



ระบบค้นหาเส้นทางรถประจำทางออนไลน์อัจฉริยะ

Smart Online Bus Route Search System

นางสาววิสาข	โกสุม	รหัส 47360227
นางสาวมณฑนา	ฤกษ์นันท์	รหัส 47362082
นางสาววลัยลักษณ์	เถื่อนยัง	รหัส 47362157

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ.....3/11 พ.ศ. 2551.....
เลขทะเบียน.....05100006.....
เลขเรียกหนังสือ.....
มหาวิทยาลัยนครสวรรค์

5093508. 2.2
มร.
93785.
2550.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์
ปีการศึกษา 2550



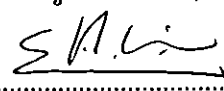
ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ	ระบบค้นหาเส้นทางการเดินรถประจำทางออนไลน์อัจฉริยะ		
ผู้ดำเนินโครงการ	นางสาววิสาข	โกสัม	รหัส 47360227
	นางสาวมณฑนา	อุกษนันท์	รหัส 47362082
	นางสาวลัษณีย์	เถื่อนยัง	รหัส 47362157
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ ธิยะมงคล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2550		

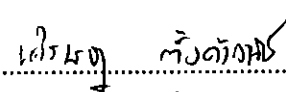
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะกรรมการสอบโครงการวิจัย


.....ประธานกรรมการ

(ดร.พนมขวัญ ธิยะมงคล)


.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ เข้มมน)


.....กรรมการ
(อาจารย์เศรษฐา ตั้งคำวานิช)

หัวข้อโครงการ	ระบบค้นหาเส้นทางการเดินทางรถประจำทางออนไลน์อัจฉริยะ		
ผู้ดำเนินโครงการ	นางสาววิสาขา	โกสัม	รหัส 47360227
	นางสาวมณฑนา	ฤกษ์นันท์	รหัส 47362082
	นางสาววลัยลักษณ์	เดือนยัง	รหัส 47362157
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ ธิยะมงคล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2550		

บทคัดย่อ

เนื่องจากปัจจุบันคอมพิวเตอร์ถือเป็นเทคโนโลยีที่มีประสิทธิภาพและมีส่วนเกี่ยวข้องกับชีวิตประจำวันมากขึ้นเรื่อย ๆ ผู้พัฒนาจึงได้ทำการพัฒนาเว็บไซต์โดยใช้ ASP.NET ในการสร้าง User Interface ติดต่อกับผู้ใช้ ระบบค้นหาเส้นทางเดินทางรถประจำทางออนไลน์อัจฉริยะจึงเป็นระบบที่มีลักษณะเป็นเว็บแอปพลิเคชัน (Web Application) ซึ่งในโครงการนี้ได้นำหลักการหาเส้นทางที่มีระยะทางสั้นที่สุด (shortest path) โดยใช้หลักการของดijkstra (Dijkstra's algorithm) และ Breath First Search (BFS) ผลที่ได้รับจากโครงการนี้คือ ได้เว็บแอปพลิเคชัน (Web Application) ซึ่งสามารถค้นหาเส้นทางไปยังสถานที่ต่างๆ ในจังหวัดพิษณุโลกโดยรถโดยสารประจำทาง ซึ่งสามารถอำนวยความสะดวกให้กับผู้ที่ต้องการเดินทางภายในพิษณุโลก

Project Title	Smart Online Bus Route Search System.		
Name	Miss Wisakha	Kosom	ID.47360227
	Miss Manthana	Rerksanan	ID.47362082
	Miss Walailak	Thuanyoung	ID. 47362157
Project Advisor	Panomkhawn Riyamongkol, Ph.D		
Major	Computer Engineering.		
Department	Electrical and Computer Engineering.		
Academic Year	2007		

ABSTRACT

The Computer is an efficient technology and involves in our life nowadays. This paper presents a designed User Interface using ASP.NET to connect with user. Smart Online Bus Route Search System in Pitsanulok is a Web Application by shortest path algorithm which is Dijkstra's algorithm and Breath First Search. The result of this project is designed and developed system to suggest the way to go to place in Pitsanulok by bus and convenient for passenger.

กิตติกรรมประกาศ

การจัดทำโครงการวิศวกรรมศาสตร์ครั้งนี้สำเร็จลงด้วยดีได้ เนื่องมาจากการได้รับความสนับสนุน แนะนำและให้ความช่วยเหลือจากบุคคลหลายท่าน ก่อนอื่นผู้จัดทำโครงการขอกราบขอบพระคุณ ดร.พนมขวัญ ธิยะมงคล ที่กรุณาเป็นอาจารย์ที่ปรึกษาด้านโครงการวิศวกรรมศาสตร์โครงการนี้ ท่านได้ให้ความสนับสนุนตลอดจนและให้คำแนะนำที่เป็นประโยชน์ชี้แนะแนวทางและขอกราบขอบพระคุณอาจารย์เศรษฐา ตั้งคำวานิช ที่ช่วยตรวจสอบเนื้อหาของโครงการและเป็นกรรมการในการสอบโครงการได้ ท่านให้การสนับสนุนให้คำปรึกษาที่เป็นประโยชน์อย่างสูงและมอบแนวทางในการแก้ไขปัญหาในการทำโครงการครั้งนี้ เพื่อให้โครงการสามารถดำเนินไปได้ด้วยดีตลอดมา และท้ายสุดขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.สุชาติ เข้มมนต์ ที่ได้เสียสละเวลาอันมีค่าในการตรวจสอบเนื้อหาและเป็นกรรมการในการสอบโครงการครั้งนี้ อีกทั้งยังให้การดูแลผู้ทำโครงการอย่างมีความเมตตาและเป็นอาจารย์ที่ปรึกษาของนิสิตชั้นปีที่ 4 คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ตลอดระยะเวลาที่ผู้ทำโครงการศึกษาอยู่ที่นี้

ขอขอบคุณสำนักงานรถเมล์บ้านเราที่ได้ให้ข้อมูลเกี่ยวกับสายรถประจำทางและเส้นทางในการเดินทางประจำทางภายในจังหวัดพิษณุโลก ขอขอบคุณสำนักงานการท่องเที่ยวแห่งประเทศไทย ภาคเหนือเขต 3 กองข่าวสารการท่องเที่ยวพิษณุโลกและศูนย์บริการนักท่องเที่ยวภายในวัดพระศรีรัตนมหาธาตุ วรมหาวิหาร (วัดใหญ่) ที่ได้ให้ข้อมูลเกี่ยวกับข้อมูลแผนที่การท่องเที่ยวในเทศบาลนครพิษณุโลกและให้ข้อมูลเกี่ยวกับการท่องเที่ยวในจังหวัดพิษณุโลก

ท้ายที่สุดนี้คณะผู้จัดทำโครงการขอกราบขอบพระคุณบิดามารดาที่ให้คำแนะนำและความช่วยเหลือให้คำแนะนำในด้านต่างๆ โดยเฉพาะอย่างยิ่งให้ความรักแก่คณะผู้จัดทำเสมอมา ทำให้คณะผู้จัดทำมีกำลังใจในการทำโครงการครั้งนี้ต่อไป ขอขอบคุณพี่ๆ เพื่อนๆ และน้องที่เป็นกำลังใจให้เสมอมา คณะผู้จัดทำจักไม่ลืมพระคุณตลอดชั่วชีวิต

คณะผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง	ฉ
สารบัญรูป.....	ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบข่ายของโครงการ.....	1
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 แผนการดำเนินงาน.....	2
1.6 ผลที่คาดว่าจะได้รับ.....	3
1.7 งบประมาณที่ใช้.....	3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีกราฟเบื้องต้น.....	4
2.2 ทฤษฎีการหาเส้นทางที่สั้นที่สุด (Shortest path algorithm)	11
2.3 หลักการ Breath-First Search (BFS)	16
2.4 หลักการเกี่ยวกับ ASP.NET.....	17
2.5 ภาษา C# ระดับพื้นฐาน.....	21

บทที่ 3 วิธีการดำเนินงาน

3.1 การค้นคว้าข้อมูล.....	23
3.2 การออกแบบเว็บเพจ.....	24
3.3 การพัฒนาโปรแกรม.....	27

บทที่ 4 ผลการดำเนินงานและการวิเคราะห์ข้อมูล

4.1 ลักษณะทั่วไปของโปรแกรม.....	33
---------------------------------	----

สารบัญ (ต่อ)

หน้า

4.2	วิธีการใช้งานโปรแกรม.....	33
4.3	วิธีการทดสอบ.....	34
บทที่ 5 บทสรุป		
5.1	สรุปผล.....	41
5.2	ปัญหา อุปสรรคและวิธีการในการแก้ปัญหา.....	41
5.3	ข้อเสนอแนะและแนวทางในการพัฒนา.....	42
เอกสารอ้างอิง.....		43
ภาคผนวก ก สายเคเบิลและเส้นทางเคเบิลโดยสาร.....		44
ภาคผนวก ข แผนที่จังหวัดพิษณุโลก.....		46
ประวัติผู้เขียน.....		49

สารบัญตาราง

ตารางที่

หน้า

2.1 ตารางสรุปความหมายของคำต่างๆที่เกี่ยวข้องกับกราฟ.....	7
--	---



สารบัญรูป

รูปที่	หน้า
2.1 กราฟ.....	4
2.2 มัลติกราฟ.....	5
2.3 กราฟไม่มีทิศทาง (Undirected graphs).....	7
2.4 กราฟมีทิศทาง (Directed graphs).....	8
2.5 กราฟเชื่อมโยง.....	8
2.6 ไม่เป็นกราฟเชื่อมโยง.....	8
2.7 กราฟมีน้ำหนัก.....	9
2.8 กราฟที่ไม่มีน้ำหนักและเป็นกราฟไม่มีทิศทาง.....	9
2.9 กราฟที่มีน้ำหนักและเป็นกราฟมีทิศทาง.....	9
2.10 กราฟบริบูรณ์.....	10
2.11 กราฟระนาบ (Planar Graph).....	10
2.12 แผนที่และบริเวณ (Maps and Regions).....	10
2.13 ขั้นเริ่มต้นในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ.....	13
2.14 กราฟที่ 1 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ.....	13
2.15 กราฟที่ 2 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ.....	13
2.16 กราฟที่ 3 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ.....	14
2.17 กราฟที่ 4 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ.....	14
2.18 กราฟที่ 5 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ.....	14
2.19 โครงสร้างของ .NET Framework.....	18
3.1 หน้าเว็บไซต์แสดงการหาระยะห่างระหว่างจุดจากสถานีรถไฟ ไป สกอ.เมืองพิษณุโลก.....	24
3.2 หน้าเว็บเพจที่ออกแบบเป็นหลักของทุกหน้าประกอบไปด้วย เมนูเชื่อมต่อให้เลือกทางด้านขวา.....	25
3.3 การเชื่อมโยงหน้าเว็บเพจ.....	26
3.4 แผนที่แสดงสายรถประจำทางที่ผ่านจุดเชื่อมต่อ.....	28
3.5 การเชื่อมโยงของสถานที่ต่างๆ และสายรถประจำทางที่ผ่าน.....	29
3.6 กราฟแสดงสถานที่ D ออก.....	30
3.7 กราฟแสดงสถานที่ A ออก.....	30
3.8 กราฟแสดงสถานที่ B ออก.....	30

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.9 กราฟแสดงสถานที่จนถึง C แล้วทำการเก็บเส้นทางย้อนกลับ.....	31
3.10 Flow – chart ของโปรแกรม.....	32
4.1 หน้าเว็บเพจแสดงหน้าแรกของระบบค้นหาเส้นทางของ รถประจำทางออนไลน์อัจฉริยะ.....	34
4.2 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อตารางเส้นทางของรถประจำทาง.....	35
4.3 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อแผนที่เทศบาลนครพิษณุโลก.....	36
4.4 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อแผนที่ท่องเที่ยวจังหวัดพิษณุโลก.....	36
4.5 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อสถานที่ท่องเที่ยวจังหวัดพิษณุโลก.....	37
4.6 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อโรงแรมในจังหวัดพิษณุโลก.....	37
4.7 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อการเดินทางในจังหวัดพิษณุโลก.....	38
4.8 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อร้านอาหารในจังหวัดพิษณุโลก.....	38
4.9 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อ Contact us.....	39
4.10 หน้าเว็บเพจแสดงผลการเลือกจุดเริ่มต้นและจุดสิ้นสุด.....	39
4.11 หน้าเว็บเพจแสดงผลการค้นหาสายรถประจำทาง.....	40

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ในสังคมปัจจุบันนี้การเดินทางได้เป็นส่วนหนึ่งในการดำเนินชีวิตประจำวันของแต่ละคน การเดินทางด้วยรถโดยสารประจำทางเป็นการคมนาคมทางหนึ่งที่จะช่วยลดปัญหาภาวะเป็นพิษ ช่วยประหยัดน้ำมัน อีกทั้งยังช่วยลดค่าใช้จ่ายในการเดินทางอีกด้วย การเดินทางด้วยรถโดยสารประจำทางจึงเป็นทางเลือกหนึ่งในการเดินทางที่มีผู้นิยมใช้บริการจังหวัดพิษณุโลกก็เป็นจังหวัดหนึ่งที่มีประชากรอยู่หนาแน่นพอสมควรเนื่องจากเป็นเมืองเก่ามีสถานศึกษาหลายแห่ง มีสถานที่ท่องเที่ยวที่น่าสนใจ ดังนั้นการเดินทางไปในที่ต่างๆ จึงมีการใช้บริการของรถโดยสารประจำทาง แต่ในบางครั้ง นักท่องเที่ยวหรือนักเรียนนักศึกษาไม่อาจทราบได้แน่ชัดว่าจะเดินทางไปยังจุดหมายที่ต้องการ จะต้องเดินทางด้วยรถโดยสารประจำทางสายอะไร จึงเป็นปัญหาหนึ่งที่เกิดขึ้นบ่อยครั้งทำให้การเดินทางไปยังจุดหมายปลายทางนั้นล่าช้าขึ้นไปอีก

ดังนั้นทางคณะผู้จัดทำจึงได้ตระหนักถึงความจำเป็นและความต้องการของบริการดังกล่าว ที่มีมากขึ้นเรื่อยๆ ในกลุ่มของผู้เดินทาง และเห็นว่าควรมี Web Application สำหรับบริการนักท่องเที่ยว นักเรียนนักศึกษาและประชาชนทั่วไปที่มีความต้องการเดินทาง เพื่อความสะดวก รวดเร็วและอำนวยความสะดวกในการเดินทางในจังหวัดพิษณุโลก โดยสามารถหาข้อมูลเส้นทาง การเดินทางในแต่ละพื้นที่ที่ต้องการได้อย่างแม่นยำ

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษาระบบการเขียนโปรแกรมในรูปแบบ Web - Application
- 1.2.2 เพื่อศึกษาทฤษฎีของ Shortest Path Algorithm และ Breath-First Search
- 1.2.3 เพื่อพัฒนาและช่วยให้การเดินทางสะดวกและรวดเร็วยิ่งขึ้น

1.3 ขอบข่ายของโครงการ

- 1.3.1 สร้าง Web page จาก ASP.Net ซึ่งผู้เดินทางเลือกค้นทางกับปลายทาง Web page จะแสดงผลเป็นสถานที่และสายรถประจำทางที่ผ่าน ในระยะทางที่สั้นที่สุด
- 1.3.2 การเดินทางด้วยรถโดยสารประจำทางนั้นจะใช้ขอบเขตในจังหวัดพิษณุโลก

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและค้นคว้าข้อมูลเกี่ยวกับการเดินทางของรถโดยสารประจำทางในจังหวัดพิษณุโลก

1.4.2 ศึกษาเส้นทางการเดินทางของรถโดยสารประจำทางในจังหวัดพิษณุโลก

1.4.3 ศึกษาและค้นคว้า Shortest path algorithm และ Breath-First Search

1.4.4 ศึกษาและค้นคว้าข้อมูลเกี่ยวกับ Web Application และการเขียนโปรแกรม ASP.Net

1.4.5 วิเคราะห์และออกแบบระบบ Web Application โดยใช้ ASP.Net

1.4.6 พัฒนาระบบ Web Application โดยใช้ ASP.Net เข้ากับ Shortest path algorithm

1.4.7 ตรวจสอบความถูกต้องและทำการปรับปรุงแก้ไข

1.5 แผนการดำเนินงาน

กิจกรรม	ปี 2551		
	มีนาคม	เมษายน	พฤษภาคม
1. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับการเดินทางของรถโดยสารประจำทางในจังหวัดพิษณุโลก	←→		
2. ศึกษาเส้นทางการเดินทางของรถโดยสารประจำทางในจังหวัดพิษณุโลก	←→		
3. ศึกษาและค้นคว้า Shortest path Algorithm และ Breath - First Search	←→		
4. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับ Web Applicationและการเขียนโปรแกรม ASP.Net	←→		
5. วิเคราะห์และออกแบบระบบ Web application โดยใช้ ASP.Net		←→	
6. พัฒนาระบบ Web application โดยใช้ ASP.Net เข้ากับ Shortest path algorithm		←→	→
7. ตรวจสอบความถูกต้องและทำการปรับปรุงแก้ไข			←→

1.6 ผลที่คาดว่าจะได้รับ

- 1.6.1 มีความสามารถในการเขียน Web Application โดยใช้ ASP.Net
- 1.6.2 สามารถใช้โปรแกรม Microsoft Visual Studio 2005 ในการเขียน Website ได้
- 1.6.4 มีความรู้ในทฤษฎี Shortest path algorithm และ Breath-First Search สามารถนำมาประยุกต์ใช้กับ Website ได้
- 1.6.5 ได้ Website ที่มีประโยชน์ต่อการเดินทางด้วยรถโดยสารประจำทางในจังหวัดพิษณุโลก

1.7 งบประมาณที่ใช้

1.7.1 ค่าวัสดุสำนักงาน	เป็นจำนวนเงิน	1,000	บาท
1.7.2 อุปกรณ์คอมพิวเตอร์	เป็นจำนวนเงิน	1,000	บาท
1.7.3 ค่าถ่ายเอกสารและปริ้นท์งาน	เป็นจำนวนเงิน	<u>1,000</u>	บาท
	รวมเป็นจำนวนเงิน	<u>3,000</u>	บาท

หมายเหตุ : ถัวเฉลี่ยทุกรายการ



บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

ในการศึกษาและการพัฒนาโครงการงานได้มีหลักการและทฤษฎีที่เกี่ยวข้องที่ต้องทำการศึกษาเพื่อใช้ในการพัฒนาโครงการงานดังต่อไปนี้

2.1 ทฤษฎีกราฟเบื้องต้น

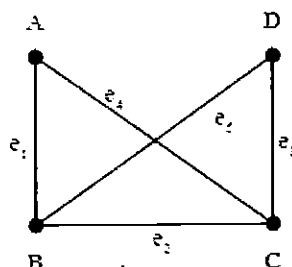
กราฟและมัลติกราฟ

ทฤษฎีกราฟสามารถใช้ในการแก้ปัญหาต่าง ๆ ในหลายสาขาได้เป็นอย่างดี ตัวอย่างเช่นเราสามารถบอกถึงความแตกต่างระหว่างสารประกอบทางเคมีที่มีสูตรโมเลกุลเหมือนกัน แต่ต่างกันที่โครงสร้างโดยใช้กราฟหรือการหาระยะทางที่สั้นที่สุดระหว่างเมืองสองเมืองในแผนที่จราจรโดยใช้กราฟที่ถูกกำหนดน้ำหนักให้แก่แต่ละเส้น โครงสร้างข้อมูลกราฟ (Graphs) ประกอบด้วย 2 สิ่งต่อไปนี้

(1) เซต V เรียกสมาชิกของ V ว่า จุดยอด (Vertex) หรือ จุด (Point) หรือ บัพ (Node)

(2) เซต E ของคู่อันดับ (Unordered Pair) ของจุดยอด เรียกคู่อันดับเหล่านี้ว่า ขอบ (Edge) เมื่อต้องการเน้นส่วนประกอบทั้งสองของ G เราจะแทนกราฟ G ด้วย $G(V, E)$ กราฟที่มีข้อมูลกำกับอยู่บนด้านของกราฟ โดยข้อมูลดังกล่าวอาจหมายถึงระยะทาง จำนวนวันหรือจำนวนเงิน เป็นต้น ทั้งนี้ขึ้นอยู่กับความสัมพันธ์ของกราฟนั้นแทนความหมายของสิ่งใด เช่น ตัวอย่างแผนที่แสดงระยะทางระหว่างเมืองต่าง ๆ

ถ้ามีขอบ $\{U, V\}$ เชื่อมจุดยอด U และ V เราจะกล่าวว่า U และ V ประชิดกัน (Adjacent) เราเขียนภาพของกราฟ โดยใช้แผนภาพระนาบเหมือนปกติ กล่าวคือ แทนจุดยอดด้วยจุด (หรือวงกลมเล็ก) และแทนขอบ $e = \{V_1, V_2\}$ ด้วยเส้น ซึ่งเชื่อมจุดยอด V_1 และ V_2 เรียก V_1 และ V_2 ว่าจุดปลาย (Endpoint) ของ e เช่น



รูปที่ 2.1 กราฟ

แทนกราฟ $G(V, E)$ เมื่อ

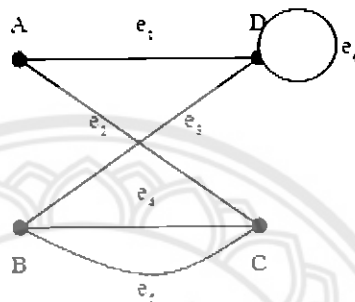
(1) V ประกอบด้วยจุดสี่จุด คือ A, B, C, D

แทนกราฟ $G(V, E)$ เมื่อ

(1) V ประกอบด้วยจุดสี่จุด คือ A, B, C, D

(2) E ประกอบด้วยขอบห้าขอบ คือ $e_1 = \{A, B\}$, $e_2 = \{B, C\}$, $e_3 = \{C, D\}$, $e_4 = \{A, C\}$ และ $e_5 = \{B, D\}$

โดยปกติเราจะแสดงกราฟด้วยการเขียนแผนภาพของกราฟ แทนที่จะแจกแจงจุดยอดและขอบของกราฟ



รูปที่ 2.2 มัลติกราฟ

รูปนี้ไม่ใช่กราฟ แต่เป็นมัลติกราฟ เพราะมีขอบสองขอบคือ e_4 และ e_5 เชื่อมจุดปลายคู่เดียวกัน (Parallel) และมี e_6 รอบเป็น (Loop) นั่นคือ เป็นขอบซึ่งจุดปลายทั้งสองเป็นจุดเดียวกัน นิยามของกราฟไม่อนุญาตให้มีขอบมากกว่าหนึ่งขอบเชื่อมจุดปลายคู่เดียวกัน และจะต้องไม่มีรอบ กล่าวอีกนัยหนึ่งคือ เราอาจนิยามกราฟว่า คือ มัลติกราฟที่ไม่มีขอบมากกว่าหนึ่งขอบเชื่อมจุดปลายคู่เดียวกันและไม่มีวงรอบ

ดีกรีของจุดยอด

ถ้า V คือจุดปลายของ e เราจะกล่าวว่า e กระทบ (Incident) กับจุด V ดีกรีของจุดยอด V ซึ่งจะเขียนแทนด้วย $\deg(V)$ ซึ่งจำนวนขอบที่กระทบกับจุด V เนื่องจากขอบแต่ละขอบจะถูกนับสองครั้งในการนับดีกรีของจุดยอดทุกจุดของกราฟ ดังนั้น เราจะได้ผลลัพธ์ต่อไปนี้ซึ่งง่ายแต่มีความสำคัญมาก

ทฤษฎีบท ผลบวกของดีกรีของจุดยอดของกราฟใดๆ เป็นสองเท่าของจำนวนขอบของกราฟ
ตัวอย่าง รูป 2.1 มี $\deg(A) = 2$ $\deg(B) = 3$ $\deg(C) = 3$ และ $\deg(D) = 2$
ผลบวกของดีกรีเท่ากับ 10

ตัวอย่าง รูป 2.2 $\text{Deg}(D) = 4$ เพราะว่าขอบ e_6 ถูกนับสองครั้ง ดังนั้น D เป็นจุดยอดคู่ เรียกจุดยอดซึ่งมีดีกรีศูนย์ ว่าจุดยอดเอกเทศ (Isolate Vertex)

กราฟสามารถนำมาประยุกต์เพื่อใช้แก้ปัญหาหลายๆ ปัญหาที่เกี่ยวกับการเคลื่อนที่จากจุดๆ หนึ่งไปยังจุดอื่นๆ ในกราฟ เช่น ปัญหาทางคมนาคม ปัญหาทางสาธารณสุขโลก ก่อนที่จะใช้กราฟแก้ปัญหาดังกล่าว เราต้องทราบความหมายของคำต่าง ๆ ดังนี้

ทางเดิน (Walk) ในมัลติกราฟประกอบด้วยลำดับของจุดยอดและขอบสลับกันในรูปแบบ

$$v_0, e_1, v_1, e_2, v_2, \dots, e_{n-1}, v_{n-1}, e_n, v_n$$

เมื่อขอบ e_i แต่ละขอบกระทบกับ v_{i-1} และ v_i ความยาว (Length) ของทางเดินคือ n ซึ่งก็คือจำนวนขอบในทางเดิน ในกรณีที่ไม่นำให้เกิดความสับสน เราจะแทนทางเดินด้วยลำดับของขอบ $(e_1, e_2, \dots, e_n)$ หรือแทนด้วยลำดับของจุดยอด $(v_0, v_1, \dots, v_n)$ ถ้า $v_0 = v_n$ เราจะเรียกทางเดินนั้นว่าทางเดินปิด C Closed Walk แต่ถ้า $v_0 \neq v_n$ เราจะเรียกทางเดินนั้นว่า ทางเดินจาก v_0 ถึง v_n หรือเรียกทางเดินระหว่าง v_0 และ v_n หรือกล่าวว่าเป็นทางเดินซึ่งเชื่อม v_0 และ v_n

รอยเดิน (Trail) คือ ทางเดินซึ่งขอบทุกขอบแตกต่างกัน และวิถี (Path) คือ ทางเดินซึ่งจุดยอดทุกจุดแตกต่างกัน วัฏจักร (Cycle) คือทางเดินปิดซึ่งจุดยอดทุกจุดแตกต่างกัน ยกเว้น $v_0 = v_n$ เรียกวัฏจักรที่มีความยาว K ว่าวัฏจักร - K (K - Cycle) วัฏจักรใดๆ ในกราฟจะต้องมีความยาวเท่ากับสามหรือมากกว่า

นิยาม ให้ G เป็นกราฟ และ u กับ v เป็นจุดยอด

1. Walk เป็นการเริ่มจากจุดยอด u ไปถึงจุดยอด v โดยที่สามารถซ้ำขอบและจุดได้ และยังสามารถเริ่มและสิ้นสุดที่จุดเดียวกัน
2. Close Walk เป็นการเดินที่เริ่มจากจุดเริ่มต้นและจุดสิ้นสุดที่จุดยอดเดียวกัน
3. Path เป็น Walk ที่เดินจากจุด u ถึง v โดยที่ห้ามซ้ำขอบ
4. Simple Path เป็น Path ที่เดินจาก u ถึง v โดยจะห้ามเดินซ้ำจุดยอด
5. Circuit ก็เป็น Close Walk แต่จะห้ามเดินซ้ำขอบ
6. Simple Circuit ถือเป็น Circuit แต่ห้ามเดินซ้ำจุดยอด ยกเว้นจุดเริ่มต้นและจุดสุดท้าย

ตารางที่ 2.1 ตารางสรุปความหมายของคำต่างๆที่เกี่ยวข้องกับกราฟ

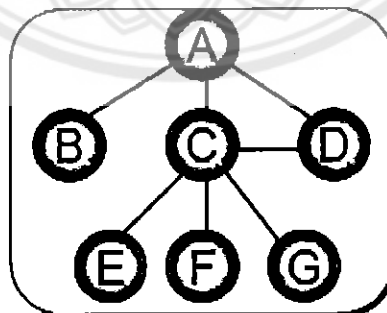
	ซ้ำจุดยอด	ซ้ำจุดยอด	เริ่มต้นและสิ้นสุดที่จุดเดียวกัน
Walk	/	/	/
Path	X	/	/
Simple Path	X	X	X
Close Walk	/	/	ต้อง
Circuit	X	/	ต้อง
Simple Circuit	X	เฉพาะจุดเริ่มต้นกับ จุดสิ้นสุดเท่านั้น	ต้อง

หมายเหตุ ได้ (Allow)
 X ไม่ได้ (No)
 ต้อง (Yes)

ประเภทของกราฟ

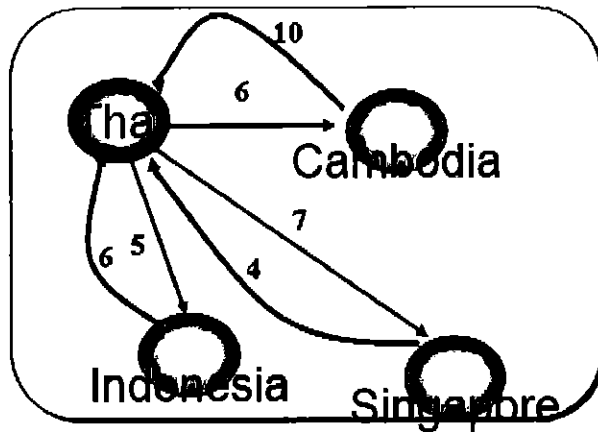
กราฟแบ่งออกเป็น 2 ประเภท ได้แก่

1. กราฟไม่มีทิศทาง (Undirected graphs) คือ กราฟที่มีด้าน (edges) ที่ไม่มีลูกศรแสดงทิศทาง ความสัมพันธ์ที่เกิดขึ้น (มองความสัมพันธ์ได้ทั้งสองด้าน) แสดงในรูปที่ 2.3



รูปที่ 2.3 กราฟไม่มีทิศทาง (Undirected graphs)

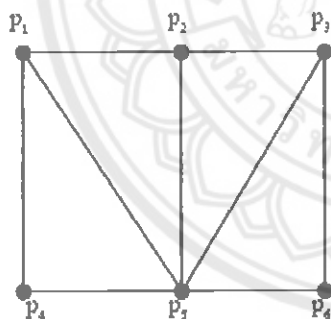
2. กราฟมีทิศทาง (Directed Graphs) คือกราฟที่มีด้าน (Edges) ที่มีลูกศรแสดงทิศทาง ความสัมพันธ์ที่เกิดขึ้น บางครั้งเรียกว่า Digraphs ดังแสดงในรูปที่ 2.4



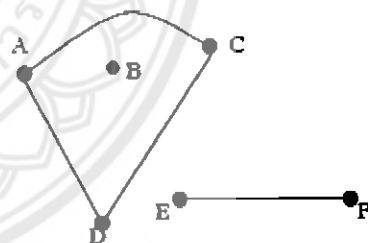
รูปที่ 2.4 กราฟมีทิศทาง (Directed graphs)

กราฟเชื่อมโยง (Connected Graph)

กราฟ G จะมีทางเดินจากจุดยอด u ถึงจุดยอด v ก็ต่อเมื่อมีวิถีจาก u ถึง v และจะเรียกกราฟว่า กราฟเชื่อมโยง (Connected Graph) ถ้ามีวิถีระหว่างจุดยอดสองจุดใดๆ กราฟในรูป 2.5 เป็นกราฟเชื่อมโยง แต่กราฟในรูป 2.6 ไม่เป็นกราฟเชื่อมโยง เพราะไม่มีวิถีระหว่าง D และ E



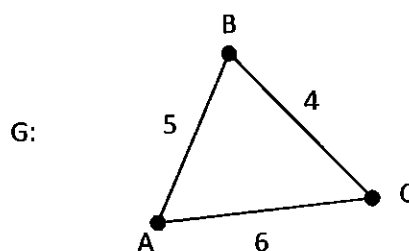
รูปที่ 2.5 กราฟเชื่อมโยง



รูปที่ 2.6 ไม่เป็นกราฟเชื่อมโยง

กราฟมีน้ำหนัก (weighted graph)

การแก้ปัญหาสถานการณ์ต่างๆ โดยใช้กราฟเป็นแบบจำลองนั้น บางครั้งจำเป็นต้องกำหนดแบบจำลองที่มีความชัดเจนเพื่อที่จะช่วยในการแก้ปัญหาได้ง่ายขึ้น เช่น ปัญหาเกี่ยวกับการเดินทางที่เกี่ยวข้องกับระยะทาง ค่าใช้จ่าย บางครั้งเราอาจกำหนดเงื่อนไขต่างๆ ลงในแบบจำลองซึ่งทำได้โดยกำหนดจำนวนที่แทนสิ่งเหล่านั้นลงในเส้นกราฟ เช่นกราฟต่อไปนี้



รูปที่ 2. 7 กราฟมีน้ำหนัก

จุด A, B, C แทนตำแหน่งของเมือง 3 เมือง ซึ่งระยะทางระหว่างเมือง

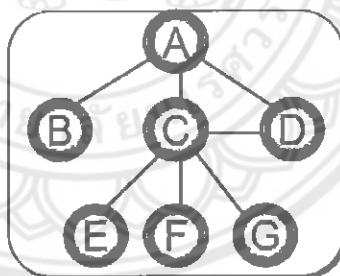
A และ B เท่ากับ 5 กม.

B และ C เท่ากับ 4 กม.

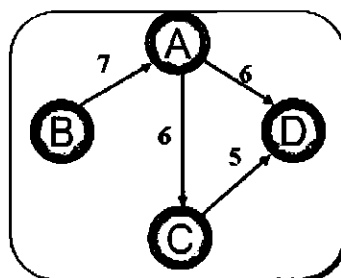
C และ A เท่ากับ 6 กม.

กราฟลักษณะดังกล่าว เรียกว่า กราฟมีน้ำหนัก ซึ่งมีความหมาย ดังนี้

1. กราฟ G เป็น กราฟมีน้ำหนัก (weighted graph) เมื่อ เส้นแต่ละเส้น (e) กำหนดด้วยจำนวนจริงที่ไม่เป็นลบ และเรียกจำนวนจริงดังกล่าวว่า น้ำหนักของเส้น แทนด้วย $w(e)$
2. วิธีที่สั้นที่สุดจากจุด U ถึงจุด V ในกราฟมีน้ำหนัก คือ วิธี U - V ที่มีผลรวมของค่าน้ำหนักของเส้นทุกเส้นในวิธี U - V มีค่าน้อยที่สุด



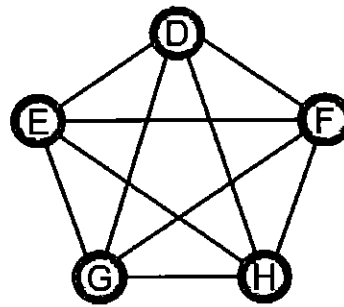
รูปที่ 2. 8 กราฟที่ไม่มีน้ำหนักและเป็นกราฟไม่มีทิศทาง



รูปที่ 2. 9 กราฟที่มีน้ำหนักและเป็นกราฟมีทิศทาง

กราฟบริบูรณ์ (Complete Graph)

คือ กราฟที่ทุก node มีเส้นเชื่อมถึงโหนดอื่นๆ ทั้งหมด



รูปที่ 2.10 กราฟบริบูรณ์

กราฟระนาบ (Planar Graph)

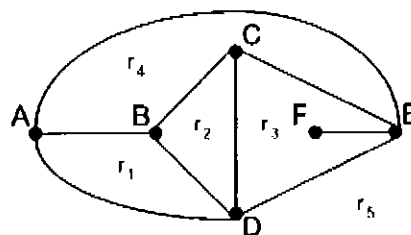
คือ กราฟที่ไม่มีขอบเขตใดๆ ตัดกัน ตัวอย่างเช่น กราฟ K_4 เพราะกราฟ K_4 สามารถเขียนโดยไม่มีขอบตัดกันได้ ดังรูปที่ 2.11



รูปที่ 2.11 กราฟระนาบ (Planar Graph)

แผนที่และบริเวณ (Maps and Regions)

คือ ระบายที่อยู่ในรูปที่ไม่มีขอบเขตใดตัดกันและอาจกล่าวว่าแผนที่นั้นเป็นแผนที่ที่เชื่อมโยง (Connected) ถ้ามีลัทธิกราฟนั้นเป็นลัทธิกราฟเชื่อมโยงแผนที่ใดๆ สามารถแบ่งระบายออกเป็นบริเวณต่างๆ ได้ จากแผนที่ต่อไปนี้ ซึ่งมีจุดยอด 6 จุด และขอบ 9 ขอบ แบ่งระบายออกเป็น 5 บริเวณ โดยมี 4 บริเวณที่มีขอบ แต่บริเวณที่ 5 อยู่นอกแผนภาพ



รูปที่ 2.12 แผนที่และบริเวณ (Maps and Regions)

ดังนั้นถ้าเราสมมุติว่าแผนที่ของเราเขียนอยู่ภายในรูปสี่เหลี่ยมผืนผ้าขนาดใหญ่ แทนที่จะเขียนอยู่ในระนาบทั้งระนาบก็จะไม่ทำให้การนับจำนวนบริเวณโดยทั่วไปแตกต่างกัน

จะเห็นว่าพรมแดนของแต่ละบริเวณของแผนที่ประกอบด้วยขอบ บางครั้งขอบเหล่านั้นประกอบกันขึ้นเป็นวัฏจักรบางครั้งก็ไม่ใช่ เช่น พรมแดนของแต่ละบริเวณ ในรูปที่ 2.12 เป็นวัฏจักร ยกเว้นพรมแดนของ r_3 และถ้าเราเดินรอบ r_3 ในทิศทางทวนเข็มนาฬิกา โดยเริ่มต้นที่จุดใดๆ เช่น เริ่มที่จุดยอด C เราจะได้ทางเดินปิด (C,D,E,F,E,C) เมื่อขอบ {E,F} ปรากฏ 2 ครั้ง เมื่อกล่าวถึงคิรีของบริเวณ r ซึ่งเขียนแทนด้วย $\deg(r)$ เราจะหมายถึงความยาวของวัฏจักรหรือทางเดินปิดที่ล้อมรอบบริเวณ r จะเห็นว่าขอบแต่ละขอบเป็นพรมแดนของบริเวณ 2 บริเวณหรือไม่ก็บรรจุอยู่ในบริเวณใดบริเวณหนึ่ง ซึ่งในกรณีหลังนี้ขอบนั้นจะปรากฏ 2 ครั้งในทางเดินที่ล้อมรอบบริเวณนั้น ดังนั้นเราได้ทฤษฎีบทสำหรับบริเวณ ดังนี้

ทฤษฎีบท ผลบวกของคิรีของบริเวณแผนที่เท่ากับสองเท่าของจำนวนขอบคิรีของบริเวณในรูปที่ 2.12 ได้แก่

$$\deg(r_1) = 3, \deg(r_2) = 3, \deg(r_3) = 5, \deg(r_4) = 4, \deg(r_5) = 3$$

ผลบวกของคิรีเท่ากับ 18 ซึ่งเป็น 2 เท่าของจำนวนขอบตามที่คาดหวังไว้

2.2 ทฤษฎีการหาเส้นทางที่สั้นที่สุด (Shortest path algorithm)

ขั้นตอนวิธีที่ใช้ในการค้นหาเส้นทางที่เหมาะสมในการเดินทางคือกฎเกณฑ์ที่ใช้ในการแก้ปัญหาการเดินทางหรือ การเลือกเส้นทางที่เหมาะสม ในการเดินทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง ขั้นตอนและวิธีการที่ใช้ในการค้นหาเส้นทางนี้มีผู้คิดค้นขึ้นหลายคนด้วยกัน เช่น Prim's Algorithm, Warshall's Algorithm และ Dijkstra's Algorithm เป็นต้น ซึ่งขั้นตอนวิธีที่คิดค้นขึ้นนี้เป็นการนำเอาทฤษฎีต่างๆเกี่ยวกับการหาเส้นทางมาผสมผสานกันเพื่อให้เกิดกฎที่ใช้ในการหาเส้นทางที่เหมาะสม สำหรับวิธีการพื้นฐานทางคณิตศาสตร์ที่ใช้แก้ปัญหาในการหาเส้นทางบนเว็บนั้นเรียกว่าวิธีการหาเส้นทางที่สั้นที่สุด (shortest path) ระหว่างโหนด ซึ่งวิธีการนี้ได้นำมาเสนอในปี ค.ศ.1959 โดย Edsger Wybe Dijkstra และวิธีการนี้มีชื่อว่า Dijkstra's algorithm เพื่อเป็นการให้เกิดเครดิตแก่ผู้คิดค้น

Dijkstra's algorithm

บริษัทแห่งหนึ่งมีคลังสินค้าที่จุด X และตลาดที่จุด Y โดยที่มีเส้นทางขนส่งจาก X ถึง Y หลายทาง เส้นทางไหนคือเส้นทางที่ประหยัดค่าใช้จ่ายมากที่สุดในการขนส่งสินค้า ปัญหาประเภทนี้สามารถแก้ได้โดย ทฤษฎีกราฟ (graph theory) ซึ่งเรียกว่า ปัญหา “ทางเดินที่สั้นที่สุด” (shortest path problem) หนึ่งในวิธีการแก้ปัญหาในเรื่องนี้คือ แบบการคำนวณของดิสตรา (Dijkstra's algorithm) ซึ่งมีประสิทธิภาพในการแก้ปัญหานี้ ให้ G แทนกราฟที่มีน้ำหนักและ

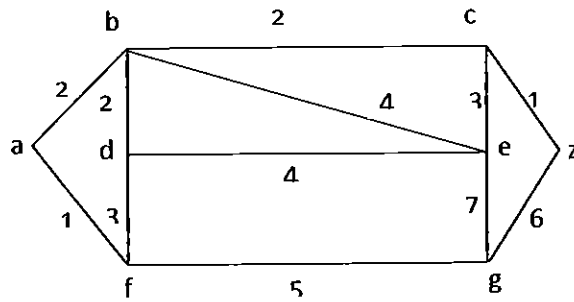
ติดต่อกัน (connected weighted graph) สมมติว่า น้ำหนักเป็นจำนวนบวกและต้องการหาการทางเดินที่สั้นที่สุดเส้นทางหนึ่งจากจุดยอด a ไปจุดยอด z

แบบการคำนวณของคิสตราเป็นการกำหนดค่าประจำให้จุดยอด ให้ $L(v)$ แทนค่าประจำจุดยอด v ณ ชั้นใดๆ จุดยอดบางจุดได้ค่าประจำจุดยอดชั่วคราวและจุดยอดที่เหลือจะได้ค่าประจำจุดยอดถาวร ให้ T แทนเซตของจุดยอดที่ได้ค่าประจำจุดยอดชั่วคราว ในการแสดงแบบคำนวณนี้ เราจะวงกลมจุดยอดที่มีค่าประจำจุดยอดถาวร เราจะแสดงต่อไปว่าถ้า $L(v)$ เป็นค่าประจำจุดยอดถาวรของจุดยอด v จะได้ว่า $L(v)$ เป็นความยาวของทางเดินที่สั้นที่สุดเส้นทางหนึ่งจาก a ไป v เมื่อเริ่มต้นทุกจุดยอดมีค่าประจำจุดยอดชั่วคราว แต่การกระทำซ้ำของแบบการคำนวณจะเปลี่ยนสภาพของค่าประจำจุดยอดค่าหนึ่งจากชั่วคราวเป็นถาวร ดังนั้นเราอาจพบแบบการคำนวณได้เมื่อ z ได้รับค่าประจำจุดยอดถาวร ณ จุดนี้ $L(z)$ จะให้ความยาวของทางเดินที่สั้นที่สุด (shortest path) เส้นทางหนึ่งจาก a ไป z

แบบการคำนวณหาทางเดินที่สั้นที่สุดของคิสตรา แบบการคำนวณนี้หาความยาวของทางเดินที่สั้นที่สุดจากจุดยอด a ไปจุดยอด z ในกราฟที่มีน้ำหนักและติดต่อกัน น้ำหนักของด้าน (i, j) ซึ่งคือ $w(i, j) > 0$ และค่าประจำจุดยอด x คือ $L(x)$ เมื่อจบการคำนวณ $L(z)$ เป็นความยาวของทางเดินที่สั้นที่สุดจาก a ไป z

1. [เริ่มต้น] ให้ $L(a) = 0$ สำหรับจุดยอดทุกจุด $x \neq a$ ให้ $L(x) = \infty$ ให้ T เป็นเซตของจุดยอดทั้งหมดในกราฟ
2. [ทำเสร็จหรือยัง] ถ้า z ไม่อยู่ใน T หยุดการคำนวณได้ ($L(z)$ เป็นความยาวของทางเดินที่สั้นที่สุดเส้นทางหนึ่งจาก a ไป z)
3. [นำจุดยอดต่อไปเข้ามา] เลือก v อยู่ใน T ที่มีค่า $L(v)$ ต่ำที่สุด ให้ $T := T - \{v\}$
4. [ปรับปรุงค่าประจำจุดยอดใหม่] สำหรับแต่ละจุดยอด x อยู่ใน T ที่ติดอยู่กับ v ให้ $L(x) := \min\{L(x), L(v) + w(v, x)\}$ ย้อนกลับไปที่ขั้นที่ 2

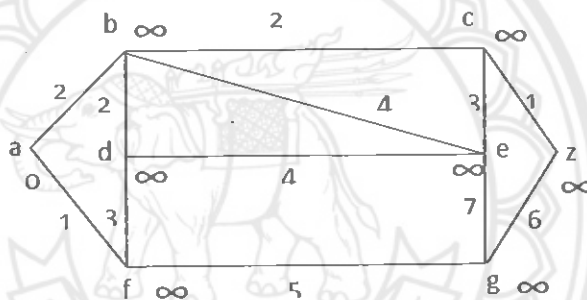
ตัวอย่าง จงหาความยาวของทางเดินที่สั้นที่สุดของกราฟรูป 2.13 จาก a ไป z



รูปที่ 2.13 ขั้นตอนเริ่มต้นในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ

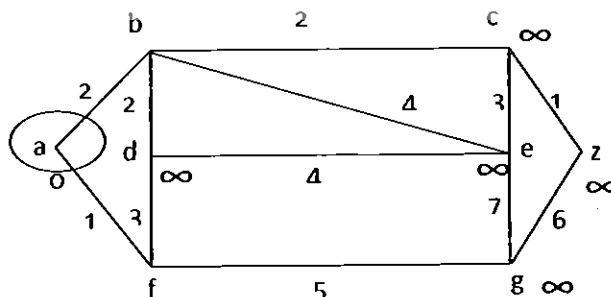
วิธีทำ

เนื่องจากจุดยอดใน T ยังไม่มีวงกลม และค่าประจำจุดยอดชั่วคราว (จุดยอดที่ไม่มีวงกลมจะมีค่าประจำจุดยอดถาวร) รูปที่ 2.14 แสดงผลลัพธ์ที่ได้จากขั้นที่ 1



รูปที่ 2.14 กราฟที่ 1 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ

ในขั้นที่ 2 นี้ z ยังไม่มีวงกลม จึงต่อไปขั้นที่ 3 เลือกจุดยอด a ซึ่งเป็นจุดยอดที่ไม่มีวงกลม และค่าประจำจุดยอดต่ำที่สุด เราวงกลมจุด a ดูรูปที่ 2.15



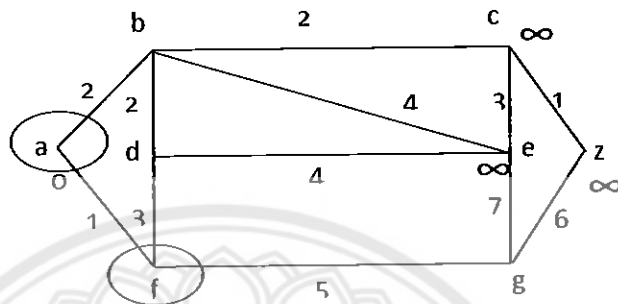
รูปที่ 2.15 กราฟที่ 2 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ

ในขั้นที่ 4 เราปรับปรุงแต่ละจุดยอดที่ยังไม่มีวงกลม b และ f ที่อยู่ติดกับ a โดยเราให้ค่าประจำจุดยอดใหม่ดังนี้

$$L(b) = \min\{\infty, 0+2\} = 2$$

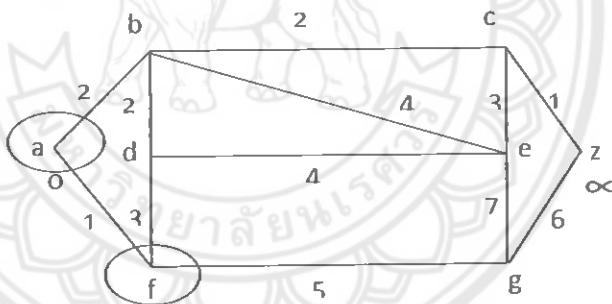
$$L(f) = \min\{\infty, 0+1\} = 1$$

รูปที่ 2.15 แล้วย้อนกลับไปที่ขั้นที่ 2 เพราะ z ยังไม่มีวงกลม ทำขั้นที่ 3 ค่อยไป เลือกจุดยอด f ซึ่งเป็นจุดยอดที่ยังไม่ได้มีวงกลมและมีค่าประจำจุดยอดต่ำที่สุด แล้ววงกลมจุด f รูปที่ 2.16

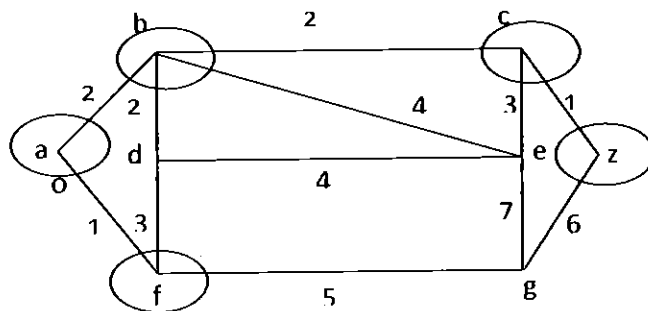


รูปที่ 2.16 กราฟที่ 3 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ

ต่อไปขั้นที่ 4 เราปรับปรุงค่าประจำจุดยอดที่ไม่มีวงกลม d และ g ที่อยู่ติดกับ f ได้ดังรูปที่ 2.17 และทำซ้ำต่อไปจนกระทั่ง จุด z ได้รับค่าประจำจุดถาวร ดังรูปที่ 2.18



รูปที่ 2.17 กราฟที่ 4 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ



รูปที่ 2.18 กราฟที่ 5 ในการหาความยาวของทางเดินที่สั้นที่สุดของกราฟ

เราจบแบบการคำนวณ เมื่อ z ถูกวงกลม (เพราะที่ z มีค่า 5 น้อยกว่าที่ c และ g) นั่นคือทางเดินที่สั้นที่สุดเส้นทางหนึ่ง คือ (a, b, c, z) มีความยาวเท่ากับ 5

ต่อไปเราจะแสดงว่า แบบการคำนวณของคิสตราถูกต้อง โดยใช้อุปนัยเชิงคณิตศาสตร์ บนจำนวนครั้งของการวนซ้ำ (loop) ซึ่งประกอบด้วยขั้นที่ 2 – 4 ที่จะแสดงว่า

ก่อนครั้งที่ 2 จะถูกกระทำ ค่าประจำจุดยอด v ที่มีวงกลม จะให้ความยาวทางเดินที่สั้นที่สุดเส้นทางหนึ่งจาก a ไป v (1)

มีทางเดินที่สั้นที่สุดเส้นทางหนึ่งที่ประกอบด้วยจุดยอดที่มีวงกลมทั้งหมดจาก a ไปจุดยอด v ที่มีวงกลม(2)

ค่าประจำจุดยอด v ที่ไม่มีวงกลมจะให้ความยาวของทางเดินที่สั้นที่สุดในรูป

$$(a, v_1, \dots, v_n)$$

เมื่อ $v = v_n$ และ $a, v_1, \dots, v_{n-1}$ มีวงกลม ถ้าไม่มีทางเดินในรูป

$$(a, v_1, \dots, v_n)$$

เมื่อ $v = v_n$ และ $a, v_1, \dots, v_{n-1}$ มีวงกลม ค่าประจำบน v จะเป็น ∞ (3)

เพราะว่าแบบการคำนวณนี้หยุดเมื่อเราวงกลมที่ z จากข้อ (1) จะได้ว่า $L(z)$ ให้ความยาวของทางเดินที่สั้นที่สุดจาก a ไปยัง z ตามต้องการ

เมื่อเริ่มต้น เราให้ค่าประจำจุดยอด a เป็น 0 และค่าประจำจุดยอดอื่นทั้งหมดเป็น ∞ ณ จุดนี้ ข้อ (1) – (3) เป็นจริง เราได้ตรวจสอบฐานของการอุปนัยว่าใช้ได้แล้ว

สมมติว่า ข้อ (1) – (3) เป็นจริง เป็นครั้งที่ 1 เมื่อมาถึงขั้นที่ 2 พิจารณาการทำซ้ำครั้งต่อไปของแบบการคำนวณ เริ่มด้วยที่ขั้นที่ 3 เราวงกลมจุดยอด v ซึ่งไม่เคยมีวงกลมแต่มีค่าประจำจุดยอดต่ำสุด เราจะแสดงว่า (1) เป็นจริงสำหรับ v นั่นคือ ค่าประจำจุดยอด $L(v)$ ขณะนี้จะให้ความยาวของทางเดินที่สั้นที่สุดจาก a ไป v

สมมติว่ามีทางเดิน $(a, v_1, \dots, v_n)$ (4)

เมื่อ $v = v_n$ และความยาวของทางเดินน้อยกว่า $L(v)$ ในแง่ของ (3) เราต้องมีจุดยอด v_j บางจุดที่ $1 \leq j < n$ ที่ไม่มีวงกลม ให้ k เป็นดัชนีที่เล็กที่สุดที่ v_k ไม่มีวงกลม เพราะน้ำหนักของทุกด้านเป็นค่าบวก ความยาว L' ของทางเดิน

$$(a, v_1, \dots, v_k) \quad \text{.....(5)}$$

จะมีค่าน้อยกว่าความยาวของทางเดินข้อ (4) เพราะ v_k เป็นจุดยอดที่ไม่มีวงกลมเพียงจุดเดียวในทางเดิน (5) ใช้ (3) ทำให้เราเห็นว่าค่าประจำ v_k ใต้อย่างมากที่สุดคือ L' แต่ $L' < L(v)$ เราได้ข้อขัดแย้งกับการเลือก v ดังนั้น (1) เป็นจริงสำหรับ v โดยการโต้แย้งทำนองเดียวกันแสดง

ว่า (2) เป็นจริงสำหรับ v

คราวนี้เราจะต้องแสดงว่า หลังจากปรับปรุงค่าประจำจุดยอดที่ไม่มีวงกลมจุดที่อยู่ชิดกับ v ที่ขั้นที่ 4 แล้วข้อ (3) เป็นจริง เริ่มด้วยแสดงว่า ถ้า x เป็นจุดยอดที่ไม่มีวงกลมและไม่อยู่ชิดกับ v จะได้ $L(x)$ เป็นความยาวของทางเดินที่สั้นที่สุดเส้นหนึ่งในรูปของ

$$(a, v_1, \dots, v_n) \dots\dots\dots(6)$$

เมื่อ $x = v_n$ และ $a, v_1, \dots, v_{n-1}$ เป็นจุดยอดที่มีวงกลมเท่านั้น

พิจารณาทางเดินที่สั้นที่สุดในรูปของข้อ (6) เพราะว่า

$$(a, v_1, \dots, v_{n-1}) \dots\dots\dots(7)$$

เป็นทางเดินที่สั้นที่สุดจาก a ไป v_{n-1} ยิ่งกว่านั้น เพราะ $v_{n-1} \neq v$ จะมีทางเดินทางหนึ่ง ที่ประกอบด้วยจุดยอดที่มีวงกลม (ไม่มีจุดยอดใดเป็น v) จาก a ไปจุดยอด v_{n-1} ซึ่งมีความยาวเท่ากับ ความยาวของ (7) ดังนั้น $L(x)$ เป็นค่าประจำที่ถูกต้องสำหรับ x

คราวนี้สมมติว่า x อยู่ติดกับ v มีทางเดิน 2 ชนิดในรูปของข้อ (6) ชุดหนึ่งมี $v_{n-1} = v$ และ อีกชุดหนึ่งมี $v_{n-1} \neq v$ การโต้แย้งในย่อหน้าก่อนมาแสดงว่า ความยาวของทางเดินที่สั้นที่สุด (6) ที่ $v_{n-1} \neq v$ คือ $L(x)$ ทางเดินที่สั้นที่สุด (6) ที่ $v_{n-1} = v$ ได้จากการนำทางเดินที่สั้นที่สุดจาก a ไป v [ซึ่งมีความยาวเป็น $L(v)$] ต่อด้วยด้าน (v, x) ทางเดินนี้มีความยาว $L(x) + w(v, x)$ ดังนั้น ความยาวของทางเดินสั้นที่สุด (6) คือ

$$\min \{ L(x), L(v) + w(v, x) \}$$

เพราะฉะนั้นหลังจากการปรับปรุงค่าประจำจุดยอดที่อยู่ติดกับ v จะได้ (3) เป็นจริง เราได้แสดงว่าข้อ (1) – (3) เป็นจริงก่อนที่จะกระทำครั้งที่ $i + 1$ ของขั้นที่ 2 ขั้นตอนการอุปนัยเป็นจริง นั่นคือ แบบการคำนวณนี้ถูกต้อง

แบบการคำนวณจากทฤษฎีของคิสตราที่ใช้หาความยาวทางเดินที่สั้นที่สุดจาก a ไป z ในการประยุกต์ส่วนมาก เราต้องการชี้ถึงทางเดินที่สั้นที่สุด การดัดแปลงแบบการคำนวณจากทฤษฎีของคิสตราเพียงเล็กน้อย จะทำให้เราสามารถหาทางเดินที่สั้นที่สุดเส้นทางหนึ่งได้ด้วย

2.3 หลัการ Breath-First Search (BFS)

BFS() ในที่นี้จะใช้อาร์เรย์ i มิติ ($d[0 \dots V-1]$) ในการเก็บสถานะของ vertices ว่าจะถูกทำเครื่องหมายหรือยัง ถ้า vertices i ยังไม่ถูกทำเครื่องหมายจะให้อาร์เรย์เก็บค่า ∞ ไว้ แต่ถ้าถูกทำเครื่องหมายแล้วจะเก็บค่าอื่น การทำงานของอัลกอริทึมจะเสร็จสมบูรณ์เมื่อทุกๆ vertices ถูกทำเครื่องหมายแล้ว และ $d[i]$ จะเก็บจำนวนของเส้นเชื่อมที่น้อยที่สุดจาก vertices ต้นทางมายัง

vertices i

Algorithm : BFS(Graph $G = (V,E)$, vertices s) // s is the source

1. for each $v \in V$ do
2. $d[v] \leftarrow \infty$ // unmark all vertices
3. $d[s] \leftarrow 0$ // mark the source
4. Enqueue (Q,s)
5. while Empty(Q) = false do
6. $v \leftarrow$ Dequeue(Q)
7. for each $u \in$ adjacent [v] do
8. if $d[u] = \infty$ then // is vertices u unmarked
9. $d[u] \leftarrow d[v] + 1$ // mark vertices u
10. Enqueue (Q,u)

ในบรรทัดที่ 1- 2 จะวนทำงานซ้ำเพื่อกำหนดค่าเริ่มต้นให้กับทุกๆ vertices ที่ไม่ได้ถูกทำเครื่องหมายไว้

ในบรรทัดที่ 3 จะทำเครื่องหมายให้กับ vertices ต้นทาง และในส่วนที่เหลือของอัลกอริทึม vertices ที่ถูกดึงออกมาจากคิวจะเป็น vertices เริ่มต้น(s)

สำหรับแต่ละ vertices v ที่ถูกดึงออกมาจากคิว ทุกๆ adjacent ของมันที่ยังไม่ถูกทำเครื่องหมาย จะถูกทำเครื่องหมายในบรรทัดที่ 9 และจะเพิ่มลงไปในคิวในบรรทัดที่ 10

สรุปการทำงานจากอัลกอริทึม BFS() มีกฎ 3 ข้อดังนี้

1. เดินทางไปยังโหนดถัดจาก vertex นั้นๆ ที่ยังไม่เคยถูก visit ทำเครื่องหมาย และเพิ่มเข้าไปในคิว
2. หากไม่สามารถทำตามกฎข้อที่ 1 ได้ เนื่องจากไม่มี vertices ที่ไม่เคยถูก visit แล้ว ถ้าเป็นไปได้ ให้ลบ v ออกจากคิว และทำให้เป็น vertex ปัจจุบัน
3. หากไม่สามารถทำตามกฎข้อที่ 1 ได้ เนื่องจากคิวว่าง ก็เป็นอันเสร็จสิ้น

2.4 หลักการเกี่ยวกับ ASP.NET

2.4.1 ความรู้เกี่ยวกับ ASP.NET

โลกในยุคปัจจุบันนี้กำลังพัฒนาเข้าไปสู่โลกแห่งการสื่อสารแบบไร้พรมแดน ในโลกแห่งเทคโนโลยีปัจจุบันนี้ หลายๆท่านคงคุ้นเคยหรือได้ยินเกี่ยวกับเทคโนโลยีและการบริการพิเศษต่างๆ มาบ้างแล้ว ไม่ว่าจะเป็นจากสื่อโฆษณา โทรทัศน์ บางท่านอาจจะเป็นผู้ใช้เทคโนโลยีนี้อย่างไม่รู้ตัว การเช็คอีเมลล์ เล่นอินเทอร์เน็ต รวมทั้งจองตั๋วชมภาพยนตร์ผ่านทางโทรศัพท์มือถือ สิ่งเหล่านี้ล้วน

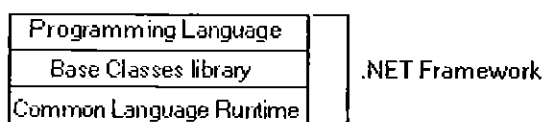
เป็นเทคโนโลยีที่ทันสมัยและน่าทึ่งอย่างยิ่ง ตัวคุณจะรู้หรือไม่ว่าสิ่งเหล่านี้เกิดขึ้นได้อย่างไร และใช้สิ่งใดในการสร้าง

สำหรับผู้พัฒนาโปรแกรมเว็บ (Web Application) ด้วย ASP (Active Server Page) จะทราบว่าการเขียนโปรแกรม ASP เข้าไปเสริม Web Page เพื่อให้เว็บทำงานกับฐานข้อมูลได้ และเปลี่ยนแปลงเนื้อหาของเว็บไซต์ตามการสั่งงานของผู้ใช้ (Dynamic Web) ได้นั้นค่อนข้างง่าย เมื่อเปรียบเทียบกับ การเขียนโปรแกรมด้วยภาษาอื่นๆ แต่ก็มีความยุ่งยากและต้องปรับปรุงให้สมบูรณ์ยิ่งขึ้นไป

Microsoft ได้ออกแบบโครงสร้างสถาปัตยกรรมในการทำงานของโปรแกรมใหม่หมด โดยให้ชื่อว่า .NET Framework ซึ่งเป็นโครงสร้างสถาปัตยกรรมของโปรแกรมที่มีข้อดีอยู่มากมาย จะขอยกกล่าวใน 3 ประเด็นหลัก คือ หนึ่งเป็นโครงสร้างที่รองรับการทำงานของโปรแกรมทุกรูปแบบ (Platform) เช่น Web Application, Windows Application, Mobile Device (PDA, Mobile Phone) ซึ่งจะเป็นประโยชน์มาก กล่าวคือ เมื่อผู้พัฒนาโปรแกรมเรียนรู้และสามารถพัฒนาโปรแกรมในรูปแบบใดรูปแบบหนึ่ง ก็จะสามารถเรียนรู้และพัฒนาโปรแกรมอีก รูปแบบหนึ่งได้อย่างรวดเร็ว

2.4.2 ความรู้เกี่ยวกับ .NET Framework

จากการที่ Microsoft ต้องการที่จะสร้าง ระบบภาษาสักอย่างที่เป็นมาตรฐานขึ้นมา เพื่อให้ทุกสิ่งทุกอย่าง เพื่อให้อุปกรณ์ทุกสิ่งทุกอย่าง สามารถติดต่อสื่อสารกันได้หมด Microsoft ได้คิดค้นระบบซึ่งหมายถึงมันนั้นมือว่าจะให้เป็นระบบมาตรฐาน ระบบนี้คือ .NET Framework ซึ่งระบบนี้ไม่ใช่ระบบปฏิบัติการ(OS) แต่เปรียบเสมือนโปรแกรมหนึ่งที่จะสามารถสร้างสภาวะแวดล้อมหนึ่ง ซึ่งสามารถทำงานใน ระบบ .NET นี้ได้ ในอนาคต Microsoft ก็หวังที่จะนำเอา ระบบ .net นี้ไปติดตั้งลงบนอุปกรณ์ทุกชนิด เพื่อให้ทุกอุปกรณ์ทุกอย่างมีระบบฯหนึ่งที่เหมือนกันหมด โดย .NET Framework นั้นมีส่วนประกอบภายในแบ่งออกเป็น 3 ชั้นใหญ่ๆ คือ



รูปที่ 2. 19 โครงสร้างของ .NET Framework

1. Programming Language : เป็นรูปแบบของภาษาที่ออกแบบ มาเพื่อให้สามารถทำงานในสภาวะที่เป็น .NET ได้โดยที่ทาง Microsoft ได้เปิดตัวภาษาหลักๆ ที่จะใช้ในการพัฒนา บน

.NET นี้ 3 ภาษา C# เป็นภาษาใหม่ที Microsoft พัฒนามาจาก C++ กับ JAVA เป็นหลัก VB.NET เป็นภาษาที่พัฒนามาจาก Visual Basic ในเวอร์ชัน 6.0 JScript.net เป็นภาษาที่พัฒนามาจาก JScript ซึ่งเป็น JavaScript ใน เวอร์ชัน ของ Microsoft

2. Base Classes Library : Library นั้นเปรียบเสมือนชุดคำสั่งสำเร็จรูปย่อยๆที่เพิ่มเข้ามา ซึ่งส่วนใหญ่จะเป็นชุดคำสั่งที่ต้องใช้งานอยู่เป็นประจำนั่นจึงมีผู้คิดค้นเครื่องอำนวยความสะดวกในการเขียนโปรแกรมซึ่ง Library ในภาษาต่างๆส่วนใหญ่จะอยู่ในรูปแบบไฟล์ include แต่ถ้าเป็น ASP สิ่งที่เป็น library ก็คือ component ต่างๆนั่นเอง ซึ่งภายในระบบ .NET จะสร้างสิ่งที่เรียกว่าเป็น Library พื้นฐานขึ้นทำให้ไม่ว่าจะใช้ภาษาใดในการพัฒนาโปรแกรมก็สามารถที่จะเรียกใช้ Library ที่เป็นตัวเดียวกันได้หมด

3. Common Language Runtime (CLR) : นับเป็น สิ่งสำคัญแทบจะที่สุดของระบบ .NET นี้ก็ว่าได้ เพราะ CLR ที่ว่านี้มีหน้าที่ทำให้โปรแกรมที่เขียนขึ้นมาด้วยภาษาต่างๆกัน กลายเป็นภาษารูปแบบมาตรฐานเดียวกันทั้งหมด เราเรียกภาษาที่ว่านี้ว่า Intermediate language (IL) ซึ่งเมื่อต้องการที่จะรันโปรแกรมใด CLR ที่ว่านี้จะตรวจสอบเครื่องที่รันว่ามีสภาวะแวดล้อมการทำงานเช่นใด หลังจากนั้นก็จะคอมไพล์เป็น โปรแกรมที่เหมาะสมต่อการทำงานของเครื่องนั้น ทำให้เราสามารถใช้งานโปรแกรมต่างๆได้อย่างมีประสิทธิภาพสูงสุดในแต่ละเครื่อง

ประโยชน์และข้อดีของ .NET Framework นั้นพอจะสรุปออกมาได้เป็นข้อๆ ดังนี้

1. เป็นระบบที่มีไลบรารีที่เป็นมาตรฐานเดียวกัน : เนื่องจากมีไลบรารีที่เป็นมาตรฐานเดียวกันทั้งหมดทำให้เราไม่ต้องกังวลว่าภาษาที่ใช้เขียนนั้นมีไลบรารีตัวนั้นตัวนี้หรือไม่ รวมทั้งไม่ต้องคอยกังวลว่าถ้าใช้ไลบรารีของภาษาหนึ่งแล้วอีกภาษาหนึ่งจะไม่มีไลบรารีตัวนั้น

2. ไม่ขึ้นกับระบบปฏิบัติการ (OS) : เนื่องจากระบบปฏิบัติการ ที่แต่ละบุคคลหรือองค์กรใช้นั้นย่อมไม่เหมือนกัน แต่ภายใน .NET Framework จะไม่มีปัญหานี้ของเพียงแค่มีระบบ .NET Framework ก็จะทำให้สามารถใช้งานโปรแกรมต่างๆได้ ซึ่งเป็นข้อดีตรงที่เราจะสามารถใช้โปรแกรมต่างๆได้ทุกระบบปฏิบัติการ

3. ใช้ในการพัฒนาได้ทุกภาษา : ทำให้เราไม่ต้องคอยมาศึกษาภาษาใหม่ๆเมื่อต้องการสร้างโปรแกรมในแต่ละครั้ง นอกจากนั้นเรายังสามารถเลือกใช้ภาษาที่เราถนัดที่สุดในการพัฒนาโปรแกรมต่างๆได้ด้วย

4. มีการควบคุมสิ่งแวดล้อมในการทำงานเป็นอย่างดี : เนื่องจากเป็นระบบที่เป็นมาตรฐาน ทำให้การควบคุมจัดสรรระบบต่างๆ ทำได้ง่ายขึ้น ไม่ว่าจะเป็นการจัดสรรหน่วยความจำ ส่วนด้านการใช้งานเครื่องก็มีความรวดเร็วมากขึ้นลด โอกาสที่เครื่องจะค้างได้เป็นอย่างดี

5. ความปลอดภัยที่มีมากขึ้น : .NET Framework สามารถกำหนดสิทธิ์การใช้งานหรือ

permission ของผู้ใช้งานได้มากขึ้นทำให้สามารถกำหนดว่า จะให้โปรแกรมในส่วนใดใช้งานได้หรือไม่ได้แล้วแต่เฉพาะบุคคล

ทั้งหมดนี้เป็นเพียงประโยชน์ส่วนหนึ่งในแนวคิดของไมโครซอฟท์ที่กำลังจะพัฒนาให้สำเร็จเท่านั้น บางข้อทำได้สำเร็จแล้วแต่บางข้อก็ยังไม่สำเร็จดังนั้นจึงต้องยอมรับดูว่า ฝันของไมโครซอฟท์จะเป็นจริงและสำเร็จได้เมื่อไร

2.4.3 ASP.NET กับ การพัฒนาโปรแกรม

ในการพัฒนา Web Application ซึ่งเป็น Platform หนึ่งของ .NET Framework นั้น Microsoft ได้ปรับปรุงเทคโนโลยีการพัฒนาโปรแกรม ASP ขึ้นใหม่โดยขยายความสามารถ ลดความยุ่งยาก ลดเวลาในการเขียนโปรแกรมของผู้พัฒนา เพิ่มความสะดวก และที่สำคัญทำให้โปรแกรมเว็บทำงานได้รวดเร็วขึ้นอย่างน่าประทับใจและเป็นไปตามมาตรฐาน .NET Framework คือรองรับภาษาในการพัฒนาที่หลากหลาย พอจะสรุปประเด็นข้อดีหลักๆ ได้ดังนี้

1. ใช้ ภาษา ใดๆ ในการเขียน script ก็ได้ : จากเดิมที่เราสามารถใช้ได้เฉพาะภาษาที่เป็น script ของ VBScript และ JScript แต่ใน ASP.NET เราสามารถใช้ภาษาที่มีรูปแบบของภาษาเต็มๆ ซึ่งในเบื้องต้น มี 3 ภาษา คือ C#, VB.NET และ JScript.Net ที่ออกมาเป็น มาตรฐาน แต่ในอนาคต Microsoft มีแผนที่จะเพิ่มตัวแปลภาษาให้ครบทุกภาษา

2. มีความยืดหยุ่นในการเขียนโปรแกรมมากขึ้น : โดยที่เราสามารถใช้ภาษาในการเขียน ASP.NET ได้มากกว่า 1 ภาษาภายในไฟล์เดียวกัน ทำให้สามารถเลือกรูปแบบของภาษาที่ง่ายที่สุดต่อการเขียนในแต่ละส่วนได้

3. ลักษณะการแปลภาษาและนามสกุลไฟล์เปลี่ยนไป : ใน ASP เวอร์ชันก่อนๆ มีลักษณะการแปลภาษาเป็นแบบอินเตอร์พรีเตอร์ (Interpreter) คือการจะทำคำสั่งใดค่อยแปลคำสั่งนั้น แต่ในเวอร์ชัน .NET นี้จะมีลักษณะเป็นคอมไพเลอร์ (Compiler) คือการแปลคำสั่งรวมทั้ง โปรแกรม นอกจากนี้นามสกุลของไฟล์ก็มีการเปลี่ยนแปลงจากเดิมที่ใช้นามสกุลไฟล์เป็น “ *.asp ” เป็น “ *.aspx ”

4. รูปแบบและการใช้งาน component ที่ง่ายขึ้น : รูปแบบของ component จะเน้นไปที่ XML มากที่สุดและที่สำคัญคือการใช้งาน component ใน ASP.NET นั้นเราสามารถอัปโหลดไฟล์ไปไว้ใน Directory ที่ผู้ดูแล server (Admin) กำหนดหลังจากนั้น component จะติดตั้งตัวเองโดยอัตโนมัติ ลดปัญหาที่เกิดจาก ASP เวอร์ชันก่อนๆ ได้เป็นอย่างดี เนื่องจากใน ASP เวอร์ชันก่อนนั้น การติดตั้ง component กระทำได้เพียงผู้ดูแล server เพียงคนเดียวเท่านั้น ทำให้เวลาต้องการใช้ component ต่างๆ ที่ server ไม่มีจึงเป็นเรื่องที่ลำบาก

5. มี Library ให้เลือกใช้ได้มากขึ้น : ใน ASP เวอร์ชันก่อนๆ นั้น Application บางอย่างสร้าง

ได้ไม่สะดวกนัก ต้องอาศัย component ต่างๆมากมาย แต่ใน ASP.NET นั้นได้เพิ่ม Library ในส่วนเหล่านี้ให้กลายเป็น พื้นฐานของการใช้งาน

6. มีคอนโทรลทำให้การใช้งานในบางสิ่งง่ายขึ้น : เป็นส่วนพิเศษที่เพิ่มเติมมาจาก ASP รุ่นก่อนๆ ที่ไม่มีส่วนที่เรียกว่าคอนโทรล ซึ่งคอนโทรลนี้จะช่วยให้เราสามารถ สร้างเว็บไซต์ได้ อย่างง่ายและมีประสิทธิภาพมากขึ้น จึงไม่ต้องกังวลว่า browser รุ่นนั้นรุ่นนี้จะรองรับกับภาษาที่เราเขียนหรือไม่

7. สามารถเรียกขอข้อมูลจาก server ได้ : ใน ASP เวอร์ชันก่อนๆ server สามารถเรียกขอข้อมูลได้จากเครื่องผู้ใช้นั้นแต่ใน ASP.NET เครื่อง server สามารถเรียกขอข้อมูลจากเครื่อง server ด้วยกันได้

8. ไม่ต้องต่อ Hardware : เนื่องจากเป็นระบบใน .NET Framework ดังนั้นจึงมีคุณสมบัติของ Common Language Runtime (CLR) ทำให้มีการคอมไพล์โปรแกรมเป็นภาษามาตรฐานที่เรียกว่า IL ก่อน ดังนั้นไม่ว่าคุณจะเล่นเครื่องปาล์มหรือคอมพิวเตอร์แบบพกพา ก็ไม่เกิดปัญหา

9. ง่ายต่อการหาข้อผิดพลาดในการเขียนโปรแกรม : หากเป็น ASP รุ่นก่อนเวลาเกิดความผิดพลาด (error) เครื่องจะบอกแค่ว่าเป็นความผิดพลาดชนิดใดบรรทัดไหนแต่ใน ASP.NET นี้เครื่องจะแสดงรายละเอียดที่มากขึ้นพร้อมแนวทางแก้ไข

10. มีการตรวจสอบเหตุการณ์ต่างๆได้ภายในเว็บเพจ : มีการตรวจสอบเหตุการณ์ต่างๆ ตั้งแต่โหลดหน้าเว็บเพจไปจนถึงปิดหน้าเว็บเพจลงทำให้เราสามารถเขียน โปรแกรม เพื่อทำการ กำหนดเหตุการณ์ต่างๆได้ง่ายขึ้น

11. แยกส่วนที่เป็น HTML กับ ASP ออกมาอย่างชัดเจน : ในเวอร์ชันก่อนๆส่วนที่เป็น HTML กับ ASP จะเขียนปนกันไปมาแต่ในเวอร์ชันนี้จะแยกส่วนกันอย่างชัดเจนว่าส่วนไหนเป็น HTML และส่วนไหนเป็น ASP

2.5 ภาษา C# ระดับพื้นฐาน

ภาษา C# จะเขียนว่า C Sharp ก็ได้ (อ่านว่าซีชาร์พ) เครื่องหมาย # ในทางคณิตศาสตร์หมายถึง ครึ่งเสียง ส่วนคำว่า Sharp หมายถึงเฉียบคม มีไหวพริบ บริษัทไมโครซอฟท์ตั้งชื่อภาษานี้ว่า C# เพื่อจะสื่อให้เห็นว่านี่คือภาษาในตระกูลภาษา C เช่นเดียวกับภาษา C++ คือเป็นภาษาที่มีต้นกำเนิด จากภาษา C หากท่านสังเกตให้ดีจะเห็นว่าสัญลักษณ์ # จะมองให้เป็นเครื่องหมาย + สื่อนำมาซ้อน กันก็ได้ แสดงเป็นนัยว่าภาษานี้ก้าวหน้ากว่าภาษา C++ ไปอีกระดับหนึ่ง (คือเป็นภาษา C+++) สัญลักษณ์ # นี้มีปรากฏในภาษาอื่นๆ ของ .NET ด้วย เช่น ภาษา J# (เดิมคือภาษา J++) และภาษา A# (เหมือนภาษา Ada) อาจเป็นเพื่อเหตุผลทางการค้า

ภาษา C# เป็นภาษาที่ใหม่มากๆ ปรากฏตัวเป็นครั้งแรกในปี 2000 และถูกอัปเดตเวอร์ชัน อยู่เสมอ มันเป็นภาษาที่ได้รับอิทธิพลจากภาษาก่อนหน้าเช่นภาษา Delphi ภาษา C++ ภาษา Java

และภาษา Eiffel ในตอนต้นภาษานี้ถูกออกแบบ และกำหนดลักษณะโดยบริษัทไมโครซอฟท์ ต่อมา ได้ถูกรับรองจากหน่วยงาน ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO แต่ปัจจุบันไมโครซอฟท์ยังพัฒนาภาษานี้อย่างต่อเนื่อง (ปัจจุบันเป็นเวอร์ชัน 3.0)

ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็น OOP อย่างถึงที่สุด ขณะเดียวกันก็ลดความซับซ้อนใน โครงสร้างของภาษาลง (เรียบง่ายขึ้นกว่าภาษา C++) และมีเครื่องแต่งตัวน้อยลง (เมื่อเทียบกับ Java)

ผู้สร้างภาษา C# คือบริษัทไมโครซอฟท์ แต่บิดาของภาษา C# คือ Anders Hejlsberg (แอนเดรส ฮายสเบิร์ก) ไมโครซอฟท์ต้องการให้ภาษา C# เป็น “อะไรที่จะอยู่ไปอีกนาน” เหมือนบริษัทรถยนต์โฟล์คที่จ้าง Ferdinand Porsche (เฟอร์ดินันด์ พอร์ช) นักออกแบบรถยนต์มีชื่อเสียงออกแบบรถโฟล์คเต่า (เป็นเหตุการณ์ที่เกิดขึ้นก่อนสงครามโลกครั้งที่สอง) ทำให้มันกลายเป็นรถยนต์คลาสสิกมาจนถึงปัจจุบัน ในทำนองเดียวกันเมื่อต้องการภาษาที่ต้องการให้กลายเป็นภาษา “คลาสสิก” บริษัทไมโครซอฟท์ตัดสินใจมอบหมายให้ Hejlsberg บรมครูนักออกแบบภาษา ผู้เคยสร้างภาษาที่กลายเป็นตำนานมาแล้วเช่น Turbo Pascal และผู้นำในทีมสร้างภาษา Delphi

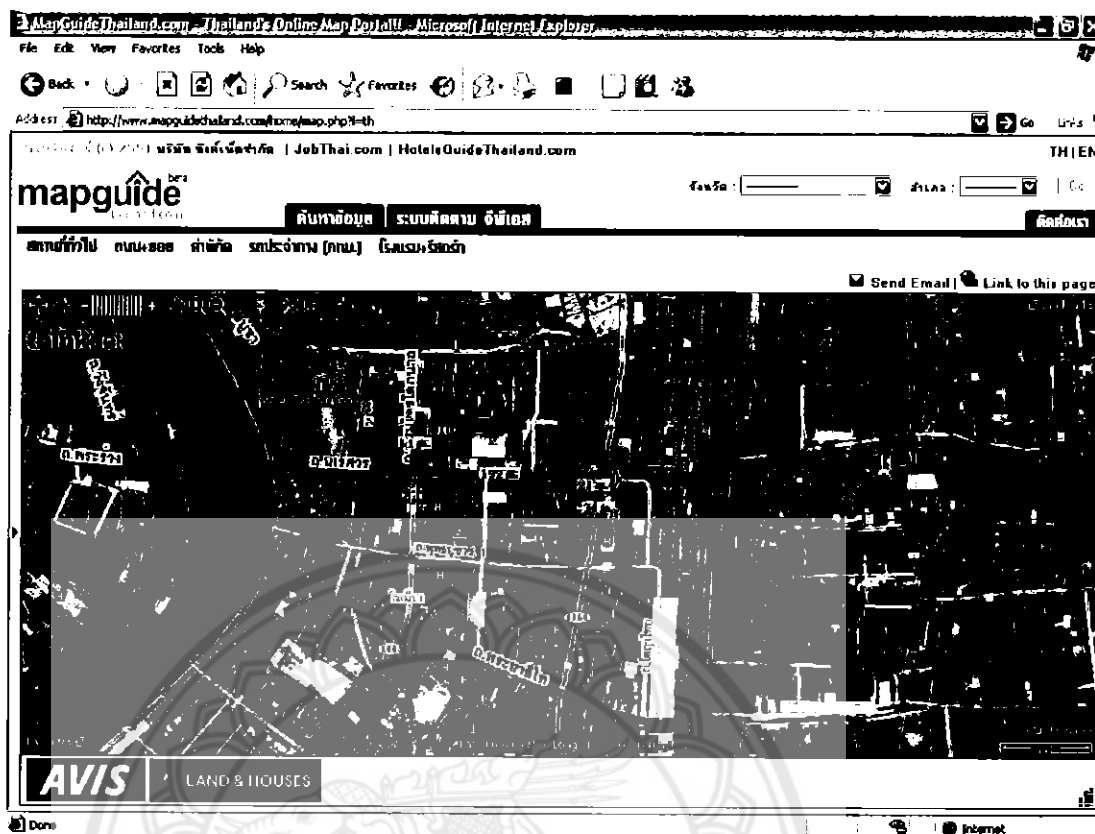
บทที่ 3

วิธีการดำเนินงาน

จากบทที่ 2 ผู้จัดทำได้เลือกภาษา C# ในการพัฒนาเว็บไซต์เพื่อบริการผู้โดยสารในจังหวัดพิษณุโลก โดยมีขั้นตอนในการดำเนินการดังต่อไปนี้

3.1 การค้นคว้าข้อมูล

ทำการรวบรวมข้อมูลของสายรถประจำทางจากศูนย์ท่ารถและสถานีรถไฟ ซึ่งเป็นศูนย์รวมของรถโดยสารประจำทางในจังหวัดพิษณุโลก ซึ่งไปทำการสอบถามและขอข้อมูลจากนายท่า ซึ่งเป็นผู้ดูแลประจำที่นั่น จากการสอบถามได้ให้ความรู้ที่เป็นประโยชน์ในการศึกษาเส้นทางจากแผนที่ และรวบรวมข้อมูลเพิ่มเติมจากการสอบถามพนักงานขับรถทำให้ทราบเส้นทางในการเดินทางที่ดียิ่งขึ้นกว่าเดิมและได้รับข้อมูลการท่องเที่ยวจากการท่องเที่ยวแห่งประเทศไทยและศูนย์บริการนักท่องเที่ยวภายในวัดพระศรีรัตนมหาธาตุ (วัดใหญ่) ในรูปแบบของแผนที่จังหวัดพิษณุโลก แผนที่ในเขตเทศบาลนครพิษณุโลก และข้อมูลการท่องเที่ยวในจังหวัดพิษณุโลก ส่วนรายละเอียดเกี่ยวกับการท่องเที่ยวได้ค้นคว้าข้อมูลจากอินเทอร์เน็ตเพิ่มเติม ในด้านของข้อมูลระยะทางได้ใช้เว็บไซต์ <http://www.mapguidethailand.com/home/map.php> แล้วกำหนดจุดลงบนแผนที่ แล้วทำการลากเส้นเพื่อทำการวัดก็จะปรากฏระยะทางขึ้นมาหน่วยเป็นเมตร ซึ่งจะได้ระยะที่ใกล้เคียงกับความจริงมากที่สุดเพราะว่าทำการวัดแบบโค้งตามถนนจริง แสดงดังภาพที่ 3.1 ซึ่งวัดระยะจากสถานีรถไฟ ไปยัง สภอ.เมืองพิษณุโลก



รูปที่ 3.1 หน้าเว็บไซต์แสดงการหาระยะทางระหว่างจุดจากสถานีรถไฟไป สกอ.เมืองพิษณุโลก

3.2 การออกแบบเว็บเพจ

3.2.1 ตรวจสอบ Requirement Specification หรือข้อมูลรายละเอียดที่ต้องการ ใส่ลงในเว็บเพจว่าเราต้องการใส่อะไรลงไปบ้าง โดยความต้องการนั้นมีดังนี้

1. ต้องการทำให้เว็บไซต์เพื่อช่วยในการตัดสินใจในการใช้บริการรถโดยสารประจำทางในจังหวัดพิษณุโลก

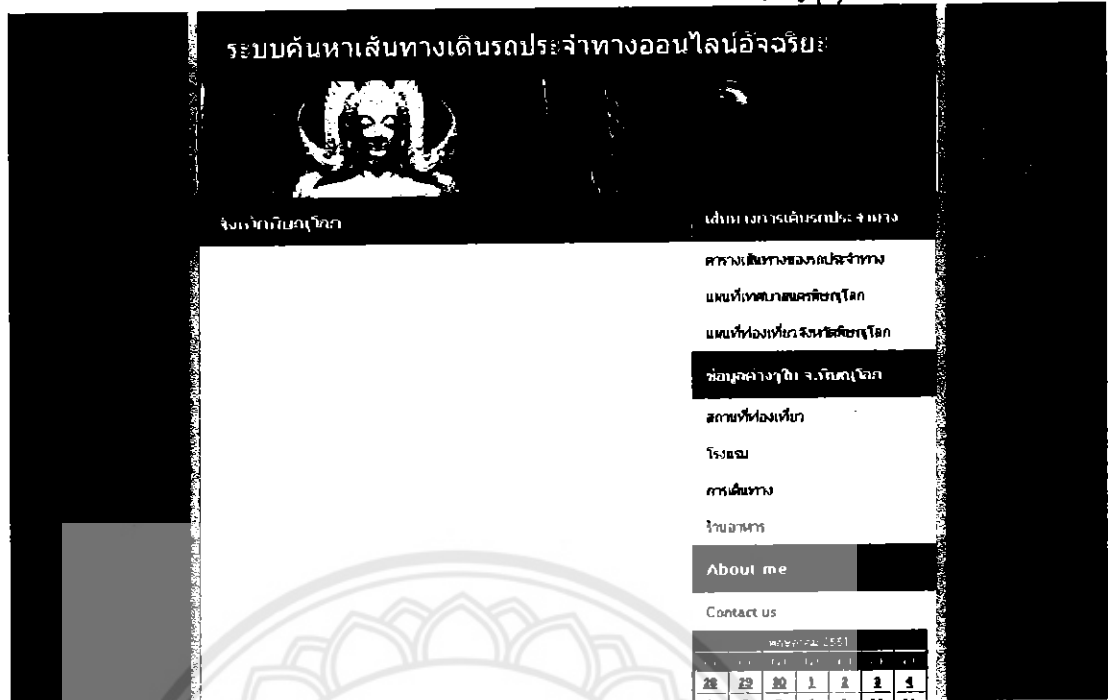
2. ค้นหาข้อมูลตามที่ User ต้องการได้

3. มีความถูกต้องของข้อมูล

4. มีประสิทธิภาพพร้อมใช้งาน

3.2.2 การออกแบบหน้าเว็บเพจ

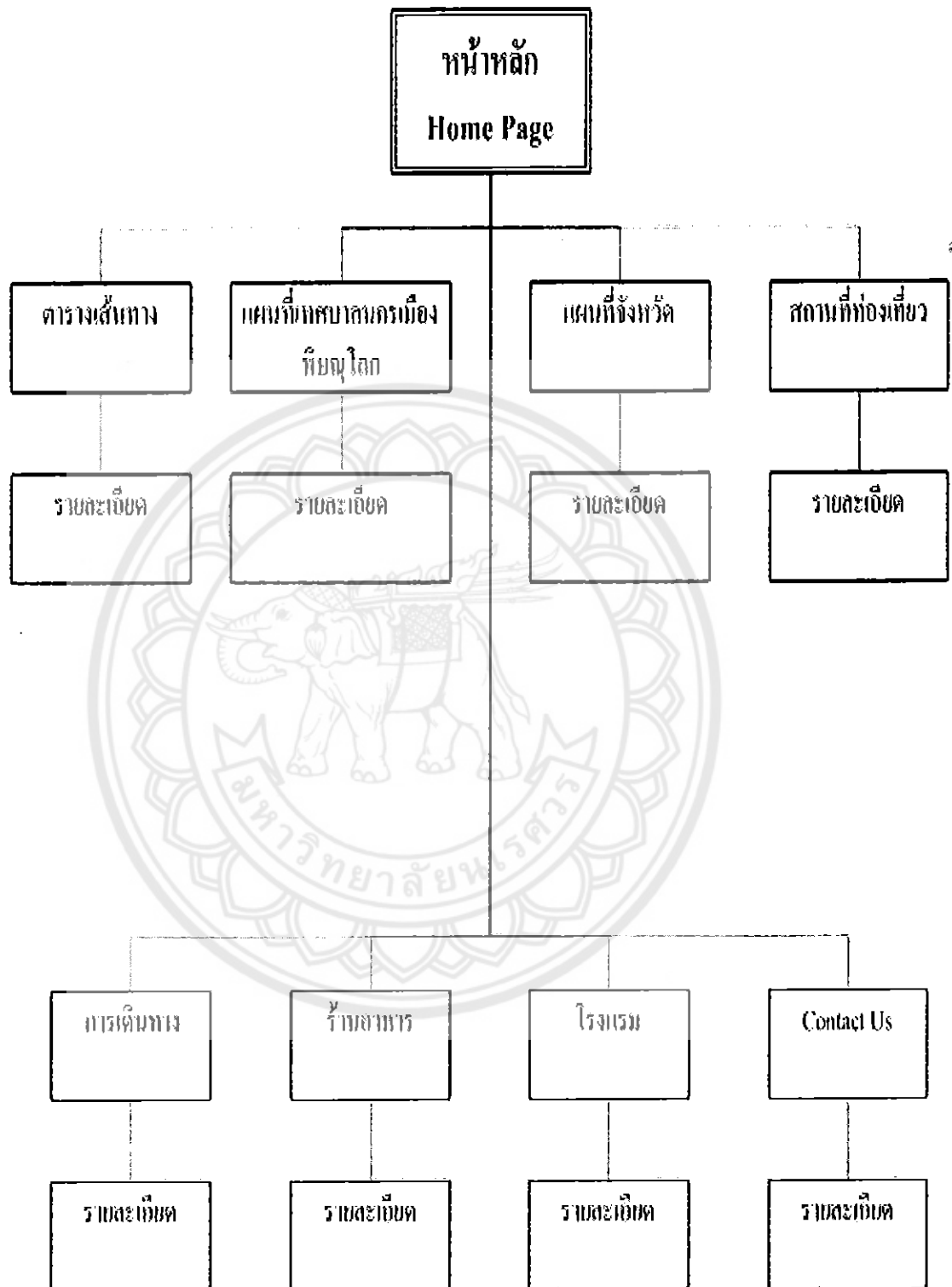
ในส่วนของหน้าเว็บเพจที่จะแสดงนั้นออกแบบเป็นส่วนแสดงข้อมูลและส่วนที่เป็นเมนูเชื่อมซึ่งจะอยู่ทางด้านขวาของเว็บเพจ แสดงดังรูปที่ 3.2 มีหัวข้อหลัก และมีหัวข้อย่อยแสดงในแนวตั้งทางด้านขวา อีกทั้งด้านบนที่เป็นชื่อของเว็บเพจคือ “ระบบค้นหาเส้นทางเดินรถประจำทางออนไลน์อัจฉริยะ” ซึ่งปรากฏในทุกหน้าของเพจนี้นจะสามารถเชื่อมต่อไปยังหน้าเว็บเพจซึ่งมีโปรแกรมค้นหาเส้นทาง ซึ่งไม่ว่าเราจะอยู่ในเพจไหน เราจะสามารถใช้บริการการค้นหาเส้นทางเดินรถประจำทางได้



รูปที่ 3.2 หน้าเว็บเพจที่ออกแบบเป็นหลักของทุกหน้าประกอบไปด้วยเมนูเชื่อมต่อให้เลือกทางด้านขวา

เว็บไซต์นี้ใช้ ASP.NET ในการสร้าง User Interface ติดต่อกับผู้ใช้ โดยแบ่งออกเป็นสองส่วนคือส่วนแรก รายละเอียดสถานที่ รายละเอียดของเส้นทางการเดินรถแต่ละสาย สุดท้ายเป็นการประมวลผลเพื่อหาสายรถประจำทางที่ดีที่สุด

3.2.3 การเชื่อมโยงหน้าเว็บเพจ



รูปที่ 3.3 การเชื่อมโยงหน้าเว็บเพจ

3.3 การพัฒนาโปรแกรม

จากการศึกษาข้อมูลเส้นทางเดินรถประจำทาง และเส้นทางในเขตเทศบาลนครพิษณุโลก แล้ว ในการพัฒนาโปรแกรมระบบค้นหาเส้นทางรถประจำทางในเขตเทศบาลนครพิษณุโลก ผู้พัฒนาได้ออกแบบโครงสร้างโปรแกรม ออกเป็น ส่วนหลักๆ คือ

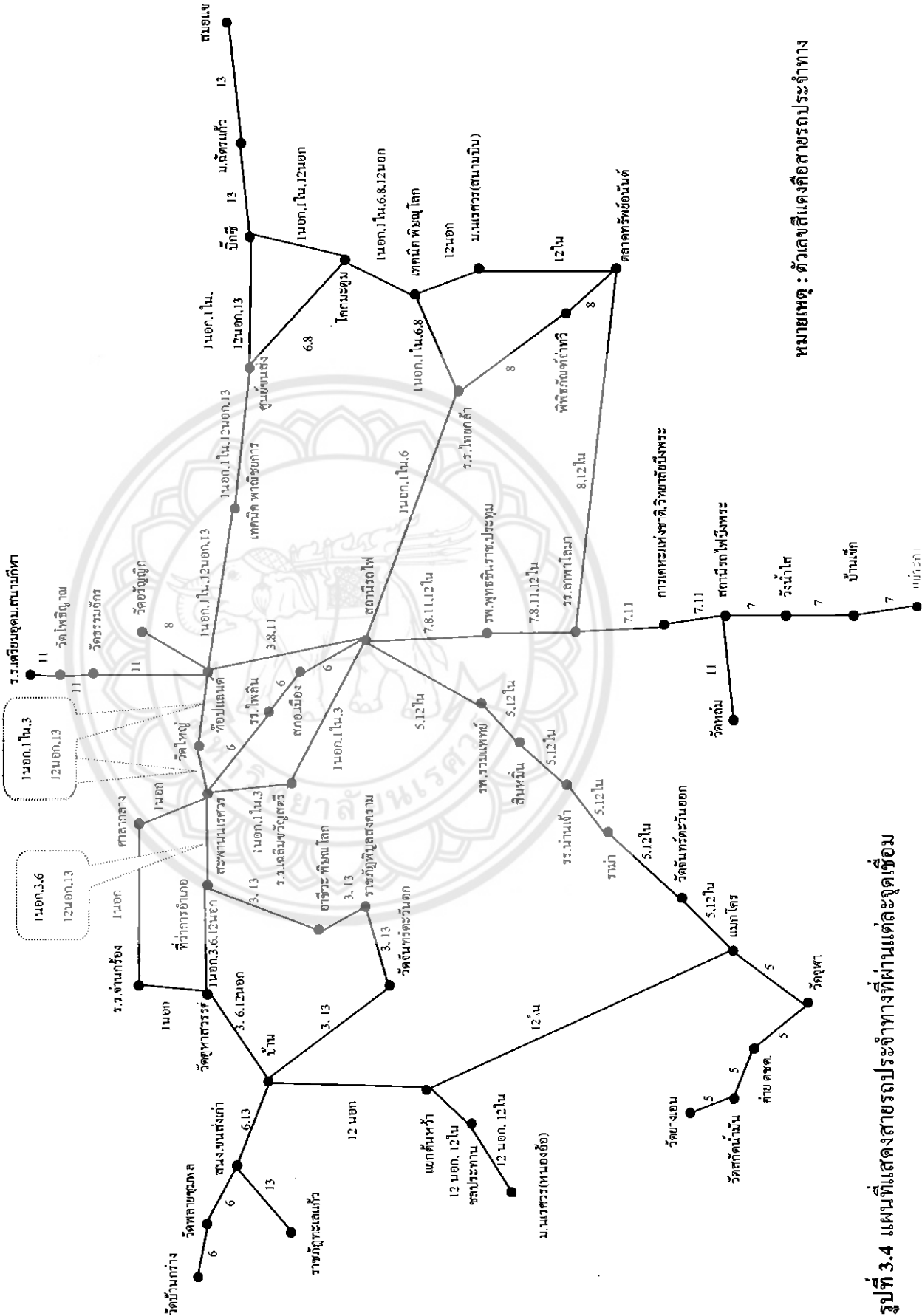
1. ส่วนการหาเส้นทางของรถประจำทาง
2. ส่วนการเปรียบเทียบเส้นทางที่หาได้จาก algorithm กับสายรถประจำทาง

3.3.1 ส่วนการหาเส้นทางของรถประจำทาง

ในส่วนนี้ทางผู้พัฒนาโปรแกรมได้ใช้หลักการของคิจสตรา (Dijkstra's algorithm) ทำการหาเส้นทางที่สั้นที่สุดจากบัพ (node) ซึ่งข้อมูลเข้า (input) มาจากการเลือกข้อมูลสถานที่ใน dropdownlist แล้วแปลงค่าที่ได้รับเป็นตัวเลข ส่งไปคำนวณหาเส้นทางที่สั้นที่สุดในส่วนของการคำนวณเส้นทางที่สั้นที่สุด แล้วแสดงค่าระยะทางที่สั้นที่สุดออกมา (output)



แผนที่แสดงสายรอบประจำทางที่ผ่านแต่ละจุดเชื่อมต่อต่างๆ ในจังหวัดพิษณุโลก



หมายเหตุ : ตัวเลขสีแดงคือสายรถประจำทาง

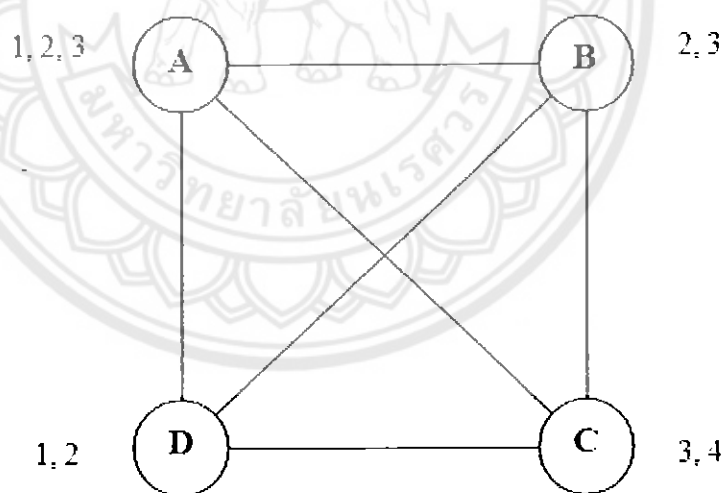
รูปที่ 3.4 แผนทแสดงสายรถประจำทางที่ผ่านแต่ละจุดเชื่อม

3.3.2 ส่วนการเปรียบเทียบเส้นทางที่หาได้จาก algorithm กับสายรถประจำทาง

ในส่วนนี้เมื่อได้เส้นทางจากส่วนการค้นหาเส้นทางเดินรถประจำทาง ที่จะขึ้นจากจุดหนึ่งไปยังอีกจุดหนึ่งนั้น เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด จะต้องมีการทำให้คอมพิวเตอร์เรียนรู้เสมือนมนุษย์เสียก่อน เราจึงได้นำหลักการปัญญาประดิษฐ์เข้ามาช่วยแก้ไขปัญหานี้ ซึ่งวิธีการค้นหาข้อมูลก็สามารถทำได้หลายวิธี หนึ่งในนั้นคือ Breath First Search จะทำให้การตัดสินใจของคอมพิวเตอร์ถูกต้องมากที่สุด

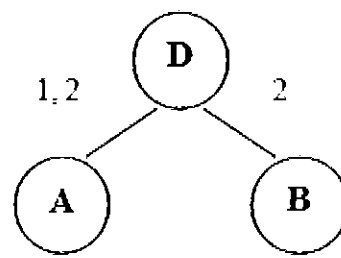
การหาสายรถประจำทาง ที่จะขึ้นนั้น ไม่ขึ้นอยู่กับการเดินทางที่ใกล้ที่สุด เพราะรถประจำทางจะวิ่งไปตามเส้นทางที่กำหนดไว้ จะมีการอ้อมไปมา ทำให้ระยะทางที่ได้เพิ่มขึ้น ซึ่งขั้นตอนที่ใช้ระบุวิธีการขึ้นรถประจำทางจากสถานที่หนึ่งไปยังอีกสถานที่หนึ่งจะเริ่มจากการพิจารณาที่จุดแรกว่า มีสถานที่ใกล้เคียงใดบ้างที่มีสายรถประจำทางเดียวกัน แล้วเก็บสายรถประจำทางที่เหมือนกันไว้ ต่อมาทำการพิจารณาในสถานที่ถัดไปเช่นเดียวกันไปเรื่อยๆ แบบ Breath First Search จนกระทั่งถึงสถานที่ปลายทางแล้วจึงหยุด หลังจากนั้นให้ไล่จากสถานที่สุดท้ายขึ้นไปยังสถานที่เริ่มต้นว่าผ่านสถานที่ใดบ้างและนั่งรถประจำทางสายใด เป็นการแสดงข้อมูลในการเดินทางด้วยรถประจำทางที่ถูกต้อง สุดท้ายจัดการข้อมูลที่ซ้ำซ้อนกันออก จะได้ผลลัพธ์ที่สมบูรณ์ออกมา

ตัวอย่างการหาสายรถประจำทางจาก D ไป C



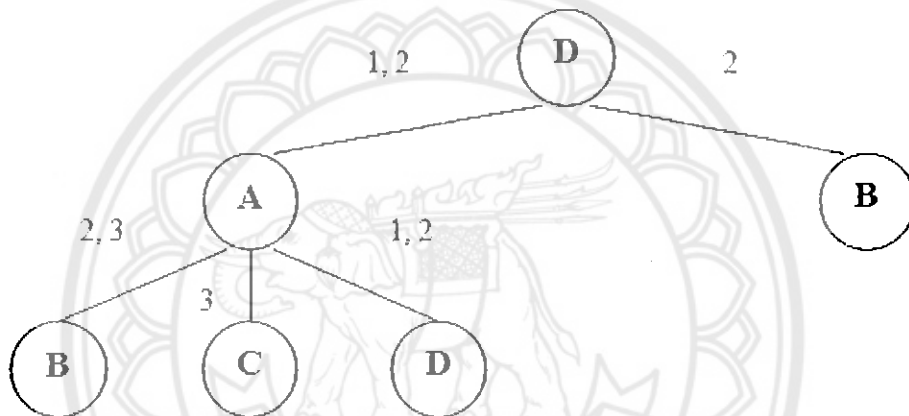
รูปที่ 3.5 การเชื่อมโยงของสถานที่ต่างๆ และสายรถประจำทางที่ผ่าน

เริ่มจากสถานที่ D มีสายรถประจำทางเหมือนกัน A และ B จึงแตกออกได้ 2 สถานที่ดังนี้



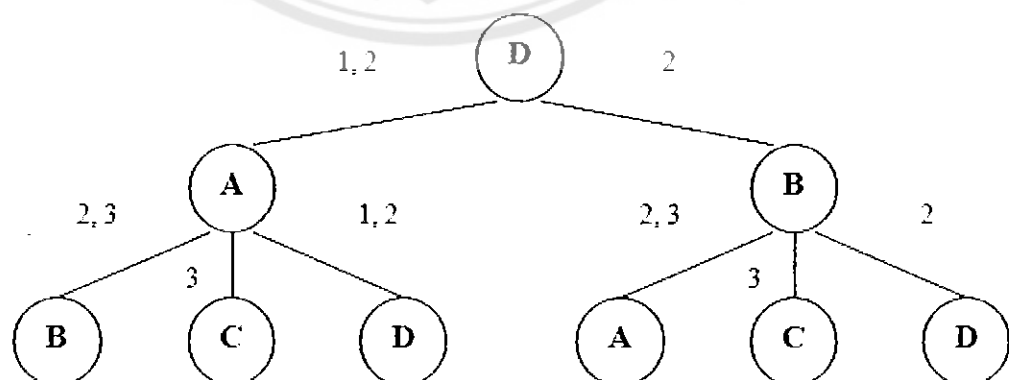
รูปที่ 3.6 กราฟแตกสถานที่ D ออก

ต่อมาสถานที่ที่มีสายรถประจำทางเหมือนกับ A คือ B, C และ D จึงแตกสถานที่ได้ดังนี้



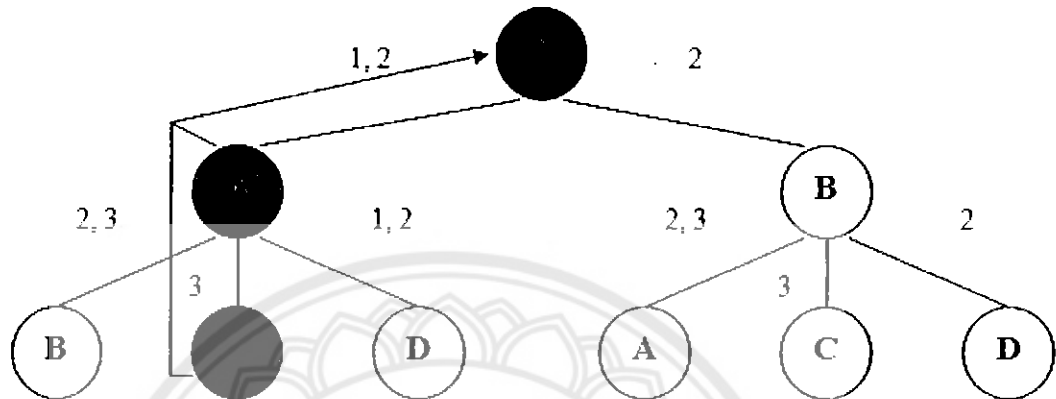
รูปที่ 3.7 กราฟแตกสถานที่ A ออก

หลังจากนั้น B ก็แตกออกได้ A, C และ D



รูปที่ 3.8 กราฟแตกสถานที่ B ออก

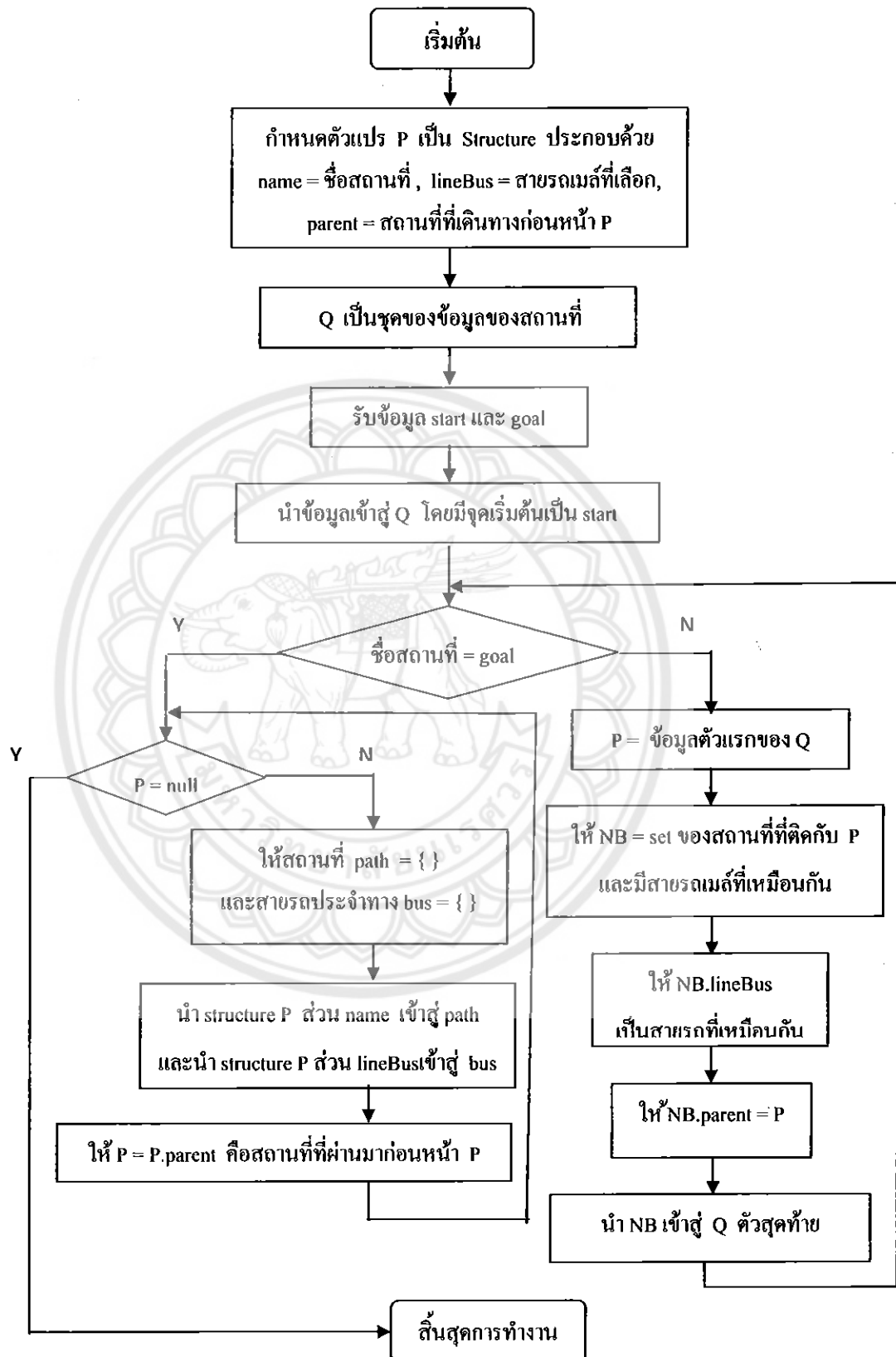
สุดท้ายเมื่อแตกถึง C จึงหยุดเพราะถึงสถานที่ที่ต้องการ แล้วทำการเก็บเส้นทางย้อนกลับ และสายรถประจำทางที่ได้เป็น $C \leftarrow (3) A \leftarrow (1,2) D$ หมายความว่า การเดินทางจาก D ไป C ต้องเริ่มจาก สถานี D นั่งรถสาย 1 หรือ 2 ไปลงสถานีที่ A แล้วต่อรถจากสถานี A ด้วยสาย 3 ไปลงสถานี C



รูปที่ 3.9 กราฟแตกสถานที่จนถึง C แล้วทำการเก็บเส้นทางย้อนกลับ

ในการทำงานจริงนั้น ถ้ามีสายรถประจำทางให้เลือกมากกว่าหนึ่งสาย โปรแกรมจะตัดสินใจเลือกสายใดสายหนึ่งให้เลย เพื่อป้องกันความสับสนที่อาจเกิดขึ้น ก็จะได้เป็น $C \leftarrow (3) A \leftarrow (1) D$ และหลักการดังกล่าวได้มาจากพื้นฐานของ Breath First Search โดยนำมาประยุกต์ใช้ในสถานการณ์จริง เพื่อให้เกิดประโยชน์ และได้ผลสำเร็จทางด้านการหาสายรถประจำทางแบบอัจฉริยะ

3.3.3 จัดทำ Flow chart diagram ของโปรแกรม



บทที่ 4

การทดสอบและการวิเคราะห์โปรแกรม

จากการดำเนินการออกแบบและพัฒนาระบบดังที่กล่าวมาแล้วในบทข้างต้น และเพื่อให้เว็บไซต์นี้ทำงานได้ตามที่ต้องการแล้ว จึงต้องมีการผลการดำเนินการและวิเคราะห์โปรแกรมให้เป็นไปตามความต้องการและจุดประสงค์ที่ตั้งไว้ ดังนี้

4.1 ลักษณะทั่วไปของโปรแกรม

ระบบค้นหาเส้นทางเดินรถประจำทางออนไลน์อัจฉริยะนี้ ทำขึ้นเพื่อเป็นตัวช่วยในการตัดสินใจให้แก่ผู้โดยสารทั่วไปที่ใช้บริการรถโดยสารประจำทางในจังหวัดพิษณุโลก เพื่อให้ผู้ใช้สามารถเดินทางไปสู่จุดหมายตามที่ผู้โดยสารต้องการ ซึ่งผู้โดยสารบางคนไม่ทราบเส้นทางเดินรถ หรือไม่รู้ว่าจะเดินทางไปยังจุดหมายปลายทางควรใช้บริการรถโดยสารสายไหน เว็บไซต์นี้จะช่วยอำนวยความสะดวกให้แก่ผู้ใช้ สามารถค้นหาสายรถประจำทางที่สามารถไปจากจุดเริ่มต้นถึงปลายทางได้ เนื่องจากปัญหาในการเดินทางส่วนมากคือการเดินทางในเส้นทางที่ไม่เหมาะสม ส่งผลให้ต้องเสียเวลาและค่าใช้จ่าย โดยในโครงงานนี้ได้นำหลักการดิคสตรา (Dijkstra's algorithm) มาใช้ และเนื่องจากปัจจุบันคอมพิวเตอร์ถือเป็นเทคโนโลยีที่มีประสิทธิภาพและมีส่วนเกี่ยวข้องกับชีวิตประจำวันมากขึ้นเรื่อย ๆ จึงได้นำคอมพิวเตอร์มาช่วยในการประมวลผลโดยใช้ ASP.NET ในการสร้าง User Interface คิดต่อกับผู้ใช้ โดยแบ่งออกเป็นสองส่วนคือส่วนแรก รายละเอียดสถานที่ รายละเอียดของเส้นทางเดินรถแต่ละสาย สุดท้ายเป็นการประมวลผลเพื่อหาระหว่างจุด 2 จุดที่ต้องการ

4.2 วิธีการใช้งานโปรแกรม

เว็บไซต์นี้เริ่มต้นด้วยหน้าเพจที่มีเมนูเชื่อมต่อต่างๆ อยู่ด้านขวาและด้านบนของเว็บโดยเมนูเชื่อมต่อด้านบนจะสามารถเชื่อมต่อไปยังหน้าของโปรแกรมที่ช่วยค้นหาเส้นทางเดินรถ และด้านขวาที่แถบเมนูก็จะทำการเชื่อมต่อไปยังรายละเอียดต่างๆ คือ

ส่วนของเส้นทางเดินรถประจำทางจะประกอบไปด้วยเมนูเชื่อมต่อย่อย 3 เมนู คือ

1. ตารางเส้นทางเดินรถประจำทาง
2. แผนที่เทศบาลนครพิษณุโลก
3. แผนที่ท่องเที่ยวจังหวัดพิษณุโลก

ส่วนของข้อมูลต่างๆ ในจังหวัดพิษณุโลกประกอบไปด้วยเมนูเชื่อมต่อย่อย 4 เมนู คือ

1. สถานที่ท่องเที่ยว
2. โรงแรม
3. การเดินทาง
4. ร้านอาหาร

ส่วนของข้อมูลต่างๆ ในจังหวัดพิษณุโลกประกอบไปด้วยเมนูเชื่อมต่อย่อย 1 เมนู คือ

1. Contact us

ในการใช้งานโปรแกรมสามารถเลือกที่หัวข้อระบบค้นหาเส้นทางเดินทางรถประจำทางออนไลน์อังกฤษจะเข้าสู่หน้าจอของโปรแกรมค้นหาเส้นทาง การใช้งานสามารถทำได้โดยเลือกต้นทางที่จะเริ่มต้น และเลือกจุดสิ้นสุดที่เราจะไป จากนั้นเลือกปุ่มค้นหาดังรูปที่ 4.1



รูปที่ 4.1 หน้าเว็บเพจแสดงหน้าแรกของระบบค้นหาเส้นทางของรถประจำทางออนไลน์อังกฤษ

โปรแกรมจะทำการหาเส้นทางการเดินทางรถประจำทางที่สั้นที่สุด โดยบอกสายรถประจำทางว่าควรขึ้นสายรถประจำทางสายใด

4.3 วิธีการทดสอบ

วิธีการที่ใช้ในการทดสอบว่าระบบสามารถใช้งานได้จริงนั้น มีดังนี้

1. ทดสอบเมนูต่างๆว่าสามารถใช้งานได้จริงหรือไม่

2. ทดสอบว่าระบบสามารถทำงานและแจ้งผลการทำงานได้จริง

ดังนั้นจากการทดสอบนั้นได้ผลว่าเมื่อเราทำการกดเชื่อมต่อที่เมนูต่าง ๆ นั้นสามารถเชื่อมต่อไปยังเพจที่ต้องการได้ ผลการทดสอบมีดังต่อไปนี้

4.3.1 การทดสอบเมนูต่างๆ

ในส่วนของเส้นทางการเดินทางประจำทางจะประกอบไปด้วยเมนูเชื่อมต่อย่อย 3 เมนูคือ

1. ตารางเส้นทางการเดินทางประจำทาง

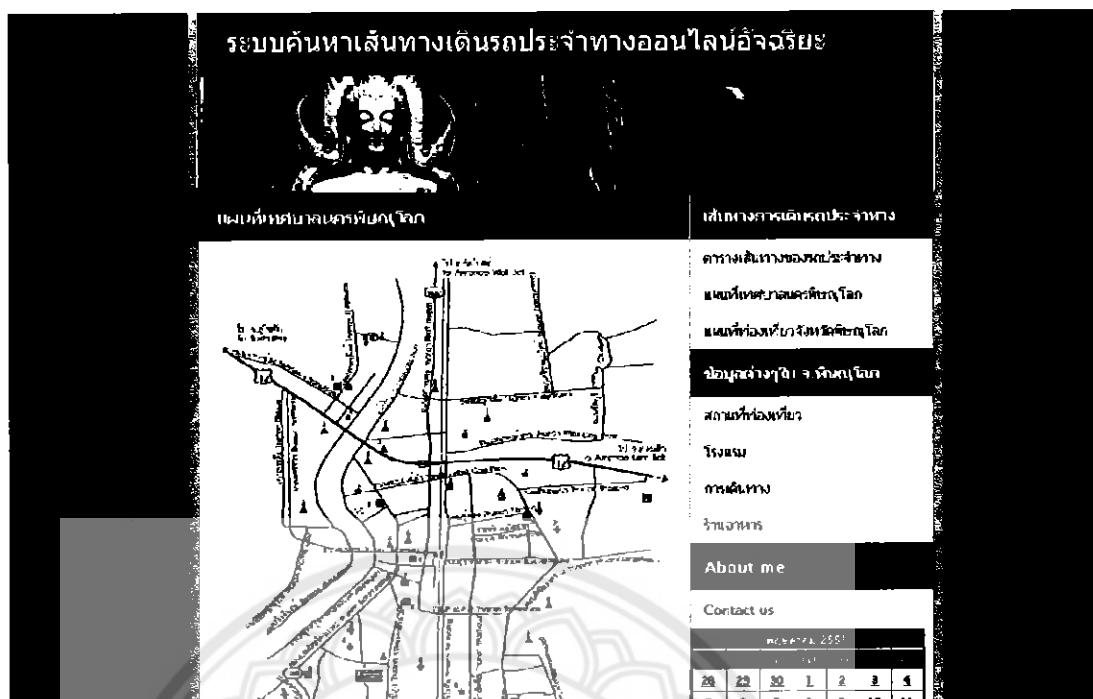
2. แผนที่เทศบาลนครพิษณุโลก

3. แผนที่ท่องเที่ยวจังหวัดพิษณุโลก

โดยเมื่อกดที่เมนูเชื่อมต่อเหล่านี้จะแสดงผลดังรูปที่ 4.2 รูปที่ 4.3 และรูปที่ 4.4



รูปที่ 4.2 หน้าเว็บเพจแสดงผลจากการกดเชื่อมต่อตารางเส้นทางของรถประจำทาง



รูปที่ 4.3 หน้าเว็บเพจแสดงผลจากการกดเชื่อมต่อแผนที่เทศบาลนครพิษณุโลก



รูปที่ 4.4 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อแผนที่ท่องเที่ยวจังหวัดพิษณุโลก

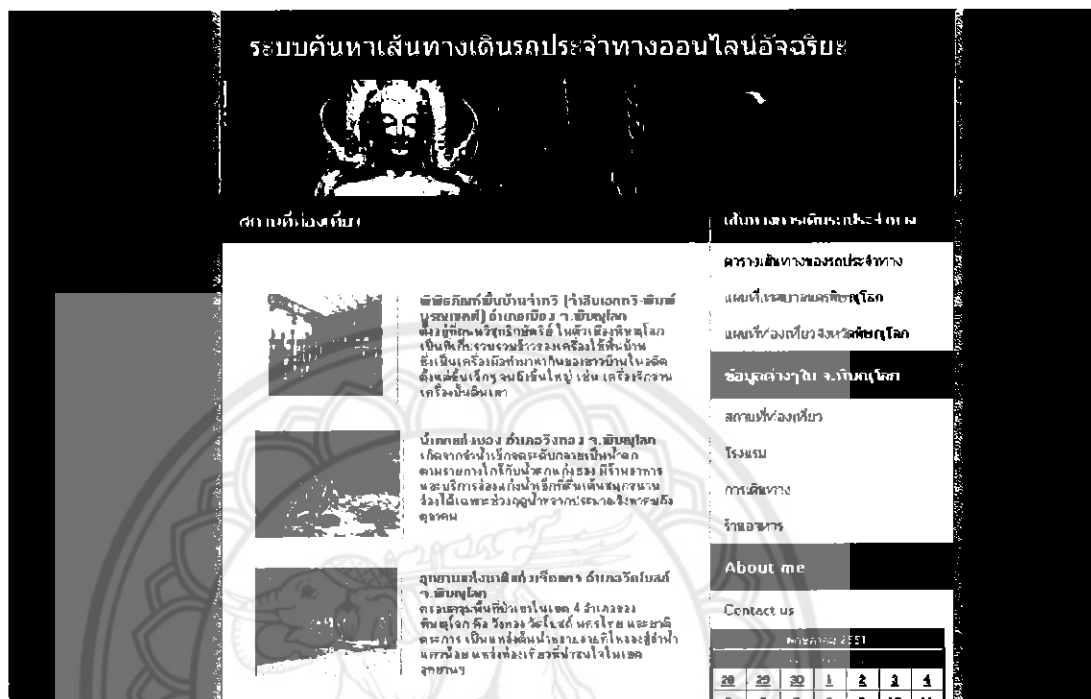
ในส่วน of ข้อมูลต่างๆ ในจังหวัดพิษณุโลกประกอบไปด้วยเมนูเชื่อมต้อย่อย 4 เมนู คือ

1. สถานที่ท่องเที่ยว
2. โรงแรม

3. การเดินทาง

4. ร้านอาหาร

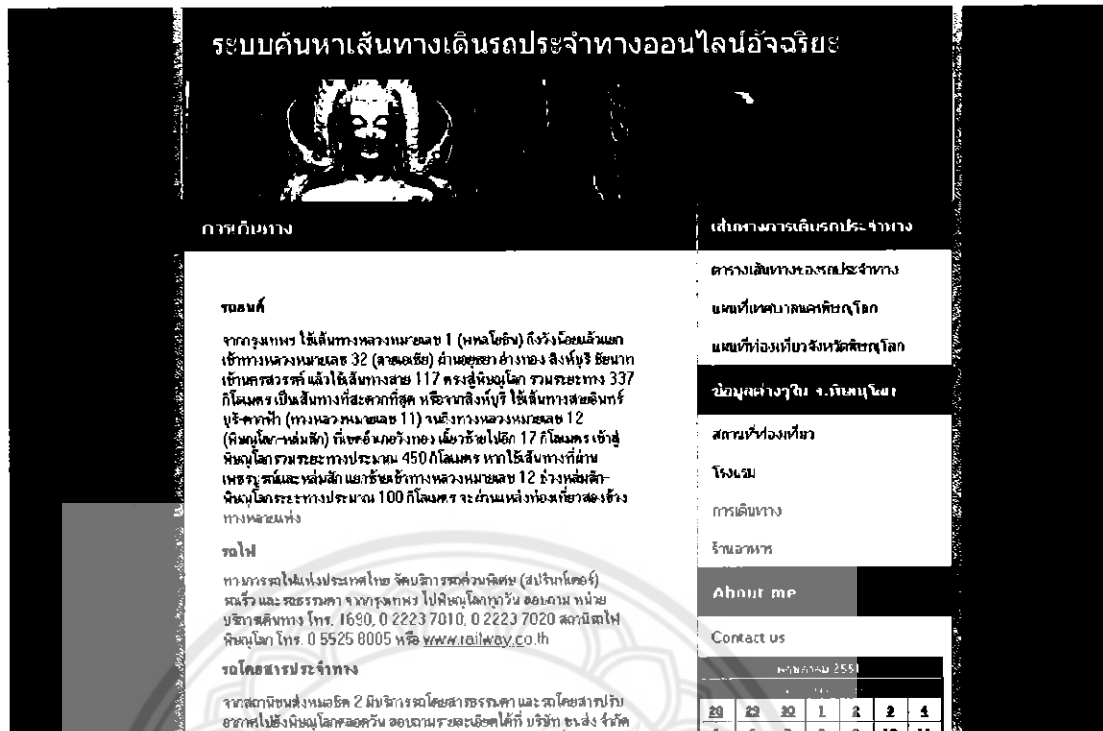
โดยเมื่อกดที่เชื่อมต่อเหล่านี้จะแสดงผลดังรูปที่ 4.5 รูปที่ 4.6 รูปที่ 4.7 และรูปที่ 4.8



รูปที่ 4.5 หน้าเว็บเพจแสดงผลจากการกดเมนูเชื่อมต่อสถานที่ท่องเที่ยวจังหวัดพิษณุโลก



รูปที่ 4.6 หน้าเว็บเพจแสดงผลจากการกดเชื่อมต่อโรงแรมในจังหวัดพิษณุโลก



รูปที่ 4.7 หน้าเว็บเพจแสดงผลจากการกดเชื่อมต่อการเดินทางในจังหวัดพิษณุโลก

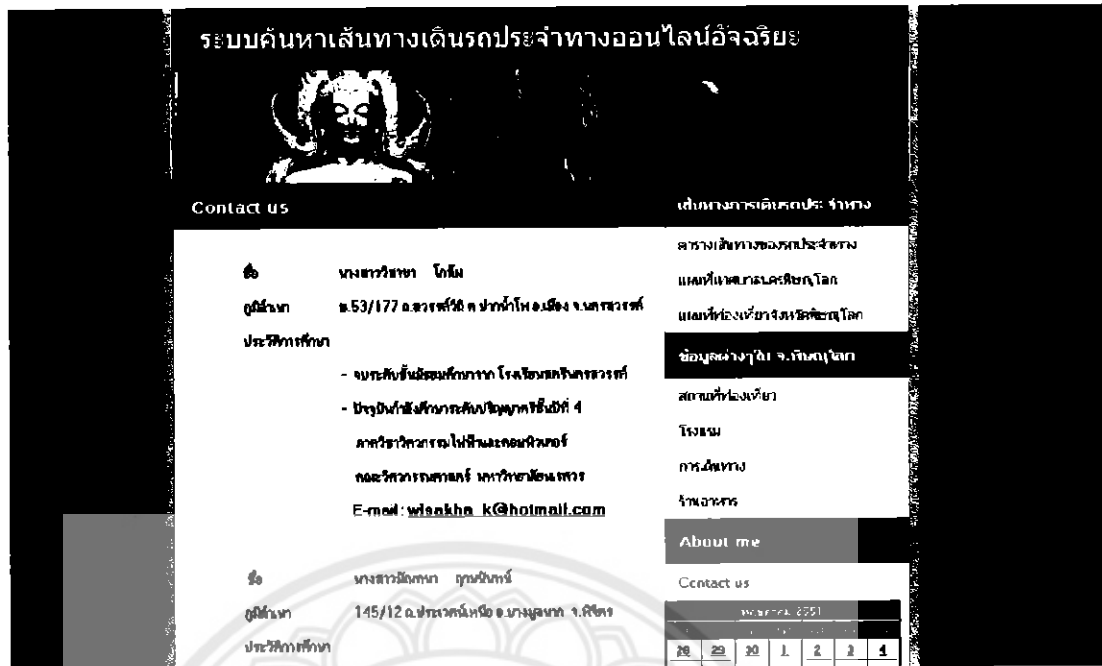


รูปที่ 4.8 หน้าเว็บเพจแสดงผลจากการกดเชื่อมต่อร้านอาหารในจังหวัดพิษณุโลก

ในส่วนของข้อมูลต่างๆ ในจังหวัดพิษณุโลกประกอบไปด้วยเมนูเชื่อมต่อย่อย 1 เชื่อมต่อ คือ

1. Contact us

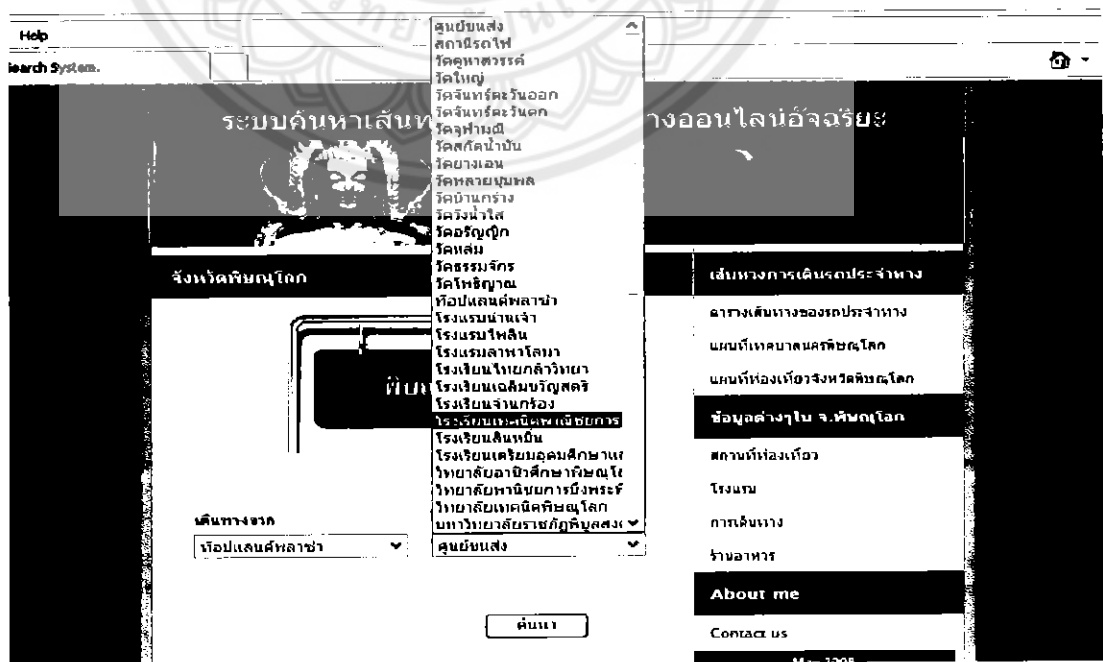
โดยเมื่อกดที่เชื่อมต่อเหล่านี้จะแสดงผลดังรูปที่ 4.9



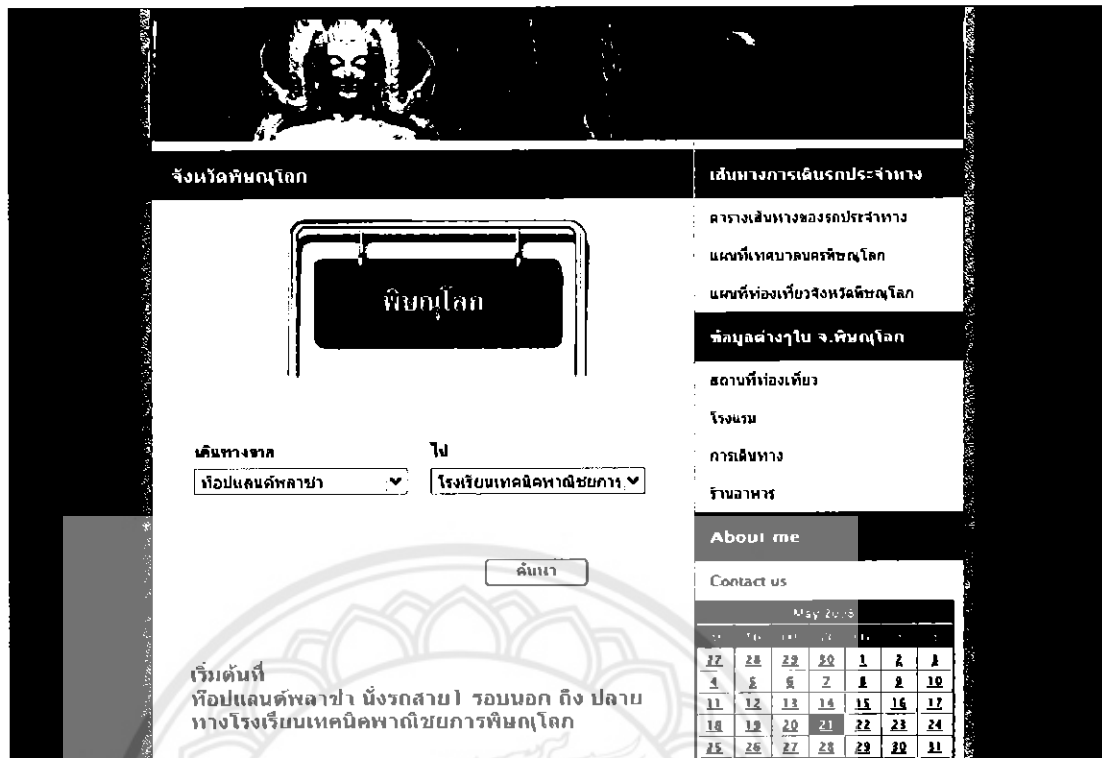
รูปที่ 4. 9 หน้าเว็บเพจแสดงผลจากการกดเชื่อมต่อ Contact us

4.3.2 การทดสอบระบบ

ทำการทดสอบทำได้โดยการเลือกสถานที่จาก Dropdown list จากหน้าเว็บไซต์ เลือกสถานที่ต้นทาง และปลายทาง และกดปุ่มค้นหา ในการทดสอบทำการเดินทางจากที่ออฟแลนด์ พลาซ่าไปยังโรงเรียนพณิชยการพิษณุโลก เมื่อกดปุ่มค้นหาจะแสดงสายรถประจำทางที่ผ่านสถานที่จุดหมายที่เราต้องการ แสดงผลดังรูปที่ 4.10 และ 4.11



รูปที่ 4. 10 หน้าเว็บเพจแสดงผลการเลือกจุดเริ่มต้นและจุดสิ้นสุด



จังหวัดพิษณุโลก

พิษณุโลก

เดินทางจาก ไป

หือปแลนด์พลาซ่า

โรงเรียนเทคนิคพาณิชยการพิษณุโลก

ค้นหา

เริ่มต้นที่
หือปแลนด์พลาซ่า นังรกลาย 1 รอบนอก ถึง ปลาบ
ทางโรงเรียนเทคนิคพาณิชยการพิษณุโลก

เส้นทางการเดินทางประจำทาง

ตารางเส้นทางของรถประจำทาง

แผนที่เทศบาลนครพิษณุโลก

แผนที่ท่องเที่ยวจังหวัดพิษณุโลก

ข้อมูลต่างๆใน จ.พิษณุโลก

สถานที่ท่องเที่ยว

โรงแรม

การเดินทาง

ร้านอาหาร

About me

Contact us

May 2018

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

รูปที่ 4. 11 หน้าเว็บเพจแสดงผลการค้นหาสายรถประจำทาง

บทที่ 5

บทสรุป

โครงการนี้ได้ทำการออกแบบขึ้นเพื่อช่วยในการตัดสินใจในการเดินทางด้วยรถประจำทางและเป็นทางเลือกหนึ่งในการใช้บริการเพื่อการท่องเที่ยวในจังหวัดพิษณุโลก ซึ่งเพิ่มความสะดวกสบายให้แก่ผู้ใช้งาน

5.1 สรุปผล

โครงการเรื่องระบบค้นหาเส้นทางเดินรถประจำทางออนไลน์อัจฉริยะ ได้จัดทำขึ้นเพื่อช่วยให้ผู้ที่ต้องการเดินทางมาเที่ยวชมสถานที่ท่องเที่ยวต่างๆภายในจังหวัดพิษณุโลกเดินทางได้อย่างสะดวก รวดเร็วและมีประสิทธิภาพ ถึงแม้ว่าไม่ใช่คนในท้องถิ่นก็สามารถเดินทางไปยังสถานที่ที่ต้องการได้ถูกต้อง โครงการนี้ได้พัฒนาเว็บไซต์ซึ่งทำให้นักท่องเที่ยวสามารถวางแผนล่วงหน้าในการเตรียมตัวเดินทางมาเที่ยวสถานที่ท่องเที่ยวต่างๆในจังหวัดพิษณุโลก

ในการเดินทางจากจุดหนึ่งไปยังอีกจุดหนึ่งสามารถขึ้นรถประจำทางได้หลายสาย ซึ่งในการขึ้นรถประจำทางบางสายจากจุดที่เราอยู่กับจุดที่เราต้องการไป อาจจะอ้อมจุดอื่นๆ อีกมากมาย ทำให้เสียเวลาในการเดินทาง ดังนั้นผู้พัฒนาจึงพัฒนาขั้นตอนในการหาเส้นทางเดินรถประจำทางที่สั้นที่สุดนั้นจะใช้ขั้นตอนวิธีของคิซตรา (Dijkstra's algorithm) เนื่องจากง่ายต่อการเขียนโปรแกรมและค่อนข้างมีประสิทธิภาพ ส่วนในเรื่องของการเขียนโปรแกรมนั้นได้ใช้ภาษา C# เนื่องจากภาษา C# สนับสนุนการเขียนโปรแกรมบนเว็บไซต์

ซึ่งผลที่ได้จากการพัฒนาโปรแกรมทำให้ได้เส้นทางสายรถประจำทางที่ผ่านจุดเริ่มต้นไปยังจุดที่ต้องการไป ทำให้สะดวกและรวดเร็วในการเดินทางแก่นักท่องเที่ยว และผู้ที่ต้องการเดินทางในจังหวัดพิษณุโลก

5.2 ปัญหา อุปสรรคและวิธีการในการแก้ปัญหา

ในการพัฒนาโครงการมีปัญหาและอุปสรรคต่างๆ หลายประการ ดังนี้

1. ต้องรวบรวมข้อมูลหลายอย่างเช่น เส้นทางเดินสายรถประจำทาง แผนที่เขตเทศบาลนครพิษณุโลกและแผนที่จังหวัดพิษณุโลกซึ่งต้องใช้เวลาในการติดต่อสอบถามอีกทั้งคณะผู้จัดทำโครงการไม่ได้เป็นคนพิษณุโลกโดยกำเนิดและการสำรวจเส้นทางรถโดยสารประจำทางเป็นไปได้ยาก เนื่องจากมีรถโดยสารหลายคัน การเก็บรวบรวมข้อมูลจึงต้องใช้เวลาและค่าใช้จ่ายในก

เส้นทางสูง แต่คณะผู้จัดทำแก้ปัญหาโดยการสอบถามข้อมูลจากพนักงานขับรถและเจ้าหน้าที่ของแหล่งข้อมูลจากศูนย์ตำรวจและสถานีรถไฟ ซึ่งเป็นศูนย์การเดินรถ และขอข้อมูลจากบริษัทเดินรถ และข้อมูลจากการท่องเที่ยวแห่งประเทศไทยเก็บไว้เป็นข้อมูลในการศึกษา นอกจากนี้การค้นหาข้อมูลของสถานที่ต่างๆ ก็ใช้เวลาในการรวบรวมเนื่องจากต้องการรูปภาพประกอบด้วย เพื่อให้ผู้ใช้ได้ประโยชน์สูงสุดจากการเข้าชมเว็บเพจ และใช้โปรแกรม จึงแก้ปัญหาด้วยการค้นหาข้อมูลจากเว็บไซต์ระดับองค์กรเพื่อให้ได้ข้อมูลที่ถูกต้อง และสามารถอ้างอิงได้

2. การหาระยะทางจากจุดถึงจุดเป็นไปได้ยาก เพราะในความเป็นจริงระยะทางจะโค้งตามเส้นทางถนน เพื่อให้ได้ระยะทางที่ใกล้เคียงกับความเป็นจริงมากที่สุดจึงทำการแก้ปัญหาโดยใช้เว็บไซต์ <http://www.mapguidethailand.com/home/map.php> แล้วกำหนดจุดลงบนแผนที่ แล้วทำการลากเส้นเพื่อทำการวัดก็จะปรากฏระยะทางขึ้นมาหน่วยเป็นเมตร

3. เนื่องจากขาดประสบการณ์ในการเขียนโปรแกรม จึงทำให้เกิดปัญหาในการทำงาน วิธีการแก้ไขจึงต้องทำการศึกษาวิธีการเขียนจากแหล่งค้นคว้าทางอินเทอร์เน็ตและจากหอสมุด

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

1. โครงการระบบค้นหาเส้นทางเดินรถประจำทางออนไลน์อัจฉริยะ ยังเป็นการศึกษาในเขตพื้นที่ในจังหวัดพิษณุโลกเท่านั้น ยังไม่ครอบคลุมสถานที่ท่องเที่ยว และสถานที่สำคัญในจังหวัดพิษณุโลกทั้งหมด แนวทางในการพัฒนาโปรแกรมต่อไปจึงแนะนำให้ทำในเขตพื้นที่กว้างออกไปมีจุดที่ละเอียดขึ้น ถ้าต้องการให้ครอบคลุมมากขึ้น จึงควรเก็บข้อมูลในส่วนที่ต้องการเพิ่มเติม

2. ทำการเขียนโปรแกรมค้นหาเส้นทางที่สั้นที่สุดบนแผนที่ประเทศไทย และนำไปใช้ได้จริง

3. ทำเว็บไซต์นี้เป็นภาษาอังกฤษเพื่อช่วยชาวต่างชาติในการใช้บริการรถโดยสารประจำทางในจังหวัดพิษณุโลก

เอกสารอ้างอิง

- [1] บัญชา ปะสีละเตสัง . คู่มือการพัฒนาโปรแกรม ASP.NET 2 ด้วย VB.NET และ C# . กรุงเทพฯ : ซีเอ็ดดูเกชั่น , 2550
- [2] Marco Bellinaso . พัฒนาเว็บไซต์ด้วย ASP.NET 2.0 วิเคราะห์ปัญหา-ออกแบบระบบ-เขียนโปรแกรม . กรุงเทพฯ : ซีเอ็ดดูเกชั่น , 2551
- [3] สัจจะ จรัสรุ่งเรือง . คู่มือ Visual C# 2005 ฉบับสมบูรณ์ . นนทบุรี : โอเคซี , 2550
- [4] D.S. Malik and M.K. Sen. **Discrete Mathematical Structures: Theory and Applications** .Messachusetts : Techsetters,Inc.2004
- [5] ดร. ขวลิศ จินอนันต์. “ Network Optimization (Dijkstra's algorithm). ” [Online]. Available : <http://mail.vcharkarn.com/varticle/1124/1>. 2007.
- [6] ลากลอย วานิชอังกูร. “ เรียนรู้ด้วยตนเอง OOP C#.NET. ” [Online]. Available : <http://thai-cs.spaces.live.com/>.2007
- [7] กลุ่มสารสนเทศสำนักน โยบายและแผนสำนักงาน คณะกรรมการการศึกษาขั้นพื้นฐาน. “ บทเรียน Microsoft SQL Server 2000. ” [Online]. Available : http://www.bopp.go.th/web_SQL/.2007
- [8] Bruno R. Preiss. “ Data Structures and Algorithms with Object-Oriented Design Patterns in C#. ” [Online]. Available : <http://www.brpreiss.com/books/opus6/html/page567.html>.2005.
- [9] Kenji Ikeda's. “ Dijkstra's Algorithm. ” [Online]. Available : <http://www-b2.is.tokushima-u.ac.jp/~ikeda/suuri/dijkstra/Dijkstra.shtml>.2000.



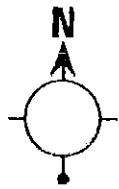
บริษัทพิษณุโลกบริการ จำกัด “รถเมล์บ้านเรา”

สายเดินรถ	เส้นทางเดินรถโดยสาร
ปอ.1 รอบเมือง (รอบนอก)	ศูนย์ขนส่ง , โรงแรมมิตรบริการลาภู , บกพ. , วัดโลกบะจุม , ร.พ.พุทธา , สถานีรถไฟ , รร.เฉลิมขวัญสตรี , สะพานเบรศวร , วัดจุ , จันทรรอง , วัดใหญ่ , ทพล.
ปอ.1 รอบเมือง (รอบใน)	ศูนย์ขนส่ง , ทพล. , ที่โปแลนด์พลาซ่า , วัดใหญ่ , รร.เฉลิมขวัญสตรี , สถานีรถไฟ , ร.พ.พุทธา , วัดโลกบะจุม , โรงแรมมิตรบริการลาภู , บกพ.
สาย 3 สถานีรถไฟ-บ้านคลอง	สถานีรถไฟ , รร.เฉลิมขวัญสตรี , สะพานเบรศวร , ราชภัฏ (วังจันทน์) , วัดจันทร์วัน ๓๓ , สามแยกบ้านคลอง , อำเภอเมือง , วัดใหญ่ , ที่โปแลนด์พลาซ่า , ตลาดสถานีรถไฟ
สาย 4 สถานีรถไฟ-สนามบิน	สถานีรถไฟ , ม.เบรศวร(สนามบิน) , วัดทองบึง , กองบิน 46 , รพ.วัดบึง , สี่แยกบ้านแขก , ตลาดสถานีรถไฟ
สาย 5 (วัดยางเอน) สถานีรถไฟ - วัดยางเอน	สถานีรถไฟ , สิบตัม , ร.พ.รวมแพทย์ , โรงแรมบ้านเจ้า , ราชภัฏ (วังจันทน์) , วัดจันทร์วัน ๓๓ , นมิด , วัดจุฬามณี , ท่าชด. , หมู่บ้านถาวรกาญจน์ , สถานีวิทยุกรมอุตุฯ , จัหวบ , วัดสวัด ป่ามื่น , วัดยางเอน , เกตุมิด 2
สาย 5 (ตึกทาม) สถานีรถไฟ-บ้านตึกทาม	สถานีรถไฟ , สามแยกบ้านแขก , ที่โปแลนด์พลาซ่า , วัดใหญ่ , วัดธรรมจักร , วัดโพธิ์ , โรงทอ , โรงเรือนพิษณุโลกศึกษา , สนามกีฬากลาง , บ้านตึกทาม
สายปอ. 6 ศูนย์ขนส่ง-วัดบ้านกร่าง	ศูนย์ขนส่ง , บกพ. , วัดโลกบะจุม , ร.พ.พุทธา , สถานีรถไฟ , ตลาดเมือง , โรงแรมไฟลิบ , วัดจุ , ตลาดบ้านคลอง , สบง.ขนส่ง , สามแยก ปช. , พิษณุโลกเมืองใหม่ , วัดหลายชุมพล , สี่แยกบาตาส , วัดบ้านกร่าง
สาย 7 สถานีรถไฟ - แม่ระกา	สถานีรถไฟ , ร.พ.พุทธา , โรงแรมลาฟาโรมา , เดอะ , วิทยาลัยบึงพระ , วังป่า , บ้านแขก , แม่ระกา , กม. 15
สาย 8 สถานีรถไฟ - อรัญญิก	ศูนย์ขนส่ง , วัดโลกบะจุม , พิพิธภัณฑสถาน , ร.พ.พุทธา , สถานีรถไฟ , ที่โปแลนด์พลาซ่า , วัดใหญ่ , วัดอรัญญิก
สาย 9 สถานีรถไฟ - นาโพธิ์แดง	สถานีรถไฟ , ตลาด , ร.พ.รวมแพทย์ , ร.พ.พิษณุเวช , ราชภัฏ (วังจันทน์) , สถานีรถไฟ , เกตุมิด
สาย ปอ. 11 วัดท่าม - สนามกีฬากลาง	วัดท่าม , สถานีรถไฟบึงพระ , วิทยาลัยบึงพระ , เดอะ , ร.พ.พุทธา , สถานีรถไฟ , ที่โปแลนด์พลาซ่า , วัดใหญ่ , วัดธรรมจักร , วัดโพธิ์ , โรงทอ , ร.พ.พิษณุโลกศึกษา , สนามกีฬากลาง , ตลาดสด อ.บ.ศ.
สาย ปอ.12, ปอ.12 รอบใน ม.สนามบิน - ม.หนองอ้อ	ม.เบรศวร(สนามบิน) , วิทยาลัยเทคโนโลยี , ท่าแยกโลกบะจุม , ศูนย์ขนส่ง , ทพล. , ที่โปแลนด์พลาซ่า , สามแยกบ้านคลอง , สามแยกวัดว่า , ตลาดประจักษ์ กม.7 , ม.เบรศวร (หนองอ้อ)
สาย ปอ.13 ราชภัฏ(ทะเลแก้ว)-หนอง	ราชภัฏ (ทะเลแก้ว) , สามแยกปช. , สบง.ขนส่ง (เก่า) , สามแยกบ้านคลอง , ราชภัฏ (วังจันทน์) , ตลาดกลาง , ที่โปแลนด์พลาซ่า , ทพล. , ศูนย์ขนส่ง , บกพ. , หมู่บ้านจันทร์วัน , ศูนย์พัฒนา ตำบลสมอแข , ไร่
สาย ปอ. 14 สบง.ขนส่งเก่า-สบง.ขนส่งใหม่	สบง.ขนส่งเก่า , บ้านคลอง , ถนนเมือง , วิทยาลัยอาชีวศึกษา , สะพานเอกา , สถานีรถไฟ , รร.ไทยกลา , ท่าแยกโลกบะจุม , บิ๊กซี , ศูนย์ขนส่ง , สามแยกเรือนแพ , โลตัส , หมู่บ้านชินลาภ , สี่แยกโรงไฟฟ้ , สบง.ขนส่งใหม่
สาย 15 รร.พพ.(ใหม่) - ศูนย์ขนส่ง - บ้านคลองเปิด	รร.พิษณุโลกศึกษา(ใหม่) , ศูนย์ขนส่งเสริมอุตสาหกรรม 2 , สบง.ไตรศิต , หมู่บ้านพิษณุโลกเมืองใหม่ , สบง.ขนส่ง(เก่า) , สี่แยกบ้านคลอง , วัดจุ , ตลาดกลาง , วัดใหญ่ , ที่โปแลนด์พลาซ่า , ทพล. , ศูนย์ขนส่ง , โลตัส , หมู่บ้านชินลาภ , บ้านคลองเปิด **
สาย ปอ. 16 ม.สนามบิน - ม.หนองอ้อ	ม.เบรศวร(สนามบิน) , ถนนชานูเวอิจ , สะพาน 1 , เดอะ , วิทยาลัยบึงพระ , บาตาส , เกตุมิดใหม่ , สี่แยกวัดสวัดป่ามื่น , ยางเอน , สี่แยกบาตาสบางระกำ , ม.เบรศวร (หนองอ้อ)
สาย 17 สถานีรถไฟ - ยางเอน	สถานีรถไฟ , ทอณา , เทศบาลใหม่ , นมิด , วัดจุ , ท่าชด. , ยางเอน , ศูนย์วิทยุ เรย์ดอมเพิวดอร์ , รร.พิษณุโลกศึกษา(ใหม่)

**เส้นทางนี้อาจมีการเปลี่ยนแปลงไปภายหลัง

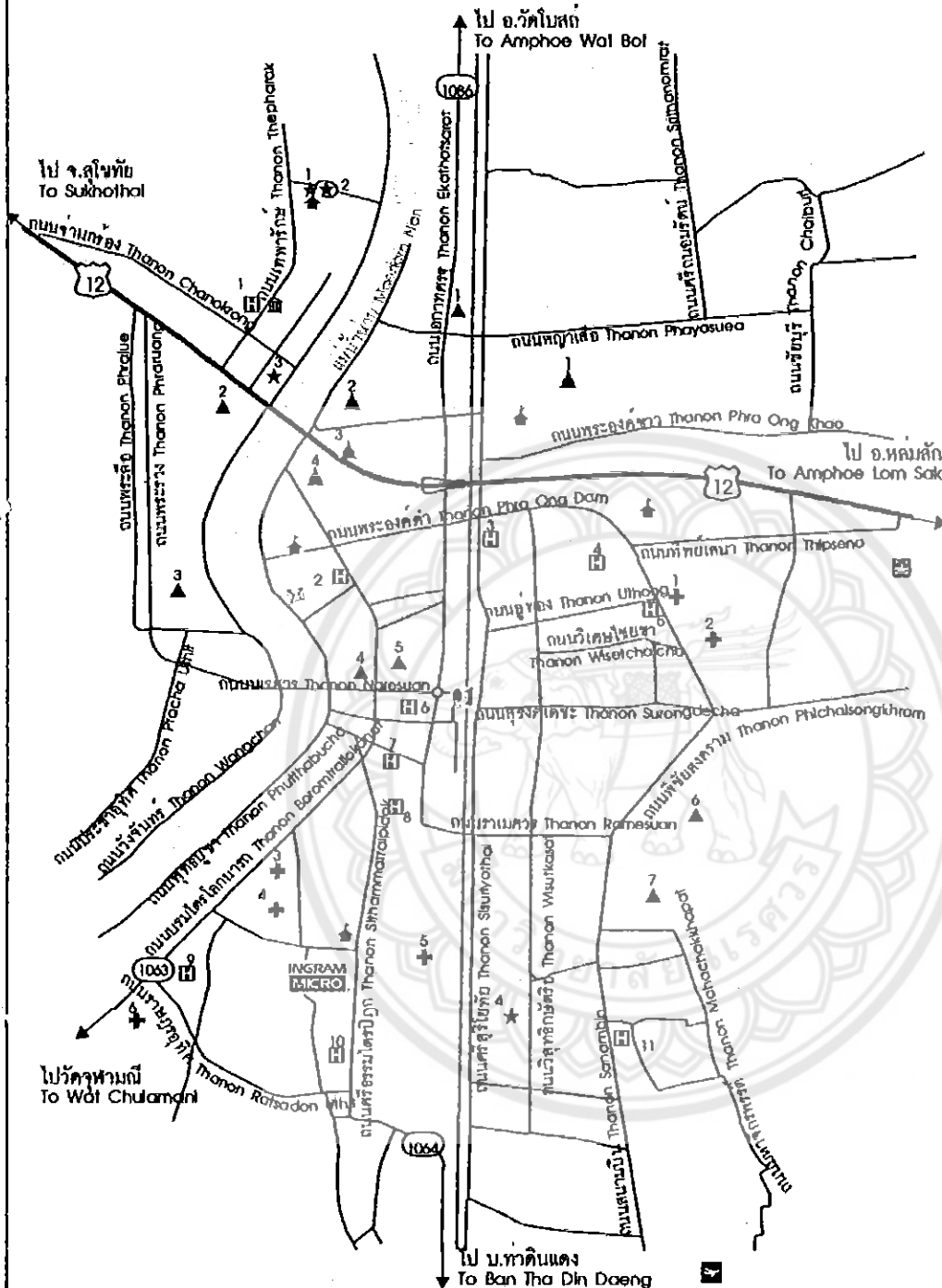
ลองถามรายละเอียดได้จากเจ้าหน้าที่ของเราทุกคน





แผนที่ อินแกรมไมโคร(ประเทศไทย) สาขาพิษณุโลก

INGRAM MICRO (THAILAND) PHITSANULOK BRANCH



Scale 1 : 25,000

INGRAM MICRO (THAILAND) LTD.
59/29 SRITUMTRIPIDOK RD.
A.MUANG PHITSANULOK 65000
TEL. 0-5522-5530-4

บริษัท อินแกรมไมโคร (ประเทศไทย) จำกัด

59/29 ถนนศรีธรรมไตรปิฎก

อำเภอเมือง จังหวัดพิษณุโลก 65000

โทรศัพท์ 0-5522-5530-4

สัญลักษณ์ Legend

	ศาลากลางจังหวัด	City Hall
	ไปรษณีย์	Post Office
	โรงเรียน	School
	สนามบิน	Airport
	สถานีขนส่ง	Bus Terminal

	สถานีรถไฟ	Railway Station
	ทางหลวง	Highway
	แหล่งน้ำ	Stream/Water
	ทางรถไฟ	Railway

โรงแรม Hotel

- 1 โรงแรมวังแก้ว Wang Kaeo Hotel
- 2 โรงแรมโกลีน Palin Hotel
- 3 โรงแรมช้างเผือก (โกลเด้นแกรนด์)
Chang Phuok Hotel (Golden Grand)
- 4 โรงแรมราชพฤกษ์ Rajchaphuek Hotel
- 5 โรงแรมอุบลทอง Uthong Hotel
- 6 โรงแรมอมรินทร์นคร Amarin Nakhon Hotel
- 7 โรงแรมศิโรตม์ Silat Hotel
- 8 โรงแรมเทพนคร Thep Nakhon Hotel
- 9 โรงแรมบ้านเจ้า Nan Chao Hotel
- 10 โรงแรมพาโลมา Paloma Hotel
- 11 โรงแรมพิษณุโลกธานี
Phitsanulok Thani Hotel

วัด Temple (Wat)

- 1 วัดเจติยาคะเชว Wat Chedi Yot Thong
- 2 วัดพระศรีรัตนมหาธาตุวรมหาวิหาร
Wat Phra Sri Rattana Mahathat Woramahawihan
- 3 วัดบางพญา Wat Bang Phaya
- 4 วัดราชบูรณะ Wat Ratchaburana

โรงพยาบาล Hospital

- 1 โรงพยาบาลรัตนเวช 1 Rattana Wei 1 Hospital
- 2 โรงพยาบาลรัตนเวช 2 Rattana Wei 2 Hospital
- 3 โรงพยาบาลธรรมแพทย์ Ruam Phaei Hospital
- 4 โรงพยาบาลพิษณุเวช Phitsanulok Wei Hospital
- 5 โรงพยาบาลจุฬารัตน Phultha Chirarat Hospital
- 6 โรงพยาบาลอินเตอร์เวช Inter Weichakan Hospital

สถานที่สำคัญ Places

- 1 กองกำกับการตำรวจท่องเที่ยวจังหวัดพิษณุโลก
Phitsanulok Provincial Tourism Police Station
- 2 ศาลจังหวัดพิษณุโลก Phitsanulok Law Court
- 3 มหาวิทยาลัยสถาบันราชภัฏพิษณุโลก
Phibun Songkhro Rajabhat University
- 4 สถานีตำรวจภูธรอำเภอเมืองพิษณุโลก
Amphoe Mueang Phitsanulok Police Station
- 5 สำนักงานเทศบาลนครพิษณุโลก
Phitsanulok Municipal Office
- 6 วิทยาลัยเทคนิคพิษณุโลก
Phitsanulok Technical College
- 7 มหาวิทยาลัยนเรศวร Naresuan University

★ แหล่งท่องเที่ยว Tourist Attraction

- 1 พิพิธภัณฑ์สมเด็จพระนเรศวรมหาราช
Samdet Phra Naresuan Museum
- 2 ศาลสมเด็จพระนเรศวรมหาราช
Samdet Phra Naresuan Shrine
- 3 ศาลหลักเมือง
City Pillar Shrine
- 4 พิพิธภัณฑ์เจ้าท้าว Chao Tawee Museum

ประวัติผู้เขียนโครงการ



ชื่อ นางสาววิสาขา โกสัม
ภูมิลำเนา ช.53/177 ถ.สวรรค์วิถี ต.ปากน้ำโพ อ.เมือง จ.นครสวรรค์ 60000

ประวัติการศึกษา

- จบระดับชั้นมัธยมศึกษาจาก โรงเรียนสตรีนครสวรรค์
- ปัจจุบันกำลังศึกษาระดับปริญญาตรีชั้นปีที่ 4

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail : wisakha_k@hotmail.com



ชื่อ นางสาวมันทนา ฤกษ์นันท์
ภูมิลำเนา 145/12 ถ.ประเวศน์เหนือ อ.บางมูลนาก จ.พิจิตร 66120

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนบางมูลนากภูมิวิทยาคม
- ปัจจุบันกำลังศึกษาระดับปริญญาตรีชั้นปีที่ 4

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail : kik_zs55@hotmail.com



ชื่อ นางสาวลัดอักษร เกื้อนยัง
ภูมิลำเนา 96 หมู่ที่ 3 ต.บ้านกล้วย อ.เมืองชัยนาท จ. ชัยนาท 17000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนชัยนาทพิทยาคม
- ปัจจุบันกำลังศึกษาระดับปริญญาตรีชั้นปีที่ 4

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail : freshy_24@hotmail.com