

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบให้บริการแผนที่

MAP SERVICES

นายพงศ์รบ สายสุวรรณ  
นายพรเทพ อาชวพงศ์

เลขหมู่.....  
เลขทะเบียน..... 62729  
วัน,เดือน,ปี..... 21 ส.ค. 2549

b..... 11629113  
i.....

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาคามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบให้บริการแผนที่  
MAP SERVICES

โดย

นายพงศ์รบ ตายสุวรรณ  
นายพรเทพ อาชวพงศ์

อาจารย์ที่ปรึกษา  
ผศ.ดร. สุรินทร์ กิตติขจรกุล

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาคามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบให้บริการแผนที่

MAP SERVICES

ผู้จัดทำ

1. นายพงษ์รบ สายสุวรรณ รหัสนักศึกษา 45010485

2. นายพรเทพ อาชวพงศ์ รหัสนักศึกษา 45010498



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบให้บริการแผนที่

|                            |                  |
|----------------------------|------------------|
| นายพงษ์รบ สายสุวรรณ        | 45010485         |
| นายพรเทพ อาชวพงศ์          | 45010498         |
| ผศ.ดร. สุรินทร์ กิตติชรกุล | อาจารย์ที่ปรึกษา |
| ปีการศึกษา 2548            |                  |

### บทคัดย่อ

การจะไปยังสถานที่ซักแห่งในปัจจุบันนั้น แผนที่ถือเป็นสิ่งที่สามารถช่วยได้อย่างมาก ระบบให้บริการแผนที่นี้ จะช่วยอำนวยความสะดวกในการหาดำเนินปัจจุบันที่อยู่ หาดำเนิน สถานที่ต่างๆ โดยโครงการนี้จะคอยส่งแผนที่ให้กับ Web Service ทำให้สามารถดูแผนที่ได้จากทั้ง Browser และ Pocket PC

นอกจากนี้ ในโครงการยังมีส่วนของการแสดงแผนที่ภายในอาคารสถานที่ ทำให้ผู้ใช้สามารถหาข้อมูลสถานที่ต่างๆ ในอาคาร เช่น ร้านอาหาร และยังสามารถเข้าไปดูรายละเอียดของร้านอาหารจนกระทั่งจองที่นั่งในร้านอาหารได้เลย ซึ่งสิ่งเหล่านี้จะช่วยให้การใช้ชีวิตในปัจจุบันมีความสะดวกสบาย และสนุกมากยิ่งขึ้น

## MAP SERVICES

Mr. Pongrob Saisuwan 45010485

Mr. Porntep Archavapong 45010498

Asst. Prof. Dr. Surin Kittitornkun Advisor

Academic Year 2004

### ABSTRACT

Location-based mobile information system (LMIS) has become a new and successful alternative for mobile users especially in Bangkok because it can provide information while the users are traveling in highly congested areas. We have implemented an LMIS for restaurants based on web services technology. The system consists of the front-end user interfaces on mobile smartphones and web browsers, the back-end, and the map server.

This project is mainly concerned with the map server providing the location on the map and interior floorplan of any restaurants in the chain. Based on the known location from the GPS, the map server responds by sending both map images where the restaurant/building is located and internal floorplan image of the restaurant/building.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก ผศ.ดร. สุรินทร์ กิตติธรรมกุล ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้ารู้สึกทราบบซึ่งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา ขอขอบคุณบัณฑิตศึกษาและบัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ที่ให้ความช่วยเหลือในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี คุณค่าและประโยชน์ที่ได้มาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

นายพงศ์รบ สายสุวรรณ

นายพรเทพ อาชวพงศ์

# สารบัญ

|   | หน้า |
|---|------|
| บทคัดย่อภาษาไทย.....  | I    |
| บทคัดย่อภาษาอังกฤษ.....   | II   |
| กิตติกรรมประกาศ.....  | III  |
| สารบัญ.....   | IV   |
| สารบัญตาราง.....  | VIII |
| สารบัญรูป.....  | VIII |
| บทที่ 1 บทนำ.....   | 1    |
| 1.1 ความสำคัญและที่มาของโครงการ.....                              | 1    |
| 1.2 วัตถุประสงค์ของโครงการ.....                                   | 1    |
| 1.3 ขอบเขตของโครงการ.....   | 1    |
| 1.4 วิธีการดำเนินการ.....   | 3    |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ.....                                | 3    |
| 1.6 ส่วนประกอบของปริญญานิพนธ์.....                                | 3    |
| บทที่ 2 ทฤษฎีที่ใช้ในโครงการ.....                                 | 5    |
| 2.1 เว็บบเซอร์วิส.....  | 5    |
| 2.2 XML(The Extensible Markup Language).....                      | 6    |
| 2.2.1 XML and HTML.....   | 6    |
| 2.3 SOAP (Simple Object Access Protocol).....                     | 7    |
| 2.4 WSDL.....   | 7    |
| 2.4.1 โครงสร้างของ WSDL.....                                      | 8    |
| 2.5 UDDI (Universal Description ,Discovery and Integration) ..... | 9    |
| 2.6 Microsoft .NET.....   | 9    |
| 2.6.1 องค์ประกอบของ Microsoft .NET.....                           | 9    |
| 2.6.1.1 ภาษาที่ใช้ในการพัฒนาแอปพลิเคชัน.....                      | 9    |
| 2.6.1.2 คอมโพเนนต์พื้นฐาน.....                                    | 10   |
| 2.6.1.3 ไดนามิกเว็บเพจ.....                                       | 10   |
| 2.6.1.4 รันไทม์ไลบรารี.....                                       | 10   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และ IV ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

|  | หน้า |
|--|------|
| 2.6.1.5 ยูสเซอร์อินเตอร์เฟซ .....                        | 10   |
| 2.6.1.6 คาด้าเบสแอคเซส .....                             | 10   |
| 2.6.2 สถาปัตยกรรม .NET .....                             | 11   |
| 2.6.2.1 เลเยอร์ .NET Framework SDK .....                 | 11   |
| 2.6.2.2 เลเยอร์ Common Language Runtime.....             | 12   |
| 2.6.2.2.1 การจัดการกับหน่วยความจำเมื่อทำการประมวลผล..... | 15   |
| 2.6.2.2.2 ระบบการตรวจจับความผิดพลาด .....                | 16   |
| 2.6.2.2.3 รูปแบบการคอมไพล์แอปพลิเคชัน.....               | 15   |
| 2.6.2.2.4 เรื่องของชนิดตัวแปร .....                      | 16   |
| 2.6.2.2.5 รูปแบบการทำงานร่วมกับภาษาอื่นๆ.....            | 17   |
| 2.6.2.3 เลเยอร์ Base Class Library .....                 | 17   |
| 2.6.2.4 เลเยอร์ Common Language Specification .....      | 18   |
| 2.6.3 ภาษาและเครื่องมือของ Microsoft .NET .....          | 18   |
| 2.6.4 พอร์ตเบบิตติของระบบ.....                           | 19   |
| 2.6.5 การเปรียบเทียบระหว่าง Microsoft.NET กับ J2EE ..... | 19   |
| 2.6.5.1 การสนับสนุนเว็บเซอร์วิส .....                    | 19   |
| 2.6.5.2 ประสิทธิภาพการทำงาน.....                         | 20   |
| 2.7 Thai Map Guide TmgX Control.....                     | 22   |
| 2.7.1 Thai Map Guide คืออะไร .....                       | 22   |
| 2.7.1 ความสามารถของ Thai Map Guide.....                  | 22   |
| 2.7.3 การนำ TmgX Control มาใช้ในโปรแกรม.....             | 23   |
| บทที่ 3 การออกแบบและพัฒนา .....                          | 25   |
| 3.1 ภาพรวมของระบบ .....                                  | 25   |
| 3.2 BuildingEditor .....                                 | 26   |
| 3.2.1 วัตถุประสงค์.....                                  | 26   |
| 3.2.2 การออกแบบพัฒนา.....                                | 27   |
| 3.2.2.1 โครงสร้างของ Building.....                       | 27   |
| 3.2.2.2 การสร้างแผนที่ภายในอาคาร.....                    | 29   |
| 3.2.2.3 การบันทึกข้อมูลที่สร้างไว้ลงคอมพิวเตอร์.....     | 30   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

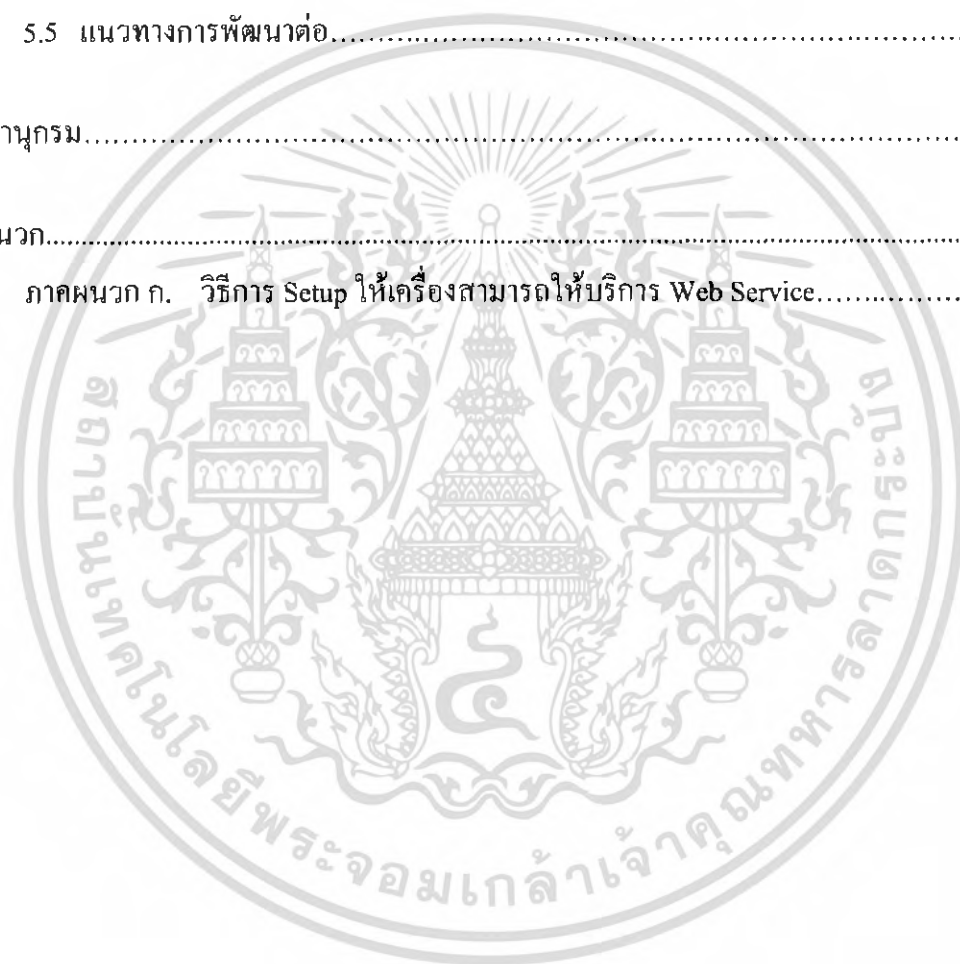
# สารบัญ (ต่อ)

|   | หน้า |
|---|------|
| 3.2.2.4 การอัปโหลดข้อมูลที่สร้างไว้ไปยังเซิร์ฟเวอร์.....                  | 32   |
| 3.2.7 เว็บเซอร์วิสที่ให้บริการข้อมูลที่สร้างจาก BuildingEditor.....       | 32   |
| 3.7.1 เว็บเซอร์วิสที่ให้บริการในการสร้างแผนที่ภายในอาคาร.....             | 32   |
| 3.7.2 เว็บเซอร์วิสที่ให้บริการในการดึงแผนที่ภายในอาคารไปใช้งาน.....       | 34   |
| 3.3 KMLLoader.....  | 35   |
| 3.3.1 วัตถุประสงค์.....   | 35   |
| 3.3.2 การออกแบบและพัฒนา.....  | 35   |
| 3.3.2.1 KML Format.....   | 36   |
| 3.3.2.2 การดึงข้อมูลจากไฟล์ KML.....                                      | 37   |
| 3.3.2.3 เว็บเซอร์วิสที่ใช้ร่วมกับ KMLLoader.....                          | 39   |
| 3.4 ThaiMapGuide Server.....  | 40   |
| 3.4.1 วัตถุประสงค์.....   | 40   |
| 3.4.2 การออกแบบและพัฒนา.....  | 41   |
| 3.4.2.1 TmgX Control.....   | 41   |
| 3.4.2.2 การเชื่อมต่อระหว่าง ThaiMapGuide Server กับเว็บเซอร์วิส.....      | 43   |
| 3.4.2.3 เว็บเซอร์วิสที่ใช้ร่วมกับ ThaiMapGuide Server.....                | 44   |
| บทที่ 4 การทดลองและผลการทดลอง.....  | 45   |
| 4.1 Building Editor.....  | 45   |
| 4.2 การทดลองใช้งาน โปรแกรม Building Editor.....                           | 45   |
| 4.2.1 วิธีการใช้งาน Tab Building.....                                     | 48   |
| 4.2.2 วิธีการใช้งาน Tab Floor.....  | 48   |
| 4.2.3 วิธีการสร้าง Polygon.....   | 52   |
| 4.2.4 การบันทึกไฟล์.....  | 55   |
| 4.2.5 การเปิดไฟล์ที่บันทึกไว้.....  | 56   |
| 4.2.6 การทดลองโดยการทำการบันทึกข้อมูลที่ได้สร้างขึ้นลงสู่คอมพิวเตอร์..... | 58   |
| 4.2.7 การทดลองอัปโหลดข้อมูลผ่าน Web Service.....                          | 60   |
| 4.3 การทดลองใช้ KMLLoader.....  | 61   |
| 4.4 การทดลองใช้ ThaiMapGuide Server.....                                  | 64   |

# สารบัญ (ต่อ)

หน้า

|   |    |
|---|----|
| บทที่ 5 บทวิจารณ์และสรุป.....                                       | 67 |
| 5.1 บทสรุป.....   | 67 |
| 5.2 วิจารณ์สิ่งที่ได้จากโครงงาน.....                                | 67 |
| 5.3 ปัญหาอุปสรรคที่เกิดขึ้น.....                                    | 68 |
| 5.4 แนวทางแก้ไข.....  | 68 |
| 5.5 แนวทางการพัฒนาต่อ.....  | 68 |
| บรรณานุกรม.....   | 69 |
| ภาคผนวก.....  | 70 |
| ภาคผนวก ก. วิธีการ Setup ให้เครื่องสามารถให้บริการ Web Service..... | 71 |



# สารบัญตาราง

| ตารางที่                       | หน้า |
|--------------------------------|------|
| 2.1 แสดงโครงสร้างของ WSDL..... | 8    |

# สารบัญรูป

| รูปที่   | หน้า |
|--|------|
| 1.1 แสดงการทำงานโดยรวมของ โปรแกรม.....   | 2    |
| 2.1 Web Service Technology.....  | 5    |
| 2.2 โครงสร้างการพัฒนาแอปพลิเคชัน.....  | 11   |
| 2.3 โครงสร้างแอปพลิเคชันแรกๆ การติดต่อกันระหว่างแอปพลิเคชันเป็นเรื่องยาก.....          | 12   |
| 2.4 แสดงการพัฒนาแอปพลิเคชันแบบ COM ที่สามารถเรียกใช้ฟังก์ชันของแอปพลิเคชันอื่นได้..... | 12   |
| 2.5 การพัฒนาแอปพลิเคชันบนเทคโนโลยี .NET คลาสภายในติดต่อกันได้โดยตรง.....               | 13   |
| 2.6 โครงสร้างการคอมไพล์จาก Visual Studio .net และเมื่อมีการนำไปใช้งานจริง.....         | 13   |
| 2.7 แสดงสถาปัตยกรรมของ Common Language Runtime.....                                    | 14   |
| 2.8 ความสามารถในการติดต่อกันแอปพลิเคชันเมื่อเขียนด้วยภาษาที่ต่างกัน.....               | 15   |
| 2.9 Implementing Sun's Java Pets tore with Microsoft .NET.....                         | 20   |
| 2.10 .Net Pet Shop vs. Revised Java Pet Performance.....                               | 21   |
| 2.11 ตัวอย่างโปรแกรมที่มี Thai Map Guide Control อยู่ภายใน.....                        | 22   |
| 2.12 Toolbox ของ Microsoft Visual Basic.NET.....                                       | 23   |
| 2.13 ภาพแสดงการเพิ่ม TmgX Control เข้าไปในโปรแกรม.....                                 | 24   |
| 3.1 แสดงโครงสร้างของระบบ.....  | 25   |
| 3.2 แสดงโครงสร้างของ Class ที่เกี่ยวข้องกับ Building.....                              | 28   |
| 3.3 แสดงภาพหน้าจอขณะใช้งานโปรแกรม ThaiMapGuide Server.....                             | 40   |
| 4.1 แสดงลักษณะและส่วนประกอบของ Tab Building.....                                       | 46   |
| 4.2 แสดง Dialog Box ที่ใช้ในการเลือกเพื่อเพิ่มรูปให้กับอาคาร.....                      | 46   |
| 4.3 แสดงภาพที่ถูกนำมาแสดงภายหลังจากถูกเลือก.....                                       | 47   |
| 4.4 แสดงส่วนที่ใช้ในการแก้ไขรายละเอียดภายนอกอาคาร.....                                 | 47   |
| 4.5 แสดงลักษณะและส่วนประกอบของ Tab Floor.....  | 48   |
| 4.6 แสดง Dialog Box ที่ใช้ในการกำหนดจำนวนชั้นให้กับอาคาร.....                          | 49   |

## สารบัญรูป (ต่อ)

| รูปที่  | หน้า |
|---|------|
| 4.7 แสดงชั้นทั้งหมด.....  | 49   |
| 4.8 แสดง Dialog Box ที่ใช้ในการตั้งชื่อชั้น.....                        | 50   |
| 4.9 แสดงค่าที่เพิ่มเข้ามาใน Combo Box ภายหลังจากทำการ New Floor.....    | 51   |
| 4.10 แสดง Dialog Box ที่ใช้ในการเลือกภาพ Background.....                | 51   |
| 4.11 แสดงการนำภาพ Background ที่ถูกเลือกมาแสดง.....                     | 52   |
| 4.12 แสดงลักษณะของ Marker ที่จะปรากฏขึ้นเมื่อทำการ Click ซ้ายบนภาพ..... | 52   |
| 4.13 แสดงลักษณะการวาดเส้นเชื่อมต่อระหว่างจุด.....                       | 53   |
| 4.14 แสดงตัวอย่างของการวาดครบ 1 โพลีกอน.....                            | 53   |
| 4.15 แสดงการเคลื่อนย้ายจุดของ Polygon.....                              | 54   |
| 4.16 แสดงภาพตัวอย่างระหว่างการใช้งาน Building Editor.....               | 54   |
| 4.17 แสดงตัวอย่างการ Clear ค่าเมื่อ Click ปุ่ม Reset.....               | 55   |
| 4.18 แสดง Dialog Box ที่ใช้ในการบันทึกข้อมูลลง File.....                | 56   |
| 4.19 แสดง Dialog Box ที่ใช้ในการเปิดข้อมูลที่ถูกเก็บไว้ใน File.....     | 57   |
| 4.20 แสดงการเปิดข้อมูลจากไฟล์.....                                      | 57   |
| 4.21 ภาพแสดงการทดสอบคำสั่ง getBuilding.....                             | 60   |
| 4.22 ภาพแสดงผลการทดสอบการใช้คำสั่ง getBuilding.....                     | 61   |
| 4.23 ภาพแสดงหน้าจอเริ่มต้นการใช้งาน KMLLoader.....                      | 62   |
| 4.24 ภาพแสดงการเปิดไฟล์ KML.....  | 63   |
| 4.25 ภาพแสดงหน้าจอหลังจากทำการอ่านไฟล์ KML เสร็จแล้ว.....               | 63   |
| 4.26 ภาพแสดงหน้าจอเริ่มต้นการใช้งาน โปรแกรม ThaiMapGuide Server.....    | 64   |
| 4.27 ภาพแสดงหน้าจอเริ่มต้นทดสอบคำสั่ง GetMapTmg.....                    | 65   |

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของโครงการ

การจะไปยังสถานที่ซักแห่งในปัจจุบันนั้น แผนที่ถือเป็นสิ่งที่สามารถช่วยได้อย่างมาก ระบบให้บริการแผนที่นี้ จะช่วยอำนวยความสะดวกในการหาตำแหน่งปัจจุบันที่อยู่ หาตำแหน่งสถานที่ต่างๆ โดยโครงการนี้จะคอยส่งแผนที่ให้กับ Web Service ทำให้สามารถดูแผนที่ได้จากทั้ง Browser และ Pocket PC

นอกจากนี้ ในโครงการยังมีส่วนของการแสดงแผนที่ภายในอาคารสถานที่ ทำให้ผู้ใช้สามารถหาข้อมูลสถานที่ต่างๆ ในอาคาร เช่น ร้านอาหาร และยังสามารถเข้าไปดูรายละเอียดของร้านอาหารจนกระทั่งจองที่นั่งในร้านอาหารได้เลย ซึ่งสิ่งเหล่านี้จะช่วยให้การใช้ชีวิตในปัจจุบันมีความสะดวกสบาย และสนุกมากยิ่งขึ้น

### 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1. เพื่อให้บริการแผนที่ประเทศไทยแก่ผู้ใช้ผ่านทางระบบเว็บเซอร์วิส
- 1.2.2. เพื่อให้บริการตำแหน่งปัจจุบันของผู้ใช้ โดยอ้างอิงตำแหน่งของ Pocket PC ที่ผู้ใช้ถืออยู่
- 1.2.3. เพื่อให้บริการแผนที่ภายในบริเวณอาคารขนาดใหญ่ เช่น ห้างสรรพสินค้า
- 1.2.4. เพื่อให้บริการในการติดตามตำแหน่งปัจจุบันของผู้ใช้ Pocket PC

### 1.3 ขอบเขตของโครงการ

โครงการนี้ประกอบด้วยส่วนต่างๆ ทั้งหมด 4 ส่วน ดังนี้

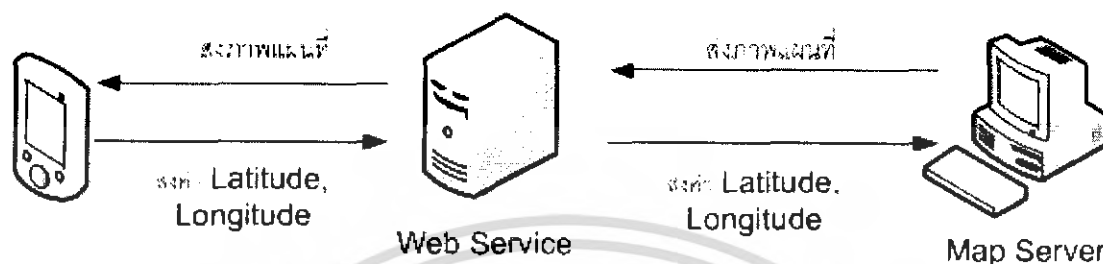
#### 1.3.1. ส่วนของการให้บริการแผนที่ประเทศไทยในแบบ 2 มิติผ่านเว็บเซอร์วิส

ในส่วนนี้จะพัฒนาบน Microsoft Visual Basic .NET ใช้ ActiveX Control ที่มีชื่อว่า Thai Map Guide (TmgX Control) ซึ่งเป็น Control ที่จะช่วยอำนวยความสะดวกในการแสดงแผนที่ ซึ่งตัว Control ตัวนี้ี้จะมีความสามารถในการจัดการกับแผนที่มากมาย อาทิ เช่น สามารถย่อ-ขยายแผนที่, สามารถเลื่อนไปมาในแผนที่ได้, สามารถหาตำแหน่ง Latitude, Longitude ได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมตัวนี้จะทำหน้าที่เป็น Server ซึ่งจะรับการติดต่อจากส่วนของ Web Service จากกลุ่มที่ทำในส่วนของ Front-End แล้วจึงทำการส่งภาพแผนที่ในตำแหน่งที่ทางฝั่ง Web Service ต้องการมากลับไปให้

สามารถอธิบายลักษณะของการทำงานได้ดังภาพ



รูปที่ 1.1 แสดงการทำงานโดยรวมของโปรแกรม

### 1.3.2. ส่วนของการสร้างแผนที่ภายในอาคารบนคอมพิวเตอร์

ในส่วนนี้จะพัฒนามัน Microsoft Visual C#.NET โดยโปรแกรมนี้จะเป็นโปรแกรมที่ใช้สำหรับสร้างแผนที่ภายในอาคาร โดยการทำงานนั้น เริ่มแรกเราจะต้องสร้างแผนที่ที่เป็นไฟล์รูปภาพขึ้นมาก่อน (อาจจะมีสร้างจากโปรแกรมวาดภาพเช่น Photoshop, CorelDraw เป็นต้น) จากนั้น ก็ให้ใช้โปรแกรมนี้ทำการโหลดภาพเข้ามาและทำการสร้างบริเวณต่างๆ ภายในอาคารเข้าไป เช่น เพิ่มรายละเอียดของการเชื่อมต่อไปยังร้านอาหาร ให้กับร้านอาหารนั้นๆ โดยหลังจากทำเสร็จแล้วก็สามารถอัพโหลดไปยัง Web Services เพื่อให้ทำการเก็บข้อมูลได้ หรือจะทำการเซฟไว้ในคอมพิวเตอร์เพื่อใช้งานในภายหลังก็สามารถทำได้

### 1.3.3. Web Service ที่ใช้ร่วมกับแผนที่ภายนอกและภายในอาคาร

ในส่วนนี้จะเป็นการพัฒนา Web Service โดยใช้ Microsoft Visual C#.NET ที่ใช้สำหรับทำงานร่วมกับหัวข้อ 1.4.1 และ 1.4.2 ยกตัวอย่างเช่น Web Service ที่ให้บริการข้อมูลแผนที่ภายในอาคาร ซึ่งเกี่ยวข้องกับ โปรแกรมที่ใช้สร้างแผนที่ภายในอาคาร เป็นต้น

การใช้ Web Service มาร่วมนั้น ทำให้สามารถทำการรวมกับโครงงานส่วนของ Backend ได้ง่ายมากขึ้น อีกทั้ง ยังทำให้ส่วนของ Front-End ซึ่งเป็นโครงงานของกลุ่ม Front-End สามารถเรียกไปใช้ได้สะดวกมากยิ่งขึ้นอีกด้วย

## 1.4 วิธีการดำเนินการ

1. ทำการศึกษาเทคโนโลยี .NET
2. ทำการศึกษาการเขียนโปรแกรมภาษา Basic บน Microsoft Visual Studio.NET
3. ทำการศึกษาการเขียนโปรแกรมภาษา C# บน Microsoft Visual Studio.NET
4. ทำการศึกษาการทำงานของ TmgX Control
5. ทำการทดลองนำ TmgX Control มาใช้ในโปรแกรมที่พัฒนาบน Microsoft Visual Basic.NET เพื่อทำการแสดงผลแผนที่ในบริเวณต่างๆ ตามที่กำหนดให้ทั้งในกรุงเทพและต่างจังหวัด
6. ทดลองเขียนโปรแกรมในการส่งข้อมูลภาพ 2 มิติใดๆ โดยใช้คำสั่งจาก Web Service เพื่อนำไปเก็บไว้ใน Database
7. ทำการศึกษาการเขียนโปรแกรมและการทำงานของ Map Editor ในแบบต่างๆ จากอินเทอร์เน็ต เพื่อนำมาประยุกต์ใช้กับโครงการ
8. วิเคราะห์และออกแบบระบบ
9. ทำการพัฒนาโปรแกรมทั้งหมด
10. วิเคราะห์ผลการพัฒนาโปรแกรมรวมทั้งทำการแก้ไขข้อผิดพลาดต่างๆ ของโปรแกรม

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1. ได้รับความรู้ความเข้าใจในการเขียนโปรแกรมบนเทคโนโลยี .NET
- 1.3.2. ได้รับความรู้ความเข้าใจในการทำ Web Service
- 1.3.3 ได้รับความรู้ความเข้าใจเกี่ยวกับการใช้โปรแกรม Thai Map Guide เพื่อให้บริการแผนที่ประเทศไทย
- 1.3.5. ได้ระบบแผนที่สำหรับให้บริการผู้ใช้

## 1.6 ส่วนประกอบของปฏิญานิพนธ์

ปฏิญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงการ ซึ่งประกอบด้วยทฤษฎีอะไรบ้าง ให้บรรยายทฤษฎีทั้งหมดโดยละเอียด

บทที่ 3 กล่าวถึงชิ้นงานของโครงการนี้ ส่วนที่ได้พัฒนาขึ้น การทำงานของระบบหรือเอกสารชิ้นงานบรรยายโดยละเอียด ทรัพยากรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 กล่าวถึงการทดลองและผลการทดลอง ของส่วนที่ได้ทำการพัฒนาขึ้น รวมทั้งผลการทดลองของระบบทั้งหมด

บทที่ 5 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงการ วิจัยสิ่งที่ได้รับจากโครงการ และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

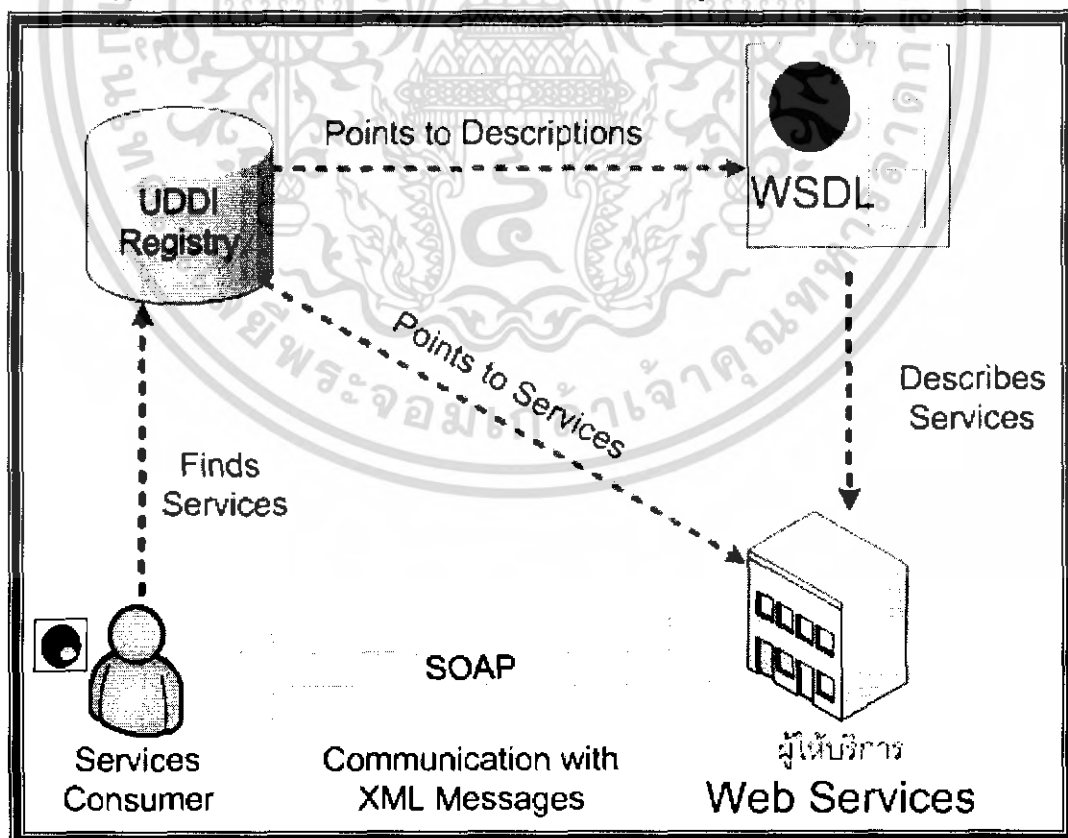
## บทที่ 2

# ทฤษฎีพื้นฐานที่ใช้ในโครงการ

### 2.1 เว็บเซอร์วิส

เว็บเซอร์วิส เป็นระบบซอฟต์แวร์ที่ออกแบบมาเพื่อสนับสนุนการทำงานระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ผ่านระบบเครือข่าย โดยที่ภาษาที่ใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์คือภาษาเอ็กซ์เอ็มแอล (XML) การอธิบายวิธีการใช้งานของเว็บเซอร์วิสนั้นจะอธิบายโดยใช้ภาษาวิสเดิล WSDL (Web Services Description Language) ซึ่งเป็นภาษา XML ประเภทหนึ่ง ระบบอื่นๆ จะสามารถติดต่อและทำงานกับเว็บเซอร์วิส โดยใช้ โปรโตคอลที่ชื่อว่า SOAP (Simple Object Protocol) ซึ่งใช้ภาษา XML เป็นมาตรฐานในการติดต่อระหว่างระบบโดยผ่านทางโปรโตคอลอื่นที่ใช้ในการส่งข้อมูลบนเว็บ อาทิเช่น โปรโตคอล HTTP

Webservice กับ Web Application ต่างกันอย่างไร สิ่งที่ทำให้ต่างกันนั่นก็คือ จุดกำเนิดและจุดประสงค์ของทั้งสองนั้นต่างกัน Webservice นั้นเกิดมาจากการที่ Web Application ถูกพัฒนาได้จากหลายภาษา เช่น asp php jsp perl ..... ทำให้การที่จะนำมารวมแล้วทำงานร่วมกันนั้นเป็นเรื่องที่ยากลำบาก (เหมือนคุยกันคนละภาษา) Webservice จึงเหมือนกับภาษาสากล ที่ทำให้แต่ละ Web Application ทำงานร่วมกันได้โดยผ่านทาง SOAP ที่มีรูปแบบเป็น XML



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษา **รูปที่ 2.1** Web service technology อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Services Consumer:**

เป็นใครก็ตามที่ต้องการเรียกใช้บริการจากผู้ให้บริการซึ่งสามารถค้นบริการที่ต้องการได้จาก UDDI Registry หรือ Service Registry หรือติดต่อจาก Provider โดยตรง

**UDDI Registry:**

ทำหน้าที่เป็นตัวกลางให้ผู้ให้บริการมาลงทะเบียนไว้ โดยใช้ WSDL ไฟล์ บอกรายละเอียดของ บริษัทและบริการที่มีให้ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้

**Provider:**

เป็นผู้ให้บริการ มีหน้าที่ในการเปิดบริการเพื่อรองรับการขอใช้บริการจากผู้ใช้บริการ (Requestor) ที่เรียกเข้ามาขอใช้

**2.2 XML (The Extensible Markup Language)**

XML (The Extensible Markup Language 1.0) เป็นภาษา Markup ที่เป็น text-based ซึ่งทำให้เป็น มาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตอย่างรวดเร็ว ผู้ที่ทำหน้าที่รับผิดชอบ และกำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) ความแตกต่างระหว่าง XML กับ HTML คือ HTML ถูกนำมาใช้ในการสร้าง เว็บไซต์ ที่สามารถแสดงผลได้โดย โปรแกรมเบราว์เซอร์ แต่ XML จะใส่ tags ได้อย่างอิสระ แล้วทำการส่ง XML ชุดนี้ไปประมวลผลยังแอปพลิเคชันใด ๆ ที่สามารถใช้ข้อมูลใน XML นี้

**2.2.1 XML and HTML**

HTML ภาษาที่ใช้ในการเขียน Web มากที่สุดนั้นเป็นเพราะมีรูปแบบที่ง่ายต่อการแสดงผลของ Browser เนื่องจาก มี tag ตายตัวที่สามารถบอกได้ว่าเมื่อเจอ tag นี้จะแสดงผลอย่างไร เช่น เมื่อเจอ tag <B>...</B> ในเอกสารก็ให้แสดงข้อความที่อยู่ระหว่าง tag เป็นตัวหนา แต่จะสังเกตได้ว่าคอมพิวเตอร์จะไม่เข้าใจว่าข้อความนั้นคืออะไร เพียงแต่รู้ว่าจะแสดงผลอย่างไร นั่นแสดงว่าไม่สามารถนำข้อมูลภายใน tag เหล่านี้ไปทำการประมวลผลใดๆ ได้เลย

XML เป็นภาษาที่มีลักษณะเป็น tag คล้าย HTML แต่ไม่ได้มุ่งที่การแสดงผล XML มุ่งที่การสื่อความหมายโดยอนุญาตให้ผู้ใช้งานสามารถกำหนด tag ขึ้นได้เองเพื่อให้สื่อความหมายทางภาษาของมนุษย์ แต่คอมพิวเตอร์เองก็เข้าใจเช่นกัน ทำให้ข้อมูลระหว่าง tag สามารถนำไปประมวลผลต่อได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3 SOAP (Simple Object Access Protocol)

SOAP กลายเป็นสิ่งที่มีความสำคัญสำหรับ Web Services อย่างรวดเร็ว เป็นโปรโตคอลที่ผู้จัดหา Web Services เลือกใช้ที่จะส่ง message ระหว่าง Web Services SOAP เป็น Transport Protocol ที่มี XML เป็นพื้นฐานและใช้ HTTP เป็นโปรโตคอลร่วมในการส่งผ่านเครือข่าย SOAP จะระบุวิธีในการเข้ารหัสส่วนหัว (Header Encoding) ของทั้ง HTTP และไฟล์ XML ไว้อย่างชัดเจนทั้งใน ส่วนของการติดต่อไปยังคอมพิวเตอร์อีกเครื่องหนึ่งและส่งผ่านข้อมูลไปให้ รวมถึงระบุวิธีที่โปรแกรมซึ่งถูกเรียกนั้นจะส่งค่าคืนกลับมาด้วย

SOAP (Simple Object Access Protocol) เป็น XML-based โปรโตคอล (lightweight protocol) และใช้ HTTP เป็นโปรโตคอลร่วม สำหรับการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (decentralized, distributed environment) SOAP ได้ กำหนดเมสเซจจิงโปรโตคอล (Messaging Protocol) ระหว่างผู้ขอบริการ (requestor) กับผู้ให้บริการ (provider) เช่น ผู้ขอบริการสามารถติดต่อแลกเปลี่ยนข้อมูลกับผู้ให้บริการโดยใช้ RMI (Remote Method Invocation) ตามวิธีการของ โปรแกรมแบบออปเจ็ค บริษัทไมโครซอฟท์, ไอบีเอ็ม, ลอรัส, ยูสเซอร์แลนด์ (User Land) และ ดีเวลลอปเปอร์เมนเตอร์ (Developer enter) ได้ร่วมกันกำหนดมาตรฐานของ SOAP ขึ้น ซึ่งต่อมาได้มีบริษัทอีก 30 กว่าบริษัทเข้าร่วมและ จัดตั้งเป็น W3C XML Protocol Workgroup ขึ้น SOAP ได้กำหนดรูปแบบพื้นฐานของการสื่อสารแบบกระจายขึ้น โดยการพัฒนา SOA แม้ว่า SOA จะไม่ได้กำหนดเมสเซจจิงโปรโตคอล (Messaging Protocol) ไว้ แต่ SOAP ได้ถูกกำหนด ให้เป็น Services-Oriented Architecture Protocol เรียบร้อยแล้ว เนื่องจากมันได้ถูกใช้ในการพัฒนา SOAP อย่างแพร่ หลายแล้วนั่นเอง จุดเด่นของ SOAP ก็คือเป็นโปรโตคอลที่เป็นกลาง กล่าวคือ ไม่มีใครเป็นเจ้าของและเป็นโปรโตคอล ที่ทำงานกับโปรโตคอลอื่นหลายชนิด การพัฒนาก่อนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการ แบบจำลองทางวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา

### 2.4 WSDL (Web Services Description Language)

WSDL (Web Services Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะการใช้บริการของ Web Services และวิธีการติดต่อกับ Web Services ความต้องการของนิยามนี้ เกี่ยวเนื่องกับความต้องการของ distributed system ที่จะกำหนด Interface Definition Language (IDL) โดยใช้ภาษา XML, WSDL เกิดจากการรวมแนวคิดของ NASSL (The Network Accessible Service Specification Language), WDS (Well-Defined Services) ของบริษัทไอบีเอ็ม, SDL (The Service Description Language) และ SCL (the SOAP Contract Language) ของบริษัทไมโครซอฟท์ ปัจจุบัน WSDL เป็นภาษา ที่อยู่ในการดูแลของ W3C (World Wide Web

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Consortium) ซึ่งยังไม่เป็นมาตรฐานที่สมบูรณ์ เวอร์ชันที่ใช้งานอยู่ใน ปัจจุบันคือ WSDL 1.1 (รายละเอียดเพิ่มเติมเกี่ยวกับ WSDL สามารถศึกษาเพิ่มเติมได้ที่ <http://www.w3c.org/TR/wsdl>)

WSDL คือ มาตรฐานสำหรับการประกาศ process ที่จำเป็นในการเรียกใช้เซอร์วิส SOAP (Simple Object Access Potocal)

### 2.4.1 โครงสร้างเอกสาร WSDL

WSDL เป็นภาษาที่อยู่ในความดูแลขององค์กร W3C (World Wide Web Consortium) version ที่มีอยู่ในปัจจุบัน คือ WSDL 1.1 ในการใช้งานจริง หากเราสร้างบริการ Web Services ก็จะมีเครื่องมือช่วยสร้างเอกสาร WSDL สำหรับ Web Services อย่างอัตโนมัติ จุดภายในเอกสารที่เราควรรู้เกี่ยวกับการติดต่อและเรียกใช้บริการของ Web Services มีจุดที่ควรรู้ ดังนี้

#### ตารางที่ 2.1 แสดงโครงสร้างของ wsdl

| Element     | Definition   |
|-------------|--|
| <portType>  | เป็นส่วนที่สำคัญที่สุดใน WSDL element อธิบาย operations ที่ web service มีให้บริการและ messages ที่เกี่ยวข้อง เทียบได้กับ function library หรือ module หรือ class ในการเขียน โปรแกรม |
| <operation> | อธิบาย method ที่ให้บริการ Web Services หนึ่งจะมี method จำนวนกี่ method ก็ได้   |
| <message>   | อธิบาย data elements ของ operation แต่ละ message อาจมีมากกว่าหนึ่งส่วนเทียบได้กับ parameter ของ function ในการเขียน โปรแกรม  |
| <types>     | อธิบายชนิดข้อมูลที่ web service ใช้ เพื่อความเป็นกลาง WSDL ใช้ XML Schema syntax ในการระบุชนิดข้อมูล   |
| <binding>   | อธิบาย format ของ message และ protocol details ในแต่ละ port  |
| <service>   | สำหรับ web server จะมี Web Services จำนวนกี่บริการก็ได้ และชื่อ Web Services ก็เป็นตัวจำแนกและบ่งบอกแต่ละบริการซึ่งห้ามมีชื่อซ้ำกัน  |

ตามทฤษฎีแล้ว ไฟล์เอกสาร WSDL แต่ละไฟล์ สามารถอธิบายคุณลักษณะของบริการ Web Services ได้มากกว่า 1 บริการ โดยแต่ละ Web Services จะมี port สื่อสารเฉพาะตัว ซึ่งบ่งบอกไว้ในเอกสาร WSDL อยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 UDDI (Universal Description, Discovery and Integration)

UDDI เป็นที่เก็บรวบรวม Web Services ต่างๆ ในอินเทอร์เน็ต ไว้ในแหล่งเดียวกัน เพื่อให้ผู้ใช้บริการสามารถค้นหาได้ง่ายๆ หากเปรียบเทียบง่ายๆ ให้มองเสมือนสมุดหน้าเหลืองที่เราใช้ในการเปิดดูเบอร์โทรศัพท์

- ผู้เริ่มก่อตั้ง UDDI ในช่วงแรกคือ IBM และ Microsoft และ Ariba ซึ่งเป็นบริษัทที่ทำธุรกิจ B2B ปัจจุบันมีบริษัทที่มีส่วนร่วมในการกำหนดมาตรฐานของ UDDI มากกว่า 70 บริษัท
- UDDI ถูกสร้างขึ้นมาเพื่อเป็นมาตรฐานในการค้นหาบริการของ WS สำหรับคู่ค้าทางธุรกิจ (Business Partner)
- UDDI Business Registry เป็นฐานข้อมูล WS ของบริษัทคู่ค้าทางธุรกิจ
- ในปัจจุบันบางบริษัทก็ตั้งตัวเองเป็นตัวแทนผู้ให้บริการ (Service brokers)

## 2.6 Microsoft .NET

คำว่า .NET มาจากโครงการของทางไมโครซอฟท์ที่ต้องการจะผนวกรวมการพัฒนาแอปพลิเคชันของทาง Desktop และทาง Web เข้าด้วยกัน ซึ่งผลิตภัณฑ์ในอนาคตทั้งหมดของไมโครซอฟท์จะทำงานอยู่บนพื้นฐานของ .NET ทั้งหมด พูดย่อแบบง่ายๆ คือ .NET นั้นถูกออกแบบมาโดยมีพื้นฐานของอินเทอร์เน็ตอยู่ในใจ (Highly distributed environment of the Internet) ทำให้อินเทอร์เน็ตเป็นพื้นฐานอยู่ภายในระบบปฏิบัติการ (Operating System) ซึ่งแนวคิดดังกล่าวนี้ได้รับแรงบันดาลใจมาจากหลาย ๆ สิ่งแต่สิ่งหนึ่งในนั้นมาจากภาษา Java โดยเฉพาะอย่างยิ่งในเรื่องของ Java Virtual Machine อาจกล่าวได้ว่า Microsoft .NET หรือเรียกสั้น ๆ ว่า .NET ก็คือแพลตฟอร์ม (สภาวะแวดล้อมของฮาร์ดแวร์หรือซอฟต์แวร์ที่โปรแกรมใช้ในการรัน) ใหม่สำหรับการสร้างแอปพลิเคชันที่มีลักษณะตั้งแต่แบบ standalone (รันบนคอมพิวเตอร์เครื่องเดียว) จนถึงแบบ web-based application และแบบ web service โดย .NET ได้ให้ .NET Framework สำหรับสร้างแอปพลิเคชันต่าง ๆ โดยมี Visual Studio .NET เป็นเครื่องมือสำหรับสร้างแอปพลิเคชันต่าง ๆ ที่สนับสนุน .NET Framework ซึ่งจะมีหลายภาษาให้เลือกใช้ เช่น Visual C++ .NET, C# (ซีชาร์ป คือ ภาษาตัวใหม่จากค่ายไมโครซอฟท์ซึ่งคล้ายภาษา Java)

### 2.6.1 องค์ประกอบของ Microsoft.NET

**2.6.1.1 ภาษาที่ใช้ในการพัฒนาแอปพลิเคชัน:** Microsoft.Net เลือกใช้ภาษา C# เป็นหลัก ไมโครซอฟท์พัฒนาภาษา C# ขึ้นมาใหม่เพื่อใช้กับ .Net โดยเฉพาะ ภาษา C# มีรากฐานมาจากภาษา C และ C++ แอปพลิเคชันที่เขียนด้วย C# อาจเลือกคอมไพล์เป็นไบท์โค้ดในฟอร์แมต Internal Language (IL) ไบท์โค้ดที่ได้จะทำงานบน Common Language Runtime (CLR) นอกจากนี้ยังอาจเลือกคอมไพล์ C# เป็นเนทีฟไค้ดเลยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.6.1.2 คอมโพเนนต์พื้นฐาน :** ไมโครซอฟท์เตรียมคอมโพเนนต์สำหรับเรียกใช้ใน Microsoft.Net ในชุด .Net Framework SDK

**2.6.1.3. ไดนามิกเว็บเพจ:** Microsoft.Net สามารถสร้างเว็บเพจแบบไดนามิกด้วย Active Server Page หรือ ASP.Net โดยอาจเลือกใช้วิชาพลศาสตร์หรือ C# มาเขียนโปรแกรมเพื่อคอมไพล์เป็นเน็ตฟโค้ดเลขก็ได้

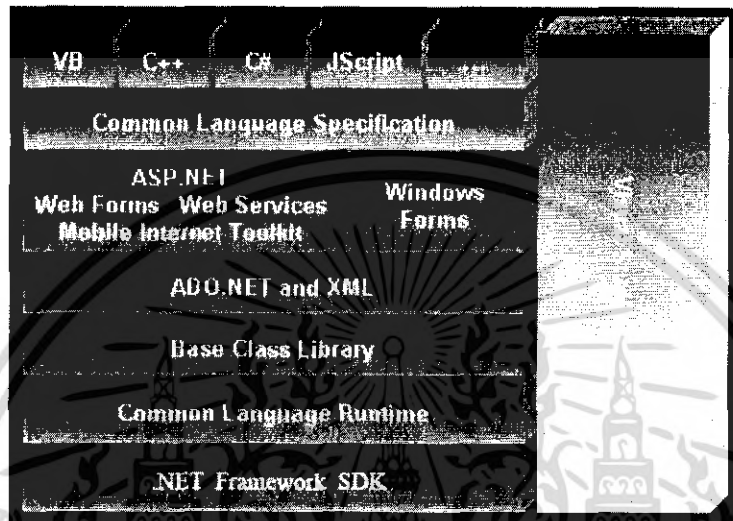
**2.6.1.4 รั้นโทมไอบรรารี:** ไอบรรารีพื้นฐานที่โปรแกรมโค้ดจะทำงานอยู่บนไอบรรารีของ Microsoft.Net ใช้ Common Language Runtime เป็นพื้นฐานให้ไบท์โค้ดทำงาน จุดเด่นของ CLR ก็คือ สนับสนุนการใช้ภาษาคอมพิวเตอร์อื่นๆ ผู้ใช้สามารถใช้ภาษา Visual Basic, Perl, Eiffel, COBOL, Fortran มาเขียนโปรแกรมได้ เพราะโปรแกรมที่เขียนจะคอมไพล์เป็นไบท์โค้ดเพื่อทำงานบน CLR

**2.6.1.5 ยูสเซอร์อินเตอร์เฟซ:** ใน Microsoft.Net มียูสเซอร์อินเตอร์เฟซคอมโพเนนต์ให้ใช้ คือ Win Forms และ Web Forms การใช้งานจะเรียกผ่าน Microsoft Visual Studio.Net ซึ่งมี IDE ทั้งสองแบบที่สามารถนำมาออกแบบแอปพลิเคชันได้

**2.6.1.6 ดาต้าเบสแอคเซส:** การติดต่อกับดาต้าเบสบน Microsoft.Net นั้นมี ADO.Net ให้ใช้ สามารถใช้งานร่วมกับ XML เพื่อนำดาต้าเบสมาใช้งานบนเว็บเพจ เว็บเซอร์วิสของ .Net จะใช้ประโยชน์จาก ADO มาก

## 2.6.2 สถาปัตยกรรม .NET

เราทราบแล้วว่า .NET เป็นนิยามการให้บริการของซอฟต์แวร์ในรูปแบบของเซอร์วิส ซึ่งจะรันได้โดยไม่ขึ้นอยู่กับอุปกรณ์ที่ใช้แสดงผล หรือระบบปฏิบัติการใดๆ ในส่วนนี้จะกล่าวถึงโครงสร้างโดยรวมทั้งหมดของการสร้างแอปพลิเคชัน .NET ( .NET Application Architecture)ซึ่งแสดงดังรูปที่ 2.2



รูปที่ 2.2 โครงสร้างการพัฒนาแอปพลิเคชัน .NET

รูปที่ 3-1 แสดงถึงสถาปัตยกรรมของแอปพลิเคชัน .NET ที่พัฒนาด้วย Visual Studio.net ถือว่าเป็นการเปลี่ยนแปลงครั้งสำคัญ ทั้งแพลตฟอร์มที่ใช้ในการพัฒนาแอปพลิเคชัน และสถาปัตยกรรมที่ใช้โดยมีเลเยอร์ล่างสุดคือ .NET Framework SDK เปรียบเสมือน Runtime Library ที่จะรันอยู่ คอยสนับสนุนการทำงานของแอปพลิเคชัน จากนั้นจะเป็นเลเยอร์ของ Common Language Runtime เป็นผลลัพธ์ของการคอมไพล์ (compile) แอปพลิเคชัน .NET เลเยอร์ถัดขึ้นมาเป็นเครื่องมือ (Tools) และเทคนิคต่างๆที่เราสามารถใช้พัฒนาแอปพลิเคชันได้ทั้งในเรื่องของ web service, ADO.net, ASP.net จะกระทั่งถึงเลเยอร์บนสุดคือ ภาษาที่ใช้ในการพัฒนาแอปพลิเคชันด้วย Visual Studio.net

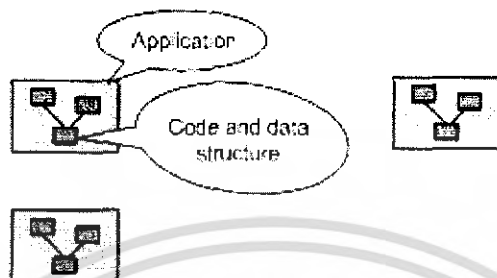
### 2.6.2.1 เวย์ .NET Framework SDK

สถาปัตยกรรม .NET Framework คือ กรอบการทำงานของการทำงานเขียน โปรแกรมที่ไม่โครซอฟต์แวร์คิดขึ้นมา เพื่อรองรับการติดต่อสื่อสาร เพื่อแลกเปลี่ยนข้อมูล (Exchange Data) ระหว่างกัน หรือแลกเปลี่ยนข้อมูลระหว่างแพลตฟอร์ม (Platform) ให้มีความสมบูรณ์ยิ่งขึ้น โดยอาศัยภาษา XML (Extensible Markup Language) ทำหน้าที่เป็นตัวกลางในการแลกเปลี่ยนข้อมูลระหว่างแพลตฟอร์มไฟล์ของฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

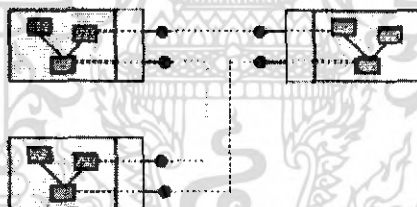
### 2.6.2.2 เดเวอร์ Common Language Runtime

ก่อนที่จะมีการพัฒนาโปรแกรมเป็น Object Oriented นั้นแอปพลิเคชันแต่ละตัวเปรียบเสมือนกล่อง ภายในแอปพลิเคชันก็จะมีโค้ด(code) มีโครงสร้างข้อมูล (data structure) ต่างๆของตัวเอง มีฟังก์ชันต่างๆของตัวเองแอปพลิเคชันนั้นๆ ดังแสดงในรูปที่ 2.3



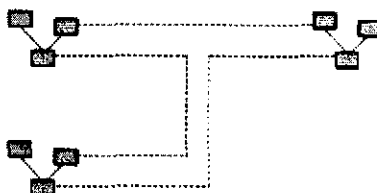
รูปที่ 2.3 โครงสร้างแอปพลิเคชันแรกๆ การติดต่อกันระหว่างแอปพลิเคชันเป็นเรื่องยาก

การที่แอปพลิเคชันต่างๆ จะมีการเรียกใช้การทำงานของกันและกัน หรือมีการส่งผ่านข้อมูลถึงกันและกัน เป็นสิ่งที่ทำได้ยาก ซึ่งอาจต้องมีการกำหนดอะไรขึ้นมาเองระหว่าง 2 แอปพลิเคชันนั้นๆ จนกระทั่งในยุคถัดมา ไมโครซอฟท์ได้คิดค้นเทคโนโลยี COM (Component Object Model) เป็นวิธีที่ทำให้เราเขียนโปรแกรมเป็นแบบ Object Oriented และเรียกใช้การทำงานที่มาจากต่างแอปพลิเคชันได้ ดังแสดงในรูปที่ 2.4



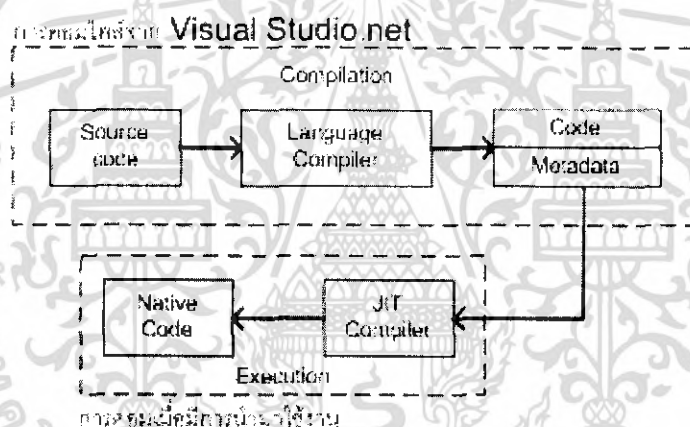
รูปที่ 2.4 แสดงการพัฒนาแอปพลิเคชันแบบ COM ที่สามารถเรียกใช้ฟังก์ชันของแอปพลิเคชันอื่นได้

หากจะอธิบายให้ง่ายขึ้น ก็คือเปรียบเทียบเราเอาแพ็คเกจอื่นหนึ่งห่อแอปพลิเคชันของเราไว้และการพูดคุยกันของแอปพลิเคชันก็พูดคุยผ่านแพ็คเกจที่เราห่อเอาไว้ จะกระทั่งมาถึงตัว Visual Studio.net ที่ได้รับการออกแบบใหม่ จะเห็นว่าจากรูปเดิม กล่องแพ็คเกจหายไป ถ้าเราพัฒนาด้วยรูปแบบเทคโนโลยี .NET นั้น คลาสต่างๆ สามารถติดต่อกันได้โดยตรง ดังรูปที่ 2.5



**รูปที่ 2.5** การพัฒนาแอปพลิเคชันบนเทคโนโลยี .NET คลาสภายในแต่ละแอปพลิเคชันสามารถติดต่อกันได้โดยตรง

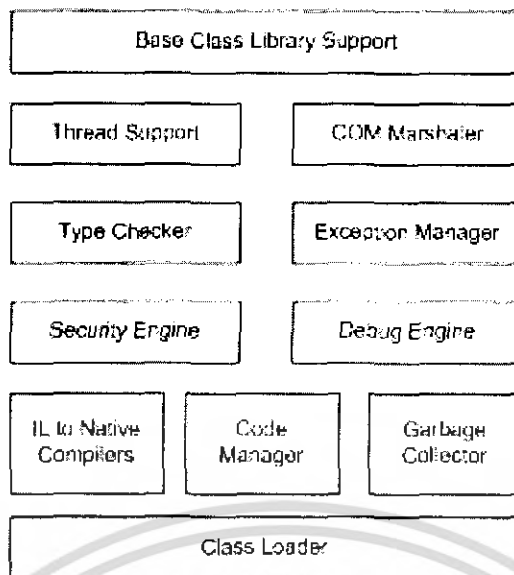
การพัฒนาแอปพลิเคชันด้วย Visual Studio.net นั้น เมื่อเราคอมไพล์สิ่งที่เราได้ จะไม่ใช่ไค้ดไบนารี (Binary Code) เลยทีเดียว แต่จะได้เป็นภาษากลางอันหนึ่งเรียกว่า Microsoft Intermediate Language (MSIL) ซึ่งเป็นภาษาในระดับเลเยอร์ล่างๆ โครงสร้างของภาษา (Syntax) จะเหมือนภาษา Assembly ภายในสิ่งที่เกิดจากการคอมไพล์ก็จะเป็น MSIL ตัวนี้ภายในตัวมันจะประกอบไปด้วย 2 ส่วน คือ ไค้ด กับตัวแอคทริบิวต์หรือพรีอเพอร์ตีต่างๆ ที่ใช้อธิบายตัวไค้คั้นซึ่งเรียกว่า Metadata ดังรูปที่ 2.6



**รูปที่ 2.6** โครงสร้างการคอมไพล์จาก Visual Studio.net และเมื่อมีการนำไปใช้งานจริง

จากนั้น เมื่อไค้คซึ่งเป็น Intermediate Language ถูกเรียกใช้งานจริงๆ จะมีตัว Compiler (ตัวแปลภาษา) ตัวหนึ่งมาทำการคอมไพล์ไค้คตัวนั้นให้เป็นไค้คไบนารี ซึ่ง Compiler ตัวนั้นจะเรียกว่า Just In Time Compiler (JIT Compiler) ที่เรียกว่า Just In Time เพราะว่าจะมีการคอมไพล์เมื่อมีการใช้งาน ฉะนั้นคลาสหรือไค้คต่างๆ ที่เราพัฒนาขึ้นแล้วจะถูกคอมไพล์มาเป็น Intermediate Language ที่มีโครงสร้างของภาษาแบบเดียวกัน เพราะฉะนั้นคลาสต่างๆ ในแอปพลิเคชันจึงสามารถทำงานได้อย่างกลมกลืนกันและไม่มีข้อติดขัดอะไร

การทำงานของ Common Language Runtime นั้นภายในตัว Common Language Runtime จะมีโมดูล (Module) ย่อยๆ ซึ่งเป็นสถาปัตยกรรมภายในดังรูปที่ 2.7



**รูปที่ 2.7** แสดงสถาปัตยกรรมของ Common Language Runtime

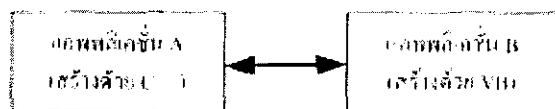
ด้านล่างสุดในรูปที่ 3-6 จะมี Class Loader ซึ่งเอาไว้โหลดโปรแกรมของเราขึ้นมาทำงานนอกจากนี้ก็จะมี Compiler ซึ่งจะทำการคอมไพล์ภาษา Intermediate Language ให้เป็นภาษาไบনারีโดยจะมีตัว Code Manager และ Garbage Collector คอยจัดการกับหน่วยความจำที่เราจองเวลาเรียกใช้งาน นอกจากนี้ก็จะมีเรื่องของระบบความปลอดภัย (Security) ในการทำงานรวมทั้ง Debug Engine ในการดัก Runtime Error และตัว Exception Manger การตรวจเช็คชนิดของตัวแปรต่างๆ และด้านบนสุดจะเป็นการใช้งานร่วมกับ Library Class ต่างๆ ซึ่งจะเตรียมมาให้เพราะฉะนั้นเลเยอร์ของ Common Language Runtime การคอมไพล์แอปพลิเคชันใดก็ตามไม่ว่าจะเป็น ASP.net หรือเขียนแอปพลิเคชันบน Windows ธรรมดาหรือจะเป็นการเขียน Web service ก็ตาม สิ่งที่ได้จากการคอมไพล์จะเป็น Common Language Runtime

นั่นคือจากการออกแบบเพื่อสนับสนุนการทำงานร่วมกันใน Common Language Runtime จึงขจัดข้อเสียของ COM ไปได้ เนื่องจากข้อจำกัดของ COM คือเป็นเพียงการเอาอะไรบางอย่างมาห่อคลาสเอาไว้เท่านั้น ดังนั้นเวลาที่เอา COM ไปใช้งานจึงค่อนข้างยุ่งยาก รวมทั้งถ้ามีการเปลี่ยนแปลง COM นั้นๆก็จะทำให้เกิดปัญหาเรื่องของการเข้ากันได้ (Compatibility) ระหว่างเวอร์ชันเดิมกับเวอร์ชันปัจจุบันแต่ถ้าเราพัฒนาด้วย Visual Studio.net ข้อเสียของ COM ทั้งหมดก็จะถูกขจัดเสีย

ความจริง Common Language Runtime เป็นหลักการทำงานที่มีวิวัฒนาการมาจาก COM อีกทีหนึ่ง เป็นการทำ Object Oriented ที่แกนของภาษาเลย โดย Visual Studio.net นั้นถูกออกแบบเพื่อสนับสนุน Object Oriented โดยเฉพาะ คลาสต่างๆที่อยู่ในแต่ละแอปพลิเคชันสามารถติดต่อกันได้โดยตรง (Inherit) ในแอปพลิเคชัน A เราอาจจะเขียนคลาสด้วยภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C++ และแอปพลิเคชัน A อาจจะติดต่อกับแอปพลิเคชัน B ซึ่งเขียนด้วยภาษา Visual Basic (VB) ได้ คือการติดต่อกัน (Inherit) ข้ามภาษานั้นสามารถทำได้ดังแสดงในรูปที่ 2.8



**รูปที่ 2.8** ความสามารถในการติดต่อข้ามแอปพลิเคชันเมื่อเขียนด้วยภาษาที่ต่างกัน ใน Visual Studio.net

ใน Visual Studio.net จะคอมไพล์เป็นภาษาเดียวกันคือ Intermediate Language ตามรูปก่อนหน้านี้ คือคอมไพล์เป็นภาษาอันหนึ่ง เพราะฉะนั้นจึงสามารถ Inherit กันข้ามภาษาได้ นอกจากนี้ยังสามารถทำงานด้วยกันกับ COM แบบเดิมที่เราเคยเขียนมาแล้วได้ด้วย ซึ่งใช้ Visual Studio เดิมได้ดั่งเดิมนี่ก็ไม่จำเป็นต้องโยนทิ้ง ใน Visual Studio.net เราสามารถเรียกใช้งาน COM ได้ตามปกติและในทางกลับกัน ใน Visual Studio 6.0 ที่มีอยู่นี้ก็สามารถเรียกใช้งานคอมโพเนนต์ที่เขียนด้วย Visual Studio.net ได้เช่นกัน คือเป็นการเข้ากันได้ทั้งสองทาง (Backward-Forward Compatibility) นี่คือข้อดีหลายๆของ .NET ทำให้เราไม่ต้องมาพัฒนาโค้ดใหม่

#### 2.6.2.2.1 การจัดการกับหน่วยความจำเมื่อทำการประมวลผล

เนื่องจากการทำงานทั้งหมดของตัว .NET จะมีการดูแลในเรื่องของ Common Language Runtime ปัญหาหนึ่งที่มีมักจะพบ เมื่อเราพัฒนาแอปพลิเคชัน คือเรื่องของ Garbage Collection ในการจัดการกับหน่วยความจำ โดยเฉพาะบางภาษาอย่างเช่น C++ ที่ต้องมีการเรียกใช้งานพอยเตอร์ (Pointer) ค่อนข้างมาก ในการใช้งานพอยเตอร์นั้นถ้าเราเขียนผิดไปนิดเดียวก็อาจจะทำให้แอปพลิเคชันของเราแฮงค์ในระหว่างรันหรือประมวลผลก็ได้ ทำให้เมื่อแอปพลิเคชันของเรารันไปเรื่อยๆหน่วยความจำส่วนนี้ก็จะขยายไปเรื่อยๆ และสักพักหนึ่งก็จะเกิดการหยุดไป

การแก้ไขข้อผิดพลาด (Debug) ตรงนี้ทำได้ยากมาก เพราะกว่าที่เราจะรู้ว่าเกิดความผิดพลาดขึ้นเราก็เสียเวลาในการรันไปแล้ว ซึ่งการจัดการกับหน่วยความจำตรงนี้ ถ้าเราพัฒนาด้วยตัวแพลตฟอร์มของ .NET จะมีเครื่องมือตัวหนึ่งเรียกว่า Garbage Collector เป็นตัวที่คอยจัดการกับหน่วยความจำให้เรา ความจริงตัว Garbage Collector จะเป็นตัวที่ทำการกำหนดค่าหน่วยความจำให้ว่างให้เราเองในส่วนที่เราไม่ใช้งานโดยอัตโนมัติ นี่คือข้อดีอันหนึ่งในการพัฒนาด้วยแพลตฟอร์มของ .NET

### 2.6.2.2 ระบบการตรวจจับความผิดพลาด

นอกจากนี้ในเรื่องของ Exception Manager หรือการดัก Runtime Error ในการสร้างแอปพลิเคชันที่มีการรันได้อย่างดีนั้น การดัก Runtime Error ที่จะเกิดขึ้นในระหว่างทำงาน ก็เป็นปัจจัยหนึ่งที่เราจะต้องคำนึงถึง ซึ่งในตัว .NET จะมี Syntax ในการดัก Runtime Error ที่มีลักษณะเป็นโครงสร้างทำให้หาได้ง่ายขึ้น เราเรียกว่า Structure Exception Handling

### 2.6.2.3 รูปแบบการคอมไพล์แอปพลิเคชัน

นอกจากนี้การคอมไพล์แอปพลิเคชันด้วย .NET มีให้เลือกหลายแบบ บางคนอาจจะคอมไพล์ให้เป็นไค้ดไบนารี (Binary Code) เลยโดยไม่ต้องคอมไพล์ให้เป็น Intermediate Language ก็สามารถจะพัฒนาได้ โดยใช้ Visual C++ .net เรียกว่า Managed C++ หรือถ้าเราเลือกใช้เครื่องมือภาษาอื่น ๆ ที่มีความง่ายในการพัฒนายิ่งกว่าคือ VB หรือ C# ก็สามารถคอมไพล์เป็น Intermediate Language ได้

ในการคอมไพล์จะไม่มีตัว Interpreter Runtime Library ต่างๆ เช่น VB ที่ต้องมี Runtime Library ของตัวเองเวลาพัฒนาด้วย Visual Studio 6.0 อันนี้ก็จะไม่จำเป็นต้องมีอีกต่อไป เราใช้ตัว Runtime อันเดียวกัน ก็คือตัว .net Framework เป็นตัว Runtime Library ที่กล่าวมาข้างต้น

### 2.6.2.4 เรื่องของชนิดตัวแปร

ภาษาทุกตัวที่เขียนในตัว .NET จะเป็นภาษาแบบ Type-Safety คือการเปลี่ยนแปลงระหว่างค่าชนิดต่างๆ เป็น String type ตัว Compiler จะตรวจสอบได้ ถ้าเรามีการติดต่อบetween ตัวแปร 2 ตัวที่มีชนิดต่างกันเช่น เราอาจต้องการเปลี่ยนชนิดของค่าในตัวแปรจาก string เป็น integer เป็นต้น ตัวภาษาใน Visual Studio 6.0 จะให้เราทำได้ แต่ใน .NET จะมีการตรวจสอบว่าถ้ามีการเปลี่ยนแปลงค่าจาก sting เป็น integer หรือการเปลี่ยนชนิดของตัวแปรอื่นๆที่จะทำให้เกิด Runtime Error นั้นตัว Compiler จะตรวจสอบได้และก็จะมีการแจ้งข้อผิดพลาดให้ทราบ

แม้แต่เรื่องของตัวแปรที่เอาไปใช้งาน ถ้าเราประกาศตัวแปรขึ้นมาแล้วเราเอาไปใช้งาน โดยไม่ได้มีการกำหนดค่าให้มันก่อน Compiler ก็จะคอยดัก นั่นคือ Compiler มีความฉลาดมากขึ้น เพื่อดักข้อผิดพลาด ณ จุดที่คอมไพล์เลย ซึ่งดีกว่าให้เกิด Runtime Error ขึ้นทำให้เราแก้ไขข้อผิดพลาดได้ยากหรือแม้แต่เรื่องอาเรย์ก็ตาม ถ้าเราประกาศอาเรย์ไว้ 100 ไอเทมถ้าบรรทัดได้ในไค้ดของเราเกิดมีการอ้างถึงอาเรย์ขึ้นมา Compiler ก็จะคอมไพล์ไม่ผ่านทันที ซึ่งจะกำหนดไว้ว่าตรงนี้เกิด Runtime Error ได้ตอนที่รัน นี่ก็ความฉลาดมากยิ่งขึ้นในตัว Compiler ของ Visual Studio.net ซึ่งจะทำให้เราพัฒนาแอปพลิเคชันได้สะดวก โดยปราศจาก Runtime Error มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเรื่องระบบรักษาความปลอดภัยนั้น ตัว .NET ก็สามารถกำหนดได้ว่าผู้ใช้ที่จะรันแอปพลิเคชันของเรานั้นมีสถานะ (permission) เป็นอะไร อาจจะต้องการสถานะเป็น admin ถึงจะรันแอปพลิเคชันของเราได้

### 2.6.2.2.5 รูปแบบการทำงานร่วมกับภาษาอื่นๆ

การพัฒนาแอปพลิเคชันใน .NET รวมไปถึงเทคนิคต่างๆ ที่กล่าวมาแล้วนั้น ไม่จำเป็นจะใช้เฉพาะภาษาตระกูลใน Visual Studio ไม่ว่าจะเป็น VB.net, C# หรือ C++ ถ้าภาษาใดก็ตามสามารถที่จะคอมไพล์มาให้เป็น Common Language Runtime ได้ ภาษานั้นก็จะสามารถใช้คุณลักษณะทั้งหมดของ .NET ได้ ซึ่งตอนนี้มีภาษาอื่นๆ ที่สามารถทำงานร่วมกับแอปพลิเคชัน .NET มากกว่า 20 ภาษาที่พัฒนาตัว Compiler ขึ้นมาเพื่อคอมไพล์ภาษาต่างๆ ให้อยู่ในรูปแบบของแพลตฟอร์ม .NET ไม่ว่าจะเป็นภาษา COBOL, PASCAL และอื่นๆ

ในตัว Visual Studio.net ก็มีเครื่องมือระดับสูง (High Level Tools) ที่จะทำให้เราพัฒนาแอปพลิเคชันได้สะดวกและง่ายมากขึ้น นี่ก็คือเรื่องของ Common Language Runtime ซึ่งถือได้ว่าเป็นเลเยอร์ล่าสุดของการคอมไพล์แอปพลิเคชันที่เขียนด้วย Visual Studio.net

### 2.6.2.3 เลเยอร์ Base Class Library

เลเยอร์ถัดมาของโครงสร้างการพัฒนาแอปพลิเคชัน .NET ต่อจาก Common Language Runtime ก็คือ Base Class Library เทคนิคต่างๆ ที่ Visual Studio.net จัดเตรียมให้เราในการใช้งานนั้น Base Class Library เปรียบเสมือน เป็นการรวบรวมเอาฟังก์ชันของ API (Application Programming Interface) ทั้งหมด สมมติว่าตอนที่เราพัฒนาแอปพลิเคชันด้วย Visual Studio 6.0 เวลาเราจะเรียกใช้งานฟังก์ชันบางอย่างในระดับสูง หรือต้องการทำงานแบบลึกๆกับระบบ เรามักจะเรียกใช้ API นี้ภาษา Visual Studio 6.0 มีความสามารถในการเรียกฟังก์ชัน API ได้ต่างกัน API บางตัวในปัจจุบันอาจจะต้องใช้ C++ ถึงจะเรียกได้ เพราะว่าต้องใช้โครงสร้างข้อมูล (Data Structure) หรือพอยเตอร์ อะไรบางอย่างที่ไม่มีใน Visual Basic แต่ API บางตัวก็สามารถเรียกจาก Visual Basic ได้

ตัว Base Class Library ก็คือ การที่เรารวบรวมฟังก์ชัน API ซึ่งกระจัดกระจายอยู่ เวลาจะเรียกใช้เราต้องไปค้นหาใน Help นั่นคือ Base Class Library พยายามที่จะรวบรวม API และฟังก์ชันทั้งหมดเกี่ยวกับระบบเข้ามาไว้ในลักษณะของ Object Oriented ทำเป็นคลาสอันหนึ่งซึ่งเป็นมาตรฐานเป็นคลาสที่สร้างมาในตัวระบบเรียบร้อย ซึ่งคลาสทั้งหมดจะอยู่ภายใต้คลาสหลักอันหนึ่งที่เรียกว่า System

ทุกอย่างที่พัฒนาด้วยภาษาใน Visual Studio.net จะเป็น Object Oriented ทั้งหมด โดยมีคลาสที่ใหญ่ที่สุดเรียกว่าคลาส System ซึ่งภายใต้คลาส System จะมีคลาสย่อยๆมากมาย ซึ่งแต่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละคลาจะสนับสนุนฟังก์ชัน API หรือสนับสนุนการทำงานที่เราต้องได้ไม่ว่าจะเป็นเรื่องของ การทำกราฟฟิค การทำเกี่ยวกับโครงสร้างข้อมูล (data structure) การทำเกี่ยวกับเรื่องเครือข่าย (network) ฟังก์ชัน API เหล่านี้จะถูกจัดกลุ่มให้เป็น Object Oriented อยู่ภายใต้ System Class การเรียกใช้งาน System Class ถ้าเป็น VB กับ C# ก็ใช้งานได้ทั้ง 2 อย่าง

เลขอร์ที่ถูกส่งขึ้นมาจาก Base Class Library รวมทั้งเป็นแนวคิดซึ่งไมโครซอฟท์ ผลักดันมากคือเรื่องของ Web Service นั่นเอง

#### 2.6.2.4 เลขอร์ Common Language Specification

เลขอร์สุดท้ายในสถาปัตยกรรม .NET ที่เราจะพูดถึงก็คือ Common Language Specification คือเรื่องของมาตรฐานภาษาบนพื้นฐาน .NET ซึ่ง Compiler จะต้อง ทำงานตาม มาตรฐานดังกล่าว เพื่อให้สามารถทำงานร่วมกับภาษาพื้นฐาน .NET และภาษาอื่นๆได้ ไมโครซอฟท์ได้ทำการปรับภาษาต่างๆ เช่น C#,VB ให้เข้ามาตรฐาน .NET นอกจากนั้น ผู้ผลิต รายอื่นสามารถพัฒนาตามข้อกำหนดนี้เพื่อให้สามารถทำงานบนพื้นฐาน .NET ได้

เลขอร์ที่ 3 ของสถาปัตยกรรม .NET ก็คือเครื่องมือที่ใช้ในการสร้างแอปพลิเคชัน หรือ หลักการที่ใช้ในการเขียน โปรแกรมต่างๆ เช่นเรื่องของ ADO.net, ASP.net ที่ใช้ในการพัฒนา แอปพลิเคชัน แต่สิ่งที่อยู่เหนือกว่าทุกอย่างก็คือภาษาที่เราใช้งาน ภาษาต่างๆที่ทำงานใน .NET นั้นมีข้อดีคือ ต้องสนับสนุนมาตรฐานเดียวกัน เรียกว่า Common Language Specification ซึ่ง ไมโครซอฟท์ได้จดทะเบียนมาตรฐานนี้เข้ากับองค์กร ECMA แล้ว ซึ่งเป็นองค์กรที่ทำงาน เกี่ยวกับการสร้างมาตรฐานกลางของระบบสื่อสารและข้อมูลคอมพิวเตอร์ อีกทั้งยังเป็นแบบเปิด (open) ด้วย เพราะฉะนั้นเจ้าของภาษาอื่นๆ ก็สามารถสร้างตัวแปลภาษา หรือ Compiler เพื่อ คอมไพล์ภาษาของเขาให้เข้ามาเป็น Common Language Specification อันนี้ได้ ในไม่ช้าเราจะ เห็นเว็บเพจเขียนด้วยภาษา COBOL ก็ได้ รวมทั้งภาษาอื่นๆด้วย นอกจากนี้ในตระกูล .NET เอง เราก็มีย Visual Basic, Visual C++ กับ Visual C# ซึ่งภาษาอื่นๆ ที่เรากล่าวถึงนั้นปัจจุบันก็มีกว่า 20 ภาษา เช่น ภาษา MEL, COBOL,PASCAL, ไอเฟล , ไพทอล, Perl, Smalltalk ภาษา ML ซึ่ง เป็นภาษาที่ใช้งานกันในต่างประเทศ ภาษาประเภท Object ทั้งหมดสามารถทำเป็นเฟลด์ฟอร์ม ของ .NET ได้เพราะว่าใน .NET นั้นผลิตทุกอย่างเป็น Object Oriented ทั้งหมด

#### 2.6.3 ภาษาและเครื่องมือของ Microsoft.NET

Microsoft.Net ใช้ภาษา C# เป็นหลักในการพัฒนาแอปพลิเคชัน C# พัฒนามาจากภาษา C และ C++ โดย C# นำแนวคิดเรื่อง Garbage collection และ Hierarchical namespace ของ Java มาใช้ อย่างไรก็ตามการจัดการ Exception ของ C# กับมีข้อยกเว้นให้โปรแกรมเมอร์สามารถสะ เว้น Exception ได้ในบางกรณีที่ต้องการ ต่างกับใน Java ที่การจัดการ Exception เป็นไปอย่าง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการเผยแพร่เท่านั้น เมื่อผู้เช่าเห็นใบเช็คใบนี้เป็นการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องครัด ดังนั้นโอกาสที่จะเกิดข้อผิดพลาดในการเขียนโปรแกรมจึงไม่มีโอกาสเกิดได้เลย นอกจากนี้ใน C# ยังสร้างใช้ตัวแปรพอยน์เตอร์ได้เหมือนในภาษา C โปรแกรมเมอร์ที่เคยใช้พอยน์เตอร์ส่วนใหญ่ทราบดีว่ามีโอกาสพลาดได้ง่าย และผลเสียที่เกิดจากการเขียนโปรแกรมโดยใช้พอยน์เตอร์ไม่รอบคอบนั้นมีมาก การตรวจสอบก็ทำได้ยาก Java ดัดปัญหาเรื่องนี้ไปเลย เพราะไม่อนุญาตให้ใช้พอยน์เตอร์ อย่างไรก็ตามมีคนแย้งว่าการที่ C# เปิดช่องให้จัดการ Exception เองนั้นทำให้เกิดความอิสระในการเขียนโปรแกรม และการใช้พอยน์เตอร์ก็เป็นประโยชน์ในการเขียนโปรแกรม และบางครั้งก็จำเป็นต้องใช้ สำหรับเรื่องภาษานี้ก็ต้องแล้วแต่ความถนัดของแต่ละคน ใครที่ชอบภาษาไหนก็อ้างเหตุผลมาเข้าข้างตัวเองได้ทั้งนั้น เพียงแต่โอกาสพลาดจากการใช้ภาษา C# มีมากกว่าถ้าไม่ระมัดระวัง

Microsoft.Net สนับสนุนการใช้ภาษาอื่นในการพัฒนาแอปพลิเคชัน ผู้ใช้สามารถเขียนโปรแกรมด้วย Visual Basic, COBOL, FORTRAN ได้ เพราะตัวคอมไพเลอร์จะแปลงออกมาเป็นไบท์โค้ด Internal Language ซึ่งโค้ดที่ได้จะทำงานบน CLR ตามที่กล่าวไปแล้ว เพียงแต่ว่าทุกอย่างที่พูดถึงนี้ต้องอยู่บนวินโดวส์เท่านั้น

เครื่องมือช่วยพัฒนาแอปพลิเคชันของ Microsoft.Net ก็คือ Visual Studio.Net ที่ทุกคนต้องยอมรับว่าเป็น IDE ที่มีประสิทธิภาพ เพราะใช้ง่ายและประสานการทำงานได้ดี

## 2.6.4 พอร์ตเทบิลิตี้ของระบบ

พิจารณา Microsoft.Net เนื่องจาก .Net ทำงานบนวินโดวส์ ดังนั้นจึงไม่มีปัญหาเรื่องการสนับสนุนฮาร์ดแวร์ แต่การทำงานบนแพลตฟอร์มอื่น ปัจจุบันยังเป็นไปไม่ได้ แม้ว่าไมโครซอฟท์บอกว่าจะพยายามพัฒนา COM เพื่อนำไปใช้บนระบบปฏิบัติการอื่นก็ตาม

อย่างไรก็ตามสิ่งสำคัญที่ต้องพิจารณาเรื่องพอร์ตเทบิลิตี้ก็คือ แพลตฟอร์ม หากงานที่ทำนั้นมีโอกาสเกี่ยวข้องกับลูกค้าที่ใช้งานแพลตฟอร์มหลายแบบ ก็ไม่ควรที่จะใช้ .Net

## 2.6.5 การเปรียบเทียบระหว่าง Microsoft.Net กับ J2EE

### 2.6.5.1 การสนับสนุนเว็บเซอร์วิส

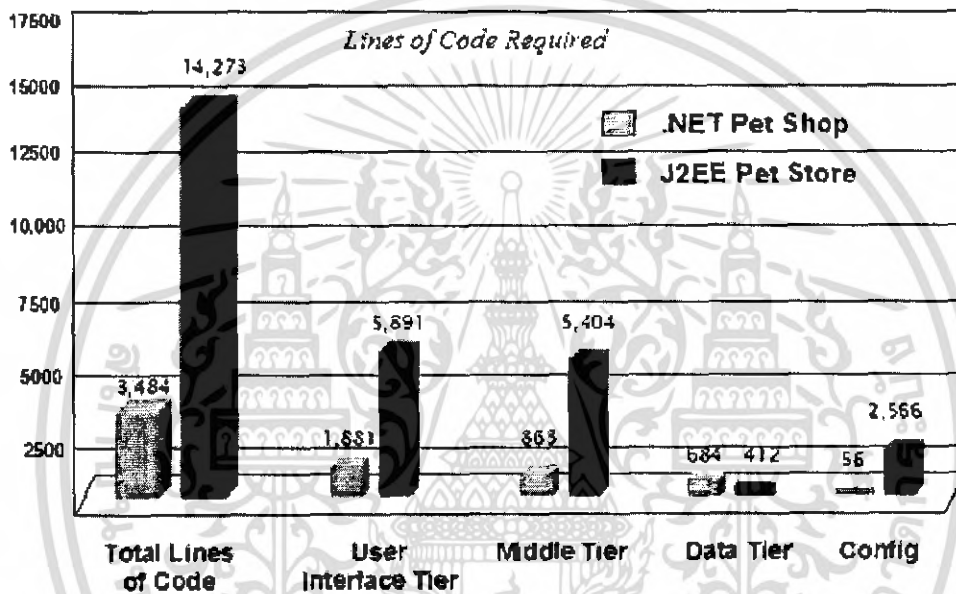
ทั้ง J2EE และ Microsoft.Net ก็มีความสามารถด้านเว็บเซอร์วิสเหมือนกัน จาวามีชุดเอพีไอที่ใช้จัดการ XML ได้ โดยตรง สนับสนุนการใช้งานร่วมกับ SOAP, UDDI และ WSDL แต่งานส่วนใหญ่ยังต้องเขียนขึ้นเอง ซึ่งเป็นงานที่ต้องใช้เวลาพอสมควร ในขณะที่การพัฒนาเว็บเซอร์วิสบน Microsoft.Net นั้นมีเครื่องมือหลายอย่างที่ช่วยในการทำงาน ตลอดเวลาที่ต้องใช้ในการทำงานและทำได้ง่ายมากกว่า J2EE พอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.5.2 ประสิทธิภาพการทำงาน

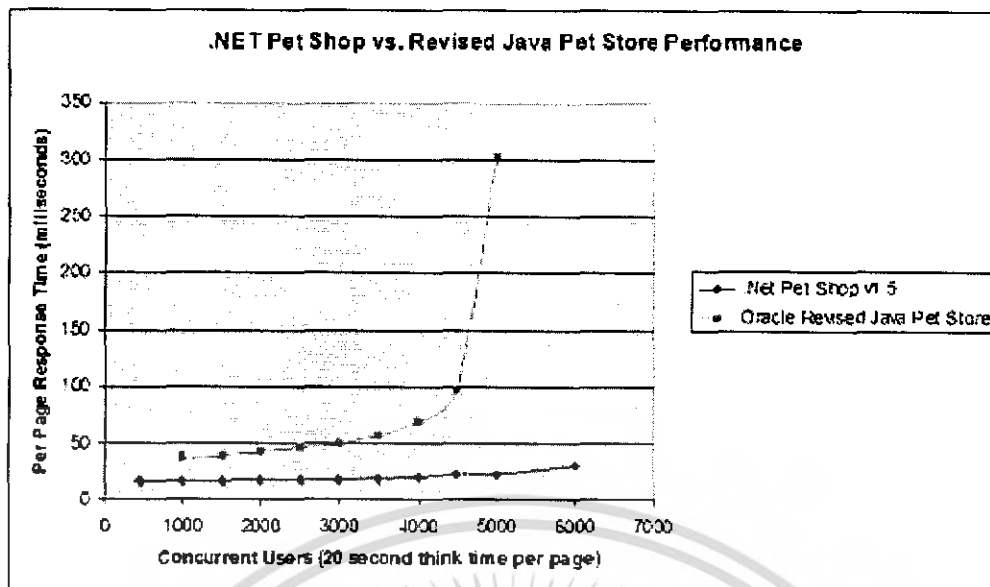
มีข้อมูลแสดงการทดสอบประสิทธิภาพของ Microsoft.Net เปรียบเทียบกับ Sun J2EE ที่แสดงอยู่ที่ <http://www.gotdotnet.com/team/compare> โดยใช้โซลูชันจาก Oracle, IBM และ Sun พัฒนาแอปพลิเคชัน (หากผู้อ่านสนใจขอให้อ่านรายละเอียดได้จากเว็บไซต์ข้างต้น) จากรูปที่ 1 และ 2 แสดงผลการทดสอบที่ปรากฏว่า Microsoft.Net ทำผลงานได้น่าประทับใจ ทั้งทั้ง J2EE ในหลายหัวข้อ ดังที่แสดงในรูปที่ 2.9 และ รูปที่ 2.10

#### Implementing Sun's Java Petstore with Microsoft .NET



รูปที่ 2.9 Implementing Sun's Java Petstore with Microsoft .Net

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



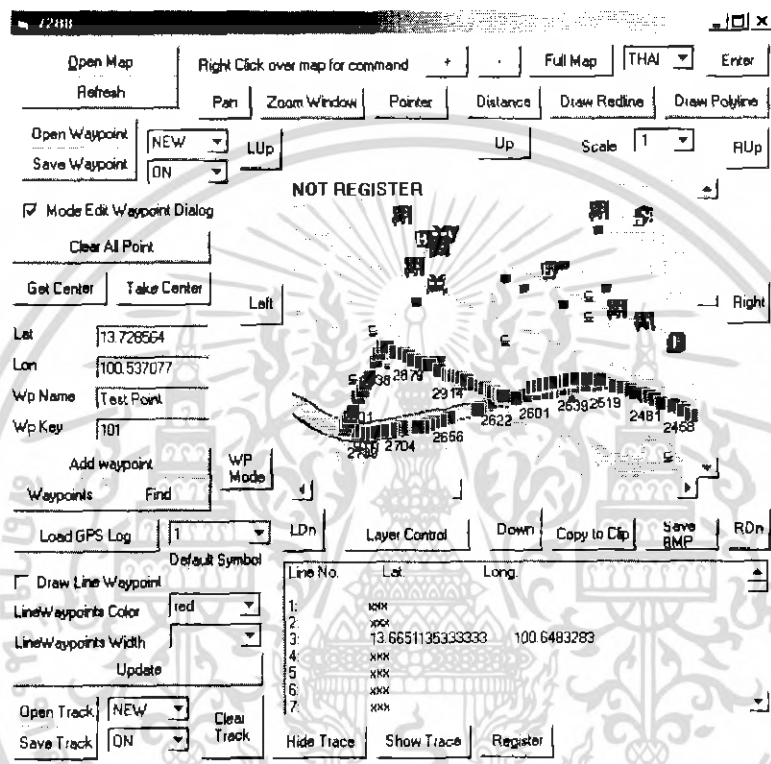
**รูปที่ 2.10** .NET Pet Shop vs. Revised Java Pet Performance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 Thai Map Guide (TmgX Control)

### 2.7.1 Thai Map Guide คืออะไร

Thai Map Guide เป็น ActiveX Control ที่ช่วยในการแสดงแผนที่ของประเทศไทย และยังสามารถในการจัดการกับแผนที่อีกมากมาย โดยเราจะใช้ Visual Basic เพื่อเรียก Control ตัวนี้ขึ้นมาใช้งาน



รูปที่ 2.11 ตัวอย่างโปรแกรมที่มี Thai Map Guide Control อยู่ใน

### 2.7.2 ความสามารถของ Thai Map Guide

1. แสดงภาพแผนที่บริเวณต่างๆ ในประเทศไทยได้
2. ข่อ-ขยายแผนที่ได้ (Zoom-in, Zoom-out)
3. เลื่อนแผนที่ไปมาได้ (Pan)
4. เลื่อนแผนที่ไปยังพิกัดที่ต้องการได้ (อ้างอิงตาม Latitude และ Longitude)
5. ควบคุมการแสดงผลข้อมูลได้ (ข้อมูลใน Thai Map Guide จะถูกแบ่งเป็น Layer อาทิ เช่น ชั้นของถนน, ชั้นของห้างสรรพสินค้า เป็นต้น)

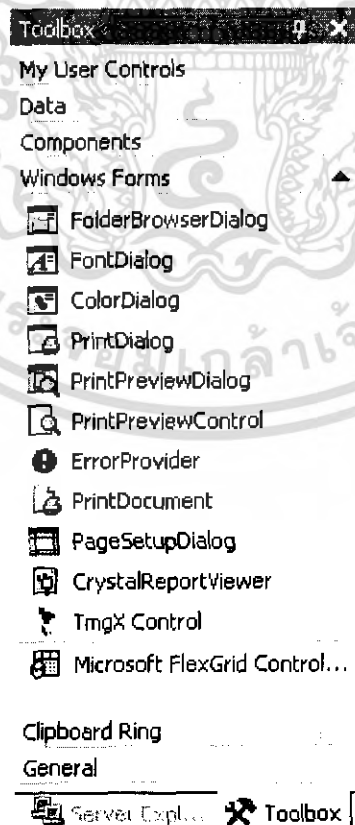
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. สามารถค้นหาตำแหน่งของสถานที่และถนนต่างๆ ในแผนที่ได้ เช่น หาตำแหน่งของร้านอาหารที่ต้องการ
7. สามารถค้นหาตำแหน่งของสถานที่ต่างๆ โดยอ้างอิงจากตำแหน่งปัจจุบันได้ ยกตัวอย่างเช่น ให้หารายชื่อร้านอาหารทั้งหมดในบริเวณรัศมีรอบตำแหน่งปัจจุบันหนึ่งกิโลเมตร เป็นต้น
8. มีเครื่องมือในการวาดเส้นและรูปต่างๆ ไปบนแผนที่ได้ เช่น การวาดเส้นแสดงเส้นทางการบินของรถ, การวาดรูปรถยนต์ลงบนแผนที่ เป็นต้น
9. เพิ่มจุดตำแหน่งเข้าไปได้ เช่น การเพิ่มตำแหน่งบ้านของตนเองเข้าไปในแผนที่ เป็นต้น
10. สามารถคำนวณระยะทางระหว่างจุดพิกัด 2 จุดได้

### 2.7.3. การนำ TmgX Control มาใช้ในโปรแกรม

ทำการ Install โปรแกรม TmgX Control ลงไปในเครื่อง หลังจากนั้น ให้เปิด Visual Basic.NET ขึ้นมา

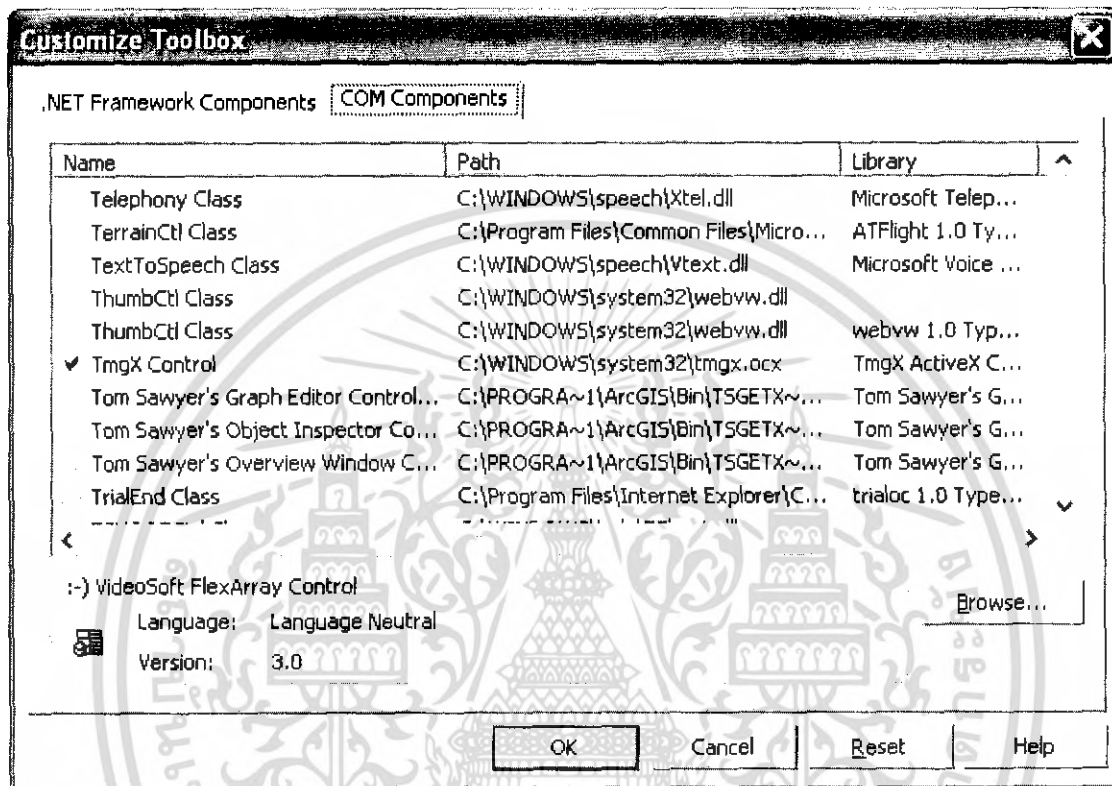
ทำการคลิกขวาบริเวณของ Toolbox ดังภาพ



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ บริษัท เทคโนโลยี จำกัด โดยภาพที่ 2.12 Toolbox ของ Microsoft Visual Basic .NET นี้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือก Add/Remove Items... เลือก Tab COM Components แล้วทำการ Browse หาไฟล์ชื่อ tmgx.ocx ซึ่งจะอยู่ใน Folder System32 ของ Folder Windows ของเรา

หลังจากเลือกจะได้ภาพดังภาพ



รูปที่ 2.13 ภาพแสดงการเพิ่ม TmgX Control เข้าไปในโปรแกรม

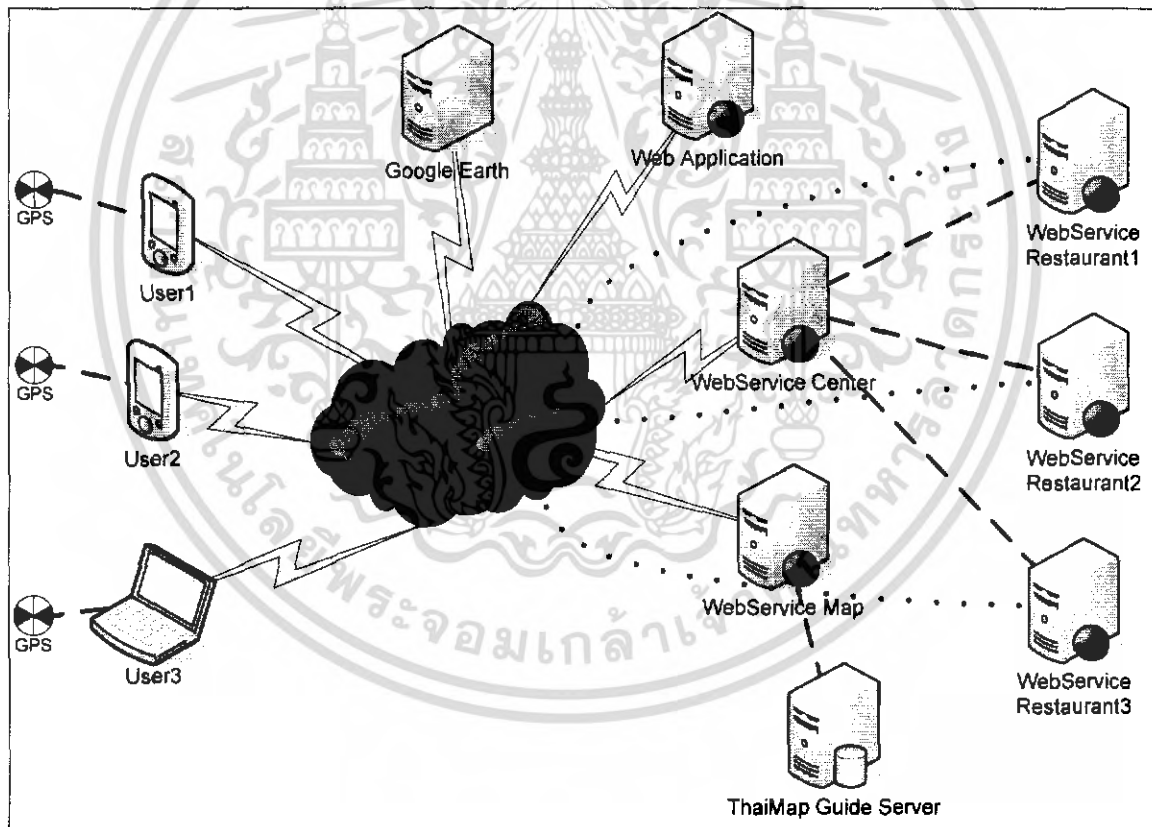
ในตอนนี้อาจจะสามารถทำการใช้ TmgX Control ได้เหมือนการเรียกใช้ Components อื่นๆ (เช่น Button) โดยการเลือก TmgX Control แล้วลากมาวางไว้ใน Form ของเรา

## บทที่ 3

### การออกแบบและการพัฒนา

#### 3.1 ภาพรวมของระบบ

1. ส่วนแรกเป็นส่วนของเว็บเซอร์วิสกลางที่ให้บริการข้อมูลร้านอาหาร และ ร้านอาหารต้นแบบที่ให้บริการเว็บเซอร์วิส
2. ส่วนที่สองเป็นส่วนของการให้บริการแผนที่ประเทศไทย และแผนที่ภายในอาคารผ่านเว็บเซอร์วิส
3. ส่วนที่สามเป็นส่วนของFront-End ที่ทำหน้าที่ติดต่อไปยังส่วนที่หนึ่งและส่วนที่สองเพื่อดึงข้อมูลออกไปแสดงผล



รูปที่ 3.1 แสดงโครงสร้างของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับส่วนของโครงการนี้จะมุ่งจัดการในเรื่องที่เกี่ยวกับแผนที่ ประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

1. BuildingEditor เป็นโปรแกรมที่ใช้ในการตั้งค่าต่างๆ ให้กับแผนที่ภายในอาคาร สถานที่ อาทิเช่น ชื่อของร้านค้า, รายละเอียดของร้านค้า เป็นต้น
2. KMLLoader เป็นโปรแกรมที่ใช้สำหรับอ่าน KML Format ซึ่งเป็น format ที่ใช้สำหรับการเก็บ Placemark (ชื่อและข้อมูลของสถานที่ต่างๆ เช่น ถนน, อาคาร) ของโปรแกรม GoogleEarth ซึ่งโปรแกรมนี้จะทำการอ่านข้อมูลต่างๆ ของไฟล์นี้แล้วทำการเซฟลง Web Service เพื่อให้โปรแกรมอื่นในโครงการสามารถเรียกใช้ได้
3. ThaiMapGuide Server เป็นโปรแกรมที่ทำหน้าที่เป็น Server ที่จะรองรับการเชื่อมต่อจาก Web Service แล้วทำการส่งภาพแผนที่ตำแหน่งต่างๆ ไปให้

ส่วนประกอบต่างๆ สามารถอธิบายได้โดยละเอียด ดังนี้

## 3.2 BuildingEditor

### 3.2.1 วัตถุประสงค์

การที่จะให้ผู้ใช้สามารถดูรายละเอียดภายในอาคารได้อย่างง่ายดายนั้น ถือว่าเป็นการเพิ่มเติมอีกมิติหนึ่งเข้าไปในโครงการ

โดยทั่วไปแล้ว อาคารสถานที่ที่มีขนาดใหญ่ เช่น ห้างสรรพสินค้า, โรงแรม, โรงเรียน สถานที่เหล่านี้มักจะมีสิ่งที่เรียกว่า Directory ซึ่งจะเปรียบดั่งเป็นแผนที่ภายในอาคารนั้นๆ ทำให้เราสามารถดูได้ว่า ภายในอาคารนี้มีสถานที่ใดบ้างและสถานที่เหล่านั้นตั้งอยู่ ณ จุดแห่งใดภายในอาคาร

แต่ข้อจำกัดของสิ่งนี้คือ Directory จะตั้งอยู่กับที่ เราไม่สามารถจะเรียกดูได้จากตำแหน่งในตอนที่เราอยู่บริเวณใดในอาคาร, ไม่สามารถเรียกดูได้ว่าสถานที่ที่เราต้องการจะไปอยู่ที่ใด นอกเสียจากเราจะเดินกลับไปยังสถานที่ตั้ง Directory นั้นอีกครั้ง

ข้อเสียอีกอย่างหนึ่งของวิธีนี้คือ บน Directory มักจะแสดงเป็นรหัสของห้องต่างๆ หลังจากดูรหัสแล้ว เราก็ต้องมาดูว่า รหัสนั้นๆ ตรงกับสถานที่ใดอีกที ซึ่งทำให้เสียเวลาอย่างมาก

BuildingEditor ถูกสร้างขึ้นบนความคิดที่อยู่บนหลักของคำว่า “สะดวก” และ “รวดเร็ว” โดยทั่วไปอาคารใหญ่ๆ นั้นจะมีภาพแผนที่ของตนเองอยู่แล้ว โปรแกรมนี้จะช่วยให้สามารถใช้ภาพแผนที่ที่มีอยู่ให้เกิดประโยชน์สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มจากผู้ใช้งานทำการโหลดภาพแผนที่ของอาคารนั้นๆ เข้ามาในโปรแกรม จากนั้นก็ให้ทำการกำหนดขอบเขตบริเวณต่างๆ เพื่อจะบอกว่าบริเวณนั้นๆ คืออะไร

อาทิเช่น การกำหนดกรอบสี่เหลี่ยมให้กับบริเวณของร้านขายอาหาร BuildingEditor ไม่ได้เพียงแค่สามารถกำหนดได้ว่าบริเวณกรอบสี่เหลี่ยมนั้นคือร้านอาหารอะไร แต่ยังสามารถสร้างช่องทางการเชื่อมต่อไปยังข้อมูลร้านอาหารของโครงการที่ทำเรื่องร้านอาหารได้อีกด้วย นั่นหมายถึง ผู้ใช้นอกจากจะทราบได้ว่า บริเวณนั้นๆ คืออะไรแล้ว ยังสามารถเลือกเข้าไปดูข้อมูลร้านอาหารและสามารถทำการจองที่ได้ในทันที

หลังจากผู้ใช้โปรแกรมทำการแก้ไขกำหนดขอบเขตบริเวณต่างๆ เรียบร้อยแล้ว ผู้ใช้ก็สามารถทำการบันทึกแล้วทำการอัพโหลดขึ้นไปยังเซิร์ฟเวอร์ส่วนกลางซึ่งมีหน้าที่เก็บข้อมูลต่างๆ ซึ่งสิ่งนี้เป็นจุดเริ่มต้นของความ “สะดวก”

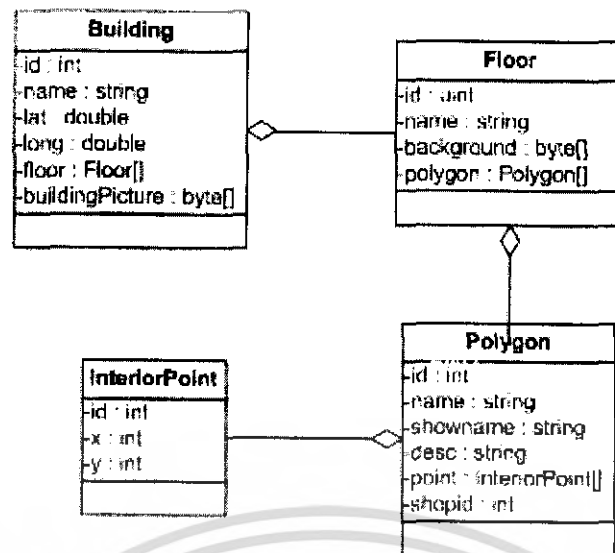
เมื่อข้อมูลถูกส่งไปยังเซิร์ฟเวอร์ส่วนกลางแล้ว ผู้ใช้ที่มี Pocket PC ก็สามารถเรียกดูได้ตลอดเวลา ซึ่งไม่ใช่เพียงแค่เรียกดูอย่างเดียว ผู้ใช้ยังสามารถค้นหาสถานที่ต่างๆ ได้อย่างรวดเร็วอีกด้วย

### 3.2.2 การออกแบบพัฒนา

BuildingEditor ได้รับการพัฒนาโดยใช้ภาษา C# บน Microsoft Visual C#.NET 2003 ซึ่งการพัฒนาโปรแกรมนี้ประกอบด้วย

#### 3.2.2.1 โครงสร้างของ Building

สำหรับโครงสร้างของ Building นั้น จะประกอบด้วยส่วนต่างๆ ที่อธิบายคุณสมบัติของ Building ซึ่งสามารถอธิบายเป็น Class Diagram ได้ดังนี้



**รูปที่ 3.2** แสดงโครงสร้างของ Class ที่เกี่ยวข้องกับ Building

Building – เป็น Class ที่ใช้เก็บข้อมูลต่างๆ ของ Building นี้

Id – เป็น ID ของ Building นี้

Name – เก็บชื่อของ Building นี้

Lat – เก็บตำแหน่ง Latitude ของ Building นี้

Lon – เก็บตำแหน่ง Longitude ของ Building นี้

Floor – เก็บค่า Array ของ Class Floor ซึ่งจะเก็บข้อมูลของแต่ละชั้น

buildingPicture – เก็บค่า Array ของ Byte ซึ่งเก็บข้อมูลรูปภาพภายนอกของอาคาร

Floor – เป็น Class ที่ใช้เก็บข้อมูลของชั้นแต่ละชั้น

Id – เป็น ID ของชั้นนี้

Name – เป็นชื่อของชั้นนี้ เช่น 1F, GF เป็นต้น

Background – ใช้เก็บภาพฉากหลังของชั้นนี้ไว้ประกอบการแสดงผล

Polygon – เป็น Array ของ Class Polygon สำหรับเก็บข้อมูลส่วนต่างๆ ของชั้น

Polygon – เป็น Class ที่ใช้เก็บข้อมูลส่วนของบริเวณในชั้น อาทิเช่น ร้านอาหาร

Id – เก็บ ID ของบริเวณนี้

Name – เก็บชื่อของบริเวณนี้

Showname – เก็บชื่อที่ต้องการจะให้เห็นแสดงของบริเวณนี้

Desc – เก็บคำอธิบายบริเวณนี้ เช่น ร้านอาหารจีน, ร้านเสื้อผ้า เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Point – เป็น Array ของ InteriorPoint ที่ใช้เก็บค่าจุดต่างๆ ที่ประกอบกันเป็น Polygon นี้

Shopid – ใช้สำหรับเก็บค่า id ของร้านค้าที่จุดนี้อ้างอิงถึง (ในโครงการนี้จะใช้เก็บ id ของร้านอาหาร เพื่อให้สามารถอ้างไปหาร้านอาหารร้านนั้นๆ ได้)

InteriorPoint – เป็น Class ที่ใช้เก็บค่า x และ y สำหรับจุดใดๆ ที่ใช้ประกอบกันเป็น 1 Polygon โดยใน Polygon หนึ่งอันนั้นจะประกอบด้วย InteriorPoint หลายตัวซึ่งนำมาเชื่อมต่อกันเป็น 1 Polygon นั้นเอง

Id – ใช้เก็บค่า id ของจุดๆ นี้

X – ใช้เก็บค่าตำแหน่งในแกน x ของจุดๆ นี้

Y – ใช้เก็บค่าตำแหน่งในแกน y ของจุดๆ นี้

### 3.2.2.2 การสร้างแผนที่ภายในอาคาร

สำหรับการพัฒนาโปรแกรม BuildingEditor ในการสร้างแผนที่นั้น หลังจากที่เรารู้จัก ออกแบบส่วนที่ใช้เก็บข้อมูลแล้ว เราก็สามารถสร้างส่วนที่ใช้สร้างแผนที่ได้โดยเริ่มจากในตอนแรก เราต้องให้ผู้ใช้ใส่ค่าข้อมูลโดยทั่วไปของตึก แล้วเก็บไว้ใน Object ของ Building

จากนั้นก็ให้ผู้ใช้สามารถสร้าง Floor ต่างๆ ได้ โดยจะเก็บเป็นรูปแบบของ ArrayList ซึ่งเป็น Class ที่ใช้เก็บ Array แบบ Dynamic ของ .NET Framework ซึ่งจะช่วยในการจัดการ Manage และ Allocate memory ให้ เราจะเก็บค่าของ Floor ชั้นต่างๆ ไว้ในนี้

ยกตัวอย่างเช่น หากเราต้องการเพิ่ม Floor เข้าไปใน Building ก็สมารถทำได้โดย

```
Floor newFloor = new Floor();
```

```
floorArray.Add(newFloor);
```

หากเราต้องการทราบว่าในขณะนี้ เรามีชั้นอยู่กี่ชั้น ก็สามารถทำได้โดยการดูค่าจาก floorArray.Count

เมื่อได้ Floor แล้ว ผู้ใช้ก็ต้องเลือกจากหลังให้กับ Floor ถึงตอนนี้ผู้ใช้ก็สามารถสร้าง บริเวณต่างๆ ภายใน Floor ได้แล้ว โดยการลากเมาส์ เมื่อลากเสร็จแล้ว ก็จะได้ Polygon ใหม่ขึ้นมา ให้ผู้ใช้กำหนดข้อมูลของตำแหน่ง บริเวณนั้นๆ ซึ่งในแต่ละจุดของการลากเมาส์ ก็คือ InteriorPoint ซึ่งจะมากลายเป็นส่วนประกอบของจุดของ Polygon นั้นๆ นั้นเอง

การทำงานของโปรแกรม BuildingEditor จะได้รับการอธิบายอีกครั้งหนึ่งในส่วนของการทดลอง

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.3 การบันทึกข้อมูลที่สร้างไว้ของคอมพิวเตอร์

ผู้ใช้สามารถบันทึกข้อมูลที่สร้างไว้ของคอมพิวเตอร์เพื่อนำมาแก้ไขต่อในภายหลังได้ โดยการบันทึกนี้จะบันทึกเป็น Format แบบ XML โดยสำหรับข้อดีของการใช้ XML คือ ทำให้เราสามารถแก้ไขข้อมูลได้ง่าย (เพราะเป็นตัวอักษรภาษาอังกฤษที่อ่านรู้เรื่อง)

วิธีการบันทึกนั้นจะทำการเก็บข้อมูลทั้งหมด โดยเริ่มจากข้อมูลชื่อสถานที่, ภาพภายนอกสถานที่ทั้งหมด, ข้อมูลของแต่ละชั้นทั้งหมด โดยเขียนแบบวนลูปไล่ไปทุกชั้น สามารถแสดงเป็นโค้ดได้ดังนี้

```
// Write output file (in xml format)
string fname = saveFileDialog1.FileName;

StreamWriter sw = new StreamWriter(fname);

// Start writing data
sw.WriteLine("<?xml version=\"1.0\" encoding=\"TIS-620\"?>");
sw.WriteLine("<Building>");

foreach(MyFloor fl in Floors)
{
    // Write floor information
    sw.WriteLine("    <Floor>");
    sw.WriteLine("        <ID>" + fl.id.ToString() + "</ID>");
    sw.WriteLine("        <Name>" + fl.name + "</Name>");
    sw.WriteLine("        <Background>" + fl.bgImage +
        "</Background>");

    foreach(MyPolygon poly in fl.Polygons)
    {
        // Polygon information
        sw.WriteLine("            <Polygon>");
        sw.WriteLine("                <ID>" + poly.index +
            "</ID>");
        sw.WriteLine("                <Name>" + poly.name +
            "</Name>");
        sw.WriteLine("                <SHOWNAME>" +
            poly.showname + "</SHOWNAME>");
        sw.WriteLine("                <DESC>" + poly.desc +
            "</DESC>");
        sw.WriteLine("                <SHOPID>" +
            poly.shopid.ToString() + "</SHOPID>");

        foreach(MarkControl mc in poly.Marks)
        {
            // Marker information
            sw.WriteLine("                    <Marker>");
            sw.WriteLine("                        <X>" +
                mc.Location.X + "</X>");
            sw.WriteLine("                        <Y>" +
                mc.Location.Y + "</Y>");
        }
    }
}
sw.WriteLine("</Building>");
sw.Close();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้เอาต์เห็นใบ้ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        sw.WriteLine("                </Marker>");
    }
    sw.WriteLine("                </Polygon>");
}

sw.WriteLine("        </Floor>");
}

// Outside building's picture
if(listBox1.Items.Count == 0)
{
    sw.WriteLine("        <BPicture>0</BPicture>");
    sw.WriteLine("        <BPicture>0</BPicture>");
}
else
{
    for( int i =0; i<listBox1.Items.Count;i++)
        sw.WriteLine(" <BPicture>" +
            listBox1.Items[i].ToString() + "</BPicture>");
    if(listBox1.Items.Count == 1)
        sw.WriteLine("        <BPicture>0</BPicture>");
}

sw.WriteLine("</Building>");
sw.Flush();
sw.Close();

```

ส่วนเวลาเปิดข้อมูลขึ้นมาอ่านนั้น สามารถทำได้โดยการใช้ .NET ช่วย โดยมีวิธีการเขียน  
ดังนี้

```

// Open file use readxml for easy implement
// instantiate new dataset
DataSet relational_data;
relational_data = new DataSet();

// read the xml file (Thai Encoding) to populate the dataset
StreamReader sr = new StreamReader(fileName, Encoding.UTF8);
relational_data.ReadXml(sr);

```

ซึ่งหลังจากโค้ดด้านบนได้ถูกทำงานแล้ว เราก็จะได้ข้อมูลทั้งหมดของแผนที่ภายในมาอยู่ในตัวแปร relational\_data ซึ่งจะแบ่งแยกการเก็บเป็น table (.NET จะทำการ generate Table รวมทั้งความสัมพันธ์ต่างๆ ขึ้นมาให้เราโดยอัตโนมัติ ทำให้เราสามารถเขียนโปรแกรมได้ง่ายดายขึ้นอย่างมาก

เราสามารถดึงข้อมูลจาก relational\_data ด้วยวิธีนี้

```
relational_data.Tables[0].Rows[0].ItemArray[0];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรทัดข้างบนอ้างอิงถึงข้อมูลของ Table ที่ 0, Row แรก และ Field แรก เมื่อรู้วิธีดึงข้อมูลแต่ละ Field แล้วเราก็สามารถทำการวนลูปเก็บค่าต่างๆ ลงตัวแปรของโปรแกรมเราและนำไปใช้ได้

### 3.2.2.4 การอัปเดตข้อมูลที่สร้างไว้ไปยังเซิร์ฟเวอร์

ส่วนนี้นั้นเป็นส่วนที่ใช้สำหรับการอัปเดตข้อมูลที่ได้สร้างไว้ ขึ้นไปยัง Web Service ส่วนกลาง โดยจะทำการอัปเดตเข้าไปใน Database ตาม Field ต่างๆ โดย สามารถเขียน ER Diagram ได้ดังนี้

สำหรับกรรมวิธีในการเขียนโปรแกรมนั้น จะทำการวนลูปอัปเดตข้อมูลทั้งหมดขึ้นไป โดยสามารถอธิบายเป็น โค้ด ได้ลักษณะคล้ายๆ กับการบันทึกลงไฟล์ XML ยกเว้นเพียงแต่จะมีการใช้ส่วนของ Web Service เข้ามาช่วยด้วย ซึ่งจะทำการอธิบายในหัวข้อต่อไป

### 3.2.2.5 Web Service ที่ให้บริการข้อมูลที่สร้างจาก BuildingEditor

Web Service ที่เกี่ยวข้องกับ BuildingEditor นั้นประกอบด้วยสองส่วนคือ

#### 3.2.2.5.1 Web Service ที่ให้บริการในการสร้างแผนที่ภายในอาคาร

สำหรับ Web Service ที่ให้บริการในการสร้างแผนที่ภายในอาคารนั้น จะนำมาใช้ในโปรแกรม BuildingEditor ตรงส่วนของการบันทึกข้อมูลไปยัง เซิร์ฟเวอร์กลาง โดยในโปรแกรม BuildingEditor จะมีการสร้าง Object ของ Class Building ซึ่งเป็น Class ของ Web Service ที่ไว้ใช้สำหรับเก็บข้อมูล Building จากนั้นก็จะทำการนำข้อมูลของอาคารที่ได้สร้างไว้มาเก็บไว้ในตัวแปรนี้ และสุดท้ายก็เรียกใช้คำสั่ง `uploadBuilding(Building bd);` ดัง โค้ดข้างล่างนี้

```
// Insert to web service
localhost.Service1 webref = new localhost.Service1();
// Upload data
webref.uploadBuilding(bd);
```

ซึ่งคำสั่ง `uploadBuilding` นั้นจะทำการดึงข้อมูลจาก Class Building แล้วนำไป Insert ลง Database เพื่อให้ส่วนของ Front-End สามารถดึงข้อมูลออกมาใช้ได้ โดยมีโค้ดดังต่อไปนี้

```
[WebMethod]
public int uploadBuilding(Building bd)
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    NpgsqlConnection aConnect;
    try
    {
        aConnect = new
NpgsqlConnection("Server=127.0.0.1;Port=5432;User
Id=postgres;Password=mclonly;Database=Interior;");
        aConnect.Open();

        // Insert building to database
        int building_id = 0;
        string sqlQuery = "insert into Building(name) values
('"+
            bd.name + "')";
        NpgsqlCommand command = new
NpgsqlCommand(sqlQuery, aConnect);

        NpgsqlDataReader dr = command.ExecuteReader();

        // Get id from of this building
        sqlQuery = "select * from Building where name =
'" + bd.name + "'";
        command = new NpgsqlCommand(sqlQuery, aConnect);
        dr = command.ExecuteReader();
        while(dr.Read())
        {
            building_id = (int) dr["id"];
        }
        foreach(Floor fl in bd.floor)
        {
            // Insert floor information
            sqlQuery = "insert into Floor values(" +
                fl.id + ", '" +
                fl.name + "', :bytesData, " +
                building_id + ")";
            command = new NpgsqlCommand(sqlQuery, aConnect);

            NpgsqlParameter param = new
NpgsqlParameter(":bytesData", DbType.Binary);
            byte[] tt = fl.background;

            param.Value = tt;
            command.Parameters.Add(param);
            command.ExecuteNonQuery();

            foreach(Polygon poly in fl.polygon)
            {
                // Polygon information
                sqlQuery = "insert into Polygon values (" +
                    poly.id + ", '" +
                    poly.name + "', '" +
                    poly.showname + "', '" +
                    poly.desc + "', " +
                    fl.id + ", " +
                    building_id + ")";
                command = new
NpgsqlCommand(sqlQuery, aConnect);
                dr = command.ExecuteReader();
                foreach(InteriorPoint p in poly.point)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            // Marker information
            sqlQuery = "insert into Point(x, y,
polygon_id, building_id) values (" +
                p.x + ", " +
                p.y + ", " +
                poly.id + ", " +
                building_id + ")";
            command = new
NpgsqlCommand(sqlQuery, aConnect);
            dr = command.ExecuteReader();
        }
    }
}

return building_id;
}
catch(Exception e)
{
    Console.Out.WriteLine(e.ToString());
    return 0;
}
}

```

### 3.2.2.5.2 Web Service ที่ให้บริการในการดึงแผนที่ภายในอาคารไปใช้งาน

การดึงข้อมูลภายในอาคารออกไปใช้นั้น ไม่ว่าจะเป็ Pocket PC หรือ Web ก็สามารดึงได้โดยผ่านคำสั่ง Web Service โดยคำสั่งเหล่านี้ประกอบด้วย

```
Building getBuilding(int building_id);
```

เป็นคำสั่งที่ใช้ดึงข้อมูลทั้งหมดภายในอาคารใดๆ ที่มี building\_id ตรงกับค่า Parameter ที่ส่งเข้าไป โดยค่าที่ส่งกลับมานั้นจะเป็น Class Building ที่สามารถนำไปใช้ได้ทันที

```
Building2[] getallBuilding(double latitude, double
longitude, int distance, string token);
```

เป็นคำสั่งที่ใช้สำหรับดึงอาคารใดๆ ก็ตามที่อยู่ห่างจากตำแหน่ง Latitude และ Longitude ที่ส่งเข้าไปเป็นระยะทาง distance โดยค่าที่ส่งกลับไปในั้นจะเป็น Array ของ Building ทั้งหมดที่ตรงอยู่ในขอบเขตของ Parameter ที่ส่งเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 KMLLoader

#### 3.3.1 วัตถุประสงค์

ในปัจจุบันนี้หากพูดถึงโปรแกรมทางด้านแผนที่ที่มีคนรู้จักมากที่สุดย่อมไม่มีใครเกิน GoogleEarth อย่างแน่นอน

GoogleEarth เป็นโปรแกรมให้บริการแผนที่ที่ใช้ภาพถ่ายจากดาวเทียม เราสามารถดูภาพบริเวณต่างๆ ในโลกนี้ได้ ไม่ว่าจะเป็น New York, Paris, กำแพงเมืองจีน หรือแม้กระทั่งอนุสาวรีย์ชัยสมรภูมิของประเทศไทยเรา

ความสามารถที่โดดเด่นอีกอย่างของ GoogleEarth คือระบบ Placemark ที่ให้ผู้ใช้สามารถเพิ่มสถานที่ต่างๆ เข้าไปในโปรแกรมได้ เช่น เพิ่มจุดที่เป็นของเรา หลังจากนั้นยังสามารถทำการบันทึกแล้วส่งให้ผู้อื่นดูได้อีกด้วย

ในประเทศไทยนั้นระบบนี้ถือว่าได้รับความนิยมอย่างมาก มีผู้คนมากมายสร้าง Placemark ของประเทศไทยขึ้นมาเพื่อแจกจ่ายให้ผู้อื่นนำไปใช้ได้ ไม่ว่าจะเป็น Placemark ของถนน, ของโรงพยาบาล, โรงเรียน และอื่นๆ อีกมากมาย ซึ่ง Format ที่ใช้ในการเก็บ Placemark นี้ก็คือ Format ที่มีชื่อว่า KML นั่นเอง

ในแผนที่ของเรานั้น ไม่ว่าจะเป็นจาก Thaimapguide หรือ Google Map API ก็พอมี Placemark ต่างๆ ให้อยู่บ้าง แต่เนื่องจากว่า KML เป็น Format ที่ได้รับความนิยมมีผู้สร้าง Placemark ต่างๆ ออกมาแจกจ่ายมากมาย หากเราจำกัดเพียงแค่ Placemark ที่เรามีอยู่ก็จะไม่เพียงพอต่อสถานที่ใหม่ๆ ที่ถูกสร้างขึ้นมาตลอดเวลา

ดังนั้น KMLLoader จึงเป็นโปรแกรมที่ถูกสร้างขึ้นมาเพื่อให้ระบบแผนที่ของโครงการนี้สามารถดึงข้อมูลของไฟล์ KML ที่สร้างจาก GoogleEarth มาอ่านแล้วทำการอัปเดตไปยังเซิร์ฟเวอร์กลาง ทำให้ผู้ใช้สามารถเพิ่มเติม Placemark ใหม่ๆ ได้ตลอดเวลา ทำให้ข้อจำกัดในเรื่องของแผนที่ที่ไม่ค่อยอัปเดตถูกทำลายไป

#### 3.3.2 การออกแบบและพัฒนา

โปรแกรมนี้ได้รับการพัฒนาขึ้นโดยโปรแกรม Microsoft Visual C#.NET 2003 โดยมีรายละเอียดการพัฒนาดังนี้

### 3.3.2.1 KML Format

KML นั้น เป็น Format ที่ใช้ใน Google Earth Client ซึ่งใช้สำหรับเก็บข้อมูลพวก Placemark (สถานที่), Ground overlays (บริเวณพื้นที่ที่สร้างเป็น Overlay ไว้เหนือแผนที่), Paths และ Polygon ซึ่งข้อมูลเหล่านี้สามารถสร้างขึ้นมาใช้ Google Earth Client ได้

สำหรับส่วนที่โครงการนี้สนใจคือส่วนของ Placemark เท่านั้น ดังนั้นจึงจะอธิบายเฉพาะ ส่วนของ Placemark

Placemark นั้นใช้สำหรับเก็บข้อมูลของตำแหน่งสถานที่ต่างๆ ตัวอย่าง Code ของ Placemark สามารถแสดงได้ดังนี้

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Placemark>
<description>Tethered to the ground by a customizable tail</description>
<name>Tethethed placemark</name>
<LookAt>
<longitude>-122.0856375356631</longitude>
<latitude>37.42240551227282</latitude>
<range>305.8880792294568</range>
<tilt>46.72425699662645</tilt>
<heading>49.06133439171233</heading>
</LookAt>
<visibility>0</visibility>
<Style>
<IconStyle>
<Icon>
<href>root://icons/palette-3.png</href>
<x>96</x>
<y>160</y>
<w>32</w>
<h>32</h>
</Icon>
</IconStyle>
</Style>
</Placemark>
</kml>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

</Style>
<Point>
  <extrude>1</extrude>
  <altitudeMode>relativeToGround</altitudeMode>
  <coordinates>-122.0856204541786,37.42244015321688,50</coordinates>
</Point>
</Placemark>
</kml>

```

โครงสร้างของ โค้ดข้างบนนี้ประกอบด้วย

XML Header

KML namespace declaration

รายละเอียดของ Placemark

ชื่อของ Placemark

Camera View ของ Placemark นี้

ค่า default ของการแสดงผลของ Placemark นี้ (ตัวอย่างนี้ ค่า 0 หมายถึง ผู้ใช้ต้องเปิดเองถึง  
จะเห็น)

Style ของ Placemark นี้ เก็บรายละเอียดของภาพ Icon และตำแหน่ง

Point เป็นส่วนที่กำหนดค่าตำแหน่งที่ปรากฏอยู่บน Surface ของ โลก

สำหรับข้อมูลที่เราสนใจ และจะดึงมาใช้เพื่ออัปเดต Database ของเรานั้น ก็คือ  
Name, Description แล้วยังมี Point

### 3.3.2.2 การดึงข้อมูลจาก KML File

หลังจากเราดาวน์โหลดไฟล์ KML มา (สมมติว่าโหลดมาจาก Web

<http://www.thaigoogleeearth.com>) แล้วเราต้องการเอาข้อมูลของสถานที่ทั้งหมดในไฟล์นั้นๆ ลง

ไปใน Database เราก็จะต้องทำการเขียน โปรแกรมเพื่อดึงเฉพาะข้อมูลที่ต้องการออกมา

โดยข้อมูลเฉพาะในที่นี้นั้นหมายถึง Placemark อันประกอบด้วยชื่อของสถานที่และ  
Latitude, Longitude ของสถานที่นั้นๆ

ในตอนแรกเราต้องทำการโหลดไฟล์ที่ต้องการจะอ่านเข้ามา โดยเขียน โค้ดดังข้างล่างนี้

```

string fname = openFileDialog1.FileName;
sr = new StreamReader(fname, System.Text.Encoding.UTF8);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตอนนี เราจะได้ตัวแปร sr ไว้เก็บค่าของข้อมูลในไฟล์ จากนั้นเราก็จะทำการ Search าคำว่า <Placemark> ซึ่งจะเป็นจุดเริ่มต้นของส่วนของข้อมูลที่เราต้องการ สุดท้ายก็ทำการดึง ข้อมูลใน tag <name>, <latitude> และ <longitude> ตามลำดับ โดยสามารถเขียนโค้ดได้ดังนี้

```
try
{
    string fname = openFileDialog1.FileName;

    sr = new StreamReader(fname, System.Text.Encoding.UTF8);
    listBox1.Items.Add("Read KML File...");

    string name, lat, lon;

    while (sr.Peek() >= 0)
    {
        string tempStr = sr.ReadLine();

        // Find placemark
        if(tempStr.IndexOf("<Placemark>") != -1)
        {
            // Get name
            int start, end;

            // Go to next line
            tempStr = sr.ReadLine();
            start = tempStr.IndexOf("<name>") + 6;
            end = tempStr.IndexOf("</name>");
            name = tempStr.Substring(start, end-start);

            // Find coordinate
            while(sr.ReadLine().IndexOf("<Point>") == -1) {};

            // Get longitude
            tempStr = sr.ReadLine();
            start = tempStr.IndexOf("tes>") + 4;
            end = tempStr.IndexOf(",");
            lon = tempStr.Substring(start, end-start);

            // Get latitude
            start = end+1;
            end = tempStr.LastIndexOf(",");
            lat = tempStr.Substring(start, end-start);

            // Input to database

            //MessageBox.Show(name + " lon = " + lon + " lat = " + lat);

            string sql;
            sql = "insert into place
(nameplace,latijude,longtijude,type) VALUES ('" +
                name + "','" + lat + "','" + lon + "','" +
                "1')";

            //    MessageBox.Show(sql,"sql :");

            command = new NpgsqlCommand(sql, conn);
            Int32 rowsaffected;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนที่มหาวิทยาลัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        try
        {
            rowsaffected = command.ExecuteNonQuery();
        }
        catch(Exception ee)
        {
            string ex = ee.ToString();
        }

    }
}
sr.Close();

listBox1.Items.Add("Insert into database complete");
}
catch (Exception ex)
{
    // catch any xmlexceptions
    listBox1.Items.Add(ex.Message);
}
}

```

ซึ่งโค้ดข้างบนนั้นจะมีขั้นตอนการทำงานคือ

1. หาคำว่า <Placemark> ในไฟล์
2. ดึงข้อมูล <name>, <latitude>, <longitude> ออกมา
3. ทำการอัปเดตข้อมูลเข้าไปยัง Database
4. ทำข้อ 1 ซ้ำไปเรื่อยจนจบไฟล์

### 3.3.2.3 Web Service ที่ใช้ร่วมกับ KMLLoader

Web Service ส่วนนี้นั้นได้ถูกพัฒนาขึ้นมาเพื่อให้ส่วนที่เป็น Front-End สามารถเรียกไปใช้งานได้ โดย Function เหล่านี้ประกอบด้วย

```

Placemark[] getallPlacemark (double latitude,double
longitude,int distance,string token);

```

ใช้สำหรับดึงข้อมูล Placemark ที่อยู่รอบบริเวณ Latitude และ Longitude ที่กำหนดให้เป็นระยะทาง distance โดยค่าที่ส่งกลับไปในนั้นจะเป็น Array ของ Placemark ที่อยู่ภายในขอบเขตที่กำหนดไว้

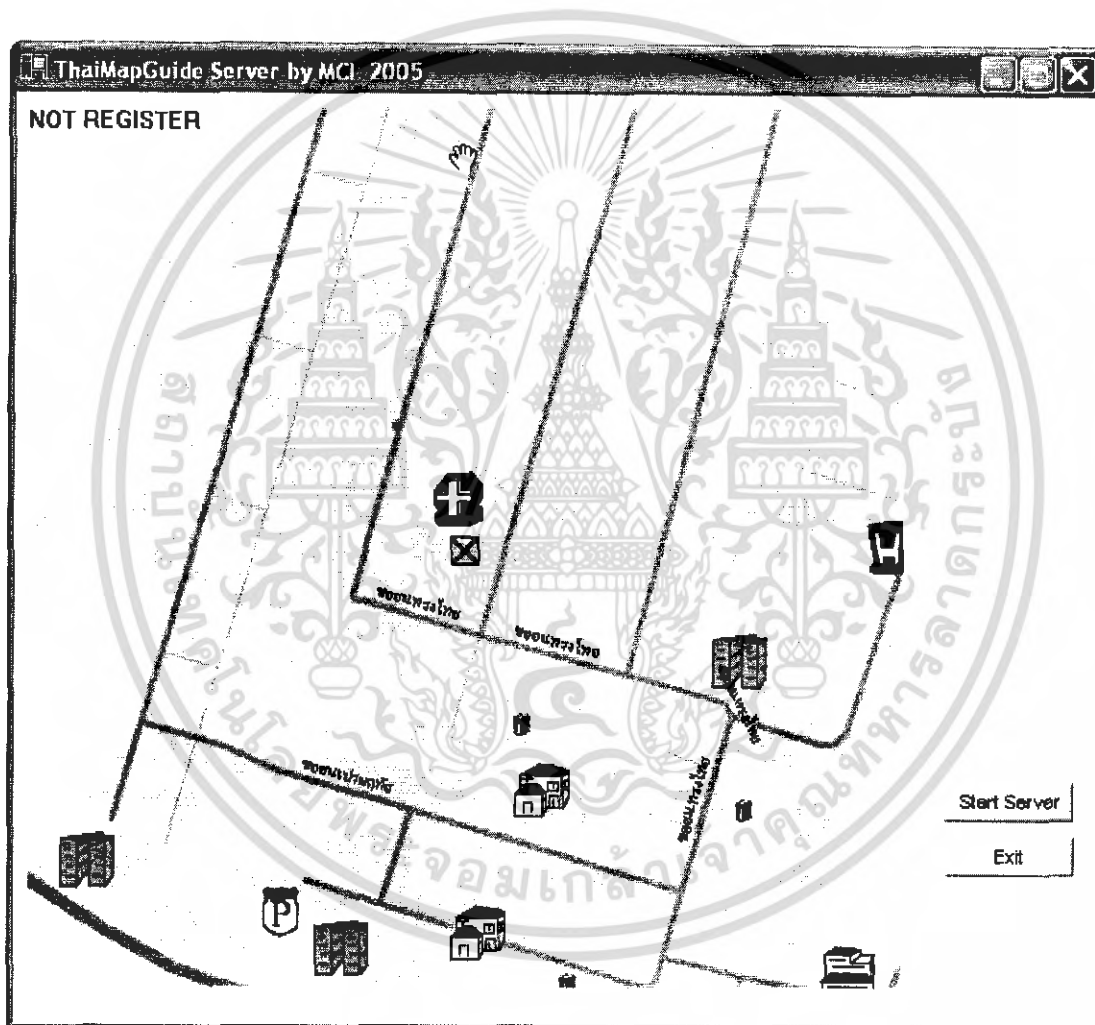
ข้อมูลของ Placemark นี้จะอยู่ในคาต้าเบสส่วนกลาง ซึ่งเมื่อเรียกใช้คำสั่ง Web Service เหล่านี้แล้ว ก็สามารถนำข้อมูล ไปใช้ได้ทั้งใน Web และ Pocket PC เพื่อใช้แสดงผลและค้นหาตามต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 ThaiMapGuide Server

#### 3.4.1 วัตถุประสงค์

แม้ว่าเราจะมีบริการแผนที่ของ Google MAP API ให้อยู่แล้ว แต่แผนที่ของ Google นั้น เป็นแบบที่เรียกว่า Satellite คือแผนที่จากดาวเทียมนั่นเอง โครงการนี้จึงได้มีการเพิ่มทางเลือกให้กับผู้ใช้ นั่นคือ นอกจากจะมีแผนที่ดาวเทียมแล้ว ยังมีแผนที่ในรูปแบบที่เป็นแผนที่ปกติให้เลือกดูได้ด้วย



รูปที่ 3.3 แสดงภาพหน้าจอขณะใช้งาน โปรแกรม ThaiMapGuide Server

ด้วยสิ่งนี้จะทำให้ผู้ใช้สามารถใช้แผนที่ได้ถึงสองแบบ ซึ่งจะสร้างความสะดวกสบายให้  
อย่างมาก

แต่เนื่องจากว่า ThaiMapGuide นั้น เป็น OCX Control ไม่ได้เป็นบริการเหมือน Google  
MAP API จึงเป็นเรื่องลำบากที่จะส่งข้อมูลแผนที่ให้กับ Pocket PC หรือเว็บที่เรียกใช้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเหตุนี้ ThaiMapGuide Server จึงได้รับการพัฒนาขึ้น เพื่อที่จะเป็น Server ที่คอยรอรับการติดต่อเพื่อขอข้อมูลแผนที่จากเซิร์ฟเวอร์กลาง โดยมีขั้นตอนการทำงานคือ เมื่อ Pocket PC หรือเว็บเรียกดูแผนที่ของ ThaiMapGuide ก็จะทำการส่งตำแหน่ง Latitude, Longitude และระดับการซูมมาผ่านทาง Web Service หลังจากนั้นเซิร์ฟเวอร์ส่วนกลางก็จะทำการส่ง TCP Message มาให้กับ ThaiMapGuide Server เมื่อ ThaiMapGuide Server ได้รับแล้วก็จะทำการอัปเดตแผนที่ที่ในตำแหน่งที่ต้องการเข้าไปยัง Database เพื่อให้ Pocket PC หรือเว็บสามารถเรียกดูได้

### 3.4.2 การออกแบบและพัฒนา

ThaiMapGuide Server ได้รับการพัฒนาโดยใช้โปรแกรม Microsoft Visual Basic.NET 2003 ด้วยเหตุผลที่ว่า ThaiMapGuide มี Document และ Code sample เป็นภาษา Basic ทั้งหมด ดังนั้นจึงเป็นเรื่องที่ง่ายและรวดเร็วกว่าถ้าจะใช้ภาษา Basic ด้วยเช่นเดียวกัน

ThaiMapGuide Server มีรายละเอียดการพัฒนาดังนี้

#### 3.4.2.1 TmgX Control

ในส่วนของการใช้ TmgX Control นี้ได้มีการอธิบายไว้บางส่วนแล้วในบทที่ 2 สำหรับในส่วนนี้นั้นจะเป็นการอธิบายถึงการใช้ TmgX Control เพื่อทำหน้าที่เป็น ThaiMapGuide Server การใช้งานนั้นเราจะสร้าง TmgX Control ขึ้นมา จากนั้นก็ทำการ Start TCP ให้เป็นแบบ Listener เพื่อรอ Message ที่จะส่งเข้ามา

จากนั้นเมื่อมี Message เข้ามาเราก็จะทำการถอด Message นั้นๆ เพื่อจะดึงเอาค่า Latitude, Longitude และ ID ของ User ที่ Request เข้ามาออกมาใช้

หลังจากเราได้ Latitude, Longitude แล้ว เราจะเอาสองค่านี้มาใช้เพื่อหาบริเวณของแผนที่ที่ถูกต้อง โดยใช้คำสั่งนี้

```
'-----Check Lat-Lon in Map?
If (Bett(lati, longi, 13.801674, 100.32588200000001,
13.556674, 100.924632)) Then
    Prov = "Bangkok Full"
    mapName.Add(Prov)
End If
```

คำสั่ง Bett เป็น Function ที่เขียนขึ้นมาเพื่อเช็คค่า จุด Latitude, Longitude ที่ส่งเข้าไปอยู่ระหว่างจุด Latitude, Longitude สองจุดใดๆ หรือไม่ โดยถ้าใช้จะส่งค่ากลับมาเป็น True แต่ถ้าไม่ใช่ก็จะส่งค่า False กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตัวอย่างโค้ดที่ยกมานี้ เป็นการชี้ว่า lati, longi อยู่ระหว่าง Latitude, Longitude ที่กำหนดหรือไม่ ซึ่งถ้าใช่ ในที่นี้ บริเวณที่ผู้ใช้ก็คือจังหวัดกรุงเทพ เราก็จะส่งให้แสดงแผนที่กรุงเทพ

สำหรับคำสั่ง `mapName.Add(Prov)` นั้น เป็นการเพิ่มรายชื่อของแผนที่เข้าไปใน `ArrayList` ซึ่งเหตุผลที่ต้องทำเช่นนี้เพราะว่า บาง Latitude, Longitude นั้นจะครอบคลุมอยู่ระหว่างสองจังหวัดหรือมากกว่านั้น แต่แผนที่ของ `ThaiMapGuide` นั้น จะแสดงเป็นจังหวัดๆ ดังนั้นในบางจุดที่ครอบคลุมอยู่ระหว่างหลายๆ จังหวัด ผู้ใช้ก็จะสามารถเลือกได้ว่า จะให้แสดงแผนที่จังหวัดใด

หลังจากได้แผนที่แล้วเราก็จะสั่งให้ `TmgX Control` เปิดแผนที่ด้วยคำสั่ง

```
TmgX1.OpenMap (mapName (0) )
```

คำสั่งนี้จะเปิดแผนที่ที่ต้องการขึ้นมา หลังจากนั้นเราก็ต้องสั่งให้แผนที่ทำการเปลี่ยนตำแหน่งให้ตำแหน่งกลางของหน้าจอเป็น Latitude, Longitude ที่ส่งมา แล้วทำการเซฟภาพลงไป ในเครื่อง (เพราะเราไม่สามารถดึงภาพออกมาใช้โดยตรงได้)

จากนั้นก็ทำการเปิดภาพขึ้นมาใหม่แล้วแปลงภาพที่เป็น `Bitmap` ให้เป็น `Byte Array` สุดท้ายก็ทำการ Upload ขึ้นไปยัง Database โดยใช้คำสั่ง `UpdateMap` ซึ่งเป็นคำสั่ง `Web Service` ซึ่งขั้นตอนเหล่านี้สามารถแสดงด้วยโค้ดข้างล่าง

```
TmgX1.SetCenter(latitude, longitude)
' Save current map
TmgX1.SaveMapToBMP("d:\testdata\test.bmp")
' Convert bitmap to byte array
imgdata = ReadBitmap2ByteArray("d:\testdata\test" & noFile &
".bmp")
' Upload byte array to database
If webref.UpdateMap(IDUser, imgdata) = 0 Then
    MsgBox("Error uploading data to server")
End If
```

คำสั่ง `ReadBitmap2ByteArray` นั้นเป็นคำสั่งที่จะแปลงภาพ `Bitmap` เป็น `Byte Array` เนื่องจากว่า ใน Database นั้นจะเก็บภาพเป็นแบบ `Byte Array` ดังนั้น จึงต้องแปลงก่อนแล้วจึงจะส่งขึ้นไปได้

### 3.4.2.2 การเชื่อมต่อระหว่าง ThaiMapGuide Server กับ Web Service

ในส่วนนี้เป็นการสร้างช่องทางการเชื่อมต่อระหว่าง Server กลาง กลับ Server ที่ทำการประมวลผล ThaiMapGuide Server

เมื่อ Server กลางได้รับ Request จากผู้ใช้ ก็จะทำการส่ง TCP Message มาให้ ThaiMapGuide Server เพื่อให้ ThaiMapGuide Server ทำการสร้างข้อมูลที่ต้องการแล้ว Upload ส่งคืน ไปให้ที่ Database ของ Server กลาง

โดยในส่วนของ ThaiMapGuide Server นั้น จะเริ่มจากการสร้าง TCP ขึ้นมาก่อน แล้วสั่งให้ทำการ Listener ไว้

```
Const portNumber As Integer = 11344
Dim tcpListener As New TcpListener(portNumber)
Dim clientdata As String
tcpListener.Start()
```

โค้ดข้างบนนี้เป็นการสั่งให้เปิด TCP Protocol ที่ Port 11344 โดยให้เป็นแบบ Listener หลังจากสั่ง Start ก็จะเป็นการเริ่มต้นทำงาน

```
'Accept the pending client connection and return a
TcpClient initialized for communication.
Dim tcpClient As TcpClient =
    tcpListener.AcceptTcpClient()
Console.WriteLine("Connection accepted.")
' Get the stream
Dim networkStream As NetworkStream =
    tcpClient.GetStream()
' Read the stream into a byte array
Dim bytes(tcpClient.ReceiveBufferSize) As Byte
networkStream.Read(bytes, 0,
    CInt(tcpClient.ReceiveBufferSize))
' Return the data received from the client to the
console.
clientdata = Encoding.ASCII.GetString(bytes)
```

โค้ดข้างบนนี้เป็นส่วนของการสั่งให้โปรแกรมทำการรอการเชื่อมต่อ (คำสั่ง AcceptTcpClient ) โดยในส่วนของโค้ดเหล่านี้จะแยกออกมาเป็นอีก Thread หนึ่งของโปรแกรม เพื่อให้โปรแกรมหลักก็สามารถทำงานได้ และส่วนของการรอคอยการเชื่อมต่อก็ยังสามารถทำงานได้ไปพร้อมๆ กันด้วย

หลังจากนั้นเราก็จะทำการดึงข้อมูลที่ Client ส่งเข้ามา โดยข้อมูลที่ดึงออกมานั้นตอนแรกจะเป็น Byte Array แต่เราจะนำมาแปลงเป็น String เพื่อนำมาวิเคราะห์ข้อมูล เราสามารถแปลง Byte Array ที่ได้รับให้เป็น String โดยใช้คำสั่ง Encoding.ASCII.GetString

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายเราก็จะได้ String ของข้อมูลที่ส่งมาจาก Client โดยในที่นี้ก็คือตัวแปรชื่อ clientData

สำหรับขั้นตอนการถอด clientData เพื่อดึงข้อมูล Latitude, Longitude และ ID User นั้นสามารถทำได้โดยวิธีดัง โค้ดข้างล่างนี้

### 3.4.2.3 Web Service ที่ใช้ร่วมกับ ThaiMapGuide Server

Web Service ในส่วนนี้ได้รับการพัฒนาขึ้นมาเพื่อให้ Pocket PC และ Web ในส่วนของ Front-End สามารถแสดงภาพแผนที่ได้ โดยในส่วนนี้ประกอบด้วยคำสั่งดังต่อไปนี้

```
public TmgData GetMapTmg(string iduser, double lat, double lon);
public int UpdateMap(string iduser,byte[] map);
```

คำสั่ง GetTmgMap เป็นคำสั่งที่ให้ส่วนของ Front-End เรียกใช้เพื่อเรียกขอภาพแผนที่ตาม Latitude, Longitude ที่ต้องการ เมื่อคำสั่งนี้ถูกเรียกใช้แล้ว Web Service ที่เป็นเจ้าของ Function ก็จะมีการติดต่อไปยัง ThaiMapGuide Server ผ่านทาง TCP/IP

เมื่อ ThaiMapGuide Server ได้รับ Message TCP/IP แล้วก็ทำการถอด Message ที่ได้รับออกมาเป็น User ID ที่ request เข้ามา, Latitude และ Longitude ที่ User ID ต้องการ จากนั้น ThaiMapGuide Server ก็จะมีการดึงภาพนั้นๆ ออกมาแล้วใช้คำสั่ง UpdateMap() ซึ่งเป็นคำสั่ง Web Service ที่ใช้สำหรับส่งภาพที่ได้ทำการ Convert เป็น Byte array แล้วขึ้นไปยัง Database ของ Server กลาง

หลังจากทำการอัปเดตภาพขึ้นไปแล้ว โปรแกรม ThaiMapGuide Server ก็จะมีการส่ง TCP Message ตอบกลับไปยัง Web Service ที่เรียกเข้ามาเพื่อบอกว่าได้ทำการอัปเดตข้อมูลที่ต้องการเข้าไปแล้ว รวมทั้งบอกกลับไปด้วยว่าภาพที่ออกมามีขนาดเท่าไร (เป็นผลต่างของระยะระหว่างขอบสองด้าน)

เมื่อฝั่ง Web Service ได้รับ TCP Message นี้แล้ว ก็จะมีการไปดึงภาพนั้นออกมาจาก Database แล้วทำการส่งกลับไปที่กับทาง Front-End ที่เรียกเข้ามา (Pocket PC หรือ Web) สุดท้ายเมื่อฝั่ง Front-End ได้รับภาพ ก็สามารถนำไปใช้ประกอบการประมวลผลได้

## บทที่ 4

### การทดลองและผลการทดลอง

ในส่วนของการทดลองนี้เราจะแบ่งออกเป็นส่วนต่างๆ ดังนี้

#### 4.1 Building Editor

Building Editor เป็น โปรแกรมที่ถูกพัฒนาขึ้น เพื่อช่วยให้ผู้ใช้สามารถจัดการกับข้อมูลทั้งภายในและภายนอกอาคารขนาดใหญ่(ศูนย์การค้า)ที่มีร้านค้าย่อยภายในได้ง่ายขึ้น ภาพรวมของโปรแกรมจะมีการสร้างลักษณะเฉพาะของข้อมูลขึ้นมา โดยแบ่งโครงสร้างออกเป็นส่วนย่อยตามลำดับได้ดังนี้

**4.1.1. Building** หรืออาคารที่เราสนใจ ภายในอาคารจะประกอบไปด้วยชั้นย่อยหลายชั้นซึ่งผู้ใช้สามารถเพิ่มลดได้ตามต้องการ

**4.1.2. Floor** หรือ ชั้น เป็นส่วนย่อยที่อยู่ภายในอาคาร แต่ละชั้นจะประกอบไปด้วย Polygon เล็กๆจำนวนมาก

**4.1.3. Polygon** ประกอบไปด้วยจุด หลากๆจุด ใช้เป็นสัญลักษณ์ใช้แทนร้านค้าแต่ละร้านที่อยู่ในชั้น โดยเราสามารถกำหนดข้อมูลและรายละเอียดต่างๆให้กับร้านค้านั้นๆได้ ข้อมูลเหล่านี้จะถูกส่งต่อไปให้กับทาง Web service

**4.1.4 Marker** หรือจุด คือ หน่วยย่อยสุดของ Building Editor เป็นส่วนประกอบที่ใช้ในการสร้าง Polygon

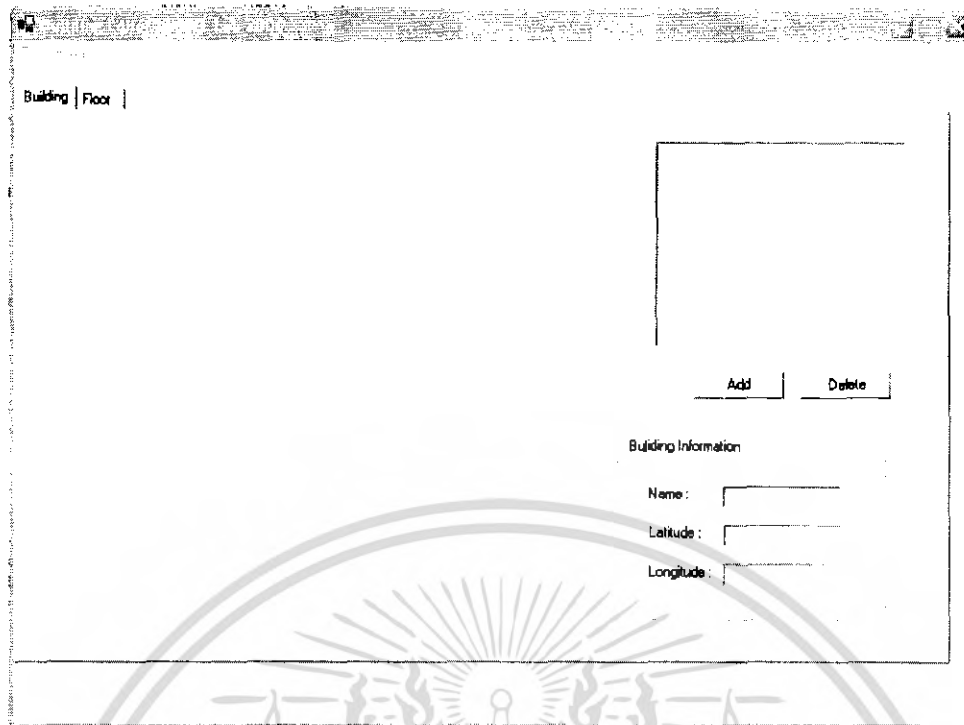
#### 4.2 การทดลองใช้งานโปรแกรม Building Editor

สำหรับโปรแกรมแก้ไขข้อมูลภายในอาคาร(Building Editor)นี้ เราได้ทำการทดลองโดยการสร้างข้อมูลของอาคาร Siam Paragon โดยอ้างอิงจากสถานที่จริง ขั้นตอนการสร้างเริ่มจากการ Scan ภาพ Background ของชั้นต่างๆ เข้ามา จากนั้น ก็ทำการโหลดภาพเหล่านี้เข้ามาในโปรแกรมของเรา แล้วเริ่มทำการวาดบริเวณต่างๆ ลงไป รวมทั้งกำหนดชื่อและรายละเอียดของสถานที่นั้นๆ ด้วย ในส่วนของการใช้งานนั้นสามารถแบ่งออกได้เป็น 2 ส่วนหลักๆคือ

##### 4.2.1.วิธีการใช้งาน Tab Building

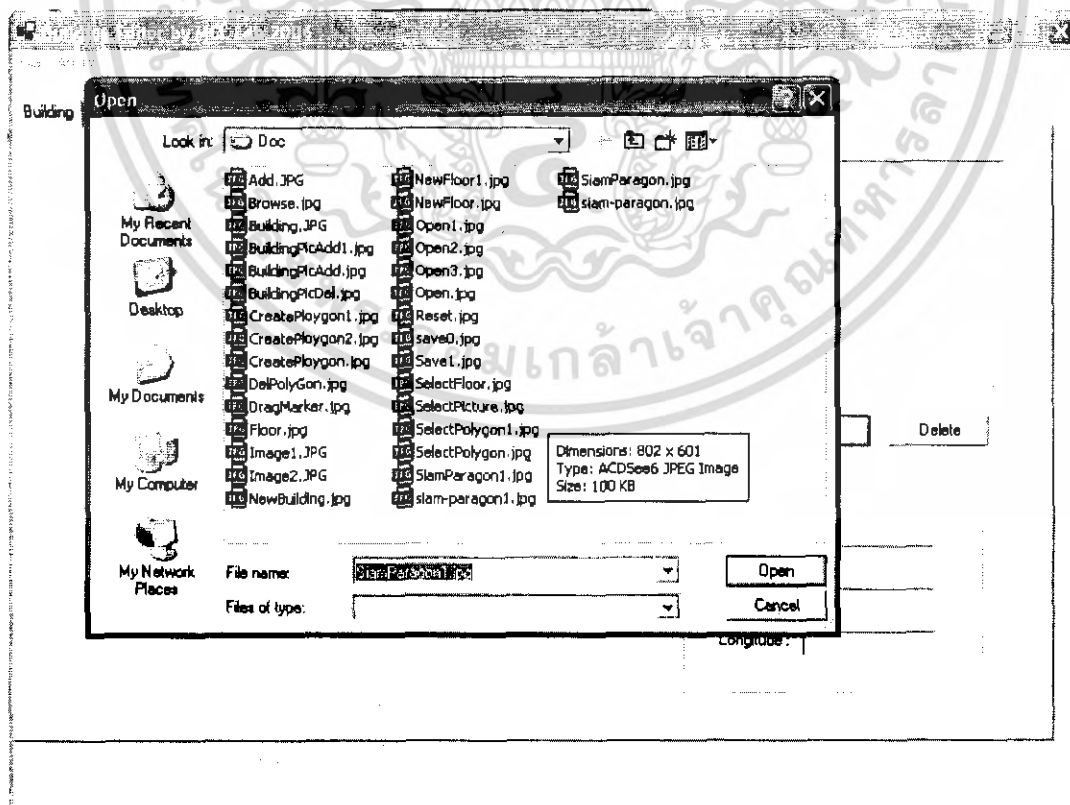
Tab building เป็นส่วนที่ใช้ในการเพิ่มรูปและข้อมูลของตัวตึก เช่น ชื่อและตำแหน่ง(การจัดการภายนอกอาคาร) ดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



#### รูปที่ 4.1 แสดงลักษณะและส่วนประกอบของ Tab Building

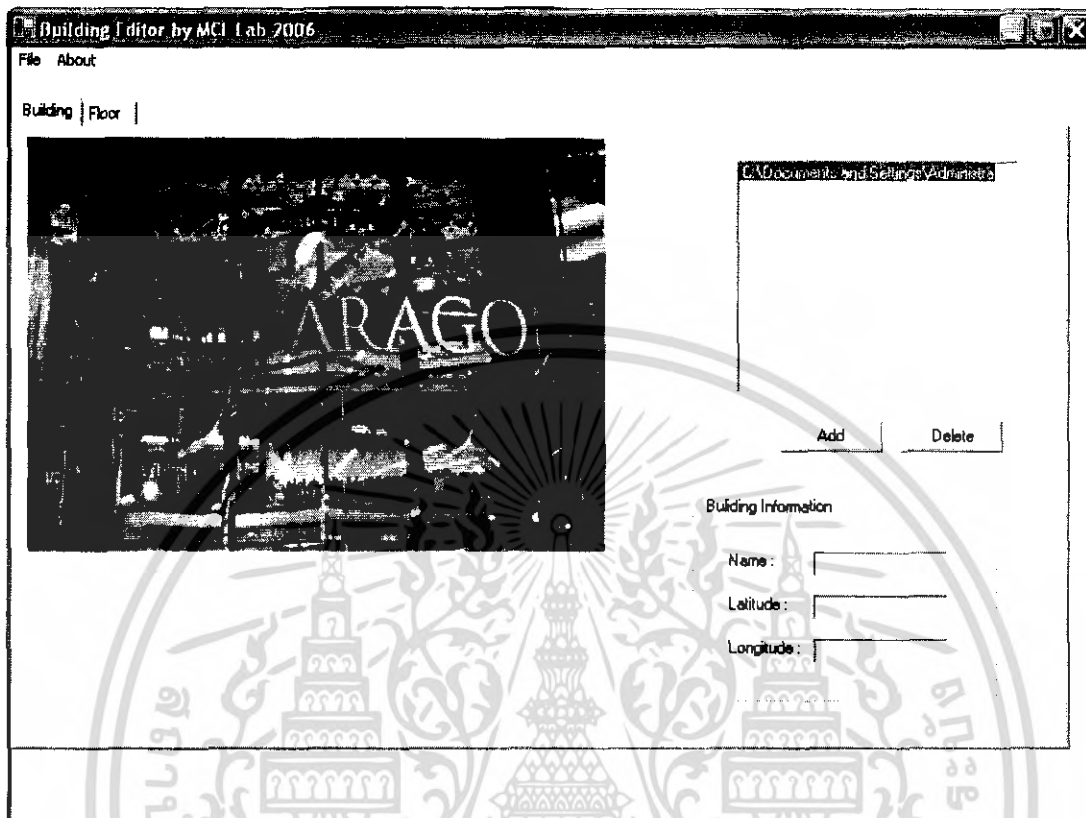
1. วิธีเพิ่มรูปให้กับอาคาร Click ที่ปุ่ม Add จะมี Dialog Box ปรากฏขึ้นมาดังรูปที่ 4.2



#### รูปที่ 4.2 แสดง Dialog Box ที่ใช้ในการเลือกเพื่อเพิ่มรูปให้กับอาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกภาพที่ต้องการ (ตามรูปที่ 4.2 คือ SiamParagon1.jpg) จากนั้นทำการ Click ที่ปุ่ม Open ภาพที่เลือกจะถูกนำมาแสดง ดังรูปที่ 4.3



**รูปที่ 4.3** แสดงภาพที่ถูกนำมาแสดงภายหลังจากถูกเลือก

2.แก้ไขข้อมูลเกี่ยวกับตึกที่ส่วน Building Information ด้านขวามือ เช่น ชื่อตึก(Name) ค่าละติจูด(Latitude) ค่าลองจิจูด(Longitude) ดังรูปที่ 4.4

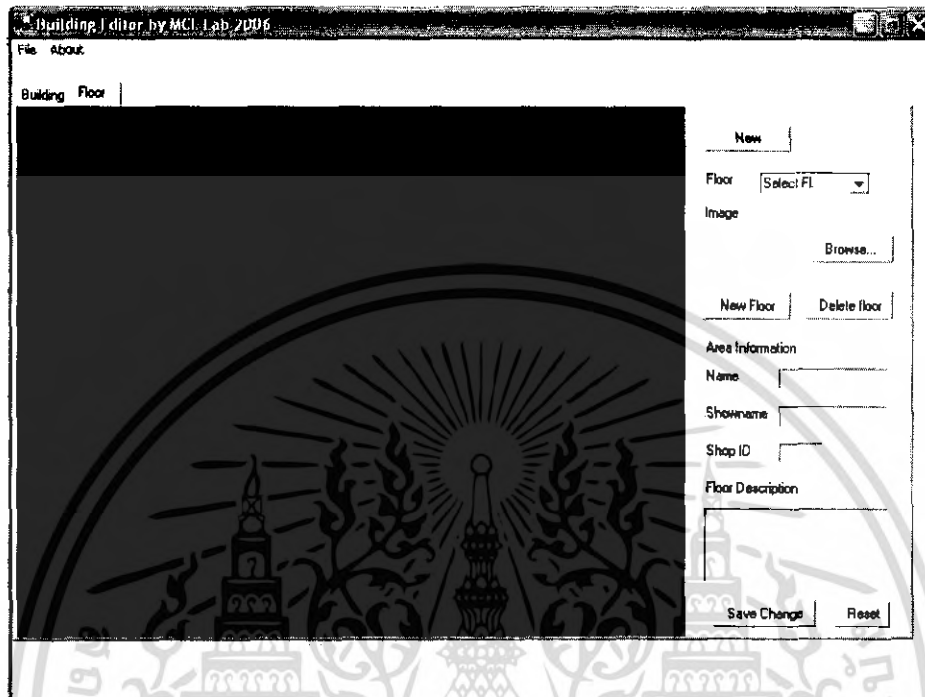
| Building Information |              |
|----------------------|--------------|
| Name :               | Siam Paragon |
| Latitude :           | 100          |
| Longitude :          | 200          |

**รูปที่ 4.4** แสดงส่วนที่ใช้ในการแก้ไขรายละเอียดภายนอกอาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.2. วิธีการใช้งาน Tab Floor

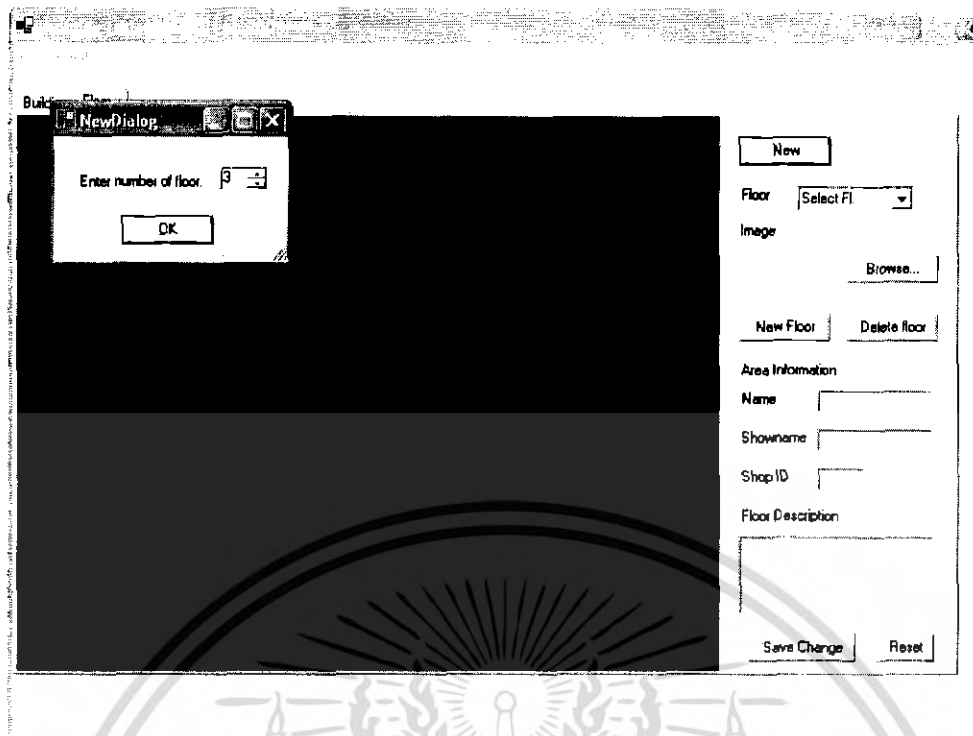
Tab Floor เป็นส่วนที่ใช้ในการสร้างข้อมูลและรายละเอียดภายในตึก เช่น ชั้นและร้านค้า ย่อยทั้งหมด(การจัดการภายในอาคาร) ดังรูปที่ 4.5



รูปที่ 4.5 แสดงลักษณะและส่วนประกอบของ Tab Floor

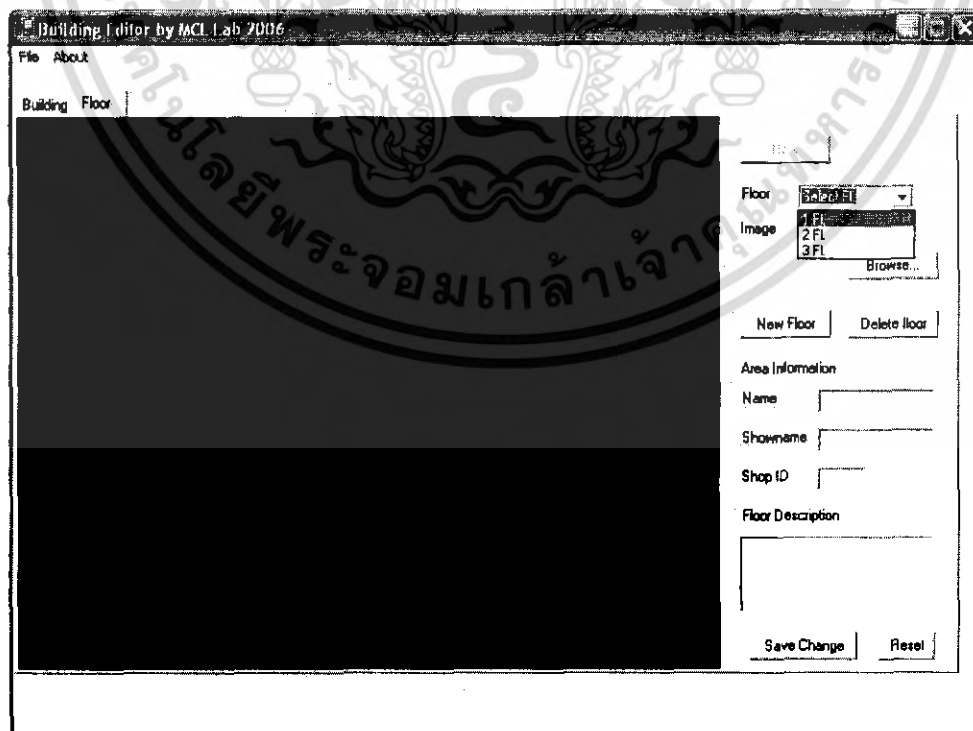
1. เริ่มจากการสร้างโครงอาคารซึ่งประกอบไปด้วยชั้น(Floor) โดยกดเลือกที่ปุ่ม New จะมี Dialog Box ปรากฏขึ้นมาดังรูปที่ 4.6 เลือกจำนวนชั้นที่ต้องการสร้างจากนั้นกด OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดง Dialog Box ที่ใช้ในการกำหนดจำนวนชั้นให้กับอาคาร

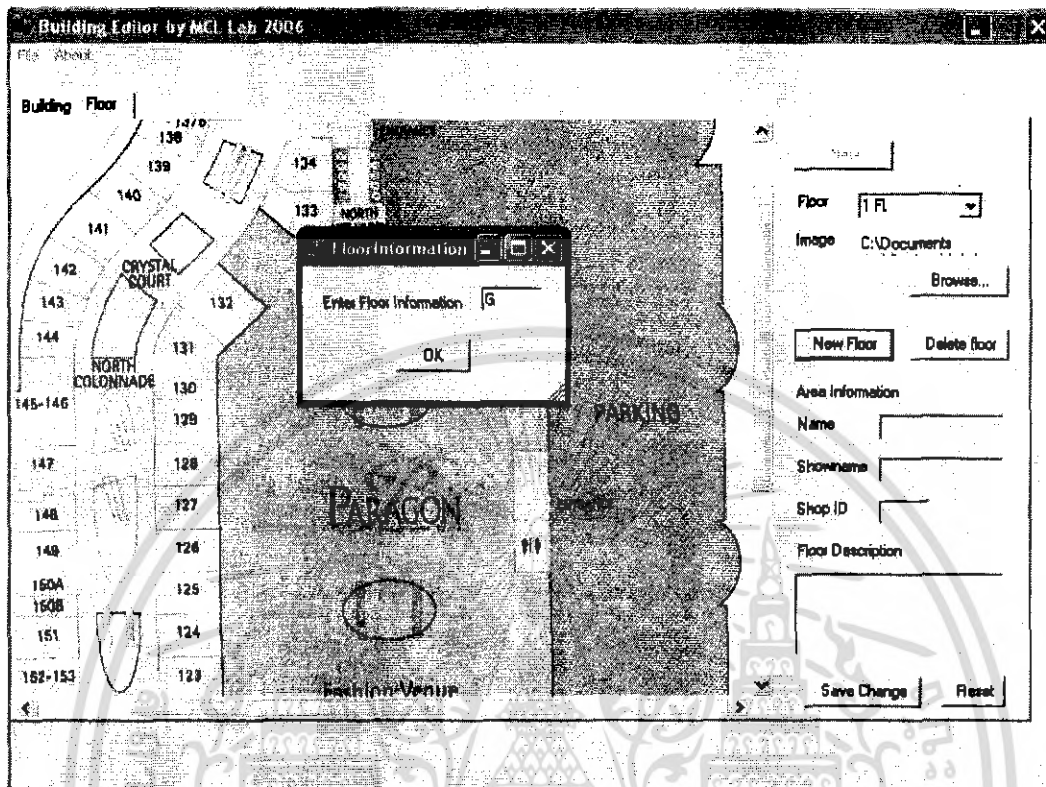
2. ที่ Floor ComboBox ด้านขวามือ จะปรากฏจำนวนชั้นเท่ากับที่เราได้เลือกไว้ใน DialogBox ดังรูปที่ 4.7



รูปที่ 4.7 แสดงชั้นทั้งหมด

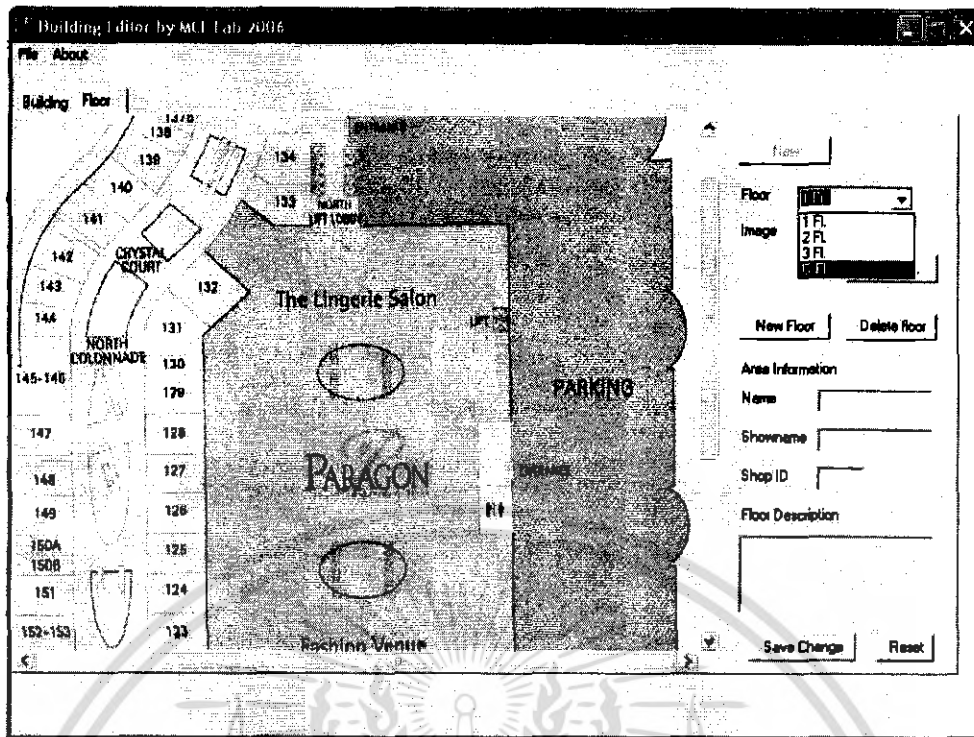
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถทำการเพิ่มหรือลดชั้นนอกเหนือจากที่เราได้กำหนดไว้ในตอนแรก ด้วยการ Click ที่ปุ่ม New Floor จะมี Dialog Boxปรากฏขึ้น ดังรูปที่ 4.8



รูปที่ 4.8 แสดง Dialog Boxที่ใช้ในการตั้งชื่อชั้น

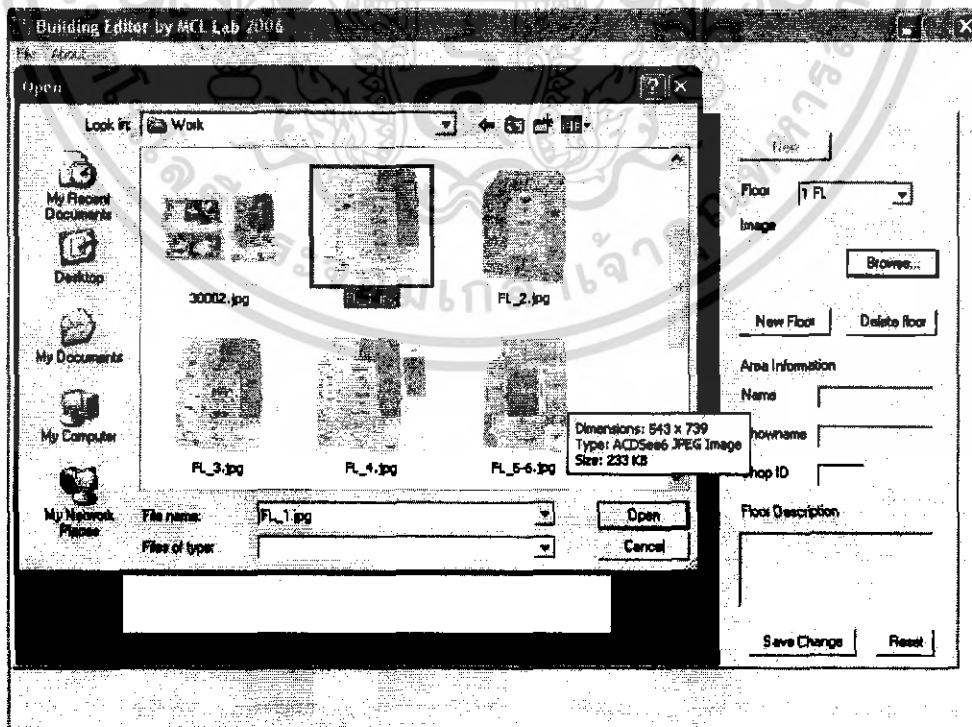
พิมพ์ชื่อชั้นที่ต้องการลงไป (จากภาพคือชั้น G) จากนั้นกด OK จะมีชั้น G ปรากฏเพิ่มขึ้น ในแถบ Combo Box Floor ด้านขวามือดังรูปที่ 4.9 จากนั้นเลือกชั้นที่ต้องการจะสร้าง



รูปที่ 4.9 แสดงค่าที่เพิ่มเข้ามาใน Combo Box ภายหลังจากทำการ New Floor

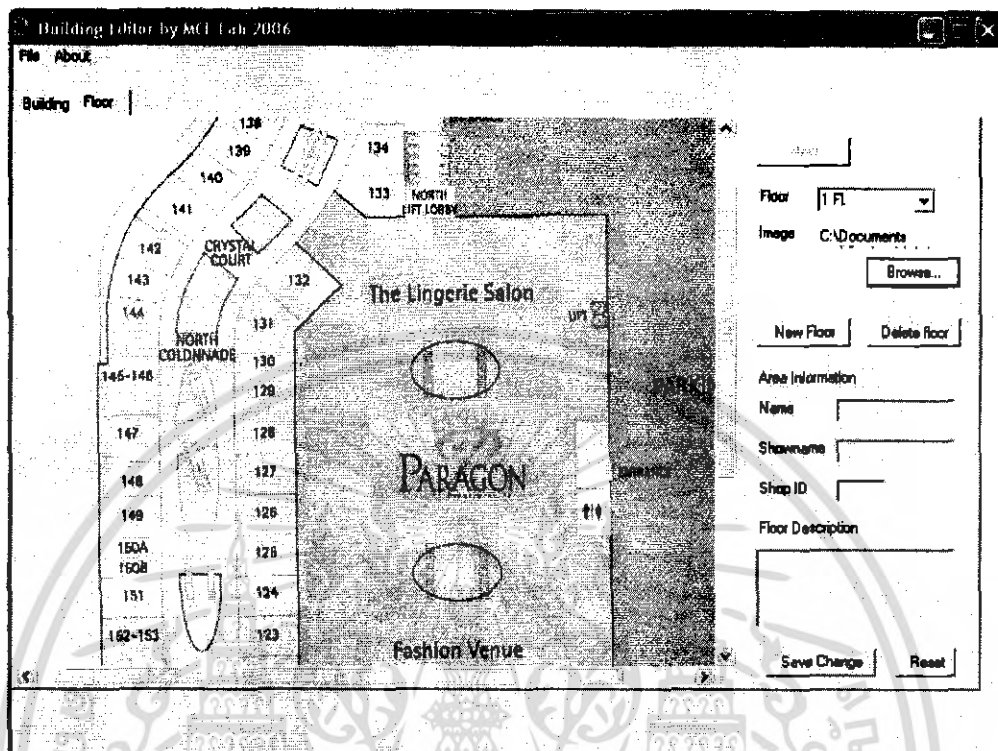
3. กำหนดภาพให้กับชั้นที่เราเลือก โดย Click Browse จะมี Dialog Box ปรากฏขึ้น ดังรูปที่

4.10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะภายในหน่วยงานเท่านั้น ไม่ควรออกให้ทั่วไปใช้ประโยชน์ด้านการค้า  
**รูปที่ 4.10** แสดง Dialog Box ที่ใช้ในการเลือกภาพ Background  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.ภาพที่เราเลือกจะถูกนำมาแสดง ดังรูปที่ 4.11



รูปที่ 4.11 แสดงการนำภาพ Background ที่ถูกเลือกมาแสดง

#### 4.2.3 วิธีการสร้าง Polygon

ต่อไปจะเป็นวิธีการสร้าง Polygon ขึ้นมาเพื่อใช้ในการกำหนดรายละเอียดและEvent ให้กับร้านค้าแต่ละร้าน

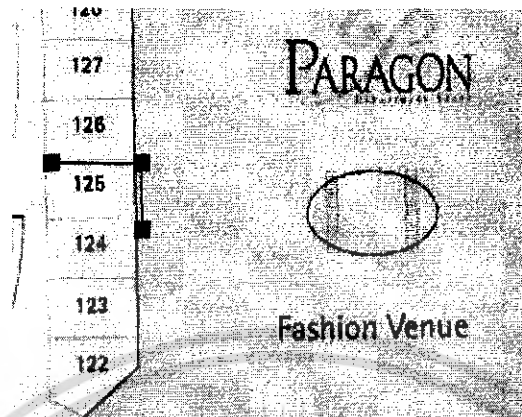
1. เริ่มต้นการสร้าง Polygon ด้วยการ Click ช้าลงบนตำแหน่งที่ต้องการบนภาพ จะปรากฏสัญลักษณ์เป็นจุดสี่เหลี่ยมดังรูปที่ 4.12



รูปที่ 4.12 แสดงลักษณะของ Marker ที่จะปรากฏขึ้นเมื่อทำการ Click ช้าบนภาพ

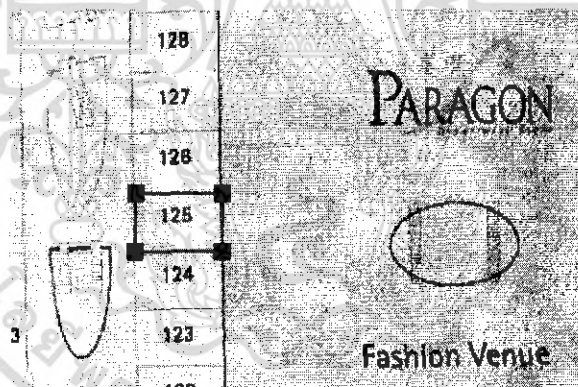
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อทำการClickซ้ายลงบนภาพที่ตำแหน่งใหม่ จะเกิดเป็น Marker ขึ้นมาอีกหนึ่งจุด พร้อมเส้นเชื่อมจากจุดปัจจุบันและจุดที่แล้วต่อกัน ไปเรื่อยๆ ดังรูปที่ 4.13



รูปที่ 4.13 แสดงลักษณะการวาดเส้นเชื่อมต่อระหว่างจุด

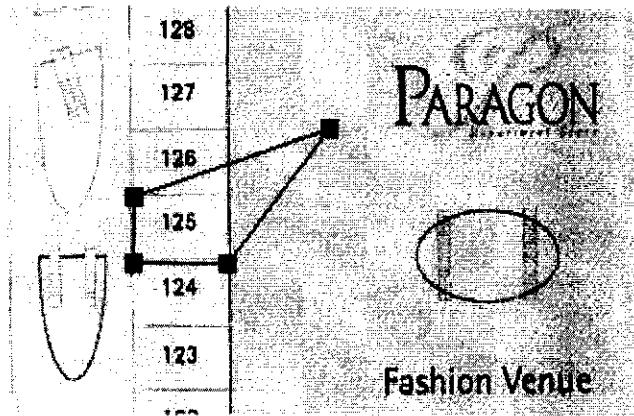
3. เมื่อ Click ลงบน marker ที่เป็นจุดเริ่มต้นของ Polygon จะเป็นการปิด Polygon ดังรูปที่ 4.14



รูปที่ 4.14 แสดงตัวอย่างของการวาดครบ 1 Polygon

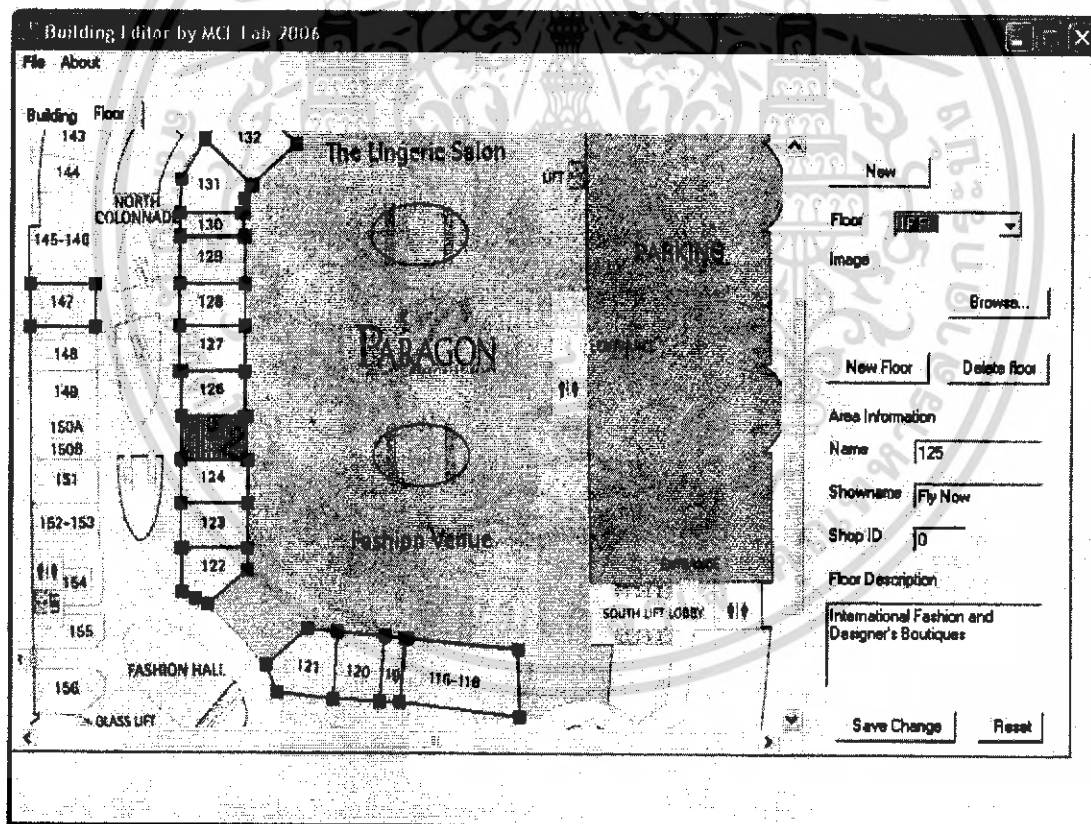
สามารถทำการเคลื่อนย้ายจุด(Marker) ของ Polygon ได้ด้วยการ Click ซ้าย ณจุดที่ต้องการค้างไว้แล้วลาก ไปวางไว้ยังที่ที่เราต้องการ ดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงการเคลื่อนย้ายจุด(Marker) ของPolygon

4. เราสามารถเลือกเพื่อกำหนดค่าและรายละเอียดต่างๆให้กับ Polygon ที่เราสร้างได้โดยการ Click ซ้ายภายในบริเวณของ Polygon ที่ต้องการดังรูปที่ 4.16



รูปที่ 4.16 แสดงภาพตัวอย่างระหว่างการใช้งาน Building Editor

จากรูปที่ 4.16 ภายในพื้นที่ของ Polygon ที่เราเลือก (Click ซ้าย) จะถูกเน้นด้วยสีแดง ทางฝั่งขวามือจะเป็นช่อง(Text Box)ให้เรากรอกรายละเอียดต่างๆให้กับ Polygon นี้ เช่น Name คือ ชื่อ Block หรือ สัญลักษณ์แทนตำแหน่งร้านค้า ShowName คือ ชื่อของร้านค้าที่จะถูกนำไปแสดงในวารสารใดๆ ทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

shopid คือ ID ของร้านค้าที่อยู่ใน Database เพื่อให้ผู้ใช้สามารถทำการดูรายละเอียดของร้านค้าใดๆ ได้ในทันที (ในที่นี้หมายถึง ID ของร้านอาหาร) Floor Description ใช้บอกประเภทของร้านค้าในแต่ละชั้น

ปุ่ม Save Change เมื่อเราทำการแก้ไขค่าเรียบร้อยแล้ว จะต้องทำการ Click ปุ่ม Save Change เพื่อเป็นการเปลี่ยนแปลงของข้อมูลด้วยทุกครั้ง

ปุ่ม Reset ใช้เพื่อทำการ Clear ค่าใน Text Box ทั้งหมดที่เราได้พิมพ์ไปแล้วของ Polygon ออกดังรูปที่ 4.17

Area Information

Name

Showname

Shop ID

Floor Description

Save Change    Reset

**รูปที่ 4.17** แสดงตัวอย่างการ Clear ค่าเมื่อ Click ปุ่ม reset

**4.2.4 การบันทึกไฟล์**

ทำการบันทึกงานลงไฟล์ โดย Click ที่แท็บ File (บน menu bar) และเลือกคำสั่ง Save

File    About

Open

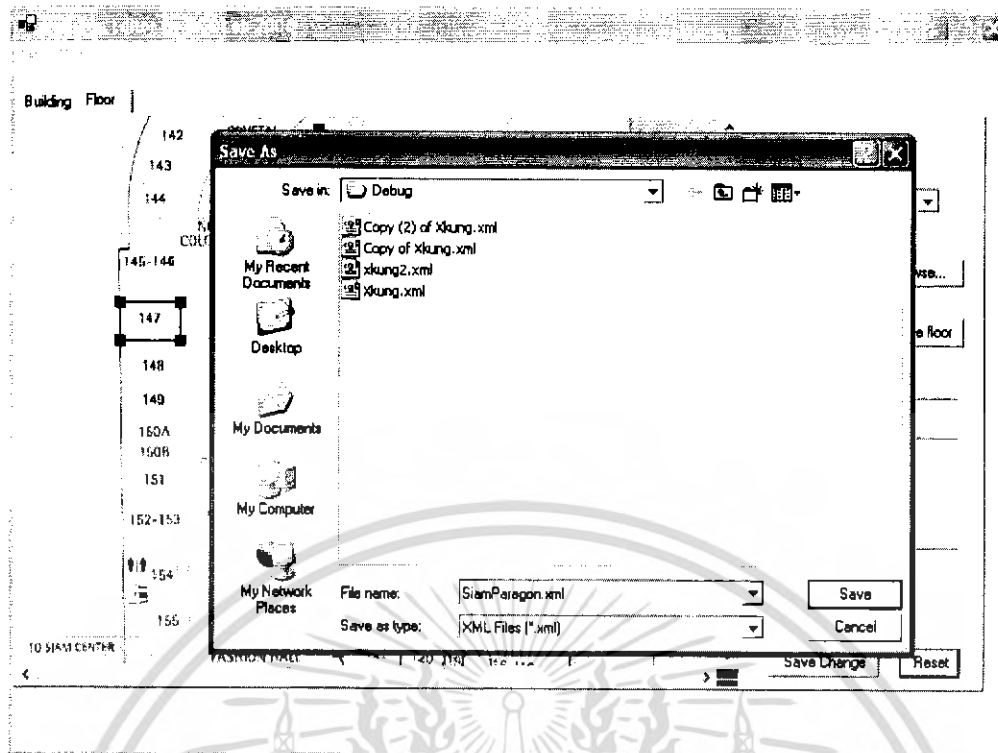
Save

Save to webservice

143

จะมี Dialog Box ปรากฏขึ้นมาดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

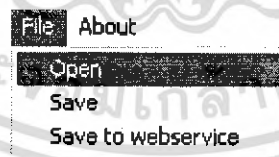


#### รูปที่ 4.18 แสดง Dialog Box ที่ใช้ในการบันทึกข้อมูลลง File

จากรูปที่ 4.8 ชื่อไฟล์ที่เรากำหนดคือ Siam Paragon จากนั้นกด Save จะได้ไฟล์ SiamParagon.xml ซึ่งใช้เปิดเพื่อทำการแก้ไขใหม่ได้

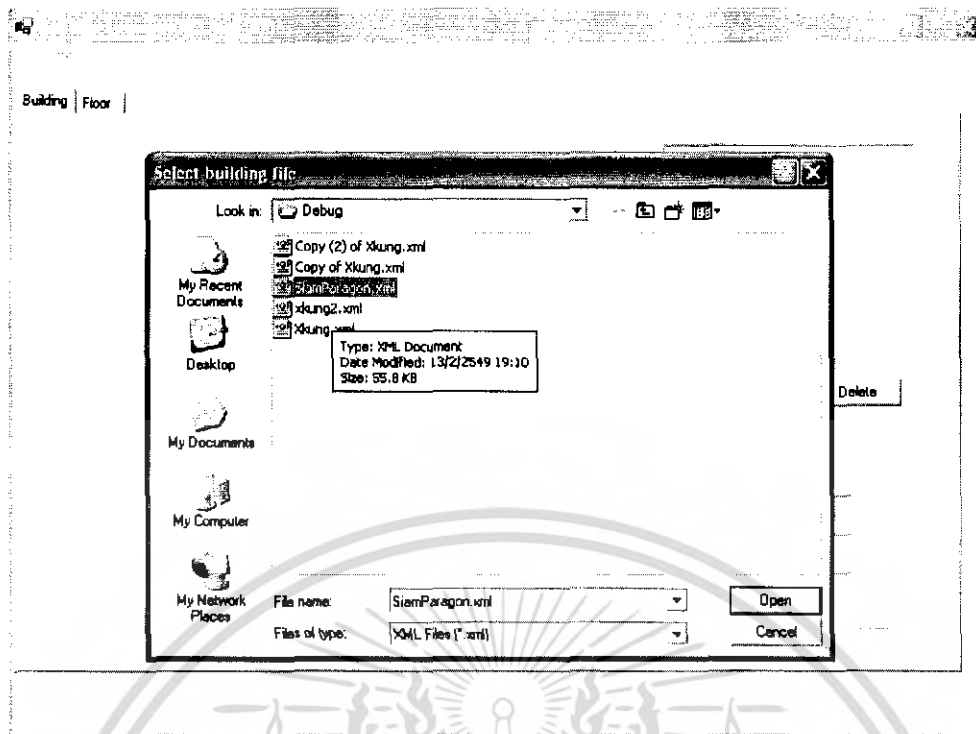
#### 4.2.5 การเปิดไฟล์ที่บันทึกไว้

สามารถเปิดไฟล์ที่เก็บไว้ได้โดย Click ที่แท็บ File (บน menu bar) และเลือกคำสั่ง Open



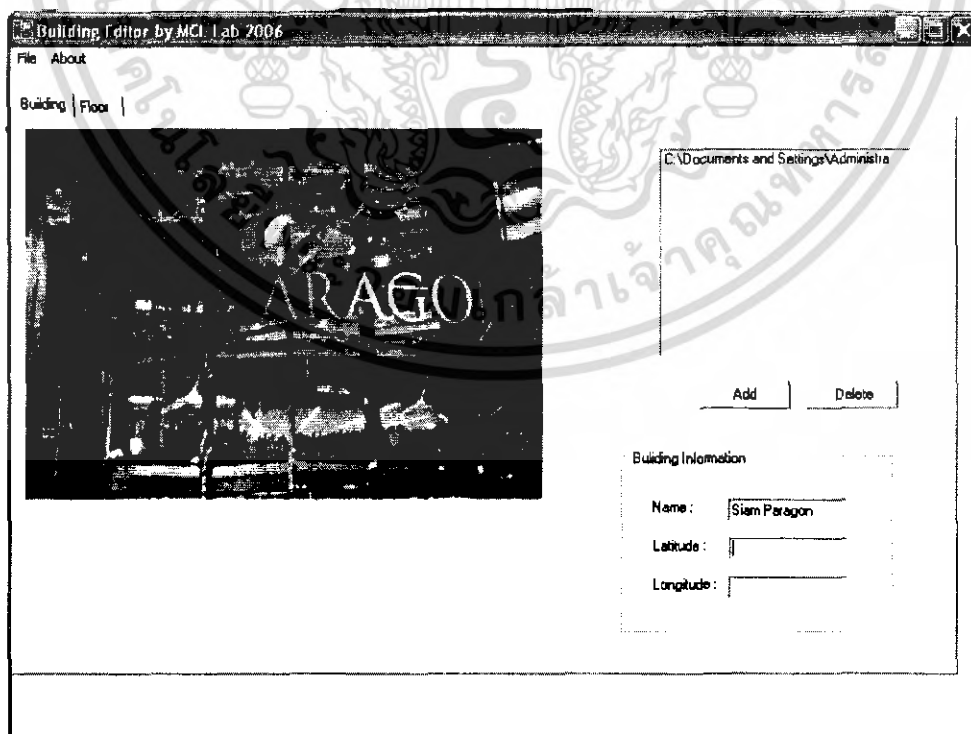
จะมี Dialog Box ปรากฏขึ้นมาดังรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 4.19** แสดง Dialog Box ที่ใช้ในการ เปิดข้อมูลที่ถูเก็บไว้ใน File

จากรูปที่ 4.19 เป็นการเลือกเปิดไฟล์ที่ชื่อว่า SiamParagon.xml ที่เราได้ทำการบันทึกไว้ เพื่อทดสอบ จะปรากฏข้อมูลต่างๆ ที่เราได้บันทึกไว้ก่อนทำการปิด ดังรูปที่ 4.20



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการวิจัยเท่านั้น ไม่ควรนำเนื้อหาไปใช้ประโยชน์ด้านการค้า  
**รูปที่ 4.20** แสดงการเปิดข้อมูลจากไฟล์  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองพบว่า Building Editor สามารถใช้ทำการสร้างข้อมูลของอาคารได้อย่างถูกต้องและรวดเร็ว

#### 4.2.6 การทดลองโดยทำการบันทึกข้อมูลที่ได้สร้างขึ้นลงสู่คอมพิวเตอร์

เมื่อทำการบันทึกแล้ว ก็จะได้ไฟล์ xml ออกมาตามต้องการซึ่งมีลักษณะของ output แบบคร่าวๆ ดังนี้

```
<?xml version="1.0" encoding="TIS-620"?>
<Building>
  <Floor>
    <ID>0</ID>
    <Name>1F</Name>
    <Background>C:\Documents and
Settings\Administrator\Desktop\Work\FL_1.jpg</Background>
    <Polygon>
      <ID>0</ID>
      <Name>147</Name>
      <SHOWNAME>Nagara</SHOWNAME>
      <DESC>International Fashion and Designer's Boutiques</DESC>
      <SHOPID>0</SHOPID>
      <Marker>
        <X>77</X>
        <Y>333</Y>
      </Marker>
      <Marker>
        <X>77</X>
        <Y>302</Y>
      </Marker>
      <Marker>
        <X>125</X>
        <Y>302</Y>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

</Marker>
<Marker>
  <X>125</X>
  <Y>333</Y>
</Marker>
</Polygon>
<Polygon>
  <ID>1</ID>
  <Name>132</Name>
  <SHOWNAME>Gems Pavillion</SHOWNAME>
  <DESC>Jewery and Watches</DESC>
  <SHOPID>0</SHOPID>
  <Marker>
    <X>205</X>
    <Y>194</Y>
  </Marker>
  <Marker>
    <X>235</X>
    <Y>164</Y>
  </Marker>
  <Marker>
    <X>272</X>
    <Y>198</Y>
  </Marker>
  <Marker>
    <X>239</X>
    <Y>228</Y>
  </Marker>
</Polygon>
</Floor>
<Floor>
  <ID>2</ID>

```

เอกสารนี้เป็นเอกสารที่ **<Name>B2</Name>** ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<Background>C:\Documents and settings\Administrator

\Desktop\Work\FL\_B2.jpg</Background>

</Floor>

<BPicture>D:\My Documents\dietbuddy1.jpg</BPicture>

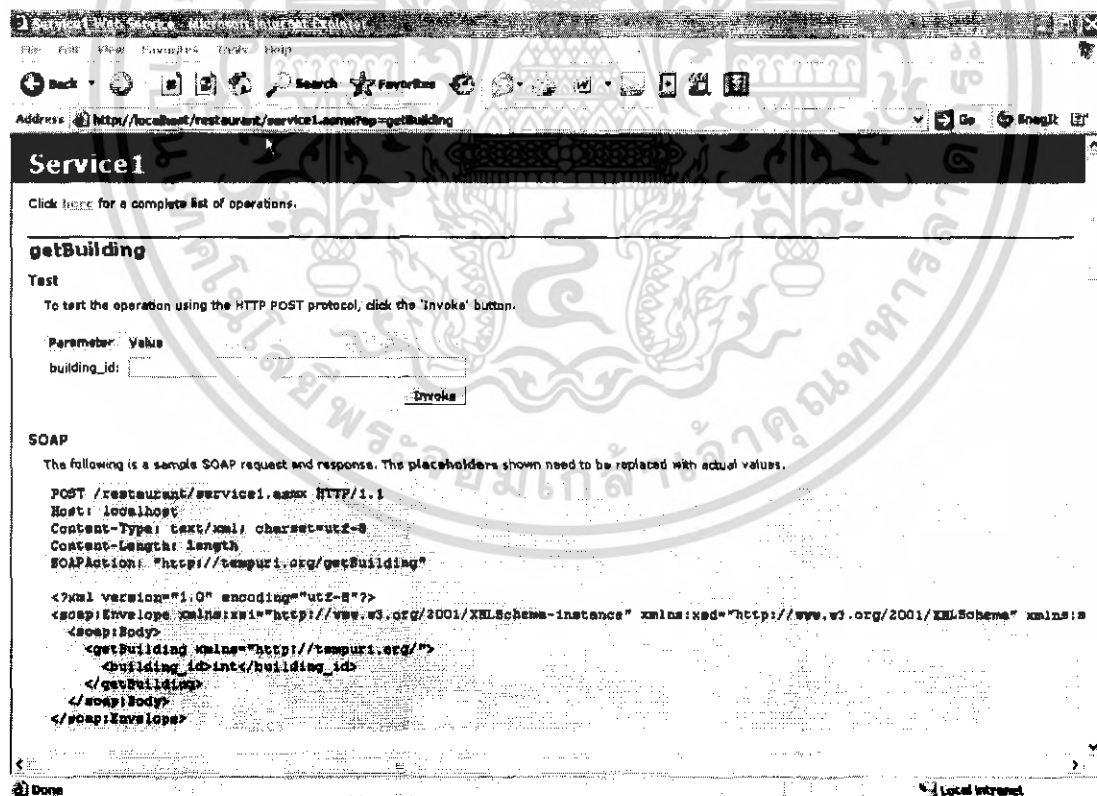
<BPicture>0</BPicture>

</Building>

หลังจากนั้นก็ทำการปิด โปรแกรมแล้วทำการเปิด โปรแกรมขึ้นมาใหม่ ทำการเปิดไฟล์ที่ได้บันทึกไว้ ก็พบว่าสามารถเปิดไฟล์ขึ้นมาแสดงผลและแก้ไขได้โดยไม่มีปัญหาใดๆ

#### 4.2.7 การทดลองอัปเดตข้อมูลผ่าน Web Service

การทดลองนี้เริ่มต้น โดยการเลือกเมนู Save to webservice ของโปรแกรม เมื่อเลือกเมนู ไปแล้ว เราก็ได้ทำการทดลอง โดยการเรียกใช้คำสั่งของ Web Service เพื่อทำการดึงข้อมูลออกมาดูว่าเป็นเช่นไร



รูปที่ 4.21 ภาพแสดงการทดสอบคำสั่ง getBuilding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในของหน่วยงานเท่านั้น ไม่ควรเผยแพร่สู่สาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<?xml version="1.0" encoding="utf-8" ?>
- <Building xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://tempuri.org/">
  <name>KKung Building</name>
  <id>59</id>
  <lat>13.758899688720703</lat>
  <lon>100.51200103789785</lon>
  - <floor>
  + <Floor>
  + <Floor>
  + <Floor>
  + <Floor>
  + <Floor>
  + <Floor>
  + <Floor>
  + <Floor>
  + <Floor>
  + <Floor>
  + <Floor>
  </floor>
</Building>

```

#### รูปที่ 4.22 ภาพแสดงผลการทดสอบการใช้คำสั่ง getBuilding

ผลการทดลองคือ ข้อมูลที่ออกมาจะเป็นข้อมูลของ Building ที่เราได้ทำการอัปโหลดเข้าไป

### 4.3 การทดลองใช้ KMLLoader

การทดลองนี้เริ่มจากทำการดาวน์โหลดไฟล์ KML จาก <http://www.thaigoogleeearth.com> มาก่อน โดยในตอนแรกไฟล์ที่โหลดมานั้นจะเป็นไฟล์ KMZ เราต้องทำการเปิดโดยใช้ WinZip ก่อนเพื่อจะดึงข้อมูล KML ที่ถูกบีบอัดไว้ภายในออกมา สำหรับตัวอย่างข้อมูลในไฟล์ KML นั้นสามารถแสดงได้ดังข้อมูลข้างล่างนี้

```
<Placemark>
```

```
<name>ถ.แจ้งวัฒนะ</name>
```

```
<LookAt>
```

```
<longitude>100.5717692219916</longitude>
```

```
<latitude>13.88823419746046</latitude>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<range>3476.527353640776</range>
<tilt>12.09988741751842</tilt>
<heading>0.007393010502167959</heading>
</LookAt>
<styleUrl>#khStyle583</styleUrl>
<Point>
<coordinates>100.5684781083375,13.89026388877073,0</coordinates>
</Point>
</Placemark>

```

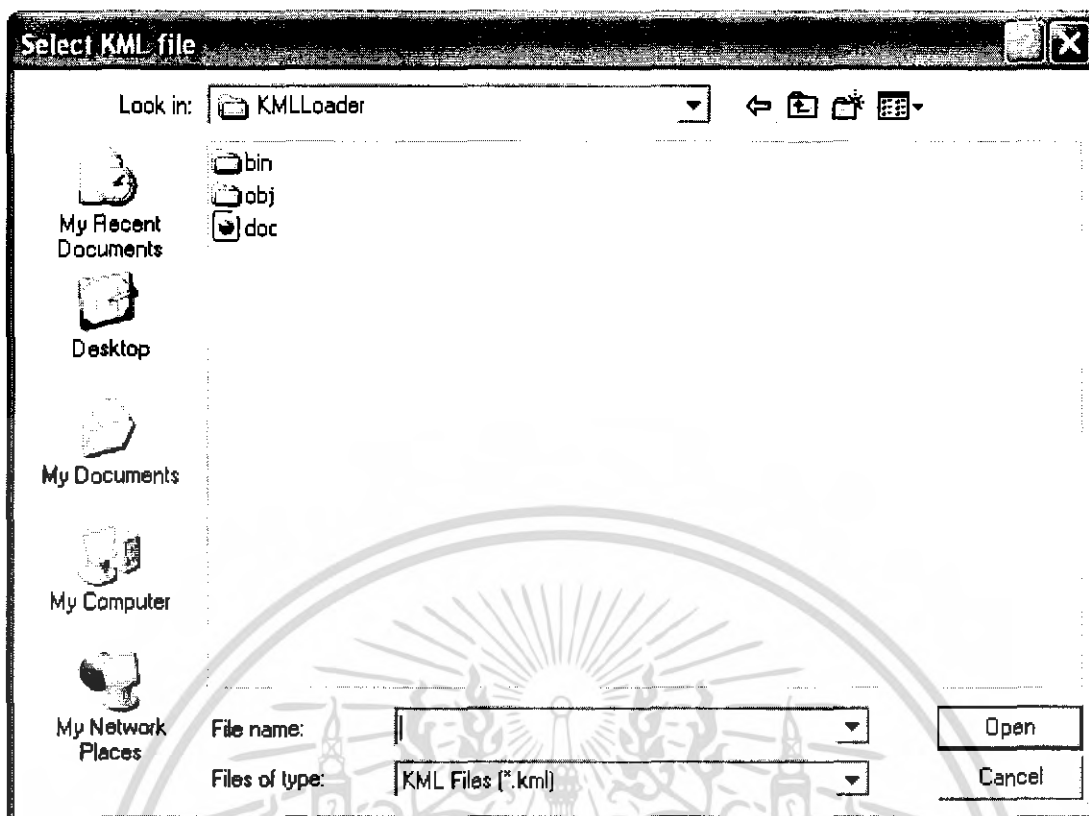
เมื่อได้ KML file มาแล้ว เราก็ทำการเปิดโดยโปรแกรม KML Loader แล้วสั่งให้อัพโหลดไปยัง Server กลาง ก็พบว่า สามารถอัพโหลดข้อมูลของ Placemark ที่ต้องการไปยัง Server กลางได้โดยไม่พบปัญหาใดๆ



**รูปที่ 4.23** ภาพแสดงหน้าจอเริ่มต้นการใช้งาน KMLLoader

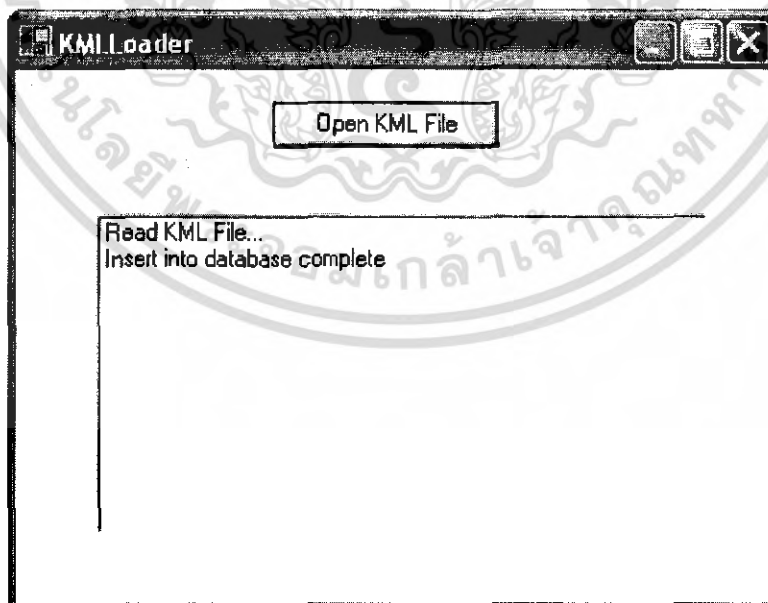
เมื่อเปิดโปรแกรมมาจะพบกับหน้าจอ ดังภาพ และเมื่อกดปุ่ม Open KML File ก็จะพบกับหน้าจอให้เลือกไฟล์ที่ต้องการเปิดดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 ภาพแสดงการเปิดไฟล์ KML

จากนั้น โปรแกรมก็จะทำการอ่านไฟล์และถ้าทำสำเร็จก็จะพบกับหน้าจอผังภาพ



รูปที่ 4.2.5 ภาพแสดงหน้าจอหลังจากทำการอ่านไฟล์ KML เสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองพบว่า หลังจากทำการเปิดไฟล์ขึ้นมาอ่านแล้ว โปรแกรมได้ทำการอัปเดตข้อมูลของ Placemark เข้าไปยัง Database ได้อย่างสมบูรณ์

#### 4.4 การทดลองใช้ ThaiMapGuide Server

การทดลองนี้เริ่มจากการเปิด โปรแกรม ThaiMapGuide Server ขึ้นมา ซึ่งจะพบกับ หน้าจอ ดังภาพ



**รูปที่ 4.2.6** ภาพแสดงหน้าจอเริ่มต้นการใช้งาน โปรแกรม ThaiMapGuide Server

การทดลองนี้เราใช้การรัน Web Service ผ่าน Visual Studio.NET ในการทดลอง โดยหลังจาก Run Web Service ขึ้นมาแล้ว เราก็ทำการเรียกคำสั่ง GetTmgMap ซึ่งจะพบกับหน้าจอ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Service1 Web Service - Microsoft Internet Explorer

Address: http://localhost/restaurant/service1.aspx?op=GetMapTmg

## Service1

Click [here](#) for a complete list of operations.

### GetMapTmg

**Test**

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

| Parameter | Value            |
|-----------|------------------|
| iduser:   | 1                |
| lat:      | 13.7468362493268 |
| lon:      | 100.583449229678 |

**SOAP**

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /restaurant/service1.aspx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/GetMapTmg"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetMapTmg xmlns="http://tempuri.org/">
      <iduser>string</iduser>
      <lat>double</lat>
    </GetMapTmg>
  </soap:Body>
</soap:Envelope>
```

### รูปที่ 4.2.7 ภาพแสดงหน้าจอการเริ่มต้นทดสอบคำสั่ง GetMapTmg

หลังจากนั้นเราก็กรอกค่าของ User ID, Latitude และ Longitude ที่อยู่ภายในประเทศไทยลงไป ซึ่งหลังจากทำการสั่งให้ Function GetTmgMap ทำงานแล้ว Web Service ก็จะมีการติดต่อไปยัง ThaiMapGuide Server แล้ว ThaiMapGuide Server ก็จะมีการวิเคราะห์ Latitude, Longitude ที่ส่งมาให้ แล้วทำการอัพโหลดภาพแผนที่ที่สอดคล้องกับจุดที่ Request มาขึ้นไปยัง Server กลาง

ผลของการทดลองเราได้ออกมาเป็นข้อมูล xml ของภาพนั้นๆ (เป็น byte array) ดังข้างล่างนี้

```
<?xml version="1.0" encoding="utf-8"?>
<TmgData xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://tempuri.org/">
  <mapdata>/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQ
NDAsLDBkSEw8UHRofHh0a...</mapdata>
  <len>0.005119999999999991</len>
</TmgData>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายเราก็ทำการดึงภาพที่ถูกอัปโหลดขึ้นไปใหม่ออกมาดู ก็พบว่า เป็นภาพที่ถูกต้องตรงตามตำแหน่งที่ร้องขอไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# บทวิจารณ์และสรุป

### 5.1 บทสรุป

ในปัจจุบัน IT ได้เข้าไปอยู่ในชีวิตของคนมากขึ้น ระบบแผนที่ที่สามารถเรียกใช้งานได้ทุกที่ทุกเวลาผ่านทาง Browser ที่มีอยู่ในคอมพิวเตอร์ทุกเครื่องและ Pocket PC ที่อยู่ติดตัวตลอดเวลา ย่อมทำให้การใช้ชีวิตสะดวกยิ่งขึ้น

โครงการที่พัฒนาขึ้นเป็นส่วนที่ช่วยในการสร้างและให้บริการดึงข้อมูลแผนที่ ซึ่งสามารถทำงานได้ตามความต้องการ นั่นคือ สามารถดึงภาพแผนที่ ณ ตำแหน่ง Latitude, Longitude ใดๆ ได้, สามารถดึงตำแหน่งของสถานที่ต่างๆ ได้ และสามารถดึงภาพแผนที่ภายในอาคารออกมาแสดงผลได้

### 5.2 วิจารณ์สิ่งที่ได้จากโครงการ

การทำโครงการชิ้นนี้ ทำให้พวกผมได้รับความรู้และประสบการณ์มากมาย สิ่งเหล่านี้ประกอบด้วย การเขียนโปรแกรมด้วยภาษา C#, การเขียนโปรแกรมด้วยภาษา Basic, การใช้เทคโนโลยี .NET ช่วยในการเขียนโปรแกรม, ความรู้ความเข้าใจเกี่ยวกับ Web Service และการพัฒนา และความรู้เกี่ยวกับการออกแบบและใช้งาน Database

นอกจากความรู้ต่างๆ เหล่านี้แล้ว พวกผมยังได้รับประสบการณ์ต่างๆ เช่น การกำหนดระยะเวลาและวางแผนการทำงาน, การหาข้อมูลใน Internet และสิ่งที่สำคัญที่สุดในการทำโครงการชิ้นนี้ก็คือ การทำงานร่วมกันเป็นทีม

สำหรับพวกผมแล้ว ทีมไม่ใช่เพียงแต่กลุ่มคนเพียงสองคนเท่านั้น เพราะโครงการนี้เป็นโครงการที่ทำร่วมกันถึงสามกลุ่ม นั่นหมายถึงมีคนถึงหกคนเลยทีเดียว การที่จะทำให้งานที่แต่ละคนทำออกมานั้นสามารถนำมารวมกันได้ และการแบ่งงานให้ทุกคนตามแต่ความถนัดและระยะเวลา สิ่งเหล่านี้ถือเป็นประสบการณ์ที่ล้ำค่าอย่างยิ่ง

ยิ่งเวลาผ่านไป การทำงานร่วมกันก็ยิ่งทำได้ง่ายมากยิ่งขึ้น อีกทั้งสภาวะแวดล้อมที่อยู่ภายในห้องวิจัยเดียวกันแทบจะตลอดเวลา สิ่งเหล่านี้ล้วนช่วยเสริมสร้างความแข็งแกร่งของทีมอย่างมาก จึงนับได้ว่า สิ่งที่ได้รับจากทำโครงการนี้มีคุณค่ามากเหลือเกิน

### 5.3 ปัญหาอุปสรรคที่เกิดขึ้น

1. การทำงานที่เป็นประกอบด้วยกลุ่มหลายกลุ่มทำงานร่วมกัน ทำให้การรวมกันในบางครั้งต้องเสียเวลาไปพอสมควรเนื่องจากว่าในตอนพัฒนาไม่ได้ตกลงกันไว้ก่อน ปัญหาเหล่านี้ อาทิเช่นว่า การตั้งชื่อตัวแปรเหมือนกัน, การตั้งชื่อ Table ใน Database เหมือนกัน เป็นต้น
2. ThaiMapGuide Control ที่ใช้ในโครงการ เป็นเวอร์ชันที่ไม่ได้ลงทะเบียน จึงทำให้การใช้งานทุกครั้งต้องมีการกดปุ่ม Later เพื่อบอกโปรแกรมว่าจะลงทะเบียนในภายหลัง และยังมีอายุการใช้งานเพียงแค่ 30 วัน ทำให้เมื่อครบ 30 วันแล้ว ต้องทำการ Format เครื่อง หรือไม่ก็ย้ายไปทำเครื่องอื่นต่อเรื่อยๆ

### 5.4 แนวทางแก้ไข

1. ปัญหาข้อที่หนึ่งนั้น ควรจะทำการตกลงสิ่งต่างๆ ให้เรียบร้อยก่อนเริ่มต้นพัฒนา
2. ปัญหาข้อที่สองนั้น หากมีผู้สนใจจะนำส่วนของ ThaiMapGuide ที่โครงการนี้พัฒนาไปใช้งานจริง ก็ต้องทำการซื้อลิขสิทธิ์ให้ได้เวอร์ชันเต็มด้วย

### 5.5 แนวทางการพัฒนาต่อ

1. เพิ่มส่วนของการ Compress ข้อมูลที่ส่งไปตาม Network เพื่อให้สามารถรับส่งข้อมูลต่างๆ ได้เร็วขึ้น
2. พัฒนาส่วน Security ให้กับส่วนต่างๆ ที่เกี่ยวข้องกับเครือข่าย อาทิเช่น การส่งข้อมูลระหว่าง Web Service กับ Pocket PC ในส่วนของการกรอก Username และ Password ก็ควรจะมีการทำ Authentication ก่อน เป็นต้น
3. พัฒนาส่วนของโปรแกรม Building Editor ให้ใช้งานได้ง่ายยิ่งขึ้น

## บรรณานุกรม

- [1] ThaiMapGuide.com [Online]. Available :  
: <http://www.thaimapguide.com/html/default.htm>. 2005.
- [2] CodeToad - XML and JavaScript Tutorial [Online]. Available :  
: [http://www.codetoad.com/xml\\_javascript\\_tutorial.asp](http://www.codetoad.com/xml_javascript_tutorial.asp). 2005.
- [3] Scrollable and RatioStretch PictureBox - The Code Project - C# Controls: [Online]. Available :  
: <http://www.codeproject.com/cs/miscctrl/ratiostretchpicturebox1.asp>. 2005.
- [4] เวชอมร จรุงผลลฐิติ, ศักดิ์ชัย เตียวศิริ “เกมส์ต่อสู้เสมือนจริง” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2548
- [5] “การพัฒนาระบบประมวลผลออนไลน์โดยใช้ Web Service” มหำบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยี สารสนเทศ บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



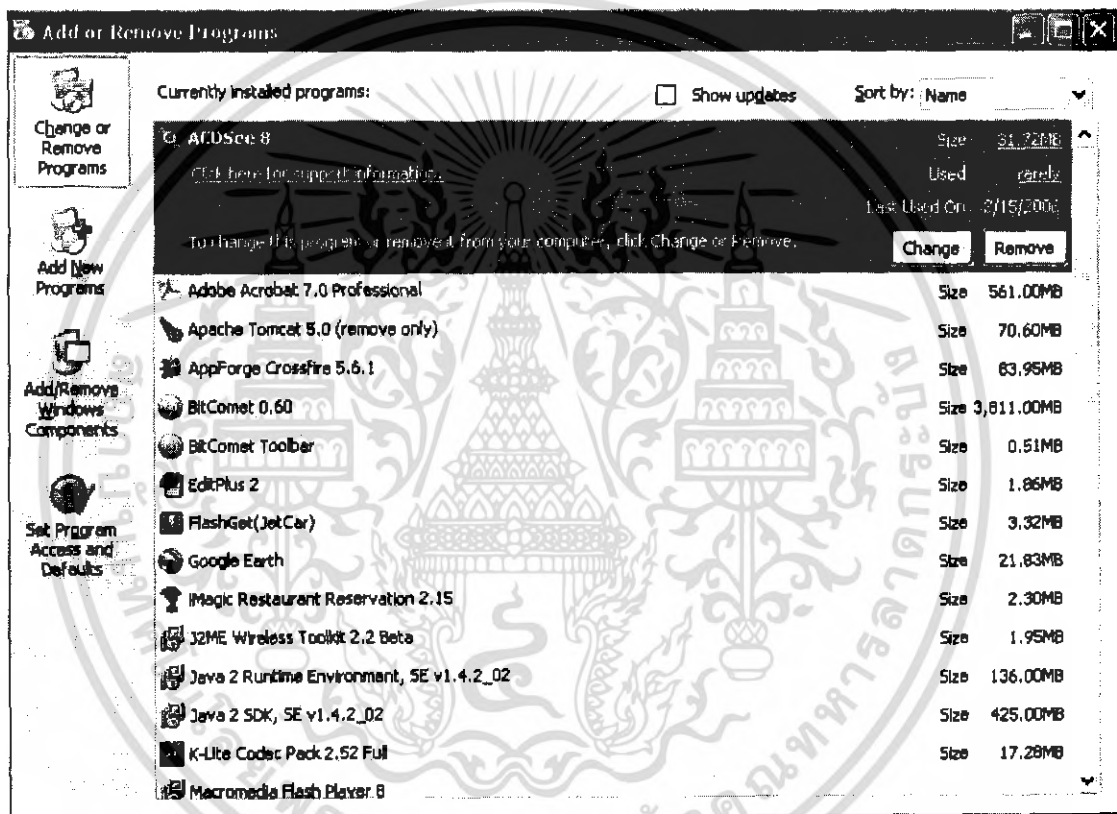
## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### วิธีการ Setup ให้เครื่องสามารถให้บริการ Web Service

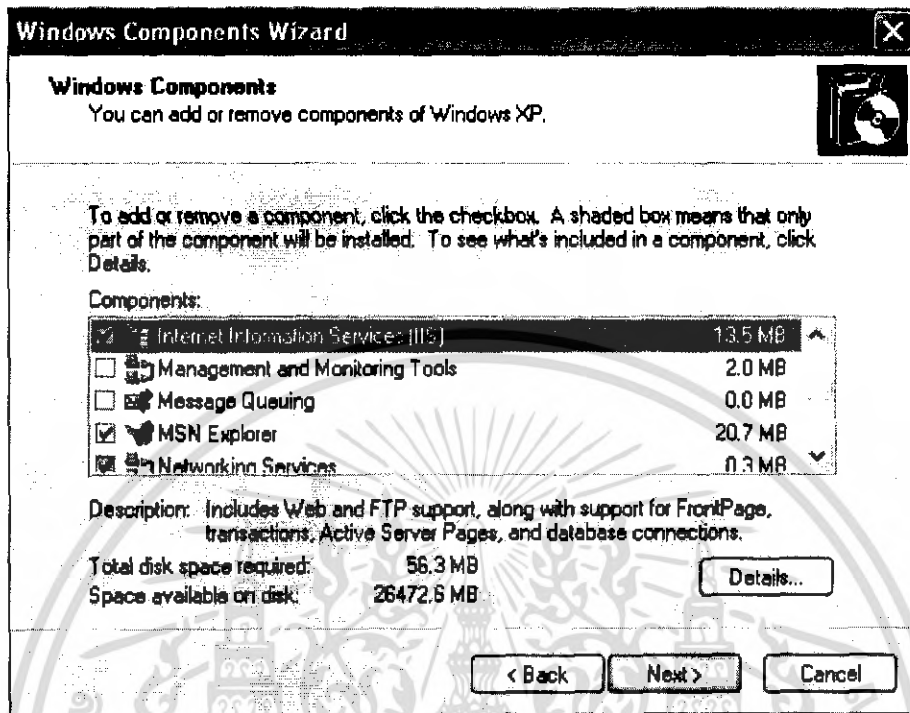
1. เราต้องลง Internet Information Service (IIS) ก่อน โดยไปที่  
Start->Setting->control Panel จากนั้นเลือก Add or Remove Programs จะปรากฏดังรูป (   
ในที่นี้เราจะต้องมีแผ่น Windows อยู่ด้วย)



รูปที่ ก-1 ภาพแสดงหน้าจอ Add or Remove Programs

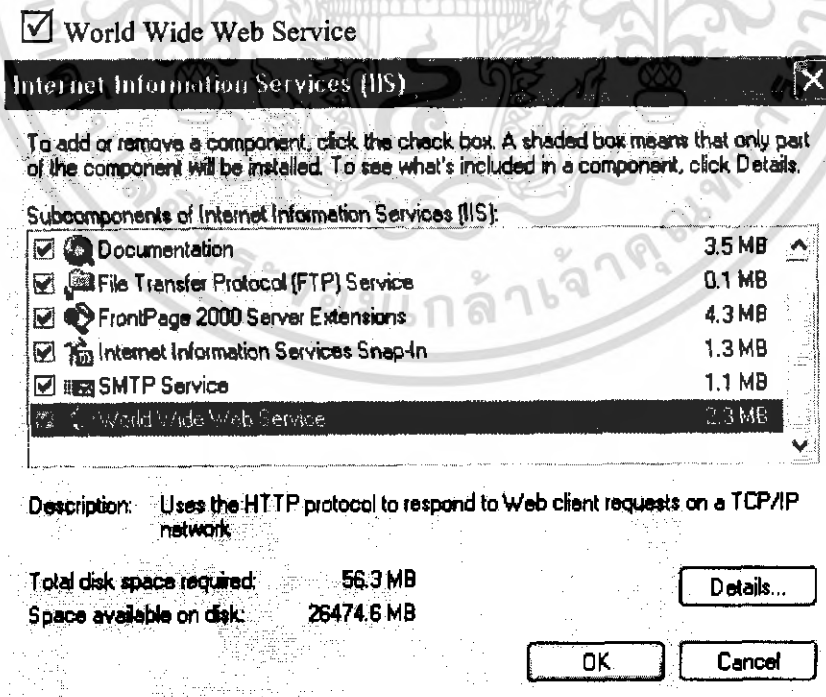
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.1 เลือก Add/Remove Windows Components ทางด้านซ้ายมือจากนั้นทำการเลือก  
เครื่องหมายถูกที่  Internet Information Services (IIS)



รูปที่ ข-2 ภาพแสดงหน้าจอ Add/Remove Windows Components

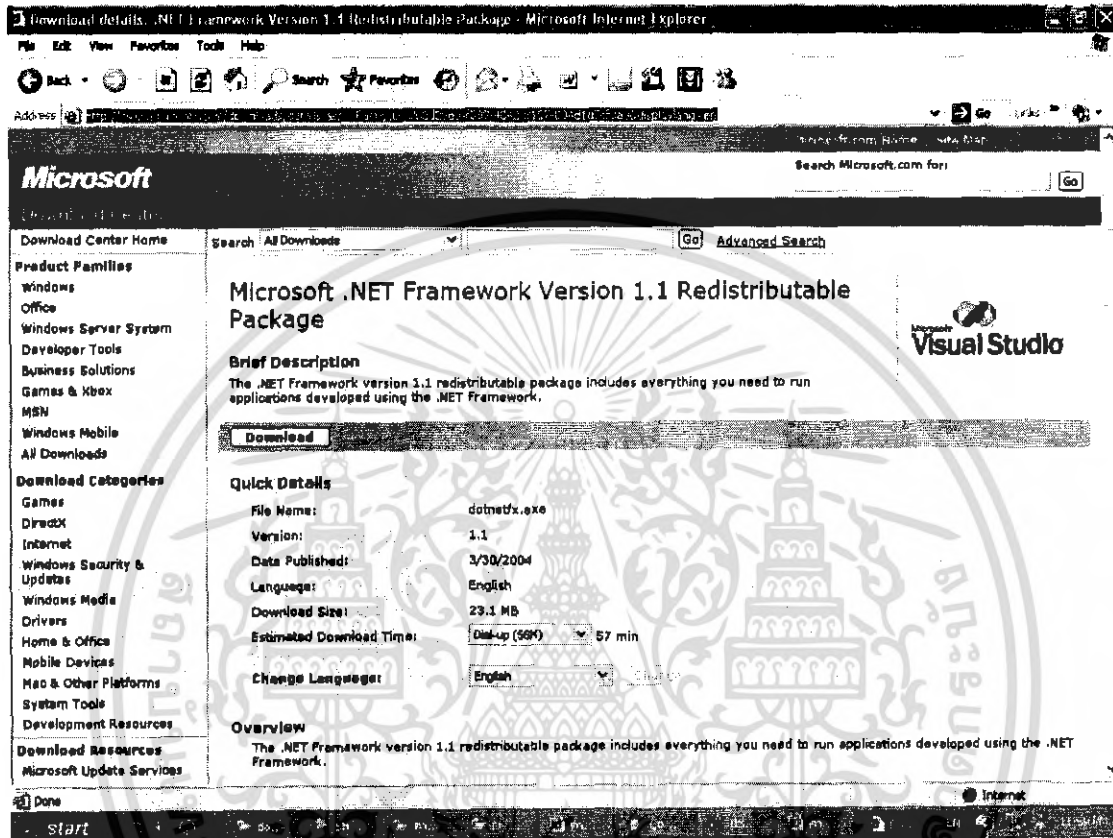
- 1.2 เลือก Detail... ทางด้านขวาล่างแล้วทำการเลือกเครื่องหมายถูกที่



รูปที่ ก-3 ภาพแสดงหน้าจอในการเลือกกล่องคอมโพเนนท์ของ IIS

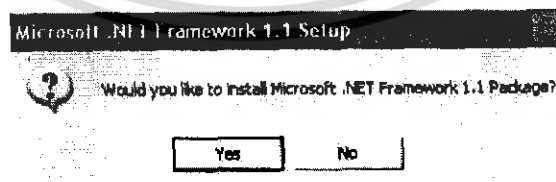
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เราต้องลง Microsoft .NET Framework version 1.1 โดยเราสามารถดาวน์โหลดไฟล์ได้จาก url : <http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&displaylang=en>



รูปที่ ก-4 ภาพแสดงเว็บเพจที่ใช้ในการดาวน์โหลด Microsoft .NET Frameworks

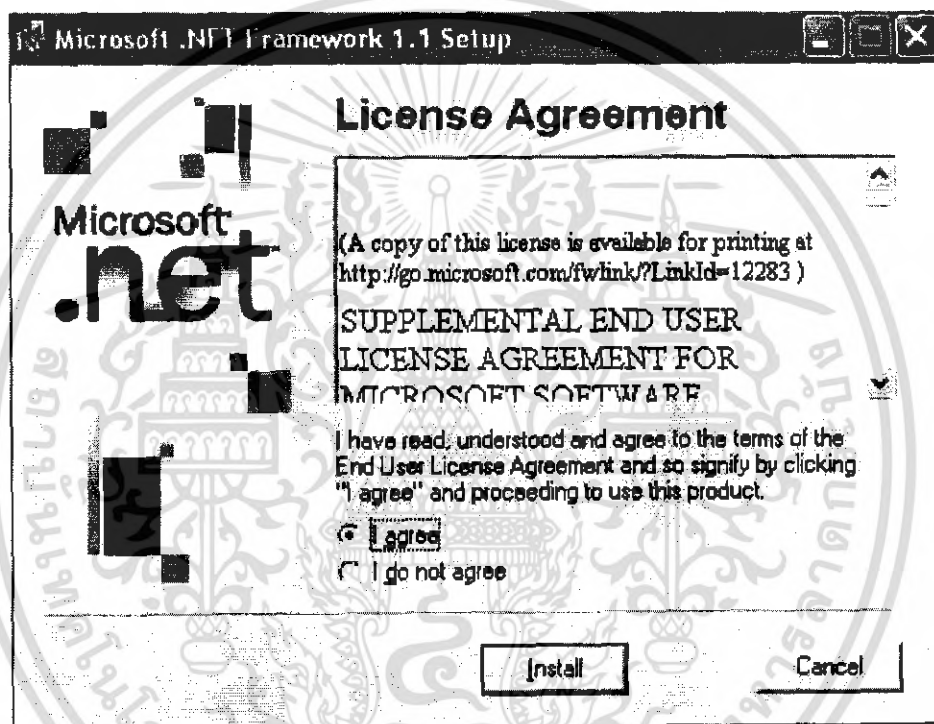
- 2.1 เมื่อดาวน์โหลดเสร็จแล้วเราจะได้ไฟล์ dotnetfx.exe ทำการดับเบิลคลิกแล้วกดปุ่ม “Yes”



รูปที่ ก-5 ภาพแสดงการดับเบิลคลิก dotnetfx.exe ไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

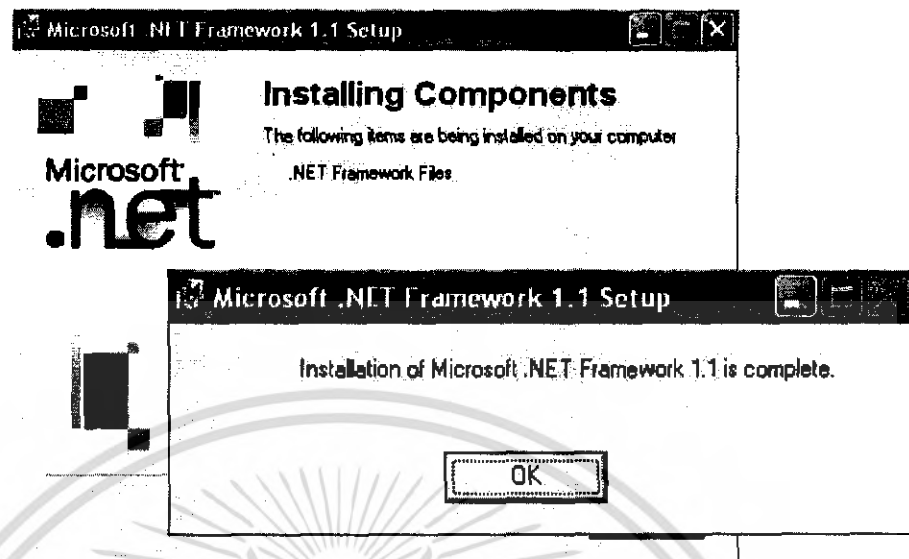
- 2.2 จากนั้นทำการเลือก I agree เพื่อทำการยอมรับข้อตกลงเพื่อที่จะลง  
Microsoft .NET Framework 1.1



รูปที่ ก-6 ภาพแสดงการยอมรับข้อตกลงในการ Install Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 รอโหลดสักระยะก็จะเสร็จสมบูรณ์ดังภาพ



รูปที่ ก-7 ภาพแสดงการลงโปรแกรมเสร็จสมบูรณ์

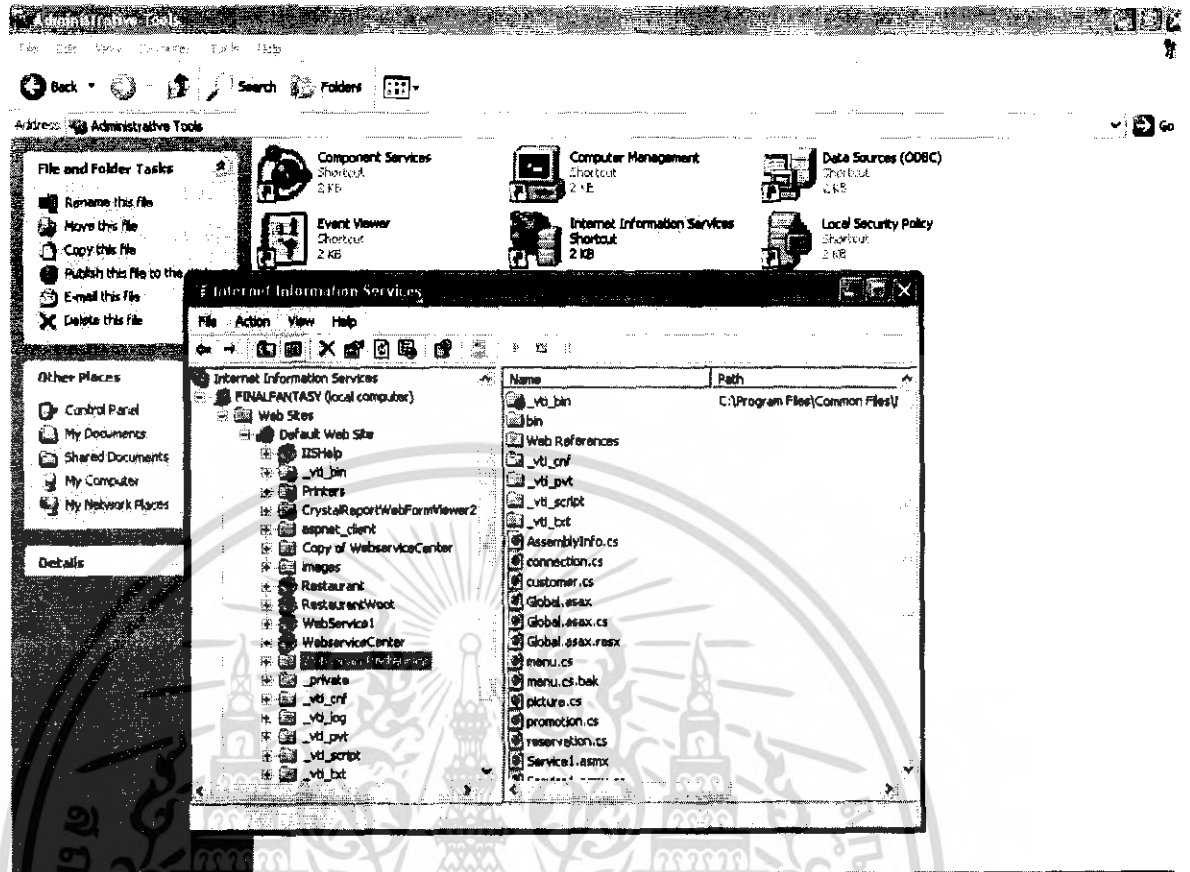
3. จากนั้นทำการทดสอบโดยนำไฟล์เดอร์ของเว็บเซอร์วิสไปไว้ที่ C:\inetpub\wwwroot (ไฟล์เดอร์นี้จะสร้างขึ้นเมื่อเราทำการลง Internet Information Services (IIS) เป็นที่เรียบร้อยแล้วเท่านั้น) โดยไฟล์ที่ใช้ในการรันเว็บเซอร์วิสได้แก่ไฟล์ดังต่อไปนี้ในรูป



รูปที่ ก-8 ภาพแสดงไฟล์ที่ต้องใช้ในการรันเว็บเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

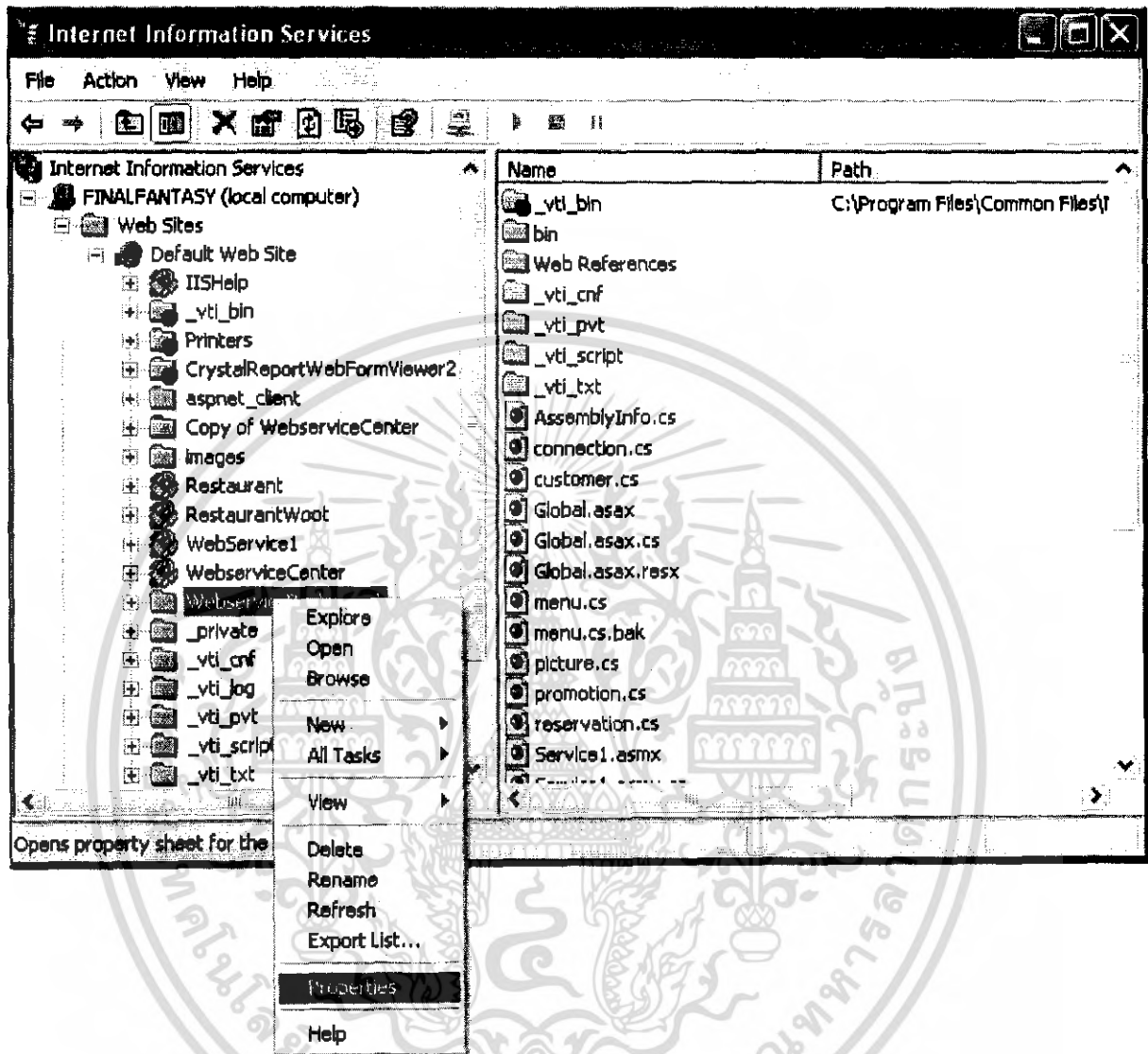
4. ให้เลือก Start->Setting->Control Panel จากนั้นให้เลือก Administrative Tools



รูปที่ ก-9 ภาพแสดงขั้นตอนเปิดบริการใน IIS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

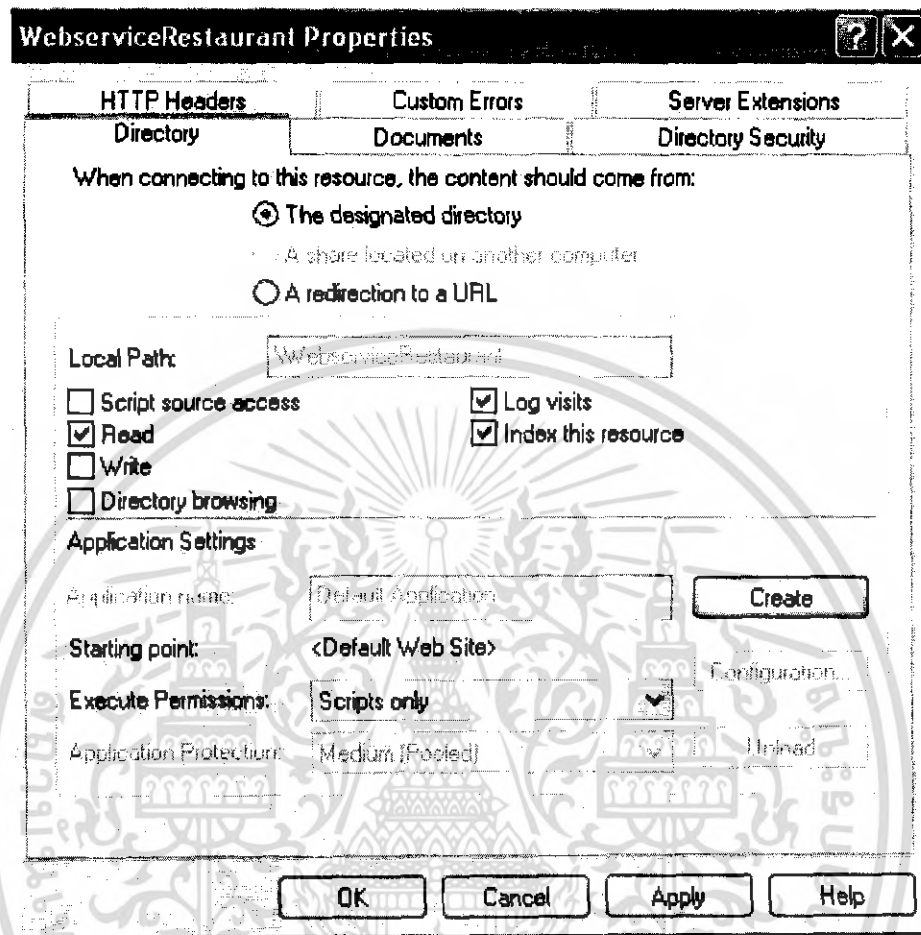
4.1 จากนั้นทำการคลิกขวาที่โฟลเดอร์ของเว็บเซอร์วิสในที่นี้ชื่อ “WebserviceRestaurant” เพื่อทำการ Create Service



รูปที่ ก-10 ภาพแสดงขั้นตอนการเปิดบริการของเว็บเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 กดปุ่ม Create เพื่อสร้างเซอวีต

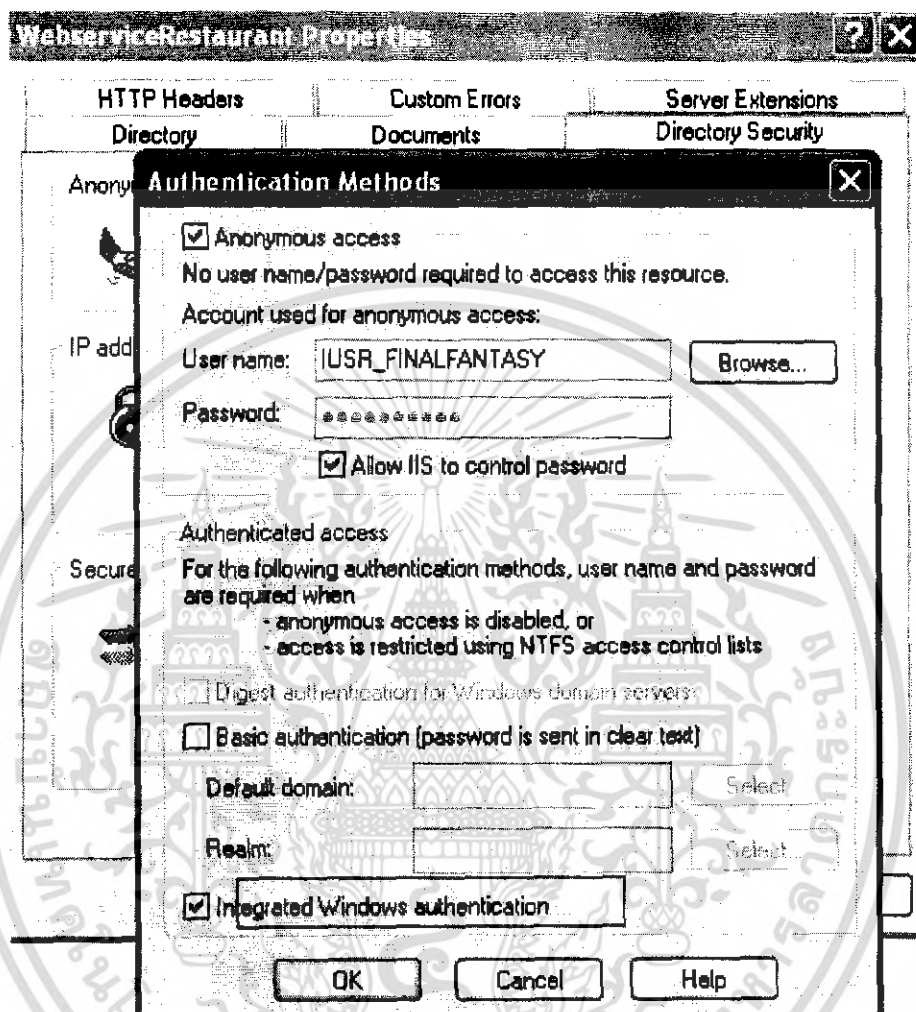


รูปที่ ก-11 ภาพแสดงขั้นตอนเปิดบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 จากนั้นเลือก Tab > Directory Security กดปุ่ม Edit เลือกเครื่องหมายถูกที่

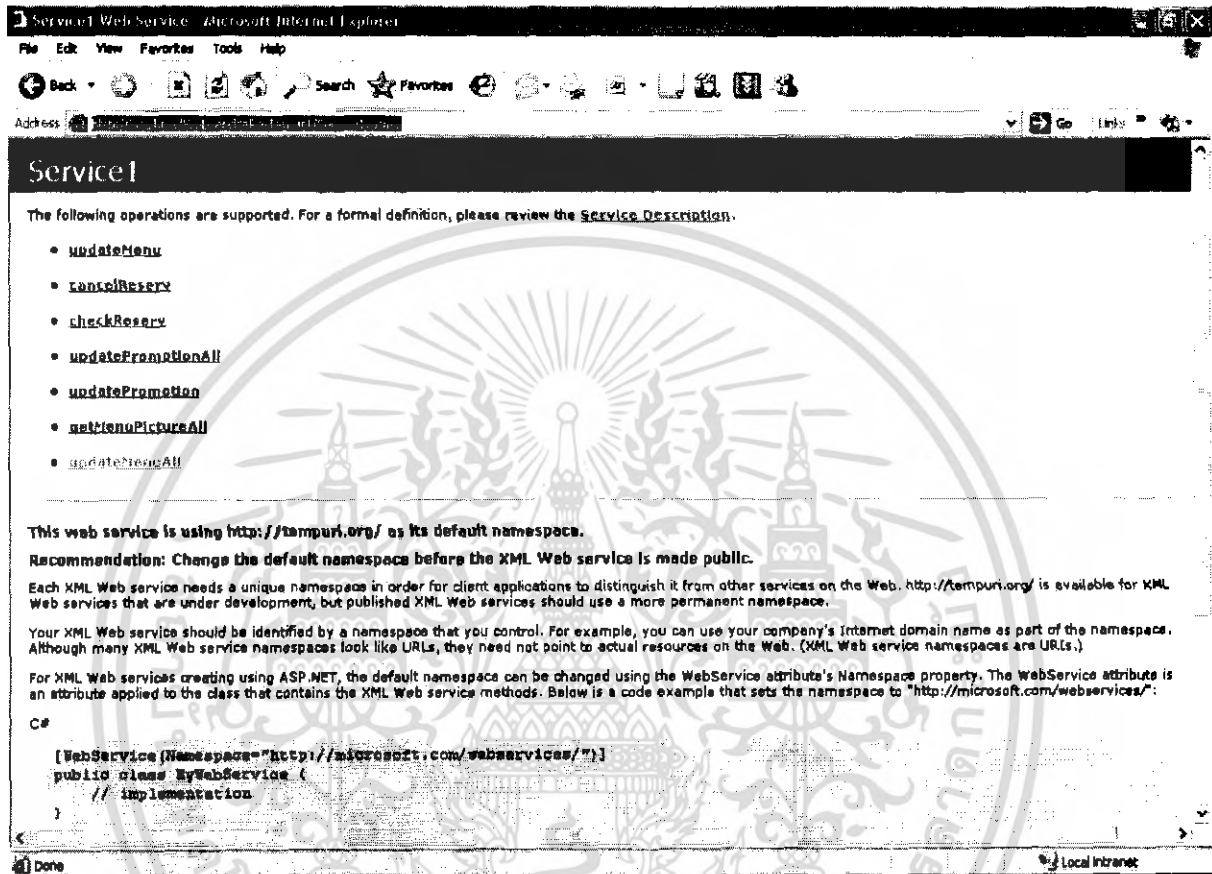
Integrated Windows authentication



รูปที่ ก-12 ภาพแสดงขั้นตอนการ Integrated Windows authentication

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จากนั้นทำการเปิด Web Browser เพื่อทำการทดสอบว่าเราสามารถเปิดเว็บเซอร์วิสของเราได้หรือยัง โดยการกรอก URL เป็น <http://localhost/WebserviceRestaurant/Service1.aspx> (ในส่วนที่ขีดเส้นใต้ นั่นคือ โพลเดอร์และชื่อไฟล์ของเซอร์วิส) ถ้าสามารถเปิดบริการได้จะปรากฏดังรูป



รูปที่ ก-13 ภาพแสดงการรันเว็บเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แนวทางและการแก้ปัญหาเมื่อไม่สามารถรันเว็บเซอร์วิสได้

- ปัญหาที่เกิดจาก Microsoft .NET Framework แนวทางในการแก้ปัญหาคือให้ทำการ Restart เครื่องถ้าไม่หายให้ทำการ Remove Microsoft .NET Framework แล้วลงใหม่



```

Server Error in '/WebserviceRestaurant' Application.

Configuration Error

Description: An error occurred during the processing of a configuration file required to service this request. Please review the specific error details below and modify your configuration file appropriately.

Server Error Message: Access is denied 'WebserviceRestaurant'

Source Error:

Line 256:         <add assembly="System.EnterpriseServices, Version=1.0.5000.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"/>
Line 257:         <add assembly="System.Web.Mobile, Version=1.0.5000.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"/>
Line 258:         <add assembly="" />
Line 259:     </assemblies>
Line 260:

Source File: C:\windows\microsoft.net\Framework\v1.1.4322\Config\machine.config     Line: 256

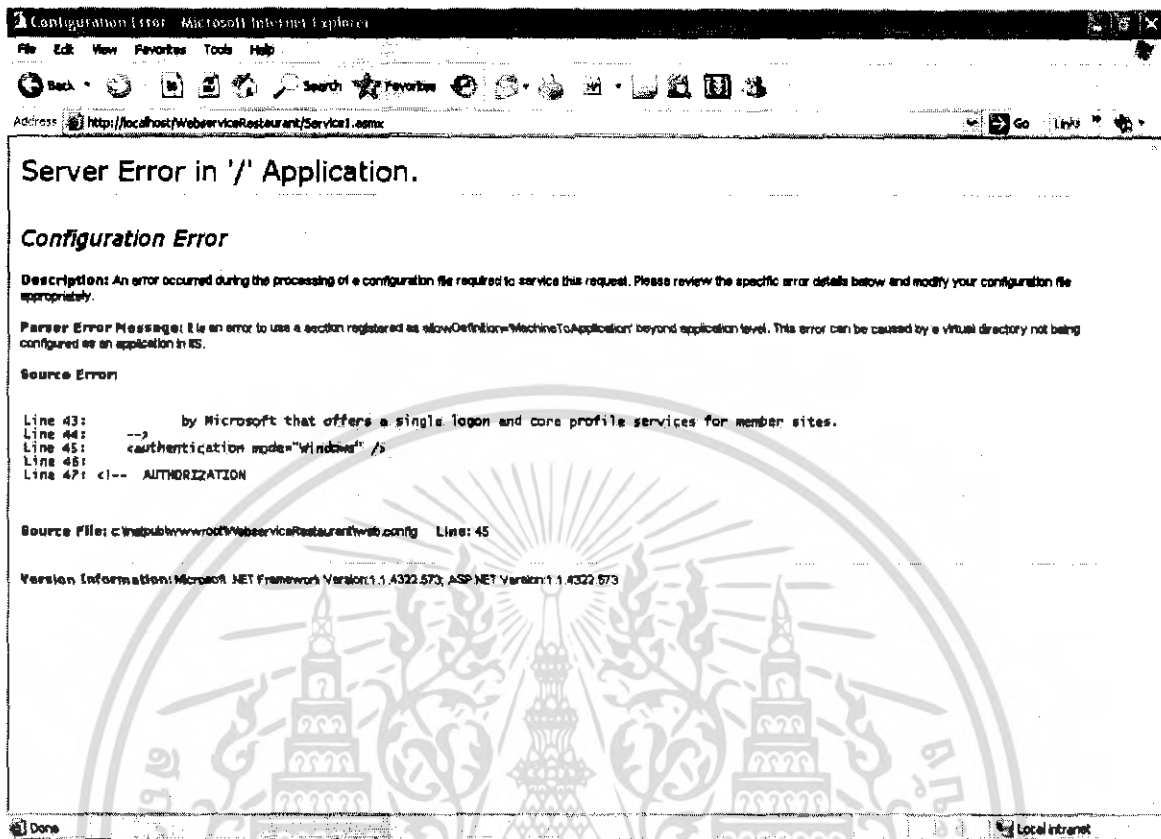
Assembly Load Trace: The following information can be helpful to determine why the assembly 'WebserviceRestaurant' could not be loaded.

--- Pre-Load state information ---
LOG: DisplayName = WebserviceRestaurant
(Partial)
LOG: AppBase = file:///c:/inetpub/wwwroot/WebserviceRestaurant/
LOG: Initial PrivatePath = bin
Calling assembly : (Unknown).
  
```

รูปที่ ก-14 ภาพแสดง Error ที่เกิดจาก Microsoft .NET Framework

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปัญหาที่เกิดจากการ ไม่เปิดบริการของเว็บเซอร์วิส แก้ไขโดยทำการเปิดเซอร์วิส (ดูขั้นตอนที่ 4)



### รูปที่ ก-15 ภาพแสดง Error ที่เกิดจากการไม่เปิดบริการของ IIS

- ปัญหาเครื่องผู้ให้บริการ ไม่สามารถเรียกเซอร์วิสผ่านเครือข่ายได้ แก้ไขปัญหาโดยการเพิ่มโหนดส่วนนี้ลงในไฟล์ Web.config

```
<system.web>
  <webServices>
    <protocols>
      <add name="HttpGet"/>
      <add name="HttpPost"/>
    </protocols>
  </webServices>
</system.web>
```

### รูปที่ ก-16 ภาพแสดงโหนดที่เพิ่มเข้าไปในไฟล์ Web.config

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้