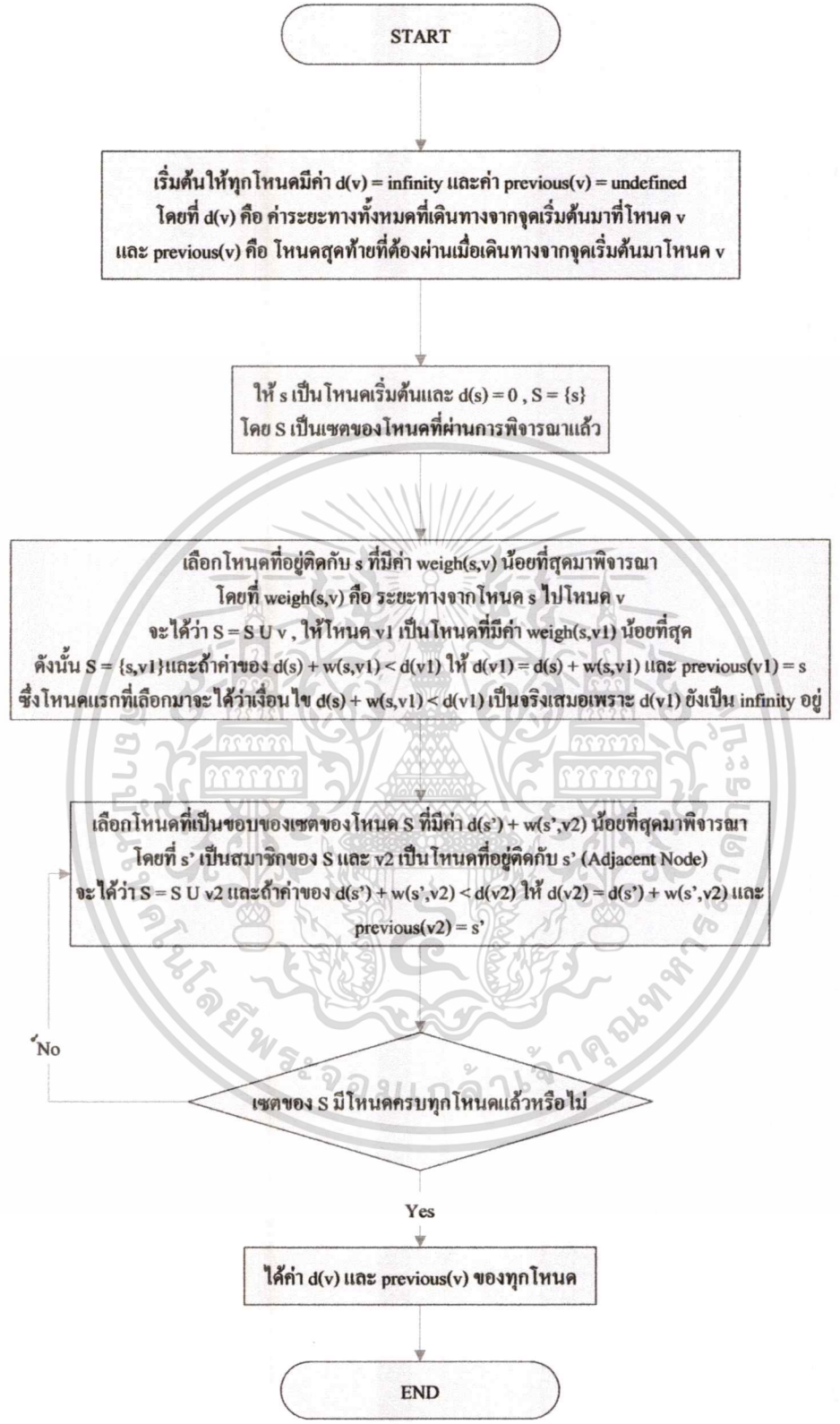
ตารางที่ 3.4 รายละเอียดของตาราง FASTEST_PATH

คีย์	ลำดับ	ชื่อ Field	ประเภท	รายละเอียด
PK, FK	1	STR_WP_NO	int	หมายเลขของจุดเริ่มต้นเส้นทาง
PK, FK	2	FIN_WP_NO	int	หมายเลขของจุดสิ้นสุดเส้นทาง
FK	3	PRE_WP_NO	int	หมายเลขของจุดที่ผ่านก่อนถึงจุดเป้าหมาย

3.7 อัลกอริทึมในการหาเส้นทาง

ระบบนำทางอัตโนมัติใช้ไดสตราอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดและเร็วที่สุด โดย input ของไดสตราอัลกอริทึมคือ โหนด และ weight ระหว่างโหนด ซึ่งโหนดทั้งหมดจะได้มาจากตาราง WP ส่วน weight จะได้จากตาราง WAY และตัวแทนของโหนดคือ WP_NO ส่วน weight ระหว่างโหนดคือ WAY_DISTANCE หรือ WAY_TRAFFIC_TIME ขึ้นอยู่กับว่าต้องการคำนวณหาเส้นทางที่มีระยะทางสั้นที่สุด หรือต้องการคำนวณหาเส้นทางที่ใช้เวลาน้อยที่สุด(เร็วที่สุด) และทั้ง WP_NO และ WAY_DISTANCE หรือ WAY_TRAFFIC_TIME จะถูกนำไปใช้เป็น input ของไดสตราอัลกอริทึมเพื่อคำนวณหาเส้นทางที่สั้นที่สุดหรือเร็วที่สุด โดย Flow Chart การทำงานของไดสตราอัลกอริทึมจะเป็นดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

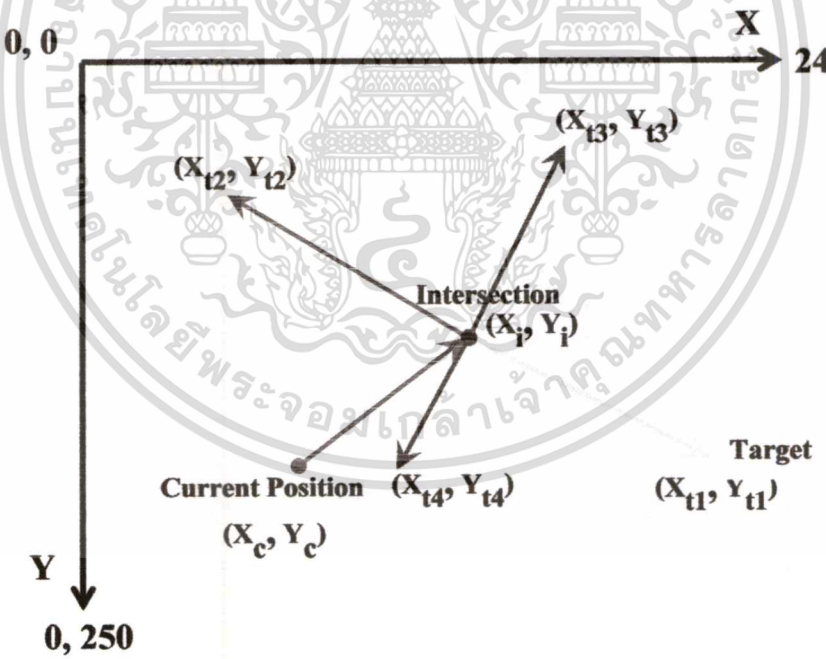


รูปที่ 3.5 รูปแสดง Flow Chart การทำงานของไดสตราอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การคำนวณหาข้อความนำทาง

ระบบนำทางอัตโนมัติสามารถคำนวณหาข้อความนำทาง (Navigation Message)ได้จาก จุด 3 จุด คือ จุดที่อยู่ ณ ปัจจุบัน (Current Point) , จุดที่เป็นทางแยกเพื่อไปยังจุดเป้าหมาย (Intersection Point) และจุดเป้าหมาย (Target Point) โดยการคำนวณหาข้อความนำทาง จากกรณีที่ จุด Current Point กำลังเคลื่อนที่เข้าหาจุด Intersection Point ซึ่งเป็นทางแยกเพื่อผ่านไปสู่จุด Target Point จะพิจารณาค่าพิกัด x และ y ของจุด Current Point กับจุด Intersection Point ว่าแนวโน้มมีค่าลดลงหรือเพิ่มขึ้นและตั้งเป็นเกณฑ์ไว้พิจารณา จากนั้นจะนำแนวโน้มที่ได้ไปเปรียบเทียบกับ จุด Intersection Point กับ จุด Target Point ว่ามีแนวโน้มเดียวกันหรือไม่ ถ้ามีแนวโน้มเดียวกันแสดงว่าการเคลื่อนที่จากจุดปัจจุบันผ่านทางแยกต้องตรงไปข้างหน้า (Go Straight) ถ้ามีแนวโน้มตรงกันข้ามกันจะต้องกลับรถ (U-turn) และถ้ามีเฉพาะแนวโน้มของค่า x หรือ y ที่ไม่เหมือนกันก็คือ เลี้ยวขวา (Turn Right) หรือ เลี้ยวซ้าย (Turn Left) แต่ในกรณีกลับรถจะไม่สามารถบอกได้ว่ากลับรถไปทางขวาหรือทางซ้ายเพราะมีแนวโน้นแบบเดียวกันไม่สามารถแยกได้



รูปที่ 3.6 รูปแสดงตัวอย่างการเคลื่อนที่จาก Current ผ่าน Intersection ไปยัง Target Point ต่างๆ

จากรูป 3.6 รูปแสดงตัวอย่างการเคลื่อนที่จาก Current ผ่าน Intersection ไปยัง Target Point ต่างๆ โดยจากตัวอย่างค่าแนวโน้มของ x และ y จากจุด Current ไปจุด Intersection ที่ใช้ในการพิจารณามีค่า x เพิ่มขึ้นและ y ลดลง ดังนั้นทำการคำนวณหาข้อความนำทางได้ดังตารางที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 ตารางแสดงข้อความนำทางของค่าแนวโน้ม x เพิ่มขึ้นและ y ลดลง

$X_i - X_c$	$Y_i - Y_c$	n	$X_{tn} - X_i$	$Y_{tn} - Y_i$	Navigate Message
+	–	1	+	+	Turn Right
		2	–	–	Turn Left
		3	+	–	Go Straight
		4	–	+	U-Turn

ดังนั้นจากเงื่อนไขและกฎข้างต้นสามารถสร้างตารางหาข้อความนำทางของทุกๆ แนวโน้มของค่า x และ y ได้ดังต่อไปนี้

ตารางที่ 3.6 ตารางแสดงข้อความนำทางทุกๆแนวโน้มของค่า x และ y

$X_i - X_c$	$Y_i - Y_c$	$X_{tn} - X_i$	$Y_{tn} - Y_i$	Navigate Message
+	+	+	+	Go Straight
		-	+	Turn Right
		+	-	Turn Left
		-	-	U-Turn
-	+	+	+	Turn Left
		-	+	Go Straight
		+	-	U-Turn
		-	-	Turn Right
+	-	+	+	Turn Right
		-	+	U-Turn
		+	-	Go Straight
		-	-	Turn Left
-	-	+	+	U-Turn
		-	+	Turn Left
		+	-	Turn Right
		-	-	Go Straight

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การพัฒนาระบบ

4.1 ขั้นตอนการพัฒนาระบบ

ระบบนำทางอัตโนมัติ พัฒนาโดยใช้ภาษา C#.NET พัฒนาในลักษณะของ Mobile Application และใช้ Frame Work ของ .NET Compact Framework 2.0 ซึ่งการพัฒนาจะทำการพัฒนาบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ และในขั้นตอนการทดสอบการทำงานของส่วนย่อยจะทดสอบผ่าน Emulator ส่วนการทดสอบแบบระบบรวมจะทดสอบบนเครื่องคอมพิวเตอร์แบบพกพา ลำดับการพัฒนาเป็นดังนี้

1. พัฒนาส่วนของการกำหนดรูปแบบข้อมูลของแผนที่และข้อมูลการจราจร และการนำเข้าสู่ระบบ
2. พัฒนาส่วนของการแสดงผลแผนที่บนหน้าจอของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพา
3. พัฒนาส่วนของการหาเส้นทางโดยใช้ไดสตราอัลกอริทึม
4. พัฒนาส่วนของการแสดงเส้นทางที่สั้นที่สุดและเร็วที่สุดให้ผู้ใช้เลือกเพื่อกำหนดเส้นทางที่ต้องการเดินทาง
5. พัฒนาส่วนของการนำทางอัตโนมัติ
6. พัฒนาส่วนของการติดต่อกับ GPS เพื่อรับพิกัดของตำแหน่งปัจจุบัน
7. พัฒนาส่วนของการติดต่อกับ Traffic Monitoring Server เพื่อรับข้อมูลการจราจร ณ เวลาปัจจุบัน
8. พัฒนาส่วนของการค้นหาสถานที่ที่สนใจ(Way point)
9. พัฒนาส่วนของเมนูการติดต่อกับผู้ใช้
10. ทดสอบความถูกต้องของระบบ

4.2 รายละเอียดการพัฒนาระบบ

4.2.1 การพัฒนาส่วนของการกำหนดรูปแบบข้อมูลของแผนที่และข้อมูลการจราจร และการนำเข้าสู่ระบบ

การพัฒนาในส่วนของการกำหนดรูปแบบข้อมูลของแผนที่และข้อมูลการจราจร และการนำเข้าสู่ระบบ สามารถอธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กำหนดรูปแบบของ file ที่เก็บข้อมูลจุดสนใจ file ชื่อว่า “wp.txt” โดยมี attribute ตรงกับ field ในตารางที่ 3.1 รายละเอียดของตาราง WP และมีเครื่องหมาย colon “:” คั่นระหว่าง attribute ต่างๆ

   {WP_POS_X}:{WP_POS_Y}:{WP_NO}:{WP_NAME}:{WP_TYPE}

   ตัวอย่างของ file “wp.txt”

   30:60:0:Casaview:Village

   60:80:1:Intersect1:Intersection

   80:50:2:Vipha:Hostpital

   30:130:3:Samsen:School

   100:125:4:Intersect2:Intersection

   140:150:5:Arois:Resteraunt

   90:180:6:LPN:Building

2. พัฒนา function ในการนำเข้า file ที่เก็บข้อมูลจุดสนใจสู่ระบบ และต่อไปจะเรียกว่าข้อมูล WP

3. กำหนดรูปแบบของ file ที่เก็บข้อมูลถนนหรือทาง file ชื่อว่า “way.txt” โดยมี attribute ตรงกับ field ในตารางที่ 3.2 รายละเอียดของตาราง WAY และมีเครื่องหมาย colon “:” คั่นระหว่าง attribute ต่างๆ

   {STR_WP_NO}:{VER_WP_NO}:{WAY_DISTANCE}:{WAY_TRAFFIC_TIME}:{WAY_NAME}

   ตัวอย่างของ file “way.txt”

   0:1:36:40:RoadName0

   1:2:36:50:RoadName1

   1:3:58:100:RoadName2

4. พัฒนา function ในการนำเข้า file ที่เก็บข้อมูลถนนหรือทางสู่ระบบ และต่อไปจะเรียกว่าข้อมูล WAY

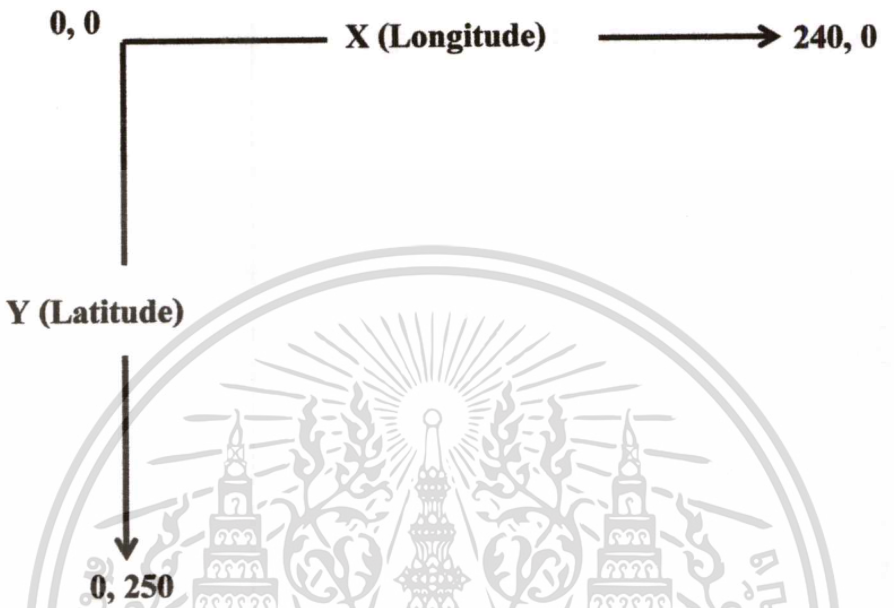
4.2.2 การพัฒนาส่วนของการแสดงผลแผนที่บนหน้าจอของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพา

การพัฒนาในส่วนของการแสดงผลแผนที่บนหน้าจอของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพา สามารถอธิบายได้ดังนี้

1. พัฒนา function ในการแสดงผลจุดสนใจบนหน้าจอของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพาซึ่งมีขนาด 240 * 250 (x * y) โดยแสดงผลเป็นรูปวงกลมสีน้ำเงิน ณ ตำแหน่งที่ระบุในข้อมูล WP และมีชื่อของจุดสนใจเป็นสีม่วงอยู่ใต้จุด หรือจะแสดงเป็นหมายเลขจุดสนใจและพิกัดแทน

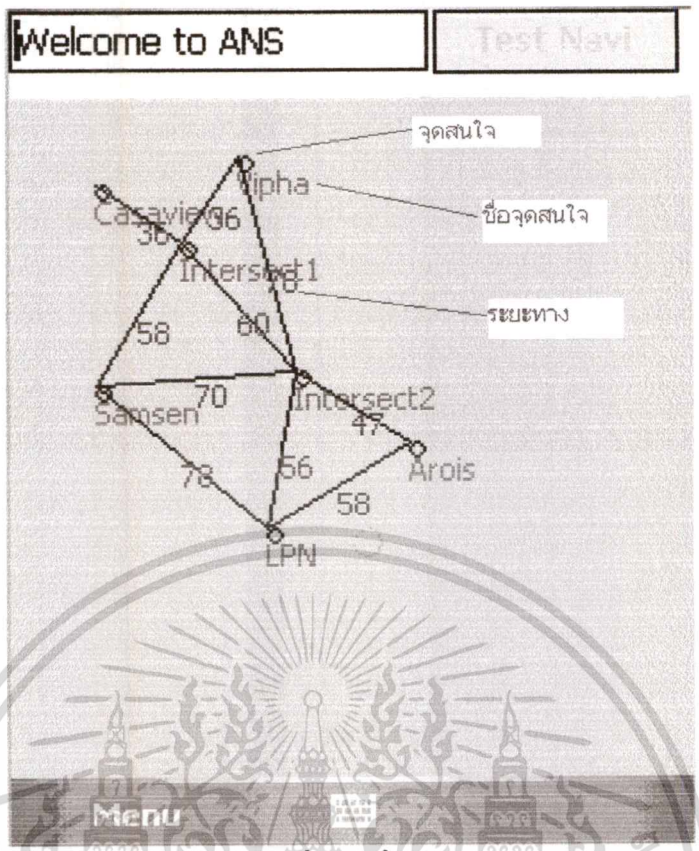
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. พัฒนา function ในการแสดงผลทางหรือถนนบนหน้าจอของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพา โดยแสดงผลเป็นเส้นสีน้ำเงินเชื่อมระหว่างจุด 2 จุด ตามที่ระบุในข้อมูล WAY และแสดงค่าระยะทางเป็นสีแดงอยู่บนเส้นด้วย หรือแสดงค่าเวลาที่ใช้ในการเดินทางอยู่บนเส้นแต่จะมีสีเทา

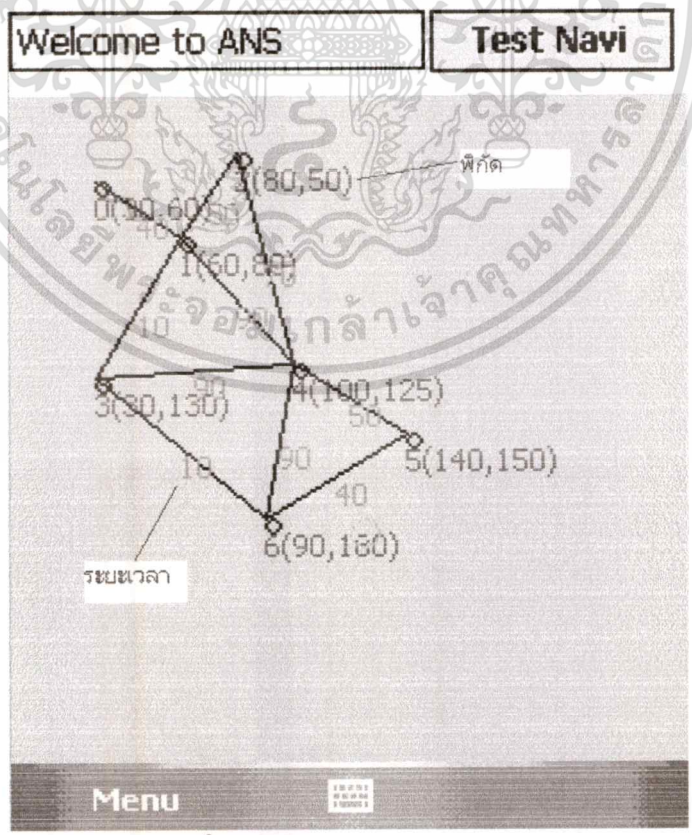


รูปที่ 4.1 แสดงลักษณะของแกน x และแกน y บนหน้าจอเครื่องคอมพิวเตอร์แบบพกพา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 หน้าจอแสดงข้อมูลแผนที่บนเครื่องคอมพิวเตอร์แบบพกพา



รูปที่ 4.3 หน้าจอแสดงพิกัดของจุดที่สนใจและเวลาการเดินทางของแต่ละถนนบนแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การพัฒนาส่วนของการหาเส้นทางโดยใช้ไดสตราอัลกอริทึม

การพัฒนาส่วนของการหาเส้นทางโดยใช้ไดสตราอัลกอริทึม สามารถอธิบายได้ดังนี้

1. ทำการพัฒนา function ในส่วนของอัลกอริทึมไดสตรา สำหรับหาเส้นทางที่มี weight น้อยที่สุด โดยรับ input parameter เป็นหมายเลขของโหนดเริ่มต้น และรับ input parameter อีกค่าคือ weight ระหว่างโหนดทุกโหนด และทำการคำนวณหาเส้นทางที่มี weight น้อยที่สุดจากจุดเริ่มต้นไปยังทุกๆโหนดตามไดสตราอัลกอริทึม

4.2.4 การพัฒนาส่วนของการแสดงเส้นทางที่สั้นที่สุดและเร็วที่สุดให้ผู้ใช้เลือกเพื่อกำหนดเส้นทางที่ต้องการเดินทาง

การพัฒนาในส่วนของการแสดงเส้นทางที่สั้นที่สุดและเร็วที่สุดให้ผู้ใช้เลือกเพื่อกำหนดเส้นทางที่ต้องการเดินทาง สามารถอธิบายได้ดังนี้

1. ทำการพัฒนาส่วนติดต่อกับผู้ใช้เพื่อรับชื่อจุดเริ่มต้นและจุดสิ้นสุดของทาง
2. เมื่อผู้ใช้ Click ปุ่ม "Find" จะทำการแปลงจากชื่อเป็นหมายเลข โดยค้นหาในข้อมูล WP ว่าชื่อจุดที่ต้องการแทนด้วยหมายเลขอะไร

รูปที่ 4.4 หน้าจอแสดงส่วนติดต่อกับผู้ใช้เพื่อรับ input จุดเริ่มต้นและจุดสิ้นสุดที่ต้องการ

3. ทำการเรียก function ในการหาเส้นทางโดยใช้ไดสตราอัลกอริทึม จากข้อ 4.2.3 โดยส่ง input parameter เป็นหมายเลขของโหนดเริ่มต้นตามที่ผู้ใช้เลือก และ input parameter อีกค่าคือ weight จะส่งเป็นเซตของระยะทาง (WAY_DISTANCE) ระหว่างโหนดทุกโหนดที่มีถนนหรือทางเชื่อมต่อกันซึ่งได้จากข้อมูล WAY และเก็บผลลัพธ์ของไดสตราอัลกอริทึมไว้ใน SHORTEST_PATH มีโครงสร้างเหมือนกับตารางที่ 3.3
4. ทำการเรียก function ในการหาเส้นทางโดยใช้ไดสตราอัลกอริทึม จากข้อ 4.2.3 โดยส่ง input parameter เป็นหมายเลขของโหนดเริ่มต้นตามที่ผู้ใช้เลือก และ input parameter อีกค่าคือ weight จะส่งเป็นเซตของระยะทาง (WAY_TRAFFIC_TIME) ระหว่างโหนดทุกโหนดที่มีถนนหรือทางเชื่อมต่อกันซึ่งได้จากข้อมูล WAY และเก็บผลลัพธ์ของไดสตราอัลกอริทึมไว้ใน FASTEST_PATH มีโครงสร้างเหมือนกับตารางที่ 3.4
5. ทำการหาเส้นทางที่สั้นที่สุดจากจุดเริ่มต้นและจุดสิ้นสุดตามที่ผู้ใช้เลือก โดยเริ่มต้นทำการค้นหาจากข้อมูล SHORTEST_PATH ซึ่งเก็บเส้นทางที่สั้นที่สุดจากจุดเริ่มต้น

เอกสารนี้เป็นกรรมสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น โดยไม่ได้รับอนุญาตเป็นลายลักษณ์อักษรจากจุฬาลงกรณ์มหาวิทยาลัย

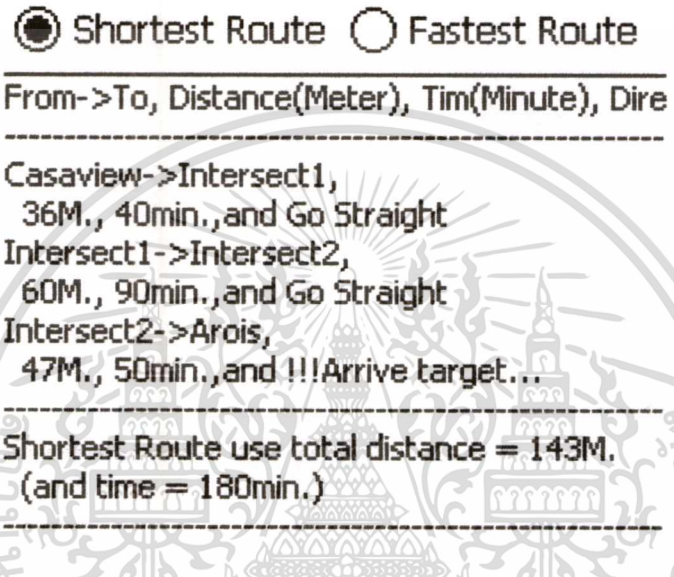
ไปยังทุกโหนดไว้ โดยให้ STR_WP_NO เป็นหมายเลขของจุดเริ่มต้น และ FIN_WP_NO เป็นหมายเลขของจุดสิ้นสุด ผลที่ได้คือ หมายเลขของจุดที่ต้องผ่าน (PRE_WP_NO) ก่อนถึงจุดสิ้นสุด เส้นทางที่ได้จะเรียงลำดับจากทางที่ผ่านก่อนถึงจุดสิ้นสุดมายังจุดเริ่มต้น ซึ่งเป็นเส้นทางย้อนกลับ ดังนั้นทำการเก็บเส้นทางจาก PRE_WP_NO ไป FIN_WP_NO ไว้ก่อนเป็นเส้นทางสุดท้าย แล้วเริ่มค้นหาต่อไป โดยให้ FIN_WP_NO = PRE_WP_NO และนำผลการค้นหาและเก็บค่า PRE_WP_NO ไป FIN_WP_NO เอาไว้อีกเช่นกัน ค้นหาไปจนได้ค่า PRE_WP_NO = STR_WP_NO แสดงว่าเป็นทางแรกที่เริ่มต้นเดินทาง ก็หยุดการค้นหาจะได้เส้นทางที่สั้นที่สุดจากจุดเริ่มต้นไปยังจุดสิ้นสุดตามที่ผู้ใช้ต้องการ จากนั้นทำการแสดงเส้นทางดังกล่าวให้ผู้ใช้พิจารณา โดยมีรายละเอียดดังนี้ From->To คือ เดินทางจากจุดไหนไปยังจุดไหน และ Distance คือ ระยะทางที่ใช้มีหน่วยเป็นเมตร (M) และเวลาที่ใช้มีหน่วยเป็นนาที (min) และแสดงข้อความนำทางด้วยเมื่อถึงจุดที่ต้องตัดสินใจว่าจะเดินทางไปทางไหน ซึ่งการคำนวณหาข้อความนำทางจะกล่าวถึงในหัวข้อเรื่อง การพัฒนาในส่วนของการนำทางอัตโนมัติ นอกจากนี้มีสรุประยะทางรวมและเวลาที่ใช้ในการเดินทางรวมเพื่อประกอบการพิจารณาเพื่อเลือกเส้นทางของผู้ใช้ด้วย

6. ทำการหาเส้นทางที่เร็วที่สุดจากจุดเริ่มต้นและจุดสิ้นสุดตามที่ผู้ใช้เลือก โดยเริ่มต้นทำการค้นหาจากข้อมูล FASTEST_PATH ซึ่งเก็บเส้นทางที่เร็วที่สุดจากจุดเริ่มต้นไปยังทุกโหนดไว้ โดยให้ STR_WP_NO เป็นหมายเลขของจุดเริ่มต้น และ FIN_WP_NO เป็นหมายเลขของจุดสิ้นสุด ผลที่ได้คือ หมายเลขของจุดที่ต้องผ่าน (PRE_WP_NO) ก่อนถึงจุดสิ้นสุด เส้นทางที่ได้จะเรียงลำดับจากทางที่ผ่านก่อนถึงจุดสิ้นสุดมายังจุดเริ่มต้น ซึ่งเป็นเส้นทางย้อนกลับ ดังนั้นทำการเก็บเส้นทางจาก PRE_WP_NO ไป FIN_WP_NO ไว้ก่อนเป็นเส้นทางสุดท้าย แล้วเริ่มค้นหาต่อไป โดยให้ FIN_WP_NO = PRE_WP_NO และนำผลการค้นหาและเก็บค่า PRE_WP_NO ไป FIN_WP_NO เอาไว้อีกเช่นกัน ค้นหาไปจนได้ค่า PRE_WP_NO = STR_WP_NO แสดงว่าเป็นทางแรกที่เริ่มต้นเดินทาง ก็หยุดการค้นหาจะได้เส้นทางที่เร็วที่สุดจากจุดเริ่มต้นไปยังจุดสิ้นสุดตามที่ผู้ใช้ต้องการ จากนั้นทำการแสดงเส้นทางดังกล่าวให้ผู้ใช้พิจารณา โดยมีรายละเอียดดังนี้ From->To คือ เดินทางจากจุดไหนไปยังจุดไหน และ Distance คือ ระยะทางที่ใช้มีหน่วยเป็นเมตร (M) และเวลาที่ใช้มีหน่วยเป็นนาที (min) และแสดงข้อความนำทางด้วยเมื่อถึงจุดที่ต้องตัดสินใจว่าจะเดินทางไปทางไหน ซึ่งการคำนวณหาข้อความนำทางจะกล่าวถึงในหัวข้อเรื่อง การพัฒนาในส่วนของการนำทางอัตโนมัติ นอกจากนี้มีสรุประยะทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวมและเวลาที่ใช้ในการเดินทางรวมเพื่อประกอบการพิจารณาเพื่อเลือกเส้นทางของผู้ใช้ด้วย

7. เมื่อผู้ใช้เลือกเส้นทางและ Click ปุ่ม "Go" จะทำการบันทึกเส้นทางที่ผู้ใช้เลือกลงสู่ข้อมูล SelectedRoute ซึ่งมีโครงสร้างเหมือนกับตาราง SHORTEST_PATH และ FASTEST_PATH จากนั้นจะเข้าสู่โหมดนำทางอัตโนมัติซึ่งจะกล่าวถึงในเรื่อง การพัฒนาส่วนของการนำทางอัตโนมัติ



รูปที่ 4.5 หน้าจอแสดงส่วนติดต่อกับผู้ใช้เพื่อแสดงผลการคำนวณหาเส้นทางแบบระยะทางสั้นที่สุด (Shortest Route)

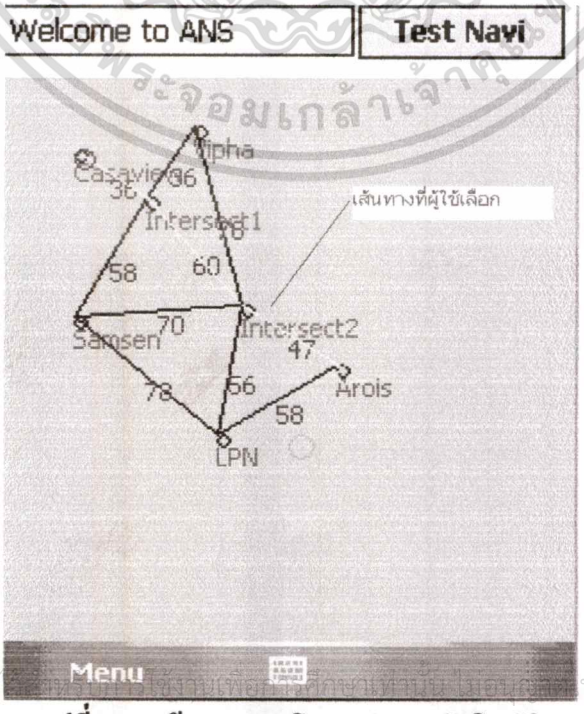
รูปที่ 4.6 หน้าจอแสดงส่วนติดต่อกับผู้ใช้เพื่อแสดงผลการคำนวณหาเส้นทางแบบใช้เวลาน้อยที่สุด (Fastest Route)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 การพัฒนาส่วนของการนำทางอัตโนมัติ

การพัฒนาในส่วนของการนำทางอัตโนมัติ สามารถอธิบายได้ดังนี้

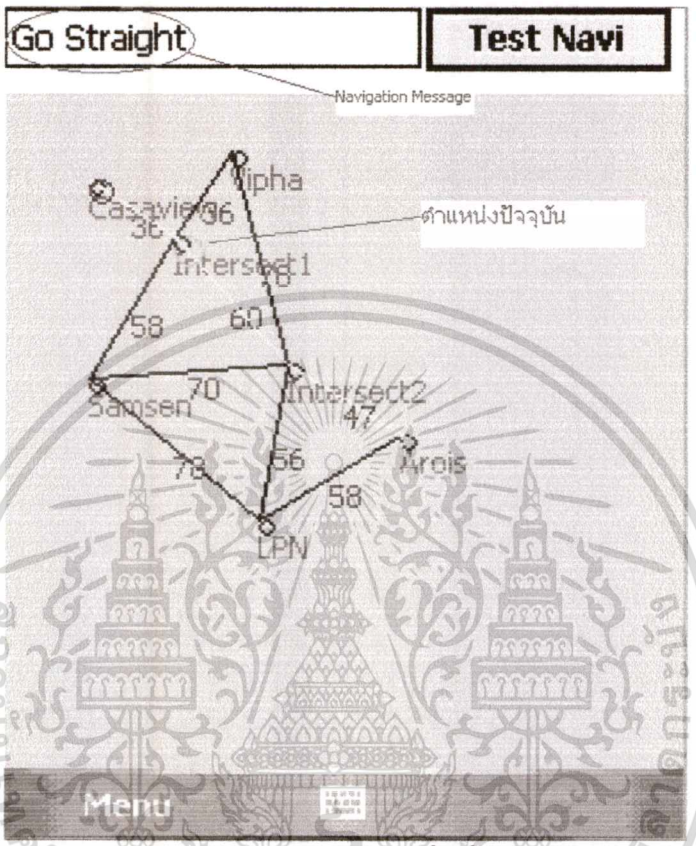
1. สร้าง function ในการหาข้อความนำทาง (Navigation Message) โดยรับ input parameter เป็นจุดปัจจุบัน (Current Point) จุดที่เป็นทางแยก (Intersection Point) และจุดเป้าหมาย (Target Point) และนำไปหาข้อความนำทาง โดยใช้เงื่อนไขจากตารางที่ 3.6 ตารางแสดงข้อความนำทางทุกๆแนวโน้มของค่า x และ y
2. เมื่อผู้ใช้เลือกเส้นทางตามที่ต้องการแล้ว Click "Go" เส้นทางจะถูกบันทึกลงใน SelectedRoute และจะนำเอาเส้นทางดังกล่าวไปคำนวณหาข้อความนำทาง ณ จุดต่างๆ ที่อยู่บนเส้นทาง ถ้าเป็นจุดเริ่มต้นจะได้ข้อความนำทางว่า "Start Go Ahead..." ถ้าเป็นจุดสิ้นสุดจะได้ข้อความนำทางว่า "!!!Arrive Target..." แต่ถ้าไม่ใช่จุดเริ่มต้นกับจุดสิ้นสุดให้เรียก function ในข้อย่อยที่ 1 และส่ง input parameter ดังนี้ คือ จุดปัจจุบัน (Current Point) ให้ส่งเป็น จุดที่ต้องผ่านก่อนถึงจุดที่ต้องการหาข้อความนำทาง หาได้จาก SelectedRoute ที่มี FIN_WP_NO เท่ากับจุดที่ต้องการหาข้อความนำทาง จะได้ค่าของ PRE_WP_NO คืนกลับไป ส่วนจุดที่เป็นทางแยก ให้ส่งเป็นจุดที่ต้องการหาข้อความนำทาง และจุดท้ายจุดเป้าหมายให้ส่งเป็นจุดที่ต้องเดินทางต่อไปจากจุดที่ต้องการหาข้อความนำทาง หาได้จาก SelectedRoute ที่มี PRE_WP_NO เท่ากับจุดที่ต้องการหาข้อความนำทาง จะได้ค่าของ FIN_WP_NO คืนกลับไป เมื่อคำนวณข้อความนำทางเรียบร้อยแล้วจะเข้าสู่โหมดนำทางอัตโนมัติ
3. โหมดนำทางอัตโนมัติจะแสดงเส้นทางใน SelectedRoute เป็นสีเหลือง



รูปที่ 4.7 หน้าจอแสดงโหมดนำทางอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

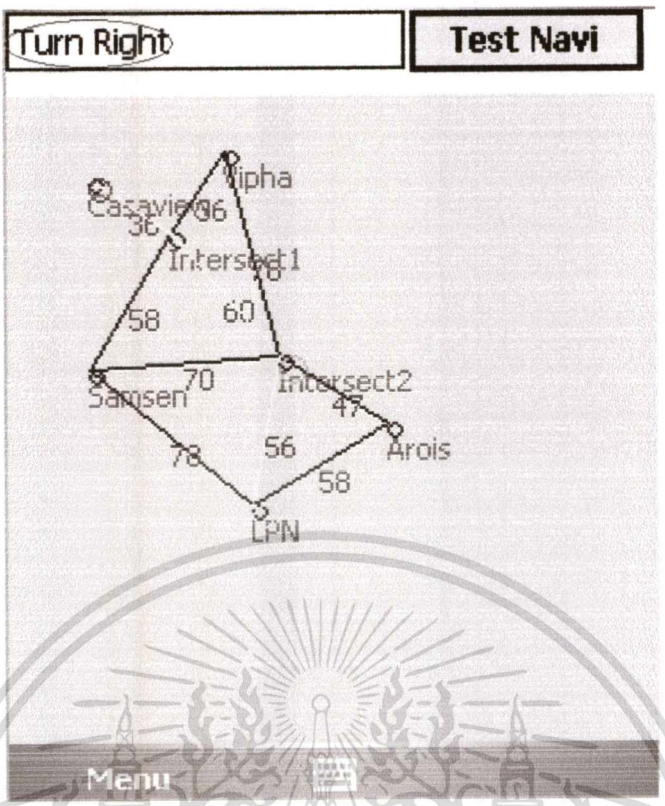
4. เมื่อเข้าสู่โหมดนำทางอัตโนมัติ จะนำเอาข้อมูลเอาตำแหน่งปัจจุบัน (แสดงด้วยวงกลมสีชมพู) ไปเทียบกับจุดที่อยู่บนทางว่าตรงกับจุดใดและแสดงข้อความนำทางของจุดนั้นที่ได้จากข้อย่อยที่ 2 บนหน้าจอให้ผู้ใช้เห็น



รูปที่ 4.8 ตัวอย่างหน้าจอแสดงข้อความนำทางเมื่อเริ่มระบบนำทางอัตโนมัติ

5. พัฒนาส่วนโหมดทดสอบการนำทางอัตโนมัติโดยเมื่อ Click ปุ่ม “Test Navi” จะย้ายตำแหน่งปัจจุบันไปอยู่บนเส้นทางที่ผู้ใช้เลือกโดยเริ่มที่จุดเริ่มต้นของ SelectedRouted และจะแสดงข้อความนำทาง ทุกครั้งที่มีการ Click ปุ่ม “Test Navi” จะย้ายตำแหน่งปัจจุบันให้เคลื่อนที่ไปเรื่อยๆบนจุดที่อยู่บนเส้นทางที่ผู้ใช้เลือกดังนั้นเมื่อตำแหน่งปัจจุบันเคลื่อนที่จะมีข้อความนำทางแสดงออกมาเมื่อถึงจุดที่ต้องตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ตัวอย่างหน้าจอแสดงแสดงข้อความนำทางในโหมดทดสอบระบบนำทางอัตโนมัติ

4.2.6 การพัฒนาส่วนของการติดต่อกับ GPS เพื่อรับพิกัดของตำแหน่งปัจจุบัน

การพัฒนาในส่วนของการติดต่อกับ GPS เพื่อรับพิกัดของตำแหน่งปัจจุบัน โดยที่ GPS Receiver เชื่อมต่ออยู่กับอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพา ผ่านทาง Bluetooth Connection สามารถอธิบายได้ดังนี้

1. ทำการพัฒนา function ในการเชื่อมต่อกับ GPS Receiver ผ่านทาง Bluetooth Connection โดยใช้ library serial port ของ .Net Compact Framework ในการสร้างการเชื่อมต่อ โดยผู้ใช้ต้องเลือก Com Port ที่เชื่อมต่อยู่กับ GPS Receiver ก่อน
2. ทำการถอดรหัสข้อมูลและนำพิกัดปัจจุบันมาเก็บไว้ โดยข้อมูลสำคัญที่สามารถบอกพิกัดปัจจุบันได้คือ ข้อมูล RMC - Recommended minimum specific GPS/Transit data ซึ่งเป็นหนึ่งในหลายๆประเภทข้อมูลที่ถูกส่งออกมาจาก GPS ตามมาตรฐานของ NMEA RMC โดยพิกัดที่ได้เป็นแบบตัวอย่างดังนี้

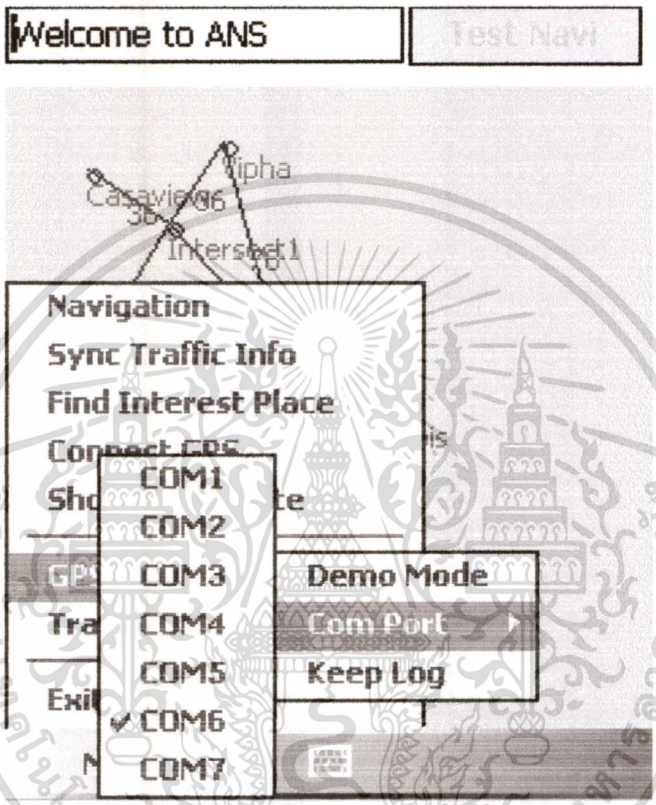
   $GPRMC,082855.000,A,1353.6250,N,10038.1164,E,0.06,253.92,230308,,,A*62

   โดยค่าที่สนใจคือ 1353.6250 N แทนด้วย Latitude = 13 Hour, 53 Minute, 6250 Second และ 10038.1164 E แทนด้วย Longitude = 100 Hour, 38 Minute, 1164 Second
3. ทำการแปลงจากค่าพิกัด latitude , longitude เป็นวินาที เพื่อเป็นค่าพิกัด y และ x บนหน้าจอของเครื่องคอมพิวเตอร์แบบพกพาตามลำดับ แต่ระบบยังไม่สามารถแปลงค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

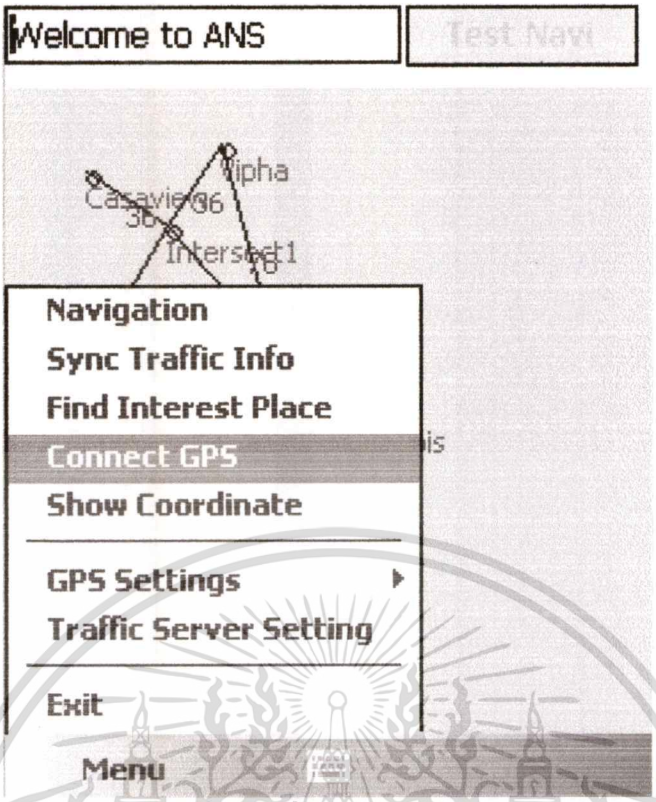
จากค่า latitude และ longitude ให้เป็นค่า y และ x ของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพาได้

4. ทำการแสดงค่าตำแหน่งปัจจุบันค่าพิกัด latitude และ longitude บนหน้าจอของเครื่องคอมพิวเตอร์แบบพกพา แต่เนื่องจากว่ายังไม่สามารถแปลงค่าของ latitude และ longitude ให้เป็นหน่วยเดียวกับแกน y และ x ได้จึงทำให้ไม่สามารถแสดงผลบนหน้าจออุปกรณ์เครื่องคอมพิวเตอร์แบบพกพาได้



รูปที่ 4.10 หน้าจอแสดงการเลือก Com Port เพื่อเชื่อมต่อ GPS Receiver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



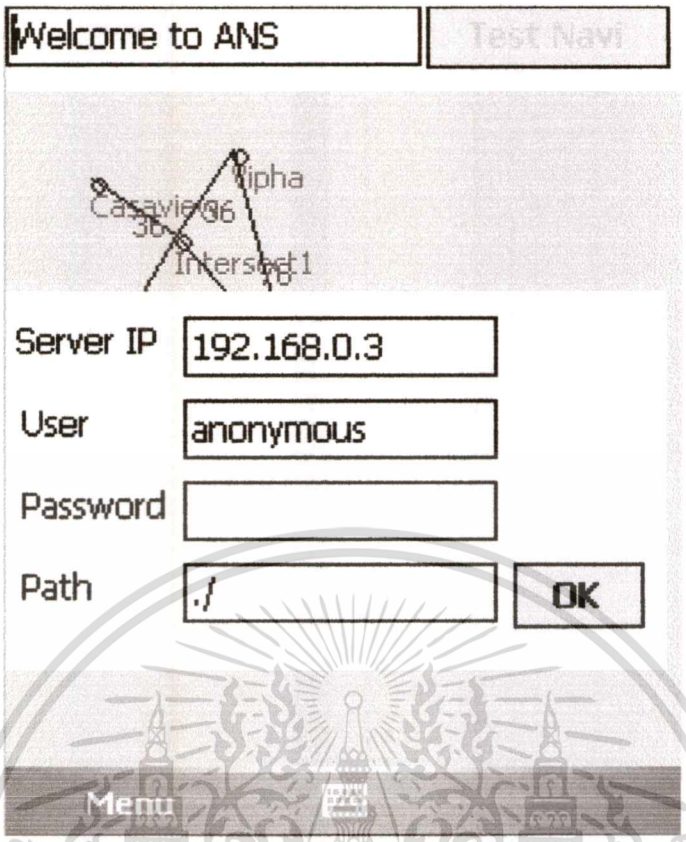
รูปที่ 4.11 หน้าจอแสดงเริ่มการเชื่อมต่อและรับข้อมูลจาก GPS Receiver

4.2.7 การพัฒนาส่วนของการติดต่อกับ Traffic Monitoring Server เพื่อรับข้อมูลการจราจร ณ เวลาปัจจุบัน

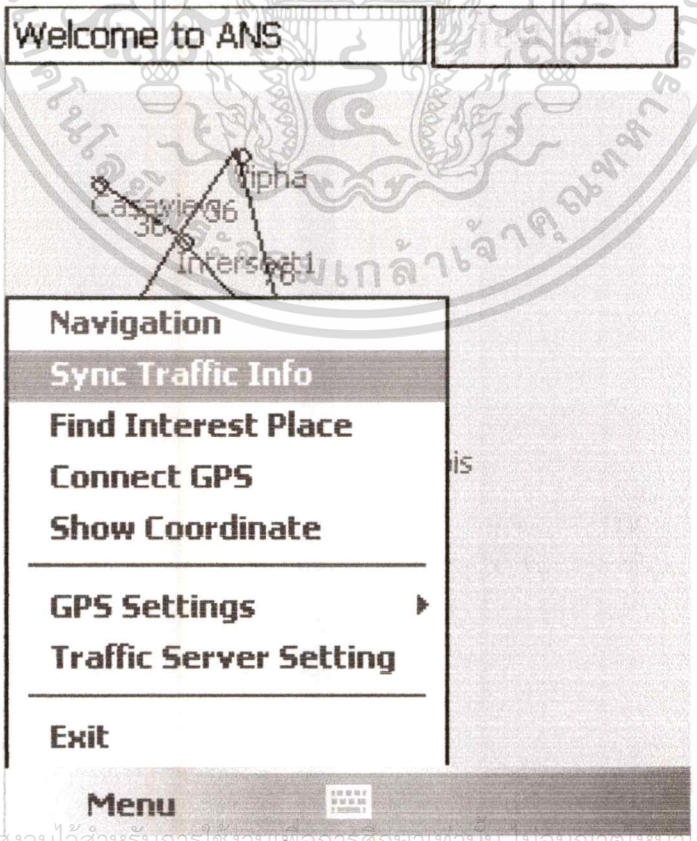
การพัฒนาในส่วนของการติดต่อกับ Traffic Monitoring Server และการ sync ข้อมูลโดยที่อุปกรณ์เครื่องคอมพิวเตอร์แบบพกพาเชื่อมต่อกับเครือข่าย ผ่านทาง GPRS Connection หรือ LAN Connection และข้อมูลการจราจรคือ เวลาที่ใช้ในการเดินทางในแต่ละถนน ซึ่งถูกจัดเก็บอยู่ใน file ชื่อ "way.txt" ซึ่งมีรูปแบบเป็นแบบเดียวกับตาราง WAY ดังที่ได้อธิบายไปแล้วในข้อ 4.2.1 โดย file "way.txt" จะถูกจัดเก็บอยู่ทั้งสองที่คือ ที่ Traffic Monitoring Server และที่อุปกรณ์คอมพิวเตอร์แบบพกพา โดยการพัฒนาการทำงานสามารถอธิบายได้ดังนี้

1. สร้าง function การติดต่อ (Connection) กับ Traffic Monitoring Server ผ่านทาง GPRS Connection หรือ LAN Connection โดยผู้ใช้ต้องกำหนด IP, User, Password และ Path สำหรับการเชื่อมต่อเพื่อ Sync ข้อมูลเสียก่อน
2. สร้าง function การรับ file (GET) "way.txt" ที่เก็บข้อมูลการจราจรในแต่ละถนนจาก Traffic Monitoring Server โดยใช้ FTP protocol และใช้ Thread ไปเรียก function นี้ เพื่อไม่ให้ไปกระทบการทำงานอื่นของ Application เพราะการรับ file อาจเกิดปัญหา Delay ขึ้นมาได้ระหว่างการ transfer file ผ่าน GPRS หรือ LAN Connection
3. เรียกใช้ function ในการนำเข้า file "way.txt" ที่เก็บข้อมูลถนนหรือทางสู่ระบบ จากข้อ 4.2.1 ข้อย่อยที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 หน้าแสดงการกำหนด Server, User, Password และ Path สำหรับการ Sync ข้อมูลกับ Traffic Monitoring Server



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

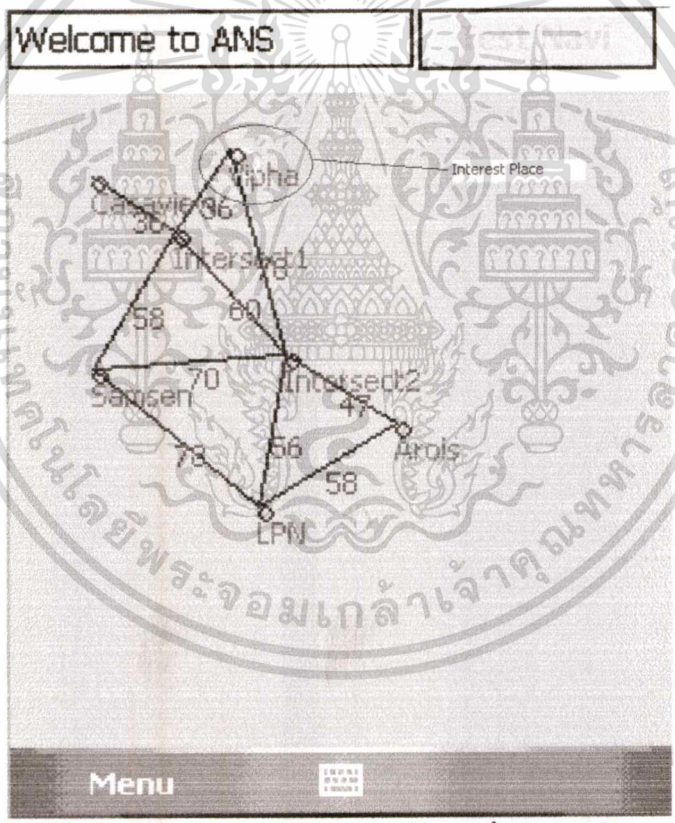
รูปที่ 4.13 หน้าแสดงเมนูส่วนเริ่มการติดต่อกับ Traffic Monitoring Server

4.2.8 การพัฒนาส่วนของการค้นหาสถานที่ที่สนใจ

การพัฒนาในส่วนของการค้นหาสถานที่ที่สนใจ สามารถอธิบายได้ดังนี้

1. สร้างหน้าจอสำหรับรับ input จากผู้ใช้เพื่อใช้ในการค้นหา
2. สร้าง function ในการค้นหาตำแหน่งของสถานที่ที่สนใจจากข้อมูล WP โดยผู้ใช้ต้องกำหนดเงื่อนไขในการค้นหาเป็นประเภทของสถานที่
3. สร้าง function ในการแสดงตำแหน่งของสถานที่ที่ผู้ใช้เลือกบนแผ่นที่

รูปที่ 4.14 ตัวอย่างหน้าจอการรับ input เพื่อค้นหาสถานที่ที่สนใจ



รูปที่ 4.15 ตัวอย่างหน้าจอการแสดงตำแหน่งสถานที่จากการค้นหาบนแผนที่

4.2.9 การพัฒนาส่วนของการติดต่อกับผู้ใช้

การพัฒนาในส่วนของกับผู้ใช้ และการระบุตำแหน่งบนแผนที่ สามารถอธิบายได้ดังนี้

1. ทำการพัฒนาส่วนติดต่อกับผู้ใช้ซึ่งประกอบด้วย
   a. ส่วนแสดงแผนที่ – โดยนำข้อมูล WP และ WAY มาสร้างเป็นจุดและเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b. เมนูหาเส้นทาง (Navigation) - ให้ผู้ใช้กำหนดจุดเริ่มต้นและจุดเป้าหมายเพื่อคำนวณเส้นทาง ระบบจะเรียกเมนูย่อยในข้อ 4.2.4

c. เมนูรับข้อมูลการจราจรอัตโนมัติ (Sync Traffic Info) – เมื่อผู้ใช้เลือกรับข้อมูลการจราจรอัตโนมัติ ระบบจะเรียกใช้งาน function การติดต่อกับ Traffic Monitoring System และการ Sync ข้อมูล ในข้อ 4.2.7

d. เมนูค้นหาสถานที่ที่สนใจ (Find Interest Place) - เมื่อผู้ใช้เลือกค้นหาสถานที่ที่สนใจ ระบบจะเรียกใช้งานเมนูย่อยการค้นหาสถานที่ที่สนใจในข้อ 4.2.8

e. เมนูกำหนดการเชื่อมต่อกับ GPS (Connect GPS) – เมื่อกำหนดการเชื่อมต่อกับ GPS ให้ไปเรียกใช้ function การเชื่อมต่อกับ GPS Receiver ผ่านทาง Bluetooth Connection ในข้อ 4.2.6

f. เมนูกำหนดให้แสดงพิกัด (Show Coordinate) – เป็นเมนูเพื่อกำหนดให้แสดงพิกัดของจุดและหมายเลขจุดแทนชื่อ

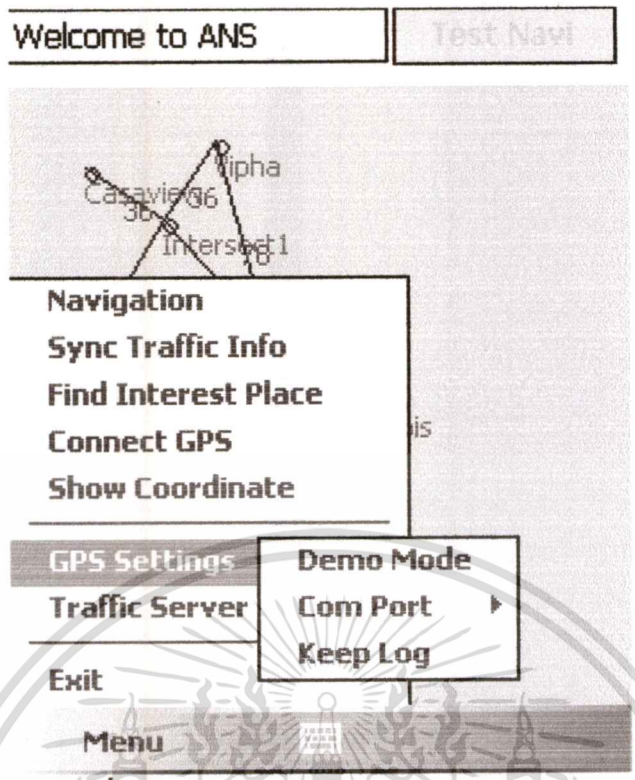
g. เมนูกำหนดการเชื่อมต่อ GPS (GPS Settings) - เป็นเมนูเพื่อกำหนดการเชื่อมต่อกับ GPS Receiver ประกอบด้วยเมนูย่อยดังนี้

    1. Com Port – เป็นเมนูให้เลือก Com Port ที่ต้องการเชื่อมต่อ โดยต้องเป็น Com Port ที่เชื่อมต่อกับ GPS
    2. Demo Mode – เป็นเมนูในโหมด Demo ให้ระบบสุ่มตำแหน่งพิกัดขึ้นมาแทนการรับค่าจาก GPS
    3. Keep Log – เป็นเมนูให้มีการเก็บ Log ของพิกัดที่รับจาก GPS Receiver ไว้ใน file GpsTracer.log ซึ่งจะถูกสร้างอยู่ที่เดียวกับ Application Program

h. เมนูกำหนดการเชื่อมต่อกับ Traffic Monitoring Server (Traffic Server Setting) - เป็นเมนูเพื่อกำหนดการเชื่อมต่อกับ Traffic Monitoring Server ซึ่งระบบจะเรียกใช้งานเมนูย่อยในข้อ 4.2.7 ข้อย่อย 3

i. เมนูออกจากโปรแกรม (Exit) – จะทำการปิด Connection ทั้งหมดทั้งที่ติดต่อกับ GPS Receiver และกับ Traffic Monitoring Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ตัวอย่างหน้าจอแสดงเมนูของระบบ

4.2.10 ทดสอบความถูกต้องของระบบ

ทดสอบความถูกต้องของระบบ สามารถอธิบายได้ดังนี้

1. ทดสอบการโหลดข้อมูลแผนที่และการแสดงผล

โดยข้อมูลของ “wp.txt” และ “way.txt” มีค่าดังนี้

“wp.txt” ใช้ข้อมูลทดสอบดังนี้

30:60:0:Casaview:Village

60:80:1:Intersect1:Intersection

80:50:2:Vipha:Hostpital

30:130:3:Samsen:School

100:125:4:Intersect2:Intersection

140:150:5:Arois:Resteraunt

90:180:6:LPN:Building

“way.txt” ใช้ข้อมูลทดสอบดังนี้

0:1:36:40:RoadName0

1:2:36:50:RoadName1

1:3:58:10:RoadName2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1:4:60:90:RoadName3

2:4:78:70:RoadName4

3:4:70:90:RoadName5

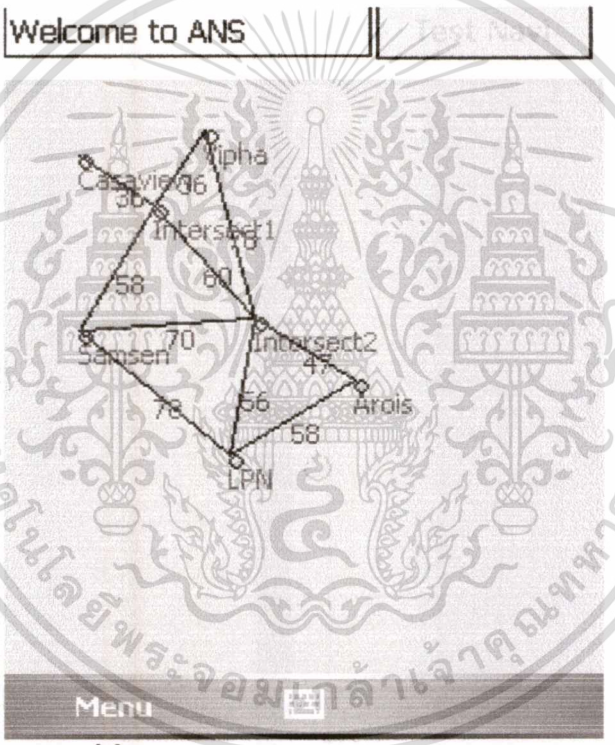
4:5:47:50:RoadName6

3:6:78:10:RoadName7

4:6:56:90:RoadName8

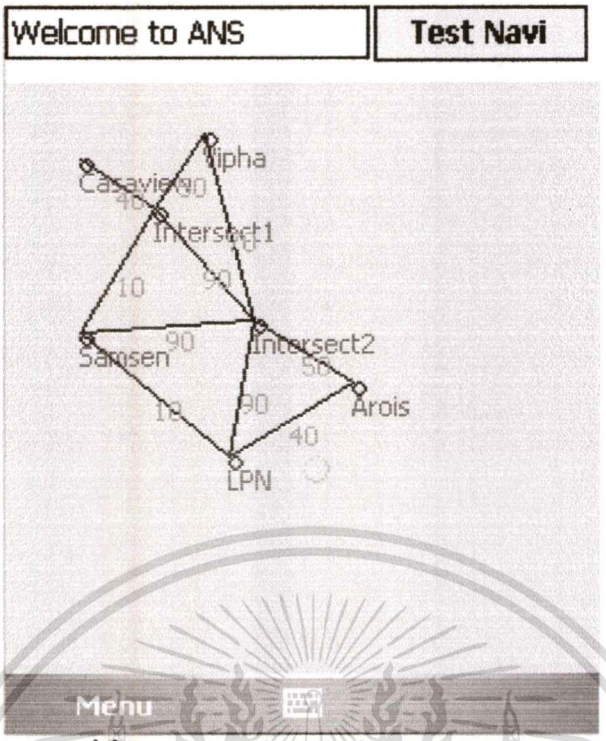
5:6:58:40:RoadName9

ระบบแสดงข้อมูลแผนที่บนหน้าจอคอมพิวเตอร์แบบพกพาได้ผลลัพธ์ดังนี้



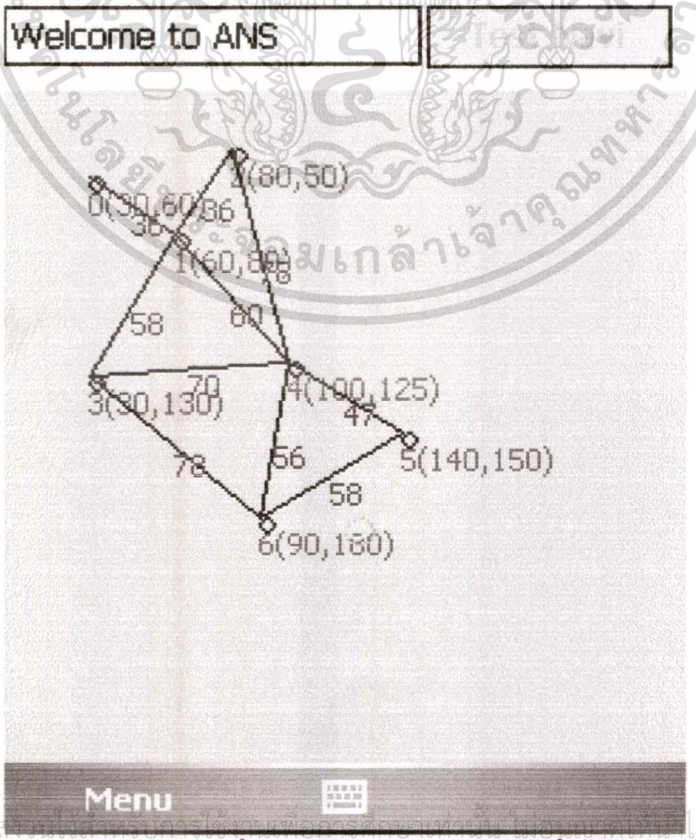
รูปที่ 4.17 หน้าจอแสดงแผนที่ที่ได้จากการโหลดข้อมูลมาจาก file “wp.txt” และ “way.txt” โดยแสดงเป็นระยะทางระหว่างจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 หน้าจอแสดงแผนที่ที่ได้จากการโหลดข้อมูลมาจาก file "wp.txt" และ "way.txt" โดยแสดงเป็นเวลาในการเดินทางระหว่างจุด

และทดลองให้ระบบแสดงพิกัด x,y เพื่อเทียบดูกับพิกัดที่ใส่ลงไปใน file ได้ผลดังรูปต่อไปนี้



รูปที่ 4.19 หน้าจอแสดงแผนที่และพิกัด x,y ของจุดสนใจที่ได้จาก file "wp.txt"

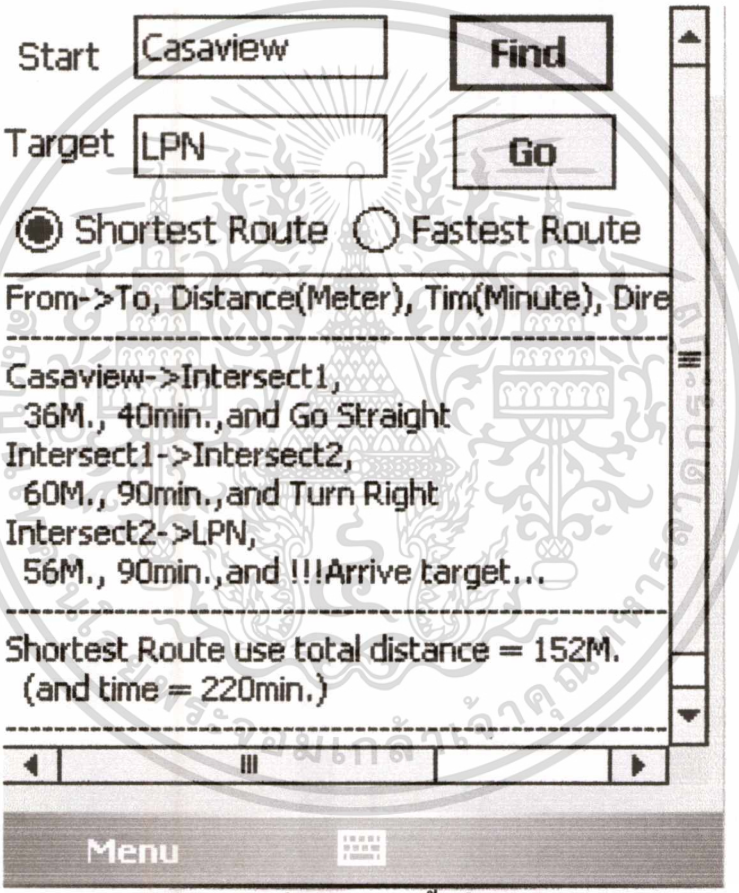
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบ – ระบบสามารถข้อมูลของจุดและถนนซึ่งได้จาก “wp.txt” และ “way.txt” ได้อย่างถูกต้อง

2. ทดสอบการค้นหาเส้นทาง

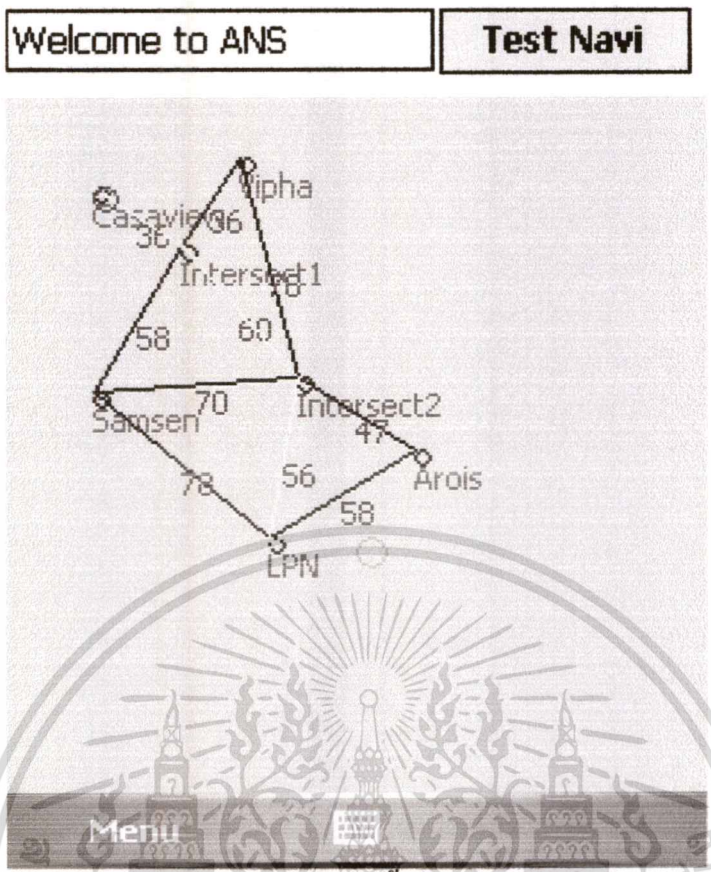
จากข้อมูล “wp.txt” และ “way.txt” จากข้อ 1 ทำการทดสอบการหาเส้นทางได้ผลดังนี้ คือ

ใส่ชื่อจุดเริ่มต้นเป็น Casaview ที่ช่อง Start และจุดที่ต้องการไป คือ LPN ที่ช่อง Target แล้ว click ที่ปุ่ม “Find” ได้ผลดังนี้

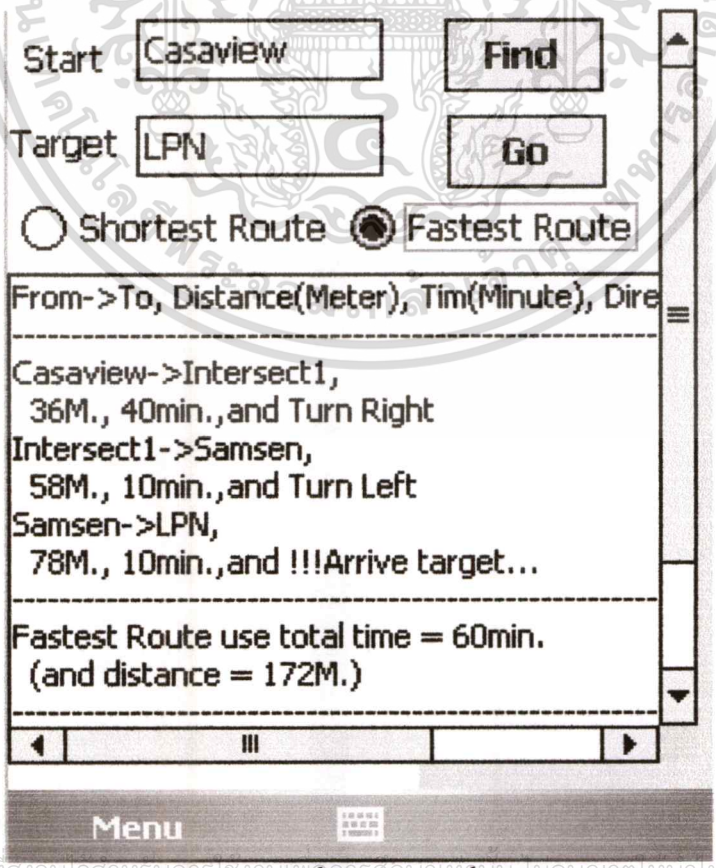


รูปที่ 4.20 หน้าจอแสดงเส้นทางที่มีระยะทางสั้นที่สุดจาก Casaview ไป LPN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

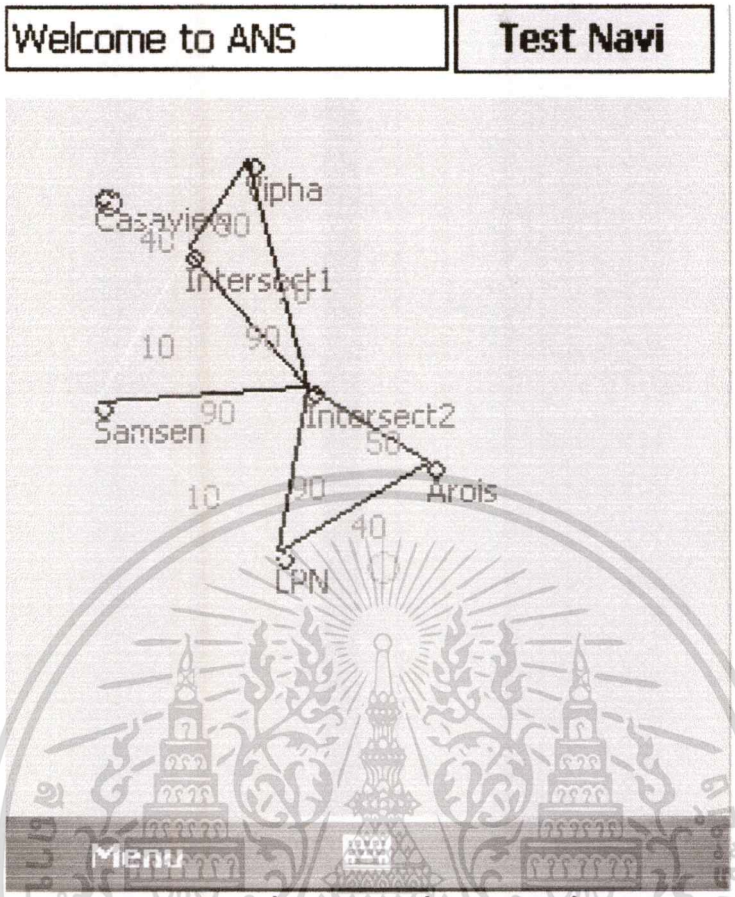


รูปที่ 4.21 หน้าจอแสดงเส้นทางที่มีระยะทางสั้นที่สุดบนแผนที่จาก Casaview ไป LPN



รูปที่ 4.22 หน้าจอแสดงเส้นทางที่ใช้เวลาน้อยที่สุดจาก Casaview ไป LPN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



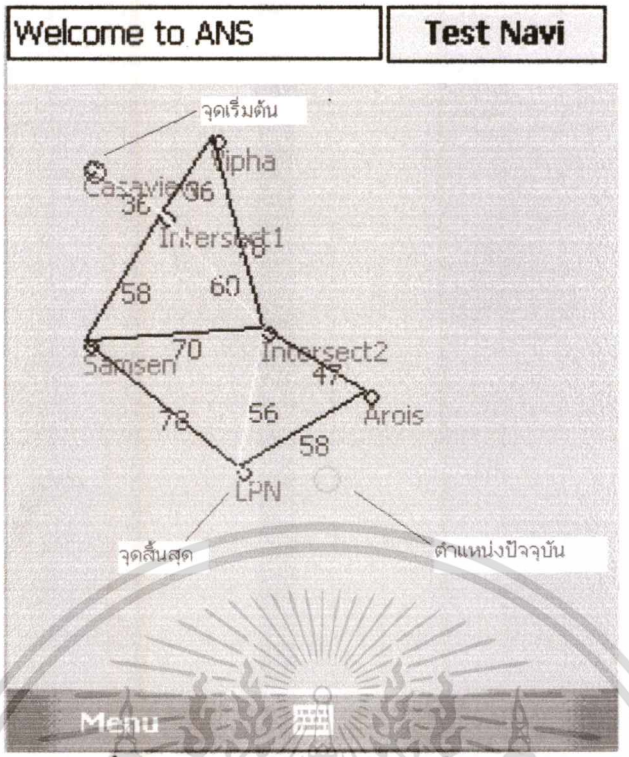
รูปที่ 4.23 หน้าจอแสดงเส้นทางที่ใช้เวลาน้อยที่สุดบนแผนที่จาก Casaview ไป LPN

ผลการทดสอบ – ระบบสามารถคำนวณเส้นทางที่มีระยะทางสั้นที่สุด และใช้เวลาน้อยที่สุดได้อย่างถูกต้อง

3. ทดสอบระบบนำทางอัตโนมัติ

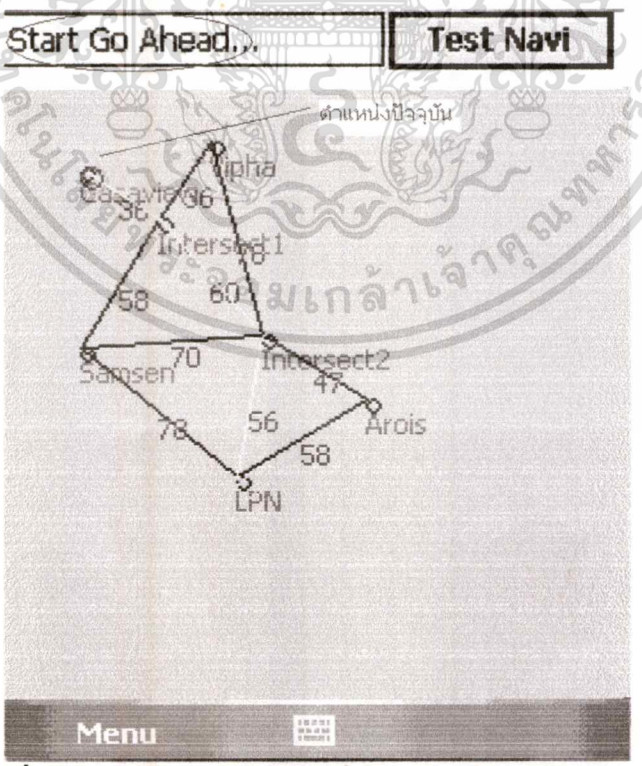
ทดสอบโดยใช้เส้นทางที่มีระยะทางสั้นที่สุดจาก Casaview ไป LPN ให้สังเกตตำแหน่งปัจจุบันซึ่งแทนด้วยวงกลมสีชมพูซึ่งจะเคลื่อนที่ไปตามจุดบนเส้นทางที่เลือกไว้ และข้อความนำทาง ณ จุดที่ตำแหน่งปัจจุบันอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 หน้าจอแสดงเส้นทางที่ผู้ใช้เลือก

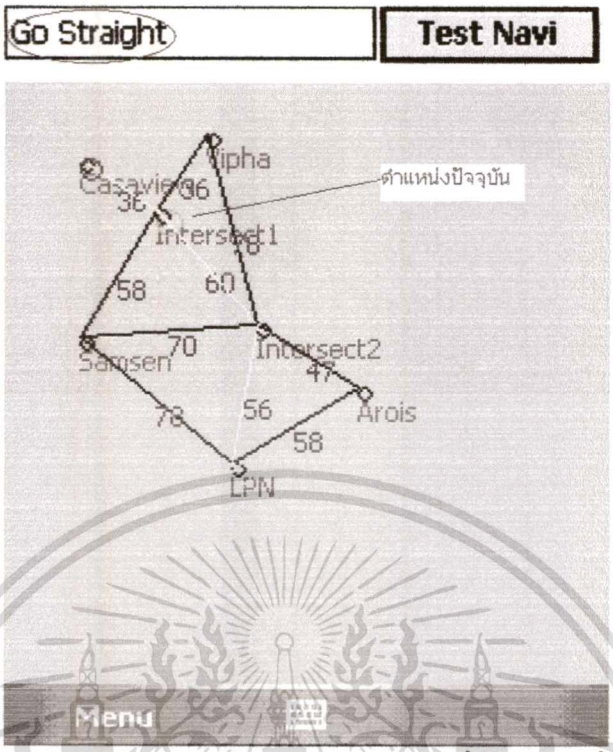
Click ปุ่ม "Test Navi" เพื่อเข้าสู่ระบบนำทางอัตโนมัติ ได้ผลดังนี้



รูปที่ 4.25 หน้าจอแสดงผลการเริ่มต้นโหมดนำทางอัตโนมัติ

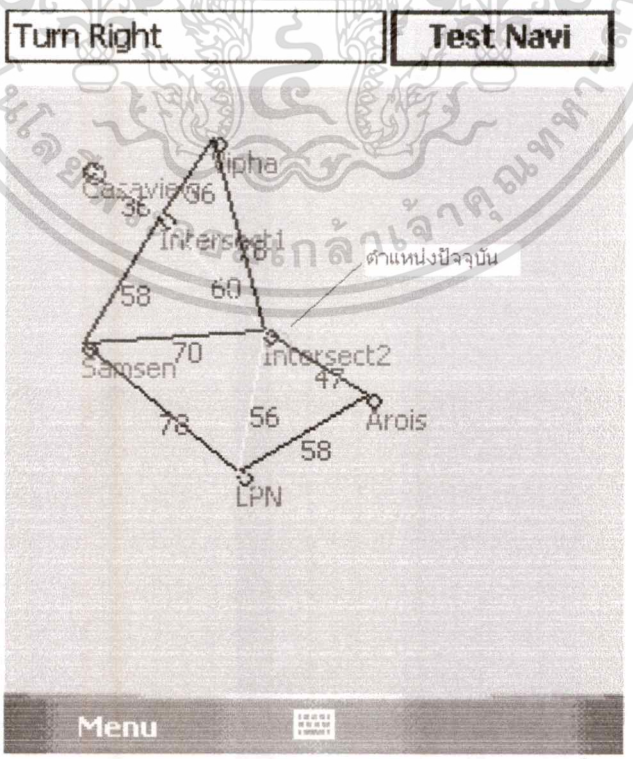
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Click ปุ่ม "Test Navi" ไปสู่โหนดถัดไป ได้ผลดังนี้



รูปที่ 4.26 หน้าจอแสดงผลข้อความนำทาง ณ จุดที่ถัดจากจุดเริ่มต้น

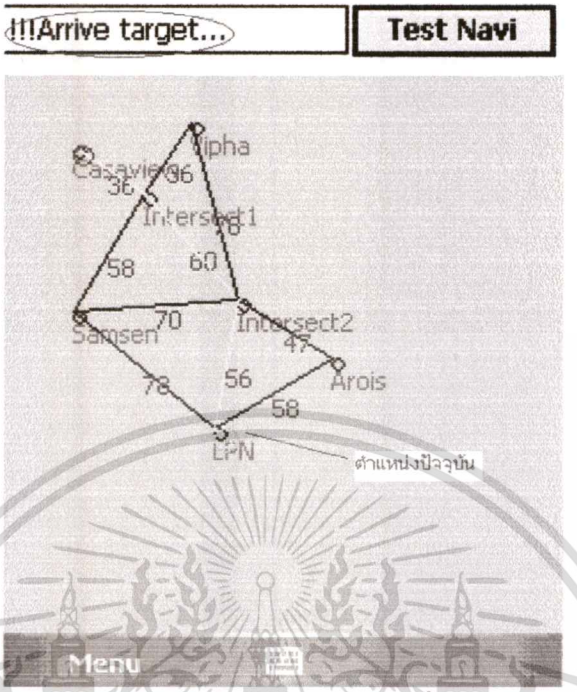
Click ปุ่ม "Test Navi" ไปสู่โหนดถัดไป ได้ผลดังนี้



รูปที่ 4.27 หน้าจอแสดงผลข้อความนำทาง ณ จุดถัดไปจากจุดเริ่มต้น 2 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Click ปุ่ม "Test Navi" แล้วถึงโหนดเป้าหมาย ได้ผลดังนี้

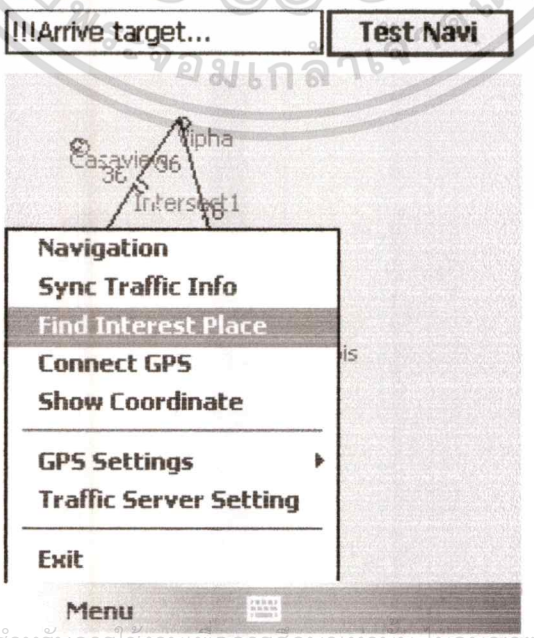


รูปที่ 4.28 หน้าจอแสดงผลข้อความนำทางเมื่อถึงปลายทาง

ผลการทดสอบ – ระบบสามารถแสดงข้อความนำทางเมื่อถึงจุดที่ต้องตัดสินใจได้อย่างถูกต้อง

4. ทดสอบระบบค้นหาสถานที่สนใจ

โดยเข้าสู่เมนูการค้นหาแล้วเลือกประเภทสถานที่ที่ต้องการค้นหาแล้วกดปุ่ม "Find" จะมีวงกลมสีขาวล้อมรอบสถานที่ที่ต้องการค้นหา



รูปที่ 4.29 หน้าจอแสดงการเข้าสู่เมนูค้นหาสถานที่ที่สนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการทดลองค้นหาโรงพยาบาลโดยเลือกประเภทสถานที่ดังนี้

รูปที่ 4.30 หน้าจอแสดงการเลือกประเภทสถานที่ที่สนใจ

ผลการค้นหาโรงพยาบาลเป็นดังนี้

รูปที่ 4.31 หน้าจอแสดงผลสถานที่ตามประเภทที่ผู้ใช้เลือก

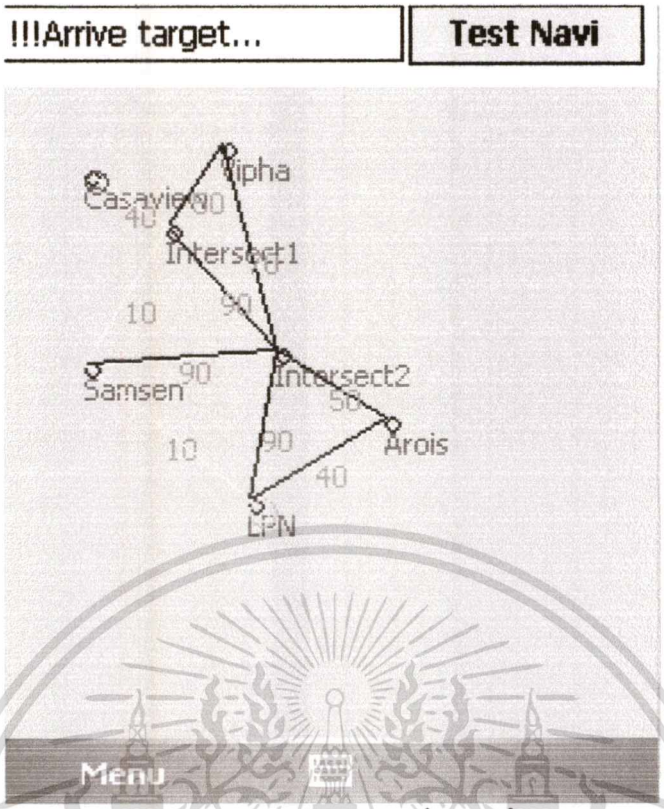
ผลการทดสอบ – ระบบสามารถค้นหาและแสดงตำแหน่งของสถานที่ที่ต้องการค้นหาได้อย่างถูกต้อง

5. ทดสอบการ Sync ข้อมูลกับ Traffic Monitoring Server

ทำการทดลองโดยการตั้งเครื่อง Server เป็น Traffic Monitoring Server ไว้ในวง Wireless LAN เดียวกันกับอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพา

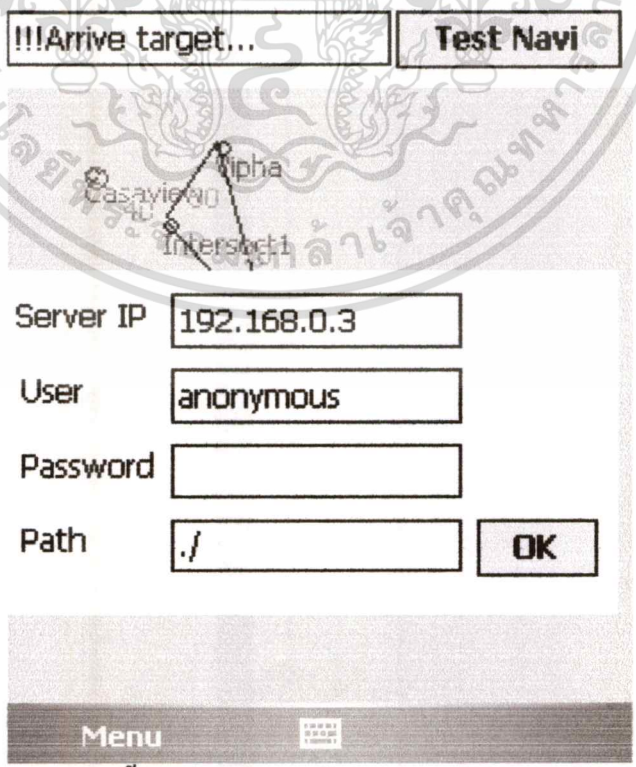
แสดงเวลาที่ใช้ในแต่ละถนนก่อนการทำลองเปลี่ยนค่าเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 หน้าจอแสดงเวลาของแต่ละถนนก่อนเปลี่ยนค่าที่ Traffic Monitoring Server

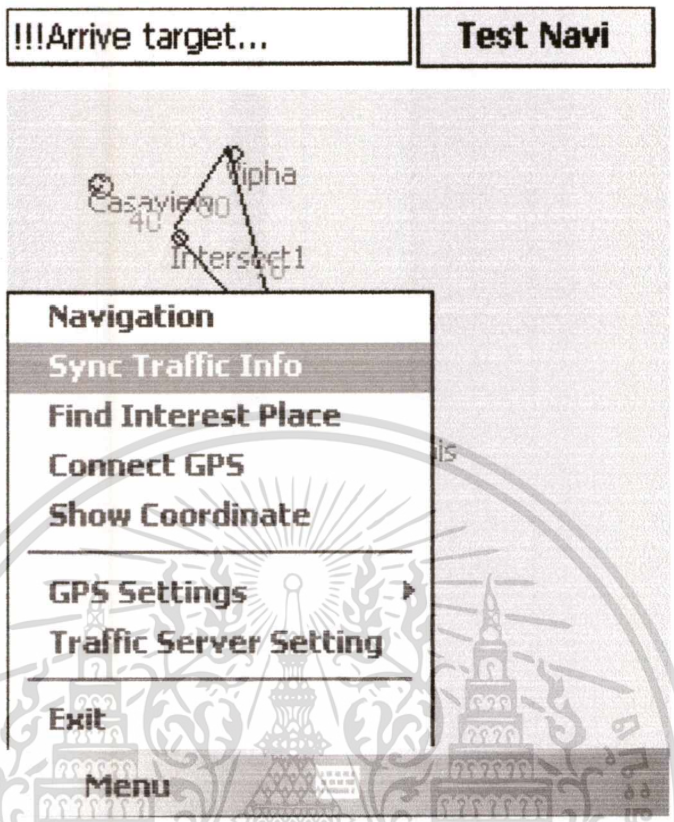
ทำการตั้งค่าให้กับการเชื่อมต่อกับ Traffic Monitoring Server ดังนี้



รูปที่ 4.33 หน้าจอแสดงการตั้งค่าสำหรับการเชื่อมต่อกับ Traffic Monitoring Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วทำการเปิดให้มีการ Sync ข้อมูลกับ Traffic Monitoring Server ดังนี้



รูปที่ 4.34 หน้าจอแสดงการเปิดให้มีการ Sync ข้อมูลกับ Traffic Monitoring Server

ทดลองเปลี่ยนค่าเวลาในการเดินทางบนถนน Interest1->Samsen และ Samsen->LPN ใน file "way.txt" ดังนี้

"way.txt" ใช้ข้อมูลทดสอบดังนี้

0:1:36:40:RoadName0

1:2:36:50:RoadName1

1:3:58:100:RoadName2

1:4:60:90:RoadName3

2:4:78:70:RoadName4

3:4:70:90:RoadName5

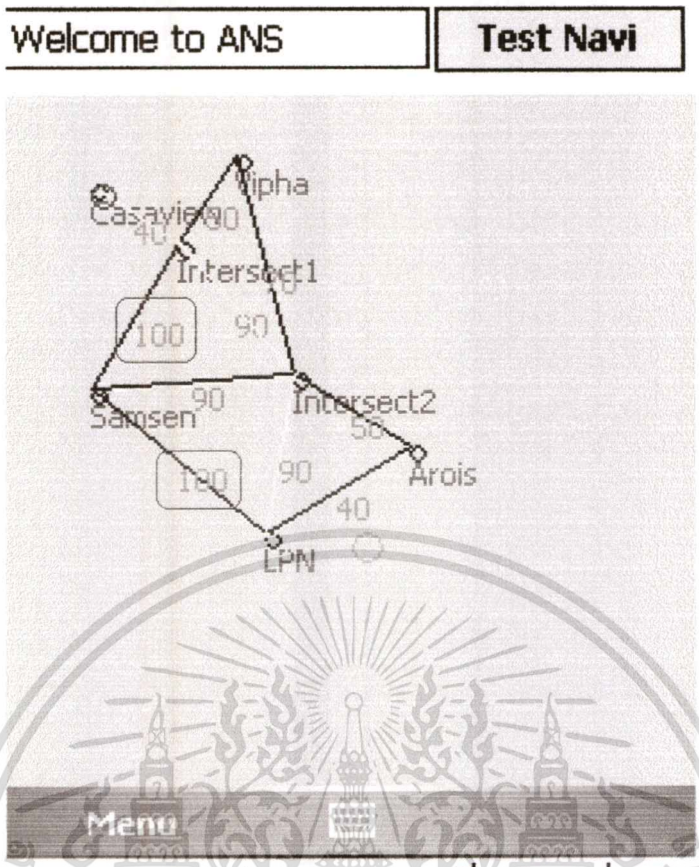
4:5:47:50:RoadName6

3:6:78:100:RoadName7

4:6:56:90:RoadName8

5:6:58:40:RoadName9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.35 หน้าจอแสดงเวลาบนถนนแต่ละถนนหลังจากเปลี่ยนค่าเวลาที่ Traffic Monitoring Server แล้ว

ผลการทดสอบ – ระบบสามารถ Sync ข้อมูลกับ Traffic Monitoring Server ได้อย่างถูกต้อง โดยเมื่อมีการทดลองแก้ไขค่าเวลาของถนนบน Traffic Monitoring Server แล้วรอประมาณ 1-2 นาที ระบบสามารถเปลี่ยนแปลงค่าเวลาและสามารถแสดงค่าบนอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพาได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการพัฒนา

จากการพัฒนาระบบนำทางอัตโนมัติ ได้ว่าระบบสามารถนำทางไปยังเส้นทางที่มีระยะทางสั้นที่สุดหรือระยะเวลาน้อยที่สุดได้โดยผู้ใช้หรือผู้ขับขี่เป็นผู้กำหนดจุดเริ่มต้นและจุดสิ้นสุด และระบบจะค้นหาข้อมูลเส้นทางที่ผู้ใช้ต้องการได้ ซึ่งเส้นทางดังกล่าวได้มาโดยการนำข้อมูลจุดที่สนใจและข้อมูลถนนไปคำนวณหาเส้นทางด้วยไดสตราอัลกอริทึม แบ่งเป็นการคำนวณหาเส้นทางที่มีระยะทางสั้นที่สุดและเส้นทางที่ใช้ระยะเวลาน้อยที่สุด เมื่อเริ่มระบบนำทางโดยผู้ใช้เลือกแล้วว่าต้องการเดินทางไปตามเส้นทางแบบใด เมื่อผู้ใช้เดินทางใกล้ถึงจุดทางร่วมหรือทางแยกระบบจะแสดงข้อความนำทาง (Navigate Message) เพื่อเตือนให้ผู้ใช้ทราบว่าแยกต่อไปข้างหน้าจะต้องเลี้ยวไปทางด้านไหน นอกจากนี้ระบบยังสามารถรับข้อมูลระยะเวลาการเดินทางของแต่ละถนนจาก Traffic Monitoring Server ได้ โดยระบบต้องทำการคำนวณเส้นทางที่มีระยะเวลาน้อยที่สุดไปยังจุดที่สุดใจใหม่ทุกครั้งที่มีการ Update ข้อมูลระยะเวลาการเดินทางเพื่อให้ได้เส้นทางใหม่ที่ใช้ระยะเวลาน้อยกว่าเดิม

5.2 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคจากการพัฒนาโครงการมีดังนี้

1. พิกัด latitude และ longitude ไม่สามารถแสดงบนหน้าจอของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพาได้ เนื่องจากว่าหน้าจอของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพามีขนาด 240 * 250 (x * y) ซึ่งเป็นจำนวนเต็ม แต่พิกัด latitude และ longitude ไม่ได้อยู่ในรูปแบบจำนวนเต็ม

2. การแสดงผลของจุดที่สนใจและถนน หรือการแสดงผลต่างๆ บนแผนที่ การพัฒนาต้องทำให้ Application สามารถวาดกราฟ จุด หรือเส้นตรงได้ ซึ่งต้องมีความรู้เกี่ยวกับการใช้ function หรือ API ที่เกี่ยวกับ Computer Graphic ทีอยู่บน .NET Compact Framework ด้วย ซึ่งผู้พัฒนาไม่ได้ให้ความสำคัญเท่าที่ควรในขั้นตอนศึกษาทฤษฎีที่เกี่ยวข้อง ทำให้ในขั้นตอนการพัฒนาต้องใช้เวลาในการศึกษาและทำความเข้าใจเป็นเวลานาน

3. ข้อมูลพิกัดที่ได้รับจาก GPS บ้างครั้งมีความคลาดเคลื่อน กรณีนี้อาจเกิดจากมีเมฆหรือตึกมาบดบังการรับสัญญาณจากดาวเทียม ทำให้การคำนวณพิกัดอาจคาดเคลื่อนได้ แนวทางแก้ไขคือ ศึกษาวิธีการลดความคลาดเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อทดลองใช้กับแผนที่ ที่มีจุดที่สนใจและถนนจำนวนมากขึ้น เมื่อใช้ระบบแบบการทำงานของโปรแกรมได้ Response time ที่ช้าลง แนวทางในการแก้ปัญหาคือต้องหาอัลกอริทึมในการหาระยะทางสั้นสุดที่ใช้เวลาในการคำนวณน้อยกว่าไดสตรา

5.3 ข้อเสนอแนะ

ข้อเสนอแนะจากการพัฒนาระบบนำทางอัตโนมัติ

1. แนวทางในการหาวิธีแสดงผลจุดที่มีค่า latitude และ longitude มากกว่า 250 และ 240 ตามลำดับ อาจทำได้โดยการใช้อัตราส่วนในการย่อค่าของ latitude และ longitude ให้อยู่ในช่วง 0-250 และ 0-240 ซึ่งสามารถแสดงบนหน้าจออุปกรณ์เครื่องคอมพิวเตอร์แบบพกพาได้

2. การได้มาซึ่งจุดสนใจหรือถนน ได้จากการสำรวจและกำหนดจุดเองโดยผู้พัฒนาในตอนต้น ซึ่งมีข้อจำกัดในการเพิ่มหรือขยายพื้นที่ที่ต้องการให้สามารถใช้ระบบนำทางได้ หากระบบมีการพัฒนาให้สามารถนำเข้าข้อมูลที่เป็นมาตรฐานเช่น KML ของ Google หรือ Sharp File ของ ESRI ก็จะทำให้ระบบสามารถครอบคลุมพื้นที่ได้กว้างกว่านี้ นอกจากนี้ถ้าระบบสามารถแสดงภาพถ่ายดาวเทียมเป็น background ในตำแหน่งที่เคลื่อนที่ไป จะทำให้การสื่อสารระหว่างผู้ใช้ชัดเจนขึ้น

3. การพัฒนาควรมีการศึกษาเรื่อง Computer Graphic และ function ในการทำงานด้าน Graphic บน .NET Compact Framework เพิ่มเติมด้วยก่อนการพัฒนา เพราะจากการที่ผู้พัฒนาประสบปัญหาต้องใช้เวลาในการศึกษาเป็นเวลานานเมื่อเข้าสู่ขั้นตอนการพัฒนาแล้ว

4. การปรับปรุงความแม่นยำของเครื่องรับ GPS ซึ่งอาจมีความคลาดเคลื่อนได้ ควรศึกษาเรื่องการลดความคลาดเคลื่อนจากทฤษฎี DOP (Dilution of Precision)

5. เมื่อมีจำนวนจุดที่สนใจและถนนเพิ่มมากขึ้นการคำนวณด้วยไดสตราอัลกอริทึม มี Response Time ช้าลง การพัฒนาต่อไปอาจต้องปรับปรุงไดสตราอัลกอริทึมหรือใช้อัลกอริทึมอื่นแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2548. **คัมภีร์วิเคราะห์และออกแบบระบบ**. พิมพ์ครั้งที่ 4. กรุงเทพฯ : สำนักพิมพ์ เคทีพี.

พิชญ จุลศิริ และคณะ. 2549. **ครบเครื่องเรื่อง GPS**. พิมพ์ครั้งที่ 1. กรุงเทพฯ:สมาร์ทโฟน.

Dale DePriest. 2008. **NMEA data** [Online]. Available:http://www.gpsinformation.org/dale/nmea.htm

Dijkstra's algorithm [Online]. Available:http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

Global Positioning System [Online]. Available:http://en.wikipedia.org/wiki/GPS

Satzinger,Jackson and Burd. 2004. **System Analysis & Design In A Changing World**. USA:Tomson Course Technology

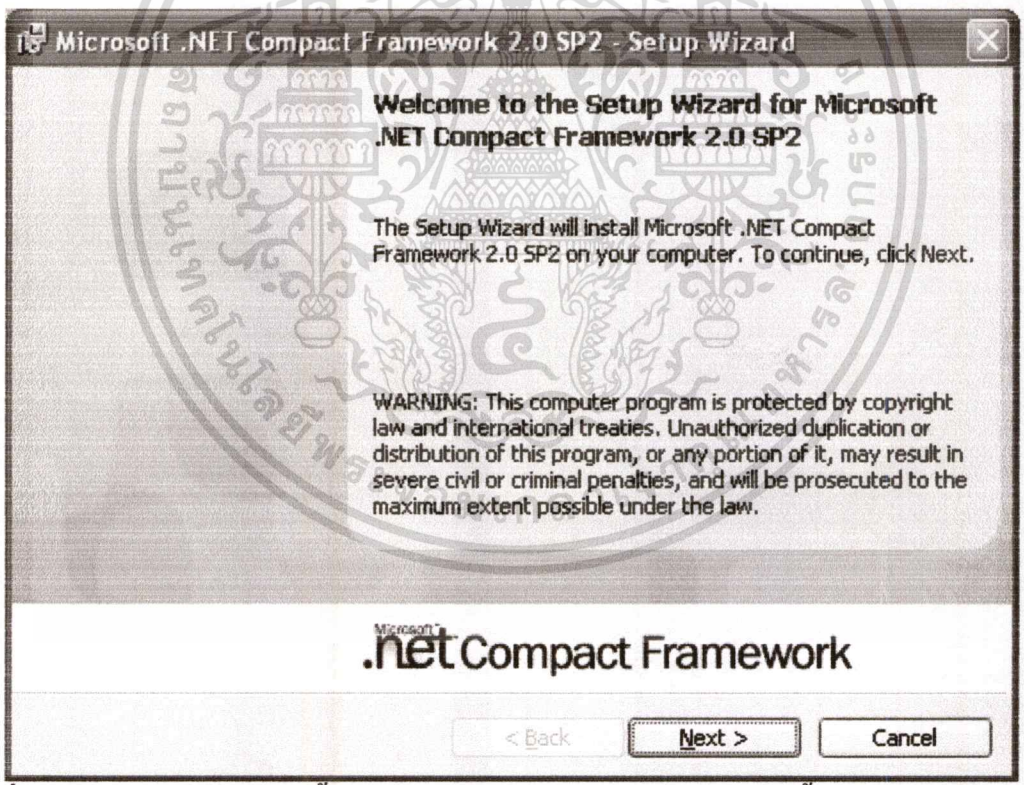
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คู่มือติดตั้งระบบนำทางอัตโนมัติ

การติดตั้งระบบนำทางอัตโนมัติมีขั้นตอนดังนี้

1. ติดตั้ง .NET Compact Framework 2.0 บนอุปกรณ์คอมพิวเตอร์แบบพกพา สามารถ download ได้ที่ http://www.microsoft.com/downloads/details.aspx?FamilyID=aea55f2f-07b5-4a8c-8a44-b4e1b196d5c0&displaylang=en (สำหรับการติดตั้งผ่าน active sync ของเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ) หรือ http://www.live-share.com/files/214189/NETCFv2.wm.armv4i.cab.html (สำหรับการติดตั้งบนอุปกรณ์คอมพิวเตอร์แบบพกพาโดยตรง)



รูปที่ ก-1 หน้าจอแสดงการติดตั้ง .NET Compact Framework 2.0 แบบติดตั้งผ่าน active sync ของเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ

2. Copy file โปรแกรมไปเก็บไว้บนเครื่องคอมพิวเตอร์แบบพกพา โดย file โปรแกรมคือ ANS.exe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการสร้าง file wp.txt และ way.txt บน root directory ของอุปกรณ์เครื่องคอมพิวเตอร์แบบพกพา

4. ทำการกำหนดตำแหน่งของจุดที่สนใจลงบนหน้าจอของอุปกรณ์คอมพิวเตอร์แบบพกพา ซึ่งมีขนาด x * y = 240 * 250 โดยการเพิ่มข้อมูลลงใน file wp.txt ดังนี้

{พิกัด x}:{พิกัด y}:{หมายเลขจุดสนใจ}:{ชื่อ}:{ประเภทจุดสนใจ}

ตัวอย่าง เช่น

30:60:0:Casaview:Village

60:80:1:Intersect1:Intersection

80:50:2:Vipha:Hostpital

30:130:3:Samsen:School

100:125:4:Intersect2:Intersection

140:150:5:Arois:Resteraunt

90:180:6:LPN:Building

โดยประเภทของจุดสนใจที่เป็นไปได้คือ Village, Intersection, Hostpital, School, Resteraunt และ Building

5. ทำการกำหนดทางหรือถนนโดยการกำหนดจุดกำเนิดของทางและจุดปลายของทาง โดยการเพิ่มข้อมูลลงใน file way.txt ดังนี้

{หมายเลขจุดสนใจที่เป็นจุดกำเนิดของทาง}:{หมายเลขจุดสนใจที่เป็นจุดปลายของทาง}:{ระยะทาง}:{เวลาที่ใช้ในการเดินทาง}:{ชื่อทาง}

ตัวอย่าง เช่น

0:1:36:40:RoadName0

1:2:36:50:RoadName1

1:3:58:100:RoadName2

1:4:60:90:RoadName3

2:4:78:70:RoadName4

3:4:70:90:RoadName5

4:5:47:50:RoadName6

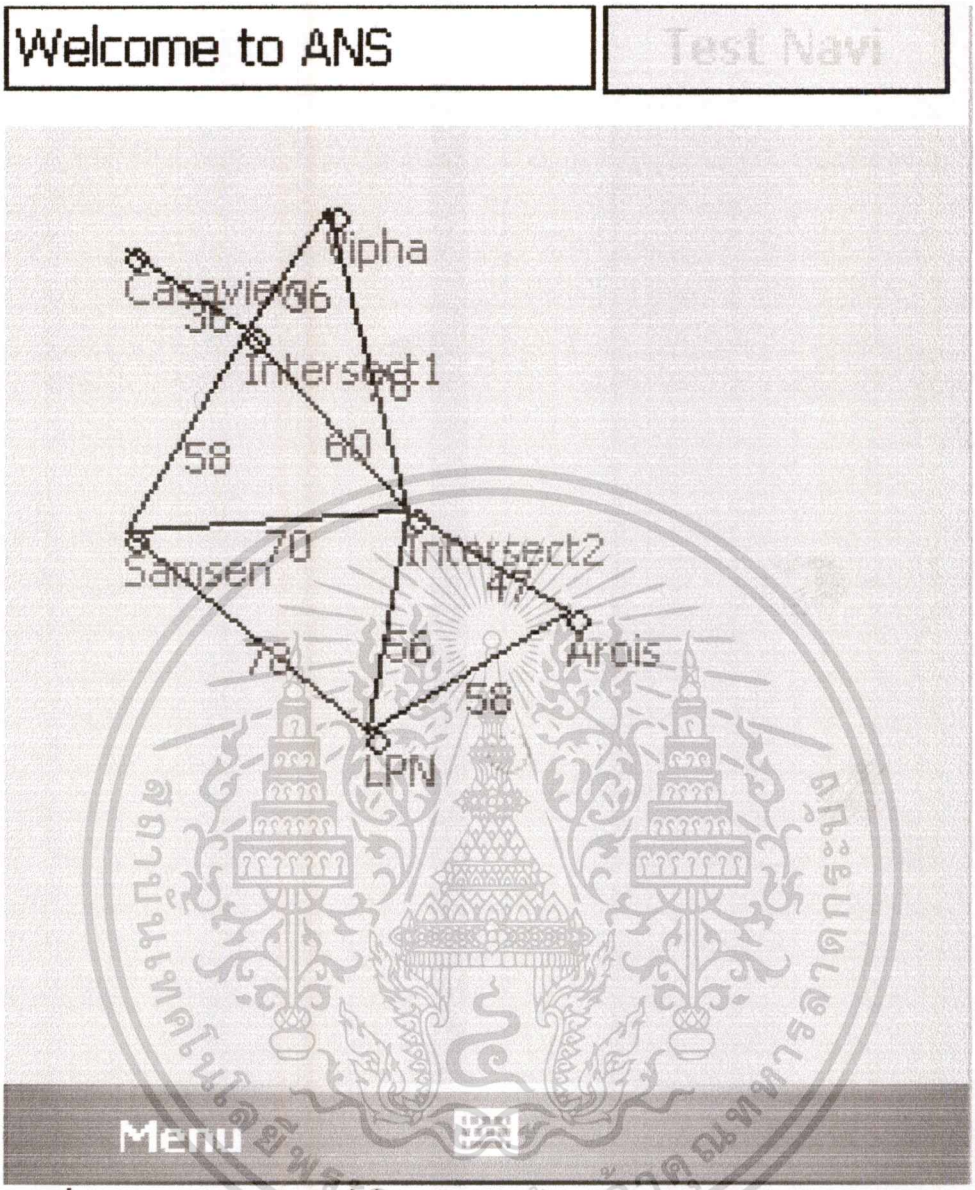
3:6:78:100:RoadName7

4:6:56:90:RoadName8

5:6:58:40:RoadName9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ทดสอบเปิดโปรแกรม จะมีแผนที่แสดงขึ้นมาให้เห็น



รูปที่ ก-2 หน้าจอแสดงแผนที่เมื่อเปิดโปรแกรมบนอุปกรณ์คอมพิวเตอร์แบบพกพา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

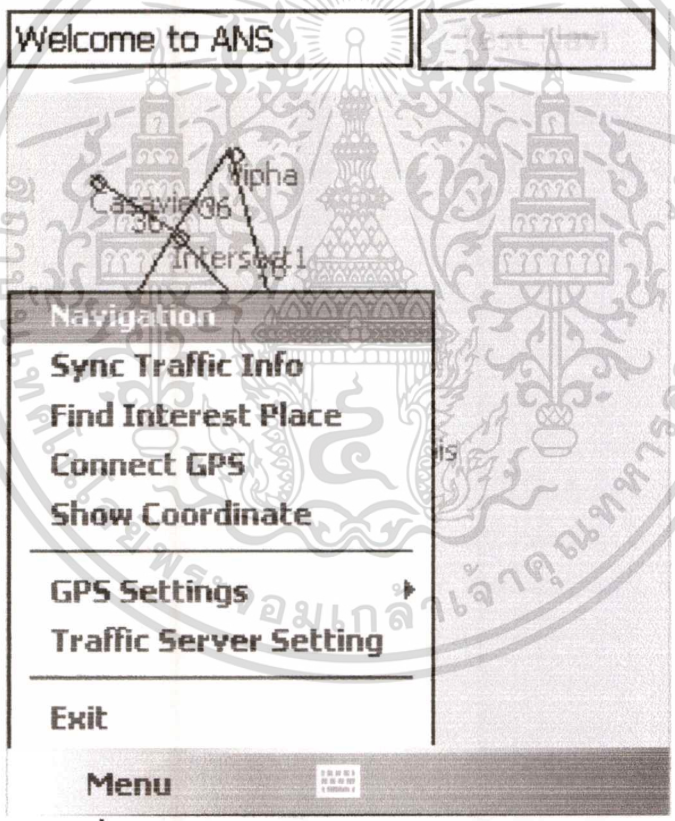
คู่มือการใช้งานระบบนำทางอัตโนมัติ

การใช้งานระบบระบบนำทางอัตโนมัติมีรายละเอียดการใช้งานดังนี้

วิธีการค้นหาเส้นทางและการทดสอบโหมดนำทางอัตโนมัติ

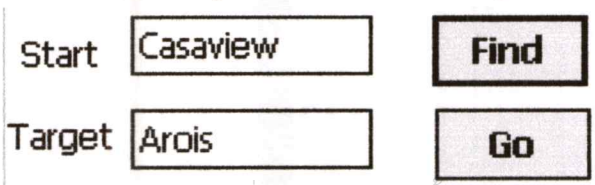
ผู้ใช้สามารถค้นหาเส้นทางและเลือกเส้นทางที่ต้องการเดินทางไปได้ และสามารถทดสอบโหมดนำทางอัตโนมัติได้ โดยทำตามขั้นตอนดังต่อไปนี้

1. เปิดโปรแกรมแล้วเลือก Menu->Navigation



รูปที่ ข-1 หน้าจอแสดงการเข้าสู่เมนูค้นหาเส้นทาง

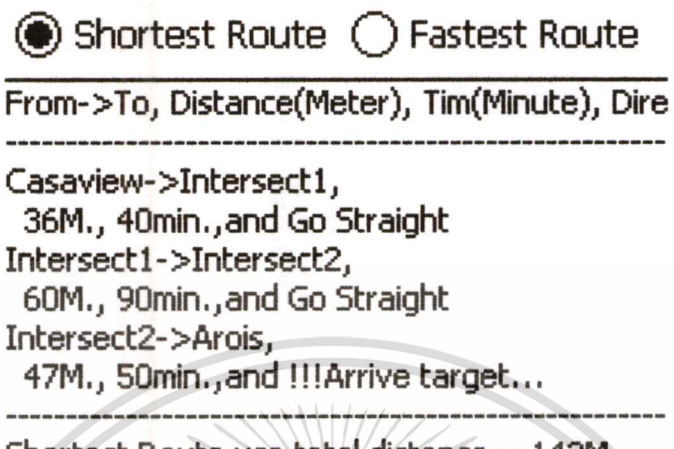
2. ใส่ชื่อจุดเริ่มต้นที่ช่อง Start และจุดที่ต้องการไปที่ช่อง Target แล้ว click ที่ปุ่ม "Find"



รูปที่ ข-2 หน้าจอแสดงการกำหนดจุดเริ่มต้นและจุดปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จะมีเส้นทางแสดงให้เลือก 2 เส้นทางคือ เส้นทางที่มีระยะทางสั้นที่สุด (เลือก "Shortest Route") และเส้นทางที่ใช้เวลาน้อยที่สุดหรือเร็วที่สุด (เลือก "Fastest Route")



รูปที่ ข-3 หน้าจอแสดงเส้นทางที่มีระยะทางสั้นที่สุด

จากรูปที่ ข-3 เส้นทางประกอบด้วยการเดินทางผ่านจุด 4 จุด และผ่านทางหรือถนน 3 ทาง คือ

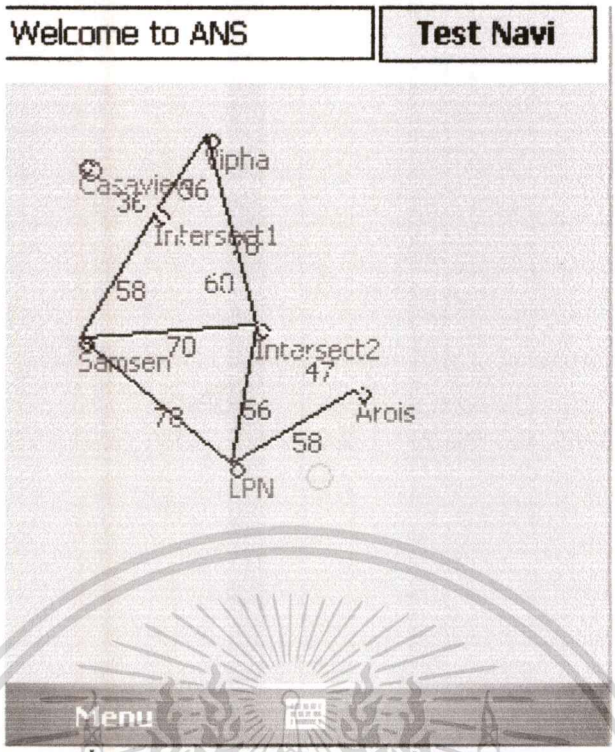
Casaview->Intersect1 เป็นระยะทาง 36 เมตร และใช้เวลา 40 นาที เมื่อถึงจุด Intersect1 ให้ตรงไป

Intersect1->Intersect2 เป็นระยะทาง 60 เมตร และใช้เวลา 90 นาที เมื่อถึงจุด Intersect2 ให้ตรงไป

Intersect2->Arois เป็นระยะทาง 47 เมตร และใช้เวลา 50 นาที เมื่อถึงจุด Arois คือถึงจุดหมายแล้ว และมีระยะทางทั้งหมด 143 เมตร ใช้เวลาทั้งสิ้น 180 นาที

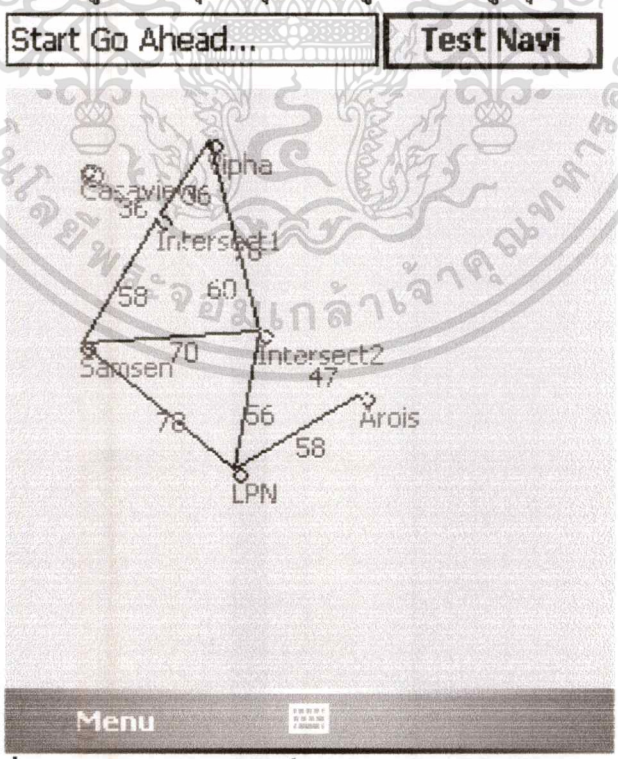
4. เลือกเส้นทาง Shortest Route หรือ Fastest Route แล้ว Click ปุ่ม "Go" จะมีหน้าจอแสดงเส้นทางที่เลือกเป็นเส้นสีเหลือง โดยวงกลมสีเขียวคือจุดเริ่มต้น และวงกลมสีส้มคือจุดปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-4 หน้าจอแสดงเส้นทางที่ผู้ใช้เลือก

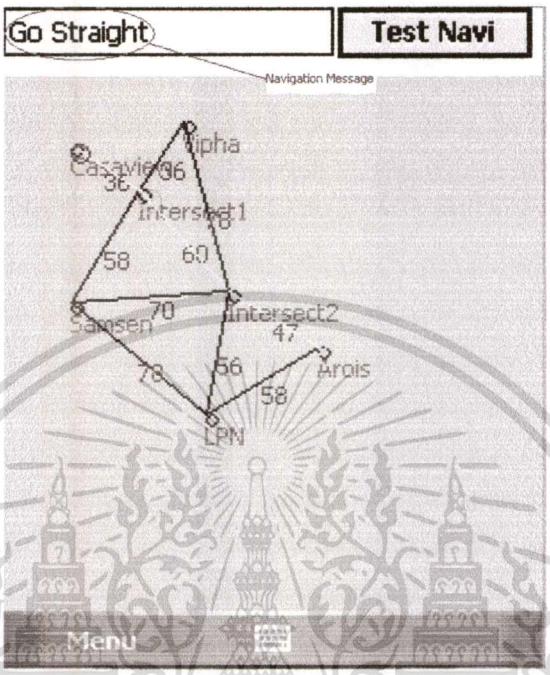
5. ทดสอบระบบนำทางอัตโนมัติโดย Click ที่ปุ่ม "Test Navi" เพื่อเข้าสู่โหมดนำทางอัตโนมัติ วงกลมสีชมพูซึ่งแทนจุดปัจจุบันของผู้ใช้จะไปอยู่ที่จุดเริ่มต้นของเส้นทาง



รูปที่ ข-5 หน้าจอแสดงการเริ่มต้นโหมดนำทางอัตโนมัติ

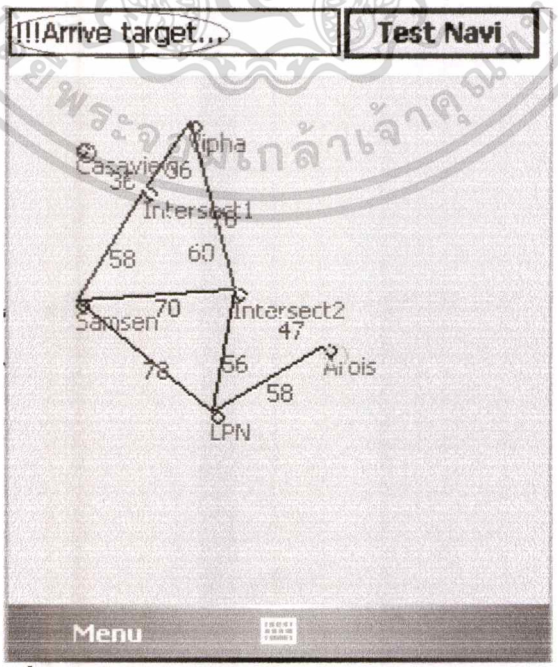
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Click ที่ปุ่ม "Test Navi" ไปเรื่อยๆ เพื่อให้จุดปัจจุบันเคลื่อนที่ไปยังจุดต่อไปที่อยู่บนเส้นทางตามลำดับ และจะมีข้อความนำทาง (Navigation Message) แสดงให้เห็นเมื่อผ่านไปยังจุดทุกจุดที่อยู่บนทาง



รูปที่ ข-6 หน้าจอแสดงข้อความนำทาง (Navigation Message)

7. เมื่อถึงจุดปลายทางจะแสดงข้อความนำทางว่า "!!!Arrive target..."



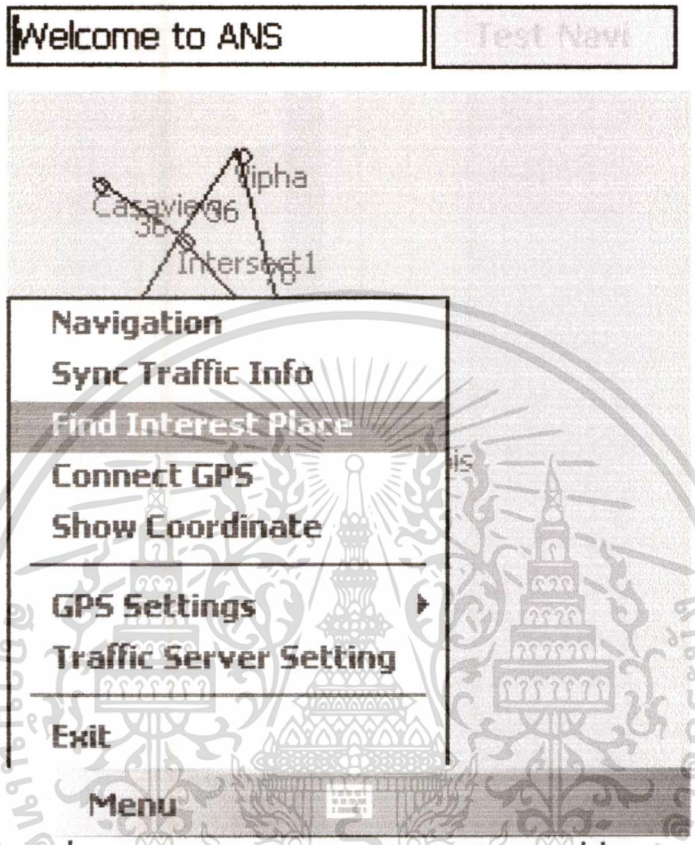
รูปที่ ข-7 หน้าจอแสดงข้อความเมื่อถึงปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการค้นหาสถานที่ที่สนใจ

ผู้ใช้สามารถค้นหาสถานที่ที่สนใจตามประเภทได้ โดยทำตามขั้นตอนดังต่อไปนี้

1. **เปิดโปรแกรมแล้วเลือก Menu->Find Interest Place**



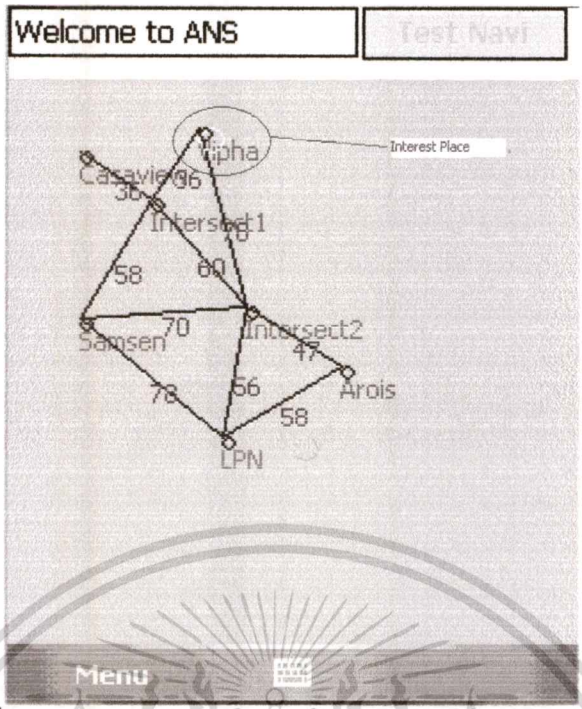
รูปที่ ข-8 หน้าจอแสดงการเข้าสู่เมนูค้นหาสถานที่ที่สนใจ

2. **Click ที่ drop down list แล้วเลือกประเภทสถานที่ที่สนใจ แล้ว click ที่ปุ่ม "Find"**

รูปที่ ข-9 หน้าจอแสดงการเลือกประเภทสถานที่ที่สนใจ

3. **จะแสดงวงกลมสีขาวบนหน้าจอลอมรอบจุดที่มีประเภทตามที่ผู้ใช้เลือก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

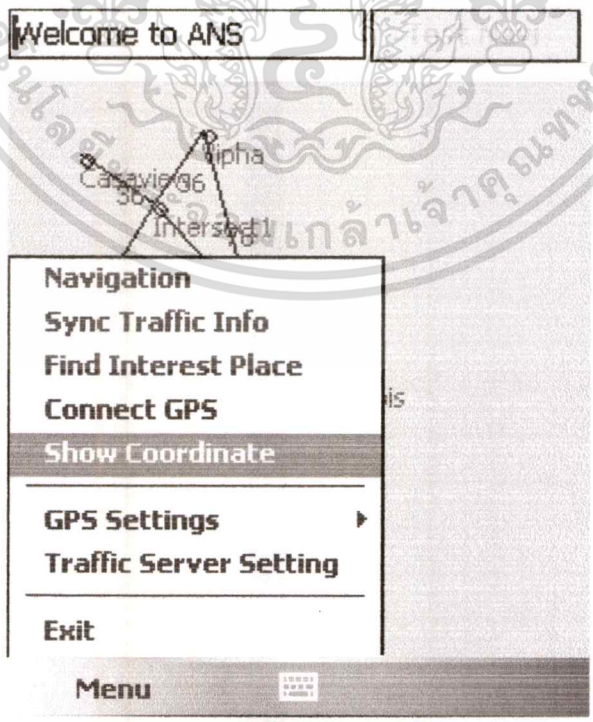


รูปที่ ข-10 หน้าจอแสดงสถานที่ตามประเภทที่ผู้ใช้เลือก

วิธีการแสดงพิกัด x,y ของจุดบนแผนที่

ผู้ใช้สามารถให้แสดงพิกัด x,y ของจุดบนแผนที่ได้ โดยทำตามขั้นตอนดังต่อไปนี้

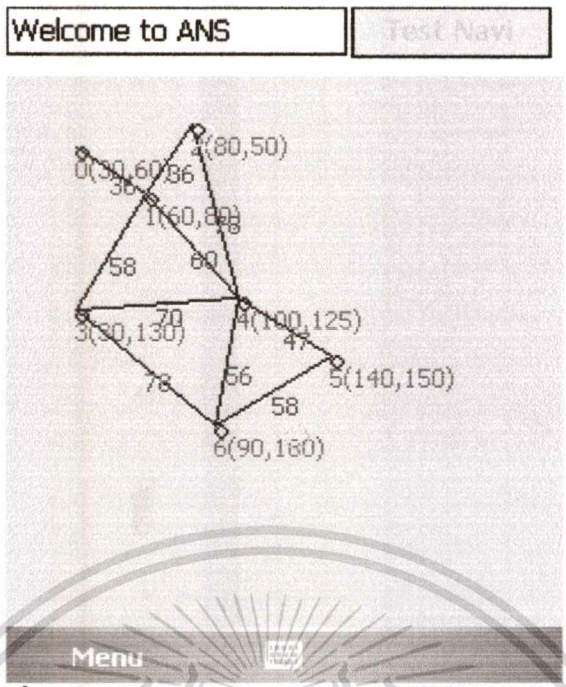
1. เปิดโปรแกรมแล้วเลือก Menu->Show Coordinate



รูปที่ ข-11 หน้าจอแสดงเมนูการเปิดให้แสดงพิกัด x,y ของจุดบนแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จะมีพิกัดและหมายเลขของจุดขึ้นมาแทนที่ชื่อจุด

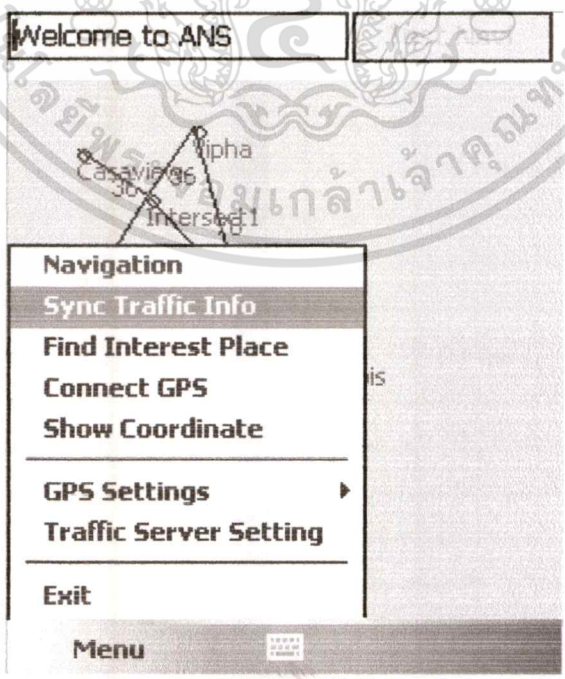


รูปที่ ข-12 หน้าจอแสดงพิกัด x,y ของจุดบนแผนที่

วิธีการเปิดการติดต่อกับ Traffic Monitoring Server

ผู้ใช้สามารถเปิดรับข้อมูลการจราจรแบบ Real-Time จาก Traffic Monitoring Server ได้ โดยทำตามขั้นตอนดังต่อไปนี้

1. เปิดโปรแกรมแล้วเลือก Menu->Sync Traffic Info



รูปที่ ข-13 หน้าจอแสดงการเปิดการติดต่อกับ Traffic Monitoring Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายภาณุวัฒน์ เรืองรักษ์
วัน-เดือน-ปี เกิด	30 เมษายน 2523
สถานที่เกิด	นครศรีธรรมราช

ประวัติการศึกษา

มัธยมศึกษาตอนปลาย	โรงเรียนเบญจมราชูทิศ
ปริญญาตรี	วิศวกรรมศาสตร์บัณฑิต (วิศวกรรมคอมพิวเตอร์)
	มหาวิทยาลัยสงขลานครินทร์ วิทยาเขต หาดใหญ่

ประวัติการทำงาน

2546 – ปัจจุบัน	บริษัท ซี ซี ซอฟแวร์ จำกัด
	ตำแหน่งนักวิเคราะห์ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้