

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

อินเตอร์เฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้าด้วยคอทเน็ตเฟรมเวิร์ค  
INTERFACE FOR POWER QUALITY METERS USING .NET FRAMEWORK



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อินเทอร์เน็ตเฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้าด้วย .NET Framework

Interface for Power Quality Meters using .NET Framework

ผู้จัดทำ

1. นายเฉลิมวัฒน์ เห่งจำชมพู รหัสนักศึกษา 48015331



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# อินเทอร์เน็ตรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้าด้วย .NET Framework

นายเฉลิมวัฒน์ เห่งามพ 48015331

ผศ.ดร. สุรินทร์ กิตติธรรมกุล อาจารย์ที่ปรึกษา

ปีการศึกษา 2550

## บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้นำเสนอการพัฒนาระบบการอินเทอร์เน็ตรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้าพัฒนาบน .NET Framework กำหนดให้ระบบทำงานบน Protocol TCP โดยแบ่งออกเป็นสามส่วนดังนี้ ส่วนแรกเป็นส่วนของ Server ทำหน้าที่บริการการเชื่อมต่อกับมิเตอร์ลูกข่าย และมีกระบวนการระบุตัวตนเพื่อให้ทราบจำนวน ตำแหน่งและข้อมูลเฉพาะของมิเตอร์ ส่วนที่สองเป็นส่วนของมิเตอร์วัดคุณภาพไฟฟ้าจะมีการร้องขอเพื่อเชื่อมต่อไปยัง Server ทุกครั้งที่มีการเปลี่ยนแปลงของไฟฟ้าบริเวณที่มีมิเตอร์ติดตั้งอยู่ เช่น ไฟฟ้าเกินตัก หรือดับ จะร้องขอไปยัง Server เพื่อส่งข้อมูลไปเก็บไว้ในฐานข้อมูล ในส่วนสุดท้าย เป็นส่วนสำหรับบริการเว็บแอปพลิเคชัน เพื่อเรียกดูข้อมูลของมิเตอร์ที่อัปเดตข้อมูลล่าสุด ระบบนี้ช่วยในการเก็บข้อมูลโดยไม่ต้องใช้มนุษย์เก็บข้อมูลที่มีมิเตอร์โดยตรง ซึ่งสะดวกรวดเร็วกว่าและสามารถนำไปใช้ประโยชน์ในการสร้างรายงานได้ง่ายขึ้น

เทคโนโลยีที่ถูกนำมาใช้เป็นฟอร์แมตกลางในการสื่อสารคือ XML (Extensible Markup Language) ซึ่งง่ายต่อการทำความเข้าใจและง่ายต่อการพัฒนาระบบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Interface for Power Quality Meters using .NET Framework

Mr. Chaleamwat Gnowchompoo 48015331

Asst. Prof. Dr. Surin Kittitornkun Advisor

Academic Year 2007

## ABSTRACT

This project study and develop interface for power quality meters using .NET Framework. Protocol TCP is used in this system. This system has 3 modules. The first is the server module. The functions of this module are connections to the clients and identify each meters. Next, module of power quality meters. The clients will send request to the server for create the connections for update database at the server when the power is changed. Finally, the latest data of each meter is showed in the web application service module. The advantages of this system are easy to store power quality meter data and easily make the report.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ขาดเสียมิได้ในความสำเร็จดังกล่าวนี้ คือ อาจารย์ สุรินทร์ กิตติธรรมกุล อาจารย์ที่ปรึกษาปริญญาบัตรที่ไม่เพียงแต่สละเวลาเท่านั้น แต่ยังสละร่างกาย และแรงใจในการเอาใจใส่ดูแลแนะนำและช่วยเหลือเสมอมาจนปริญญาบัตรฉบับนี้ได้สำเร็จ ลุล่วงด้วยดี อีกทั้งคณาจารย์ทุกท่านที่ถ่ายทอดวิชาความรู้มาให้ ซึ่งในความกรุณาของอาจารย์ที่มี ต้องขอบพระคุณอย่างสูง

และต้องขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้มีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่งซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆ ด้าน อันที่หาที่เปรียบมิได้ ข้าพเจ้าขอรำลึกในพระคุณ และขอกราบขอบพระคุณมา ณ ที่นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 วิธีการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีและเทคโนโลยีพื้นฐาน.....	3
2.1 มิเตอร์วัดคุณภาพไฟฟ้า.....	3
2.1.1 หลักการทำงาน.....	4
2.2 มาตรฐานของไฟล์ข้อมูล.....	5
2.2.1 มาตรฐาน IEEE 1159.3 Power Quality Data Interchange Format (File.PQIDF).....	6
2.3 ภาษาC#.....	8
2.4 ASP.NET.....	8
2.4.1 จุดเด่นของ ASP.NET.....	9
2.4.2 ส่วนประกอบของเว็บไซต์ใน ASP.NET.....	10
2.4.3 วงจรชีวิตของเว็บฟอร์ม.....	11
บทที่ 3 การออกแบบและพัฒนาระบบ.....	15
3.1 การออกแบบตารางฐานข้อมูล.....	16
3.2 Protocol ที่ใช้ในการเชื่อมต่อ.....	17
3.3 ขั้นตอนการทำงานของระบบการอินเตอร์เฟซของมิเตอร์คุณภาพไฟฟ้า.....	19
3.4 ระบบมิเตอร์จำลอง.....	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
3.5 ระบบแสดงมิเตอร์ที่อัปเดตข้อมูลล่าสุดตามจำนวนที่กำหนดผ่านเว็บ.....	21
3.6 การสร้างและการลบมิเตอร์ออกจากระบบ.....	22
3.7 ฟอร์มแม่ตไฟล์ที่ใช้ในการสื่อสารของระบบ.....	24
บทที่ 4 ผลการทดลอง.....	38
4.1 การใช้งาน.....	38
4.2 ผลการทดลองใช้งาน Meter Simulator.....	42
บทที่ 5 บทวิจารณ์และสรุป.....	45
5.1 บทวิจารณ์และสรุป.....	45
5.2 ปัญหาที่เกิดขึ้นในการพัฒนาและแนวทางการแก้ปัญหา.....	45
5.3 ข้อจำกัดของระบบ.....	45
5.4 แนวทางการพัฒนา.....	46
บรรณานุกรม.....	47
ภาคผนวก ก. การติดตั้ง IIS (Internet Information Server).....	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงชนิดของไฟล์ทั้งหมดใน .NET Framework 1.X.....	11
3.1 แสดงตาราง SagSwellTable ในฐานข้อมูลของระบบ.....	17



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

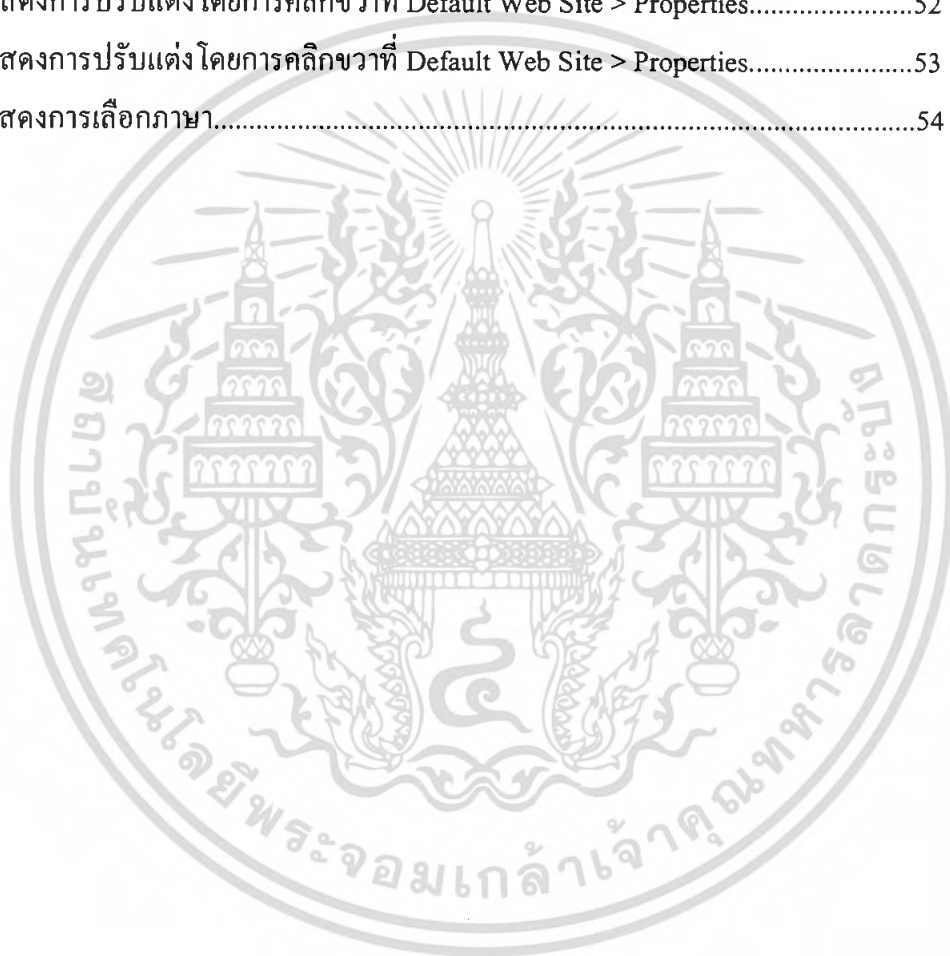
## สารบัญรูป

รูปที่	หน้า
2.1 ภาพตัวอย่างมิเตอร์วัดคุณภาพไฟฟ้า .....	4
2.2 ตัวอย่างภาพมิเตอร์วัดคุณภาพไฟฟ้าที่ใช้งานจริง.....	4
2.3 ภาพการทำงานเมื่อมิเตอร์เชื่อมต่อกับคอมพิวเตอร์.....	5
2.4 The power quality concept.....	5
2.5 Logical Record Structure.....	6
2.6 โครงสร้างภายในไฟล์.....	7
2.7 ตัวอย่างโครงสร้างไฟล์.....	7
2.8 โครงภายในของ Web Server IIS .....	10
2.9 การจัดการกับการเรียกไปยังเว็บฟอร์มของ IIS.....	12
2.10 วงจรชีวิตของเว็บฟอร์ม.....	12
2.11 ตัวอย่างแสดงขั้นตอนการไหลค้ำจากการ Post Back.....	13
3.1 User Case Diagram.....	16
3.2 เส้นทางการเชื่อมต่อ.....	18
3.3 การทำงานของระบบมิเตอร์เมื่อจะทำการส่งข้อมูล.....	18
3.4 ขั้นตอนการสร้าง Passive Sockets .....	19
3.5 ขั้นตอนการสร้าง Active Sockets .....	19
3.6 การทำงานของระบบการส่งข้อมูล.....	20
3.7 การทำงานของเว็บ.....	21
3.8 แสดงรูปเว็บแสดงข้อมูลมิเตอร์ที่อัพเดทล่าสุด.....	22
3.9 ขั้นตอนการทำงานในการสร้างมิเตอร์ลงในระบบ.....	23
3.10 แสดงกระบวนการ Serialize และการ Deserialize.....	37
4.1 ภาพแสดง Meter Simulator ในโหมด Real time.....	38
4.2 ภาพแสดง Meter Simulator ในโหมด Logfile.....	39
4.3 แสดงลักษณะกราฟไฟตก.....	40
4.4 ตัวอย่างผลลัพธ์ของไฟล์ที่ถูกสร้างจากมิเตอร์ซิมูเลเตอร์.....	43
4.5 แสดงตัวอย่างไฟล์ซึ่งส่งมาจากมิเตอร์ซิมูเลเตอร์.....	43
4.6 แสดงผลการทำงานของระบบอินเตอร์เฟซรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้า.....	44
ก.1 แสดงเมนูสำหรับการติดตั้ง IIS ลงบนวินโดว XP.....	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป(ต่อ)

รูปที่	หน้า
ก.2 แสดงขั้นตอนการติดตั้งเมื่อระบบถามหาตัวติดตั้ง Windows XP.....	49
ก.3 แสดงการเลือกโฟลเดอร์ที่จะติดตั้ง.....	49
ก.4 แสดงช่วงเวลาในการติดตั้ง.....	50
ก.5 แสดงโครงสร้างโฟลเดอร์ภายใน IIS.....	51
ก.6 แสดงการปรับแต่งโดยการคลิกขวาที่ Default Web Site > Properties.....	52
ก.7 แสดงการปรับแต่งโดยการคลิกขวาที่ Default Web Site > Properties.....	53
ก.8 แสดงการเลือกภาษา.....	54



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่

## บทนำ

### 1.1 ความเป็นมาและความสำคัญ

เนื่องจากในปัจจุบันอัตราการใช้ไฟฟ้าในประเทศไทยนับว่าเพิ่มขึ้นอย่างรวดเร็ว ทำให้มาตรฐานการให้บริการของการไฟฟ้า แก่ผู้ใช้บริการเช่น เขตนิคมอุตสาหกรรมซึ่งมีความต้องการการใช้ไฟฟ้าที่สูงกว่าเขตที่อยู่อาศัยในชุมชนทั่วไปรวมถึงต้องการมาตรฐานของไฟฟ้าที่สูงกว่าด้วย มาตรฐานในที่นี่มีความหมายถึงคุณภาพของไฟฟ้าซึ่งเป็นเครื่องชี้วัดว่าในพื้นที่นั้นมีความคงที่เที่ยงตรงอัตราไฟตกไฟเกินหรือไฟดับมีน้อยที่สุด การที่จะแก้ปัญหาเหล่านี้ได้ต้องมีการเก็บข้อมูลและต้องทราบอย่างรวดเร็วเพื่อให้สามารถหาแนวทางในการควบคุมและป้องกันการเกิดเหตุการณ์ไฟฟ้าตกหรือ ไฟเกิน

มิเตอร์วัดคุณภาพไฟฟ้าเป็นเครื่องมือที่ใช้วัดค่าคุณสมบัติทางไฟฟ้ามันจะทำงานเก็บค่าคุณสมบัติทางไฟฟ้า เมื่อเกิดเหตุการณ์ไฟฟ้าตก ไฟเกินและไฟดับ โดยค่าที่ได้จะถูกเก็บไว้ในมิเตอร์เป็นไฟล์ซึ่ง มิเตอร์ชนิดนี้สามารถส่งข้อมูลออกมาผ่านโมเด็มได้ หรือจะใช้คนไปเก็บที่ตัวมิเตอร์ก็ได้ ดังนั้นเราจึงพัฒนาระบบนี้ขึ้นมาเพื่อรับข้อมูลของมิเตอร์ผ่านอินเทอร์เน็ตมาเก็บไว้ที่ฐานข้อมูลเพื่อสามารถนำข้อมูลนี้มาใช้ประโยชน์ต่อไป

### 1.2 วัตถุประสงค์ของปริญญาานิพนธ์

1. ช่วยในการควบคุมคุณภาพทางไฟฟ้าในสถานที่ ที่ต้องการได้
2. สะดวกในการเรียกดูข้อมูลของมิเตอร์ได้ทุกที่ถ้าหากมีคอมพิวเตอร์ที่ได้ทำการเชื่อมต่ออินเทอร์เน็ตอยู่
3. เพื่อสามารถเก็บสถิติของข้อมูลไฟฟ้าตามแหล่งที่เราต้องการได้
4. ลดค่าใช้จ่ายในการใช้เจ้าหน้าที่เพื่อเก็บข้อมูลของมิเตอร์วัดคุณภาพไฟฟ้า
5. เก็บข้อมูลของมิเตอร์วัดคุณภาพไฟฟ้าได้อย่างรวดเร็ว

### 1.3 ขอบเขตของปริญญาานิพนธ์

1. สามารถค้นหาและแสดงผลมิเตอร์วัดคุณภาพไฟฟ้าผ่านเว็บได้ตลอดเวลา
2. สามารถเก็บข้อมูลของมิเตอร์ได้ตลอดเวลา
3. แสดงมิเตอร์ที่อัพเดทข้อมูลล่าสุดได้
4. มิเตอร์ที่สามารถส่งข้อมูลมาเก็บได้ต้องเป็นสมาชิกของระบบเท่านั้น
5. สามารถลงทะเบียนเพิ่มมิเตอร์ในการเก็บข้อมูลได้ของระบบได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ระบบสามารถบันทึกข้อมูลที่ได้รับจากมิเตอร์ฐานข้อมูลได้ถูกต้อง

#### 1.4 วิธีการดำเนินงาน

1. ค้นหาหาข้อมูลที่จะใช้ในการทำงานและระบบที่มีการใช้งานจริงอยู่แล้ว
2. นำข้อมูลที่ได้มาวิเคราะห์หาความต้องการของระบบ
3. ออกแบบระบบงานและจัดหาเครื่องมือที่เหมาะสมกับงาน
4. พัฒนาโปรแกรม
5. นำโปรแกรมมาทดลองใช้กับข้อมูลจริง
6. หาข้อผิดพลาดเพื่อปรับปรุงแก้ไขให้ถูกต้อง
7. จัดทำเอกสารประกอบ
8. สรุปผลและนำเสนอโครงการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและเทคโนโลยีพื้นฐาน

การทำงานของระบบ อินเทอร์เน็ตกับมิเตอร์วัดคุณภาพไฟฟ้านั้น จะทำงานในลักษณะ Server-Client จึงแบ่งเป็น 2 ส่วน ส่วนของServer ทำหน้าที่รองรับการเชื่อมต่อจาก Client ซึ่งในที่นี้ จะใช้ Protocol Tcp ในการเชื่อมต่อ แล้ว Client ซึ่งเป็นมิเตอร์วัดคุณภาพไฟฟ้าทำการส่งรหัสซึ่งเป็นค่าของคุณลักษณะทางไฟฟ้า เข้าไปเก็บลงฐานข้อมูล ฐานข้อมูลที่ใช้คือ MSSQLServer โดยเครื่องมือที่ใช้ในการพัฒนาคือ ภาษา C# .NET(C Shap.NET) ซึ่งใช้พัฒนาทั้งสองส่วน

### 2.1 มิเตอร์วัดคุณภาพไฟฟ้า

ในปัจจุบันการใช้ไฟฟ้าในประเทศไทยของเรานั้นนับว่าเป็นเรื่องที่ขาดไม่ได้เลย ไม่ว่าจะ เป็นภาคธุรกิจอุตสาหกรรมต่างๆล้วนมีปริมาณการใช้ไฟฟ้าที่มากขึ้นเรื่อยๆ รวมทั้งใน การทำงาน ของคนทุกคน ปัจจุบันล้วนเกี่ยวกับการใช้ไฟฟ้าไม่ว่าในทางตรงหรือทางอ้อม และอาจจะมีบาง สถานที่ ที่มีความจำเป็นต้องใช้ไฟฟ้าตลอดเวลา เช่น โรงพยาบาล โรงงานอุตสาหกรรมที่ต้องทำงาน ตลอดเวลา เขตอุตสาหกรรม สนามบิน สถานที่เหล่านี้มีความจำเป็นมากที่จะต้องได้รับไฟฟ้าที่มี คุณภาพดี

คุณภาพในการใช้ไฟฟ้านั้นเราอาจมีเกณฑ์ในการใช้วัดจากความถี่ของการเกิดเหตุการณ์ ผิดปกติซึ่งทำให้ไฟฟ้าไม่สามารถจะจ่ายให้กับผู้ใช้ได้อย่างถูกต้องตามความต้องการเหตุการณ์ เหล่านี้สามารถตรวจวัดได้จากอุปกรณ์ซึ่งเรียกว่า “มิเตอร์วัดคุณภาพไฟฟ้า” มิเตอร์ชนิดนี้ทำงาน ต่างจากมิเตอร์ที่ใช้คิดค่าบริการที่ติดตามเสาไฟฟ้า มิเตอร์ชนิดนี้จะใช้วัดค่าที่เกิดขึ้นจากการใช้ ไฟฟ้าของของผู้ใช้ โดยมาตรฐานค่าที่มิเตอร์จะวัดออกมาจะประกอบด้วยค่าเหล่านี้

- Voltage (phase-phase, phase-neutral)
- Current (phase-phase, phase-neutral)
- Active, Reactive and Apparent Power
- Active, Reactive and Apparent Energy
- Power Factor and  $\cos \varphi$
- Frequency



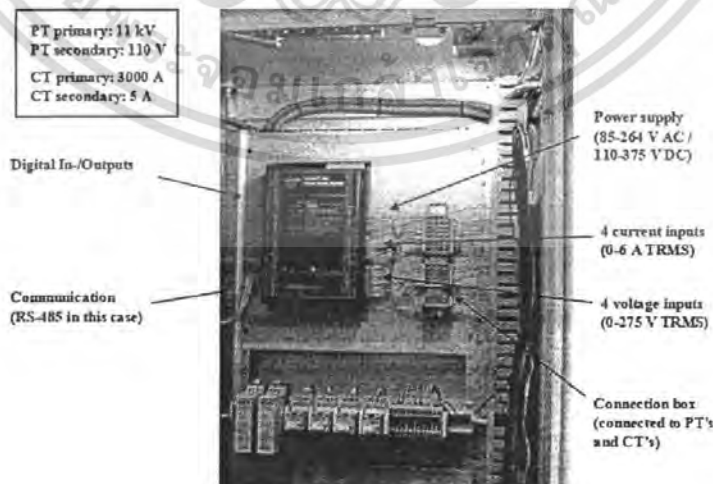
รูปที่ 2.1 ภาพตัวอย่างมิเตอร์วัดคุณภาพไฟฟ้า

### 2.1.1 หลักการทำงาน

มิเตอร์วัดคุณภาพไฟฟ้าจะถูกติดตั้งอยู่ตามนิคมอุตสาหกรรมหรือสถานที่ซึ่งจะต้องมีการวัดคุณภาพไฟฟ้าอย่างสม่ำเสมอ โดยค่าที่มิเตอร์วัดคุณภาพไฟฟ้านั้นที่กั้นกำหนดให้บันทึกตามเหตุการณ์ที่เกิดขึ้นก็ได้ หรือกำหนดให้บันทึกทุกวินาที หรือจะให้บันทึกตลอดเวลา

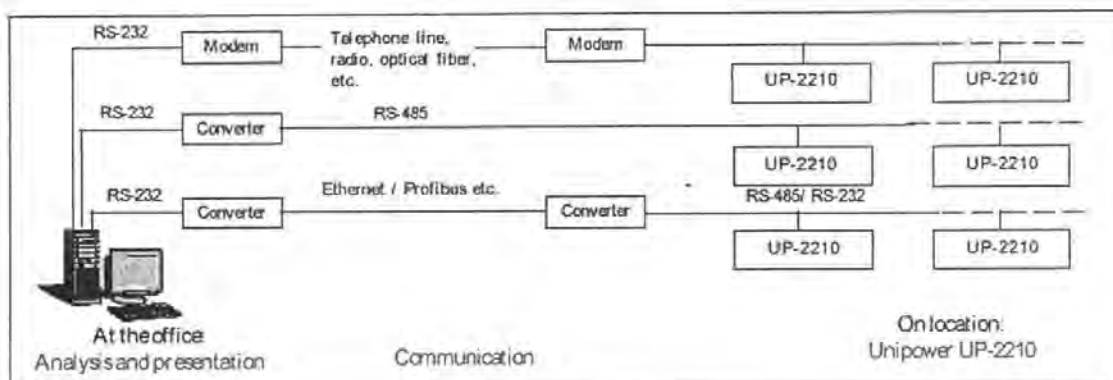
เมื่อมิเตอร์ทำการบันทึกค่า ค่าเหล่านั้นจะถูกบันทึกลงในหน่วยความจำภายในของมิเตอร์ ซึ่งจากมิเตอร์ที่อ้างอิงตัวอย่าง มีหน่วยความจำขนาด 4 MB เก็บได้ประมาณ 20 วัน และสามารถเชื่อมต่อตัวมิเตอร์เข้ากับอินเตอร์เน็ตเพื่อทำการส่งค่าที่ทำการบันทึกมาที่เครื่องคอมพิวเตอร์ ผ่านพอร์ต RS-232, RS-485 เข้ากับ Modem

#### Typical Substation Application



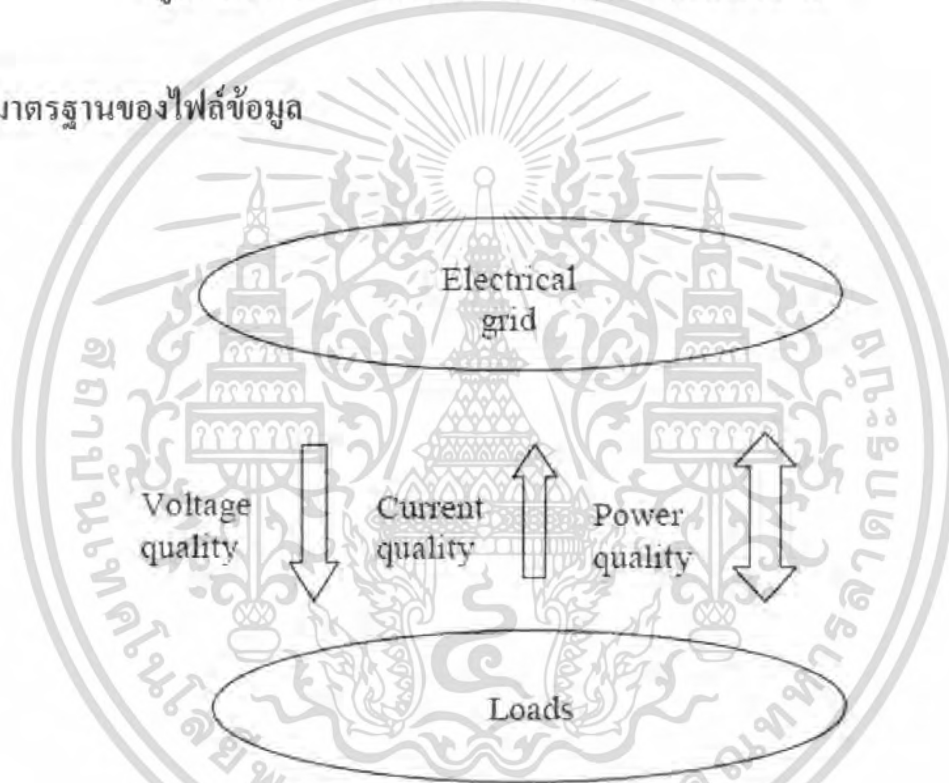
รูปที่ 2.2 ตัวอย่างภาพมิเตอร์วัดคุณภาพไฟฟ้าที่ใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ภาพการทำงานเมื่อมิเตอร์เชื่อมต่อกับคอมพิวเตอร์

## 2.2 มาตรฐานของไฟล์ข้อมูล



รูปที่ 2.4 The power quality concept

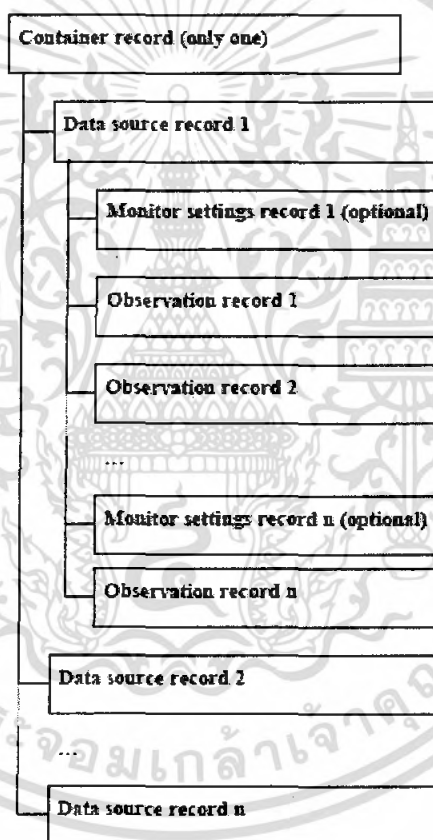
หลังจากมิเตอร์คุณภาพไฟฟ้าบันทึกค่าซึ่งเป็นคุณลักษณะทางไฟฟ้ามาแล้ว ไฟล์ที่มิเตอร์วัดคุณภาพไฟฟ้าบันทึกมานี้จะมีนามสกุลเป็น .PF4 เช่น BNL1-19-010903.PF4 (ไฟล์ตัวอย่างซึ่งได้จาก การไฟฟ้าส่วนภูมิภาค เดินทางไปเก็บจากมิเตอร์โดยตรง) ซึ่งชื่อเริ่มต้นของไฟล์นี้จะเป็นการระบุถึงแหล่งที่มา ต่อมาเป็นวันเวลาในการบันทึกไฟล์นี้

ไฟล์เหล่านี้จะต้องนำมาอ่านแล้วแปลงค่าต่างๆลงเก็บในฐานข้อมูลของ Server ซึ่งค่าที่อยู่ในภายในจะประกอบด้วยค่าคุณลักษณะต่างๆทางไฟฟ้า การที่จะสามารถอ่านไฟล์นี้ได้เราต้องทราบมาตรฐานของฟอร์แมตที่ถูกต้อง ซึ่งไฟล์ตัวอย่างที่นำมาศึกษานี้ เป็นไฟล์ที่ถูกบันทึกจากมิเตอร์วัดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณภาพไฟฟ้าของบริษัท Uni Power ซึ่งฟอร์แมตของไฟล์นี้เป็นมาตรฐานซึ่งเป็นลิขสิทธิ์เฉพาะของบริษัทนี้เท่านั้นจึงไม่มีเอกสารที่อธิบายถึงฟอร์แมตของไฟล์ใดๆทั้งสิ้น จากการศึกษาได้พยายามนำมาถอดรหัสด้วยวิธีต่างๆเพื่อหาฟอร์แมตที่ถูกต้อง แต่ไม่สามารถทำได้สำเร็จ ต่อมาจึงค้นหามาตรฐานซึ่งคาดว่าน่าจะเป็นมาตรฐานของไฟล์นี้ จากการค้นคว้าดังนี้

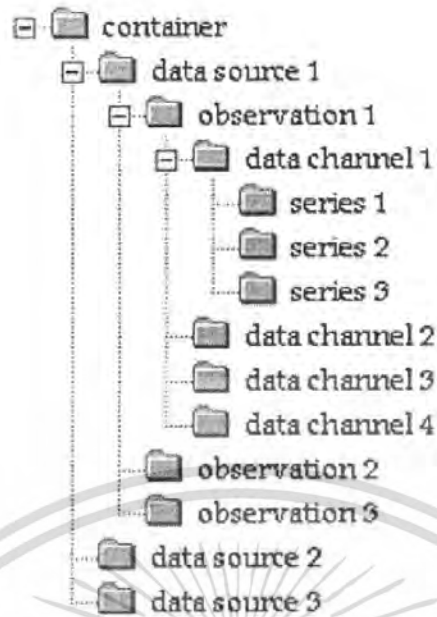
### 2.2.1 มาตรฐาน IEEE 1159.3 Power Quality Data Interchange Format (File.PQIDF)

มาตรฐานนี้กล่าวถึงการส่งข้อมูลของมิเตอร์วัดคุณภาพไฟฟ้า รวมถึง ค่าที่เก็บอยู่ในไฟล์ว่าประกอบด้วยพิวส์ใดบ้างเช่น เวลาที่สร้าง มิเตอร์ที่สร้างชื่ออะไร ฯลฯ ค่าที่บรรจุอยู่ในรูปแบบใด จากสามรูปแบบคือ Scalar ,Vector ,Collection รูปแบบการเก็บมีลักษณะของ Tree หรือโครงสร้างข้อมูลแบบ Link ดังรูป เป็นโครงสร้างทางลอจิกของไฟล์



รูปที่ 2.5 Logical Record Structure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 โครงสร้างภายในไฟล์

Tag name	Element type	Physical type	Contents
tagSeriesValues	Vector	Double-precision real (8-byte)	Count: 1024
			0: 0
			1: 176.896845388036
			2: 353.287255203502
			3: 529.66485798664
			4: 705.723410464688
			5: 881.85686154886
			6: 1056.45941621555
			7: 1230.92559923331
			8: 1404.65031869717
			9: 1577.52892933208
			10: 1749.45729552733
			11: 1920.33185406389

รูปที่ 2.7 ตัวอย่างโครงสร้างไฟล์

จากการทดลองทำให้ทราบว่าไฟล์นามสกุล .PF4 นั้นไม่สามารถใช้ มาตรฐานเดียวกับ IEEE 1159.3 อ่านได้ การแก้ปัญหาในส่วนของฟอร์แมตนี้เราได้ทำการจำลองไฟล์ในรูปแบบ โครงสร้าง XML ไฟล์มีโครงสร้างใกล้เคียงกับไฟล์ .PF4 มากที่สุดโดยสร้างเป็น XML ไฟล์ ที่บรรจุ ค่าเหล่านี้ เพื่อให้ง่ายต่อการอ่านเพื่อบันทึกลงฐานข้อมูลอีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ภาษา C#

ภาษา C# จะเขียนว่า C Sharp ก็ได้ (อ่านว่าซีชาร์พ) เครื่องหมาย # ในทางคณิตศาสตร์หมายถึง เครื่องเสียง ส่วนคำว่า Sharp หมายถึงเฉียบคม มีไหวพริบ บริษัทไมโครซอฟท์ตั้งชื่อภาษานี้ว่า C# เพื่อจะสื่อให้เห็นว่านี่คือภาษาในตระกูลภาษา C เช่นเดียวกับภาษา C++ คือเป็นภาษาที่มีต้นกำเนิดจากภาษา C หากท่านสังเกตให้ดีจะเห็นว่าสัญลักษณ์ # จะมองให้เป็นเครื่องหมาย + สื่ออันนำมาซ้อนกันก็ได้ แสดงเป็นนัยว่าภาษานี้ก้าวหน้ากว่าภาษา C++ ไปอีกระดับหนึ่ง (คือเป็นภาษา C++++) สัญลักษณ์ # นี้มีปรากฏในภาษาอื่นๆ ของ .NET ด้วย เช่นภาษา J#(เดิมคือภาษา J++) และภาษา A# (เหมือนภาษา Ada) อาจเป็นเพื่อเหตุผลการการค้า

ภาษา C# เป็นภาษาที่ใหม่มาๆ ปรากฏตัวเป็นครั้งแรกในปี 2000 และถูกอัปเดตเวอร์ชัน อยู่เสมอ มันเป็นภาษาที่ได้รับอิทธิพลจากภาษาก่อนหน้าเช่นภาษา Delphi ภาษา C++ ภาษา Java และ ภาษา Eiffel ในตอนต้นภาษานี้ถูกออกแบบ และกำหนดลักษณะโดยบริษัทไมโครซอฟท์ ต่อมาได้ถูกรับรองจากหน่วยงาน ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO แต่ปัจจุบันไมโครซอฟท์ยังพัฒนาภาษานี้อย่างต่อเนื่อง (ปัจจุบันเป็นเวอร์ชัน 3.0)

ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi ภาษา C++) มาปรับปรุงเพื่อให้ความเป็น OOP อย่างถึงที่สุด ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีเครื่องแต่งตัวน้อยลง (เมื่อเทียบกับ Java)

เหตุที่เลือกใช้ภาษานี้เพราะ เป็นความถนัดภาษาที่เป็นตระกูล C และเนื่องจากเดิมที่เคยเขียนภาษา java มาก่อนทำให้ง่ายที่จะปรับเปลี่ยนมาเขียน ภาษา C# และระบบนี้มีความจำเป็นที่ต้องมีความเข้ากันได้กับระบบเดิมซึ่งเป็นฐานข้อมูล MSSQLServer ซึ่งเป็นฐานข้อมูลที่สามารถทำงานกับภาษาในตระกูลวิซวลได้ดี และระบบเว็บซึ่งโปรแกรมส่วนนี้ต้องไปเชื่อมต่อได้พัฒนาด้วย .NET Framework เช่นกัน

## 2.4 ASP.NET

ASP.NET คือ ภาษาสคริปต์ตัวหนึ่งพัฒนาโดยไมโครซอฟท์ที่ใช้พัฒนาเว็บแอปพลิเคชัน โดยพัฒนาต่อเนื่องจาก ASP 3.0 เดิม หรือที่เรียกว่า Classic ASP หลังจากที่มีการใช้งานมานาน 5 ถึง 6 ปี ปัจจุบันได้มีการพัฒนา ASP เวอร์ชันใหม่ขึ้นมาผสานกับเทคโนโลยีใหม่ที่เรียกว่า .NET Framework โดยได้เพิ่มเติมส่วนของการติดต่อแบบ XML ผ่านโปรโตคอล SOAP, USDL, UDDI หรือการพัฒนาติดต่อสื่อสารแบ่งปันข้อมูลผ่านอินเทอร์เน็ต โดยใช้งานผ่านระบบปฏิบัติการและดีไวซ์ต่างๆ และให้ชื่อใหม่ว่า ASP.NET ซึ่งมีโครงสร้างเดิมของภาษา Classic ASP เพียงเล็กน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.1 จุดเด่นของ ASP.NET

จากการกล่าวของไมโครซอฟต์นั้น ASP.NET จะช่วยเพิ่มประสิทธิภาพในการทำงานกับเว็บเซิร์ฟเวอร์เนื่องจากได้พัฒนาระบบแคชที่ใช้ทำงานขึ้นมาใหม่ซึ่งจะทำให้ทำงานได้เร็วขึ้น รวมทั้งเพิ่มเครื่องมือในการตรวจสอบจัดการโปรเซสต่างๆ ได้อย่างใกล้ชิดขึ้น และสำหรับนักพัฒนานั้นช่วยการทำงานในการสร้างโค้ดให้สั้นและก็เร็วขึ้นด้วย แทนที่จะแยกโค้ด HTML ออกจากโค้ดโปรแกรมโดยมีการผนึกรวมกันโดยเรียกว่าเซิร์ฟเวอร์คอนโทรล (Server Control)

ประโยชน์ของ .NET Framework คืออะไร ความง่าย ใช้เวลาพัฒนาน้อยลง นำไปใช้และบริหารง่ายขึ้น และมีการปรับปรุงประสิทธิภาพให้ดีขึ้น

#### 1. สามารถใช้ภาษาโปรแกรมใดก็ได้

.NET Framework เปิดโอกาสให้นักพัฒนาใช้ภาษาโปรแกรมใดพัฒาก็ได้ และใช้กับแอปพลิเคชันที่เขียนขึ้นด้วยภาษาโปรแกรมใดๆ เพื่อรวมเข้าด้วยกันอย่างสมบูรณ์

#### 2. เขียนโปรแกรมน้อยลง

.NET Framework นั้นเน้นการออกแบบที่ใช้คอมโพเนนต์อย่างมาก นักพัฒนาจึงให้ความสำคัญกับการเขียน business logic ได้อย่างเต็มที่ โดยไม่จำเป็นต้องเขียนคำสั่งของ IDL หรือรีจิสตรี

#### 3. มี XML/SOAP เป็นแกนกลาง

.NET Framework ถูกสร้างขึ้นเพื่อการนำเสนอซอฟต์แวร์ในรูปแบบของบริการ จึงถูกสร้างขึ้นมาบนมาตรฐานร่วมกันระหว่าง XML และ SOAP ดังนั้น ด้วยการเพิ่มการเรียกเมธอดเข้าไป .NET Framework ก็สามารรถแปลงให้กลายเป็นบริการ XML Web ได้อย่างเต็มรูปแบบ

#### 4. รันแอปพลิเคชันที่น่าเชื่อถือได้มากขึ้น

.NET Framework ได้รวมเทคโนโลยีเพื่อให้แอปพลิเคชันมีความน่าเชื่อถือได้มากยิ่งขึ้น ตัวอย่างเช่น หน่วยความจำแคช และโปรเซส จะถูกจัดการโดย .NET Framework เพื่อให้แน่ใจได้ว่า จะไม่มีเหตุการณ์การรั่วไหลของหน่วยความจำเกิดขึ้น ASP.NET จะติดตามการทำงานของเว็บแอปพลิเคชันที่ทำงานอยู่ และสามารถเริ่มทำงานแอปพลิเคชันขึ้นใหม่ได้โดยอัตโนมัติภายในช่วงเวลาที่ยูสเซอร์ระบบกำหนดไว้

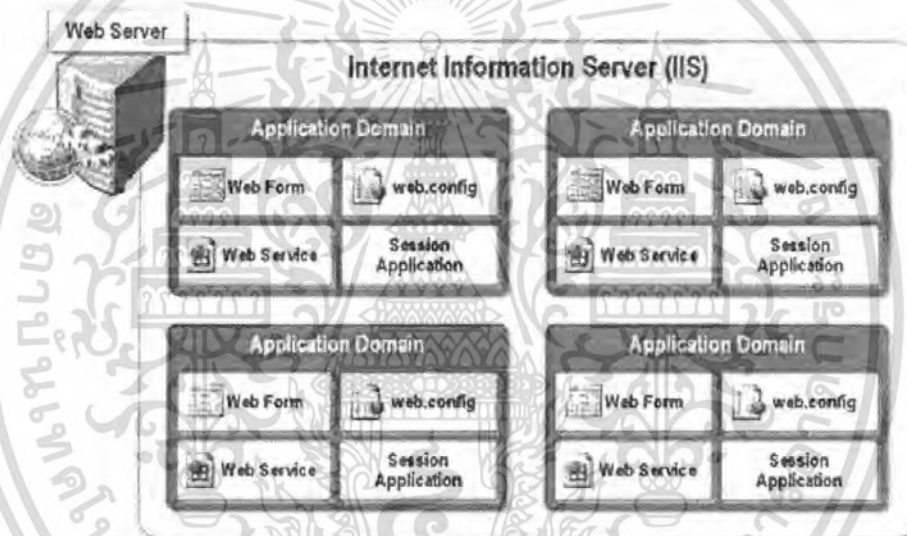
#### 5. ประสิทธิภาพที่ดีขึ้น

.NET Framework จะช่วยปรับปรุงประสิทธิภาพของเว็บแอปพลิเคชันให้ดียิ่งขึ้น โดย ASP.NET ได้รวมคุณสมบัติในการคอมไพล์และการทำแคชระดับสูงไว้ ซึ่งช่วยปรับปรุงประสิทธิภาพขึ้น 2-3 เท่าเมื่อเปรียบเทียบกับแอปพลิเคชัน Active Server Pages เดิมที่มีอยู่

## 2.4.2 ส่วนประกอบของเว็บไซต์ใน ASP.NET

แอปพลิเคชันโดเมน ในเว็บไซต์แต่ละเว็บไซต์ ไม่ว่าจะเป็นเว็บไซต์ที่รันบน ASP .NET 1.x หรือ 2.0 จะประกอบด้วยไฟล์หลายๆ ไฟล์ ที่ทำงานภายใต้หน่วยความจำกลุ่มเดียวกัน และทำงานภายใต้ค่าคอนฟิก (Configuration Setting) เดียวกันที่เรียกว่า แอปพลิเคชันโดเมน

ในแต่ละเว็บไซต์ จะมีแอปพลิเคชันโดเมนเป็นของตัวเอง แต่ละแอปพลิเคชันโดเมนจะไม่สามารถมองเห็นข้อมูลของแอปพลิเคชันโดเมนอื่นได้ เช่น ค่าของตัวแปรเซสชัน หรือตัวแปรแอปพลิเคชัน แม้ว่าจะมีหลายๆ แอปพลิเคชันโดเมนทำงานอยู่ในเซิร์ฟเวอร์ตัวเดียวกันก็ตาม ซึ่งตรงนี้ทำให้เว็บไซต์มีความปลอดภัยในตัวเอง แต่ถ้าเกิดมีเว็บไซต์บางตัวเกิดตายไป ก็จะไม่ส่งผลกระทบต่อเว็บไซต์อื่นที่อยู่คนละแอปพลิเคชันโดเมน ภาพที่ 2.8 แสดงภาพแอปพลิเคชันโดเมนบนเว็บเซิร์ฟเวอร์



รูปที่ 2.8 โครงภายในของ Web Server IIS

ไฟล์ที่มีในเว็บไซต์ ASP .NET 1.x มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

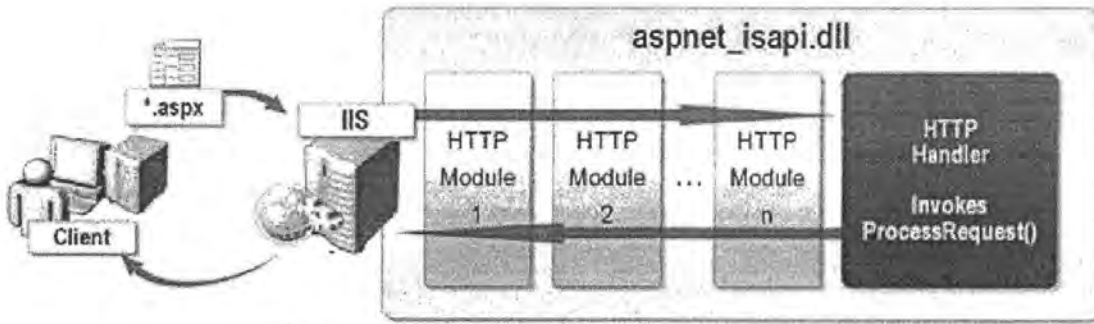
ตารางที่ 2.1 แสดงชนิดของไฟล์ทั้งหมดใน .NET Framework 1.X

ไอคอน	ชนิดไฟล์	นามสกุล	รายละเอียด
	เว็บฟอร์ม	*.aspx	ใช้ออกแบบหน้าเว็บเพจ และติดต่อกับ Code Behind ได้
	เว็บยูสเซอร์ คอนโทรล	*.ascx	ใช้สร้างคอนโทรลที่ใช้สำหรับเว็บฟอร์ม
	เว็บเซอร์วิส	*.asmx	ใช้สร้างเว็บเซอร์วิส
	HTML	*.html	หน้า HTML ที่สามารถบรรจุไคเอนสคริปต์ได้
	สไตล์ชีต	*.css	ใช้กำหนดสไตล์ของหน้า HTML
ไอคอน	ชนิดไฟล์	นามสกุล	รายละเอียด
	คลาส	*.cs	คลาสเปล่าที่ให้นำไปเขียนโค้ดต่อ
	ไฟล์ Global Application	*.asax	ไฟล์สำหรับจัดการกับเหตุการณ์ (Event) ต่างๆ ของเว็บไซต์ ทำหน้าที่คล้ายกับไฟล์ global.asa ของคลาสสิก ASP
	XML	*.xml	ไฟล์ XML เปล่า
	สกีมาของ XML	*.xsd	ไฟล์สกีมาที่ใช้สำหรับตรวจสอบความถูกต้องของ XML
	เท็กซ์	*.txt	ไฟล์ข้อความเปล่า
	รีซอร์ส	*.resx	ไฟล์ที่ใช้เก็บข้อมูลของ .NET
	เดต้าเซต	*.xsd	ใช้สร้างสกีมา จากคลาสเดต้าเซต
	คริสตัล รีพอร์ต	*.rpt	ไฟล์ที่ใช้สร้างรายงานสำหรับแอปพลิเคชันเว็บ และวินโดวส์
	XSLT	*.xslt	ไฟล์ที่ใช้แปลงข้อมูล XML ให้เป็นรูปแบบต่างๆ

### 2.4.3 วงจรชีวิตของเว็บฟอร์ม

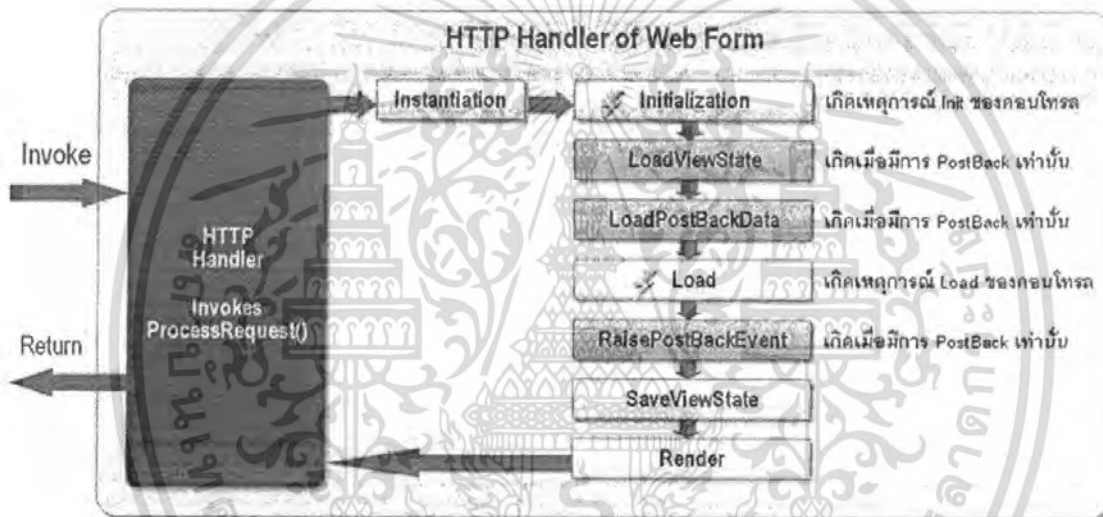
เมื่อไคลเอนต์มีการร้องขอไฟล์แบบไดนามิกไปที่ IIS การเรียกนี้จะถูกส่งต่อไปให้กับ aspnet\_isapi.dll จากนั้นจะส่งผ่านเป็นทอดๆ ให้กับ HTTP Module และสุดท้ายก็จะมาทำงานที่ HTTP Handler ที่เป็นตัวรันเว็บฟอร์ม ดังภาพที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การจัดการกับการเรียกไปยังเว็บฟอร์มของ IIS

เมื่อการร้องขอไฟล์เว็บฟอร์มมาถึง HTTP Handler ของเว็บฟอร์ม และจะมีการเรียกเมธอด ProcessRequest และเข้าสู่วงจรชีวิตของเว็บฟอร์มดังภาพที่ 2.9



รูปที่ 2.10 วงจรชีวิตของเว็บฟอร์ม

จากรูปที่ 2.10 เราจะแบ่งขั้นตอนการทำงานย่อยของเมธอด ProcessRequest ออกได้เป็น 8 ขั้นตอน ดังนี้

**ขั้นตอนที่ 1 Instantiation**

ในขั้นตอนนี้ ASP .NET จะมีการสร้างอ็อบเจกต์ของเว็บฟอร์มขึ้นมาโดยอัตโนมัติ ไม่ว่าจะเว็บฟอร์มของเราจะมีได้คแบบ Inline หรือ Code Behind ก็ตาม โดย ASP.NET จะเข้าไปทำการอ่านแท็กต่างๆ ที่อยู่ในมุมมองของ Source ในไฟล์ .aspx แล้วทำการสร้างอ็อบเจกต์เว็บฟอร์มขึ้นมา ก่อน จากนั้น จึงทำการสร้างอ็อบเจกต์ของคอนโทรลแต่ละตัวที่อยู่ในเว็บฟอร์ม

**ขั้นตอนที่ 2 Initialization**

หลังจากที่มีการสร้างลำดับชั้นของคอนโทรลเสร็จแล้ว จะเข้ามาในส่วนของเหตุการณ์เริ่มต้น (Initialization) ของทั้งเว็บฟอร์ม และของคอนโทรลภายในเว็บฟอร์ม เหตุการณ์นี้ตรงกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด OnInit ของทั้งเว็บฟอร์ม และคอนโทรล โดยเมธอด OnInit ของคอนโทรลจะถูกเรียกก่อนจนครบทุกตัว จากนั้นจึงทำการเรียกเมธอด OnInit ของเว็บฟอร์ม

สำหรับเมธอด OnInit ของเว็บฟอร์ม เราสามารถเห็นเมธอดนี้ได้ ในตอนที่เพิ่มเว็บฟอร์มเข้ามาใหม่ในโปรเจกของ Visual Studio .NET 2003 เราจะสังเกตเห็นส่วนของ Web Form Designer generated code ซึ่งถ้าคลิกเข้าไปดูภายในจะพบว่ามีเมธอด OnInit อยู่ภายใน และเราสามารถเข้าไปเขียนโค้ดเพิ่มเติมในส่วนนี้ได้ แต่สำหรับ Visual Studio 2005 ส่วนนี้จะถูกซ่อนไว้ ถ้าหากเราต้องการเขียน โค้ดเพิ่มในส่วนนี้เราต้องเพิ่มเมธอด OnInit นี้เข้าไปเองดัง โค้ดตัวอย่าง

```
protected override void OnInit(EventArgs e){
    base.OnInit(e);
    //Add your custom OnInit here
    ...}
```

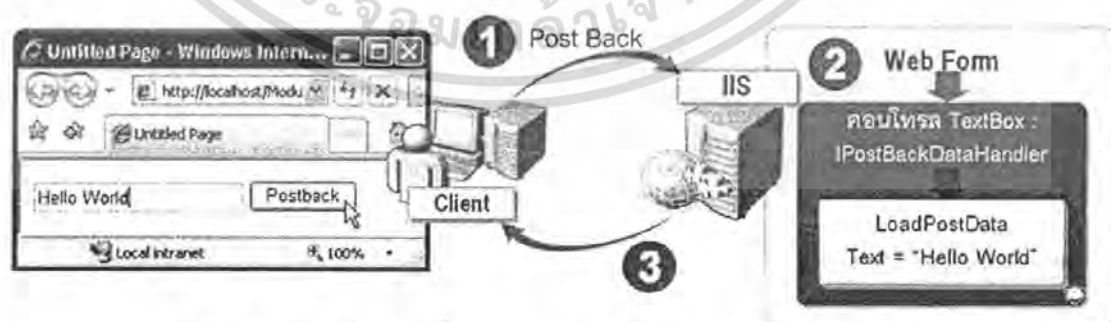
ตัวอย่างการสร้างเมธอด OnInit ของเว็บฟอร์ม

ขั้นตอนที่ 3 โหลด View State

การโหลดค่าจาก View State จะทำเฉพาะตอนที่เว็บฟอร์มมีการ Post back เท่านั้น ในขั้นตอนนี้ ASP .NET จะทำการโหลดค่าจาก View State แล้วนำค่านี้ไปกำหนดให้กับคอนโทรลในเว็บฟอร์มตามลำดับชั้นของอ็อบเจกต์เว็บฟอร์ม และคอนโทรลในเว็บฟอร์ม

ขั้นตอนที่ 4 โหลด Post Back Data

จะเกิดเมื่อเว็บฟอร์มมีการ Post back เท่านั้น ขั้นตอนนี้เว็บฟอร์ม จะทำการโหลดค่าจาก HTTP POST Headers แล้วส่งต่อค่าที่เหมาะสมให้กับเซิร์ฟเวอร์คอนโทรลแต่ละตัวที่ทำการอิมพลีเมนต์อินเตอร์เฟซ IPostBackDataHandler หลังจากนั้นเซิร์ฟเวอร์คอนโทรลจะทำการเรียกเมธอด LoadPostData เพื่อดึงค่าที่เว็บฟอร์มส่งให้ไปกำหนดพร้อมเพอดี Text ให้ตัวเอง ตัวอย่างของการโหลดค่าในขั้นตอนนี้ดังภาพที่ 2.11



รูปที่ 2.11 ตัวอย่างแสดงขั้นตอนการโหลดค่าจากการ Post Back

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.11 มีขั้นตอนการไหลค้ำจากการ Post Back ดังนี้

1. โคลเอนต์ร้องขอไฟล์เว็บฟอร์ม ซึ่งบรรจุคอนโทรล TextBox เพื่อให้กรอกข้อความ และคอนโทรล Button ซึ่งเป็นปุ่มที่ทำให้เกิดการ Post Back ไปที่เซิร์ฟเวอร์ โดยไม่มีส่วนของโค้ดที่แก้ไขค่าในคอนโทรล TextBox เลย

2. เมื่อการ Post Back มาถึงที่เว็บเซิร์ฟเวอร์ เว็บฟอร์มจะถูกรันขึ้นมา และเมื่อถึงขั้นตอนการไหลค้ำ Post Back Data เว็บฟอร์มตรวจคว่ำมีคอนโทรลตัวใดที่ทำการอิมพลิเมนต์อินเตอร์เฟส IPostBackDataHandler บ้าง ซึ่งในที่นี้มีเพียงคอนโทรล TextBox เท่านั้น จากนั้นเว็บฟอร์มจะทำการอ่านจาก HTTP POST Headers เฉพาะที่เป็นของ TextBox แล้วนำค่าที่ได้ไปกำหนดให้กับพรอปเพอร์ตี้ Text ซึ่งในที่นี้ก็คือคว่ำ “Hello World”

3. เมื่อรันเว็บฟอร์มครบทุกขั้นตอนแล้ว ก็จะส่งผลลัพธ์คืนกลับไปให้โคลเอนต์ ที่โคลเอนต์จะเห็นว่าคว่ำใน TextBox นี้ จะพบคว่ำว่า “Hello World” เหมือนกับสถานะตอนที่มีการ Post Back ไปที่เว็บเซิร์ฟเวอร์

ขั้นตอนที่ 5 Load

เมธอดนี้เป็นของเว็บฟอร์ม ซึ่งนักพัฒนาเว็บจะรู้จักเป็นอย่างดี เหตุการณ์นี้ตรงกับเหตุการณ์ในโพซีเซอร์ Page\_Load ขั้นตอนนี้จะถูกเรียกทั้งตอนที่มีการ Post Back และไม่มีการ Post Back

ขั้นตอนที่ 6 Raise Post Back Event

เป็นเหตุการณ์ที่เกิดกับคอนโทรลที่มีความสามารถในการ Post Back ได้เช่น เหตุการณ์ Click ของ Button เหตุการณ์ TextChanged ของคอนโทรล TextBox คอนโทรลที่สามารถสร้างการ Post Back ได้นั้นต้องมีการอิมพลิเมนต์อินเตอร์เฟส IPostBackDataHandler ที่เราได้รู้จักกันไปแล้ว

ขั้นตอนที่ 7 Save View State

ขั้นตอนนี้เว็บฟอร์มจะเรียกเมธอด SaveViewState ของคอนโทรลแต่ละตัวที่อยู่ภายใต้เว็บฟอร์มตามลำดับชั้น เพื่อทำการบันทึกสถานะของคอนโทรลล่าสุดที่อาจมีการเปลี่ยนแปลงไปเนื่องจากขั้นตอนก่อนหน้านี้ ผลของการบันทึกนี้จะถูกเก็บอยู่ในรูปตัวหนังสือที่เข้ารหัสแบบ Base-64 แล้วเก็บใน Hidden Field ที่จะถูกสร้างในขั้นตอนถัดไป

ขั้นตอนที่ 8 Render

ขั้นตอนนี้ เว็บฟอร์มจะทำการแปลงข้อมูลในขั้นตอนที่ผ่านมาออกมาในรูปแบบภาษา HTML เพื่อส่งต่อให้กับโคลเอนต์ ในขั้นตอนนี้เว็บฟอร์มจะเรียกเมธอด RenderControl ของคอนโทรลแต่ละตัวที่อยู่ภายใต้เว็บฟอร์มตามลำดับชั้น สำหรับ ขั้นตอนทั้ง 8 ที่กล่าวมานี้ เป็นขั้นตอนที่สำคัญที่มีความเกี่ยวข้องกับ View State ซึ่งได้ละขั้นตอน PreInit ขั้นตอนPreRender และขั้นตอน Unload ไว้

## บทที่ 3

### หลักการออกแบบ

จากทฤษฎีที่ได้กล่าวในบทที่แล้วทำให้เราสามารถจะกำหนดคุณลักษณะพื้นฐานของระบบการอินเตอร์เฟสกับมิเตอร์ได้แล้วพอสมควร แต่ในการพัฒนาระบบของรายงานเล่มนี้ถูกจำกัดด้วยข้อกำหนดบางประการซึ่งทำให้เราต้องทำการเพิ่มหรือลดคุณลักษณะของระบบ เพื่อสอดคล้องกับเงื่อนไขเหล่านี้ได้

1. เรื่องของมาตรฐานของไฟล์ PF4 ที่ถูกส่งออกจากมิเตอร์
2. ไม่มีมิเตอร์วัดคุณภาพไฟฟ้าที่ใช้ในการทดลองจริง
3. ระบบต้องพัฒนาเพื่อเชื่อมต่อและเข้ากันได้กับฐานข้อมูลซึ่งเป็นของซอฟต์แวร์ที่ใช้

#### จัดการมิเตอร์

จากข้อจำกัดดังกล่าวทำให้ระบบนี้สามารถเขียนความต้องการเบื้องต้นได้ดังนี้

#### ความต้องการของระบบ

ระบบอินเตอร์เฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้า เป็นระบบที่เชื่อมต่อกับมิเตอร์ผ่านทาง Internet โดยโปรแกรมที่ทำหน้าที่อินเตอร์เฟสต้องมีความสามารถในการประมวลผล การรับข้อมูลของมิเตอร์(Receivelog) การถอดรหัสไฟล์(Extractlog) รวมถึงการนำข้อมูลที่ได้อัปโหลดลงฐานข้อมูล(Data Persistent)

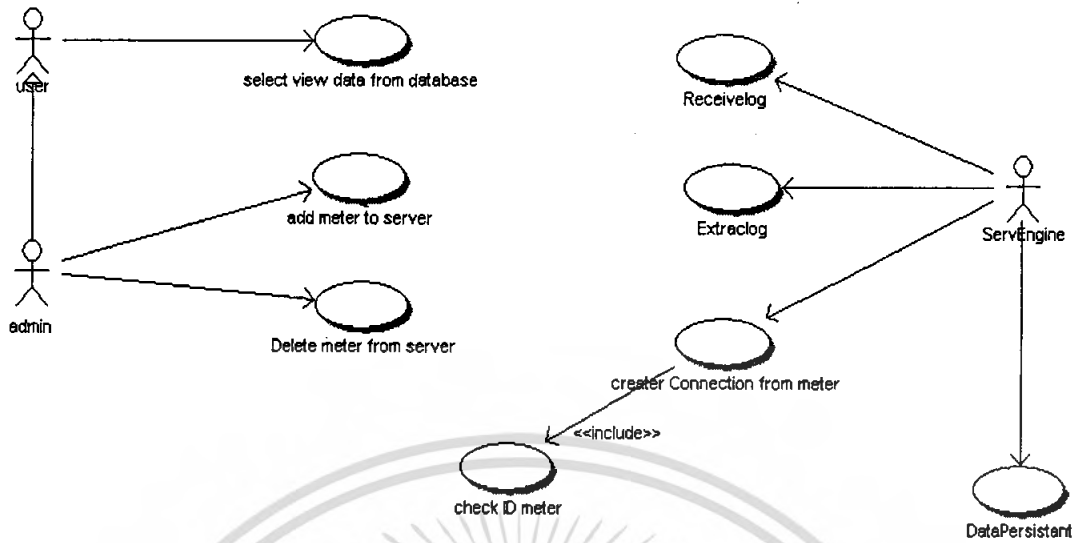
#### ความสามารถของระบบ

1. ระบบสามารถแปลงข้อมูลซึ่งเป็นคุณลักษณะทางไฟฟ้าเข้าสู่ฐานข้อมูลได้
2. ผู้ใช้สามารถอ่านค่าคุณลักษณะทางไฟฟ้าที่รวบรวมไว้ในฐานข้อมูลออกมาทางเว็บได้
3. มิเตอร์คุณภาพไฟฟ้าต้องทำการส่งข้อมูลมายังเครื่อง Server
4. มิเตอร์คุณภาพไฟฟ้าแต่ละตัวจะมีรหัสเฉพาะที่ใช้จำแนกมิเตอร์นั้นๆ
5. ระบบจะทำงาน โดยรอรับการเชื่อมต่อเข้ามาของมิเตอร์ลูกข่าย
6. ระบบสามารถทำการตรวจสอบว่าการเชื่อมต่อนั้นมาจากมิเตอร์ลูกข่ายของตนหรือไม่
7. ระบบจะตัดการเชื่อมต่อหากมิเตอร์ได้ทำการส่งข้อมูลสำเร็จ
8. สามารถเพิ่มหรือลบมิเตอร์ที่เป็นลูกข่ายออกจากระบบได้

#### คุณสมบัติของระบบ

1. ประหยัดงบประมาณในการใช้งานคนในการเก็บข้อมูลของมิเตอร์คุณภาพไฟฟ้า
2. ประหยัดเวลาที่ใช้ในการเก็บข้อมูลจากมิเตอร์
3. ระบบสามารถรองรับมิเตอร์ที่เชื่อมต่อได้เป็นจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 User Case Diagram

### 3.1 การออกแบบตารางฐานข้อมูล

ฐานข้อมูลที่ใช้ในการบันทึกค่าของคุณลักษณะทางไฟฟ้าที่ได้รับจากมิเตอร์วัดคุณภาพไฟฟ้า ใช้ MSSQLServer(Microsoft SQL Server) ซึ่งในส่วนของฐานข้อมูลในส่วนการพัฒนาจำเป็นต้องใช้ ฐานข้อมูลเดียวกับซอฟต์แวร์ซึ่งมาพร้อมกับ มิเตอร์วัดคุณภาพไฟฟ้าซึ่งเป็นมาตรฐานของมิเตอร์ของบริษัทนั้น ในรายงานนี้จึงจะไม่กล่าวถึง การออกแบบ หรือโครงสร้าง ของฐานข้อมูลอาจจะมีกรกล่าวถึงเฉพาะตารางที่สำคัญๆ ที่เราจะนำมาใช้ในในงานเท่านั้น อย่างเช่น ตาราง SagSwellTable เป็นตารางที่เก็บข้อมูลของ ไฟฟ้าตก หรือไฟฟ้าเกิน ที่มิเตอร์ส่งมา มีโครงสร้างดังนี้

ตารางที่ 3-1 แสดงตาราง SagSwellTable ในฐานข้อมูลของระบบ

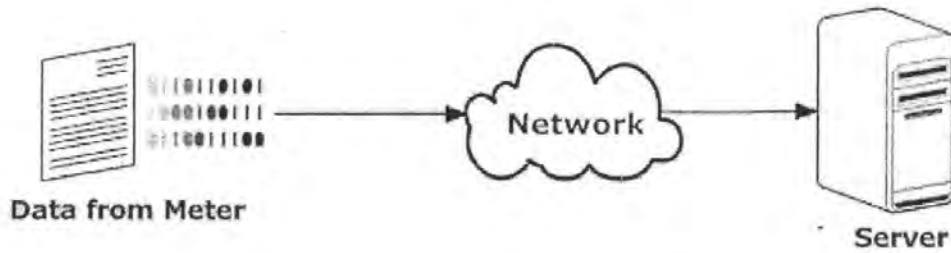
Column Name	Type	User Type	Key	Not Null	Default	Rule
MeasureNo	int		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
StartTime	datetime		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
Resolution	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
Method	int		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
TrigPhase	int		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
Duration	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
Depth	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
Uref	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
Deviation	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
PreTrig	int		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
UW2R	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
IW2R	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
U1	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
U2	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
U3	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
U4	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
I1	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
I2	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
I3	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
I4	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
sagno	int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	{null}	{null}
UW2R2	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
UW2R3	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
UW2R4	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
IW2R2	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
IW2R3	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
IW2R4	float		<input type="checkbox"/>	<input checked="" type="checkbox"/>	{null}	{null}
Length	int		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
UW2R5	float		<input type="checkbox"/>	<input checked="" type="checkbox"/>	{null}	{null}
IW2R5	float		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}
U5	image		<input type="checkbox"/>	<input checked="" type="checkbox"/>	{null}	{null}
I5	image		<input type="checkbox"/>	<input type="checkbox"/>	{null}	{null}

### 3.2 Protocol ที่ใช้ในการเชื่อมต่อ

การที่มีเตอร์วัดคุณภาพไฟฟ้าจะทำการเชื่อมต่อกับคอมพิวเตอร์ผ่านทางอินเตอร์เน็ตนั้น จำเป็นต้องมีการกำหนดวิธีการในการเชื่อมต่อ นั่นคือ Protocol นั้นเองโดยเงื่อนไขของ การส่งข้อมูลของมิเตอร์นั้นมีดังนี้

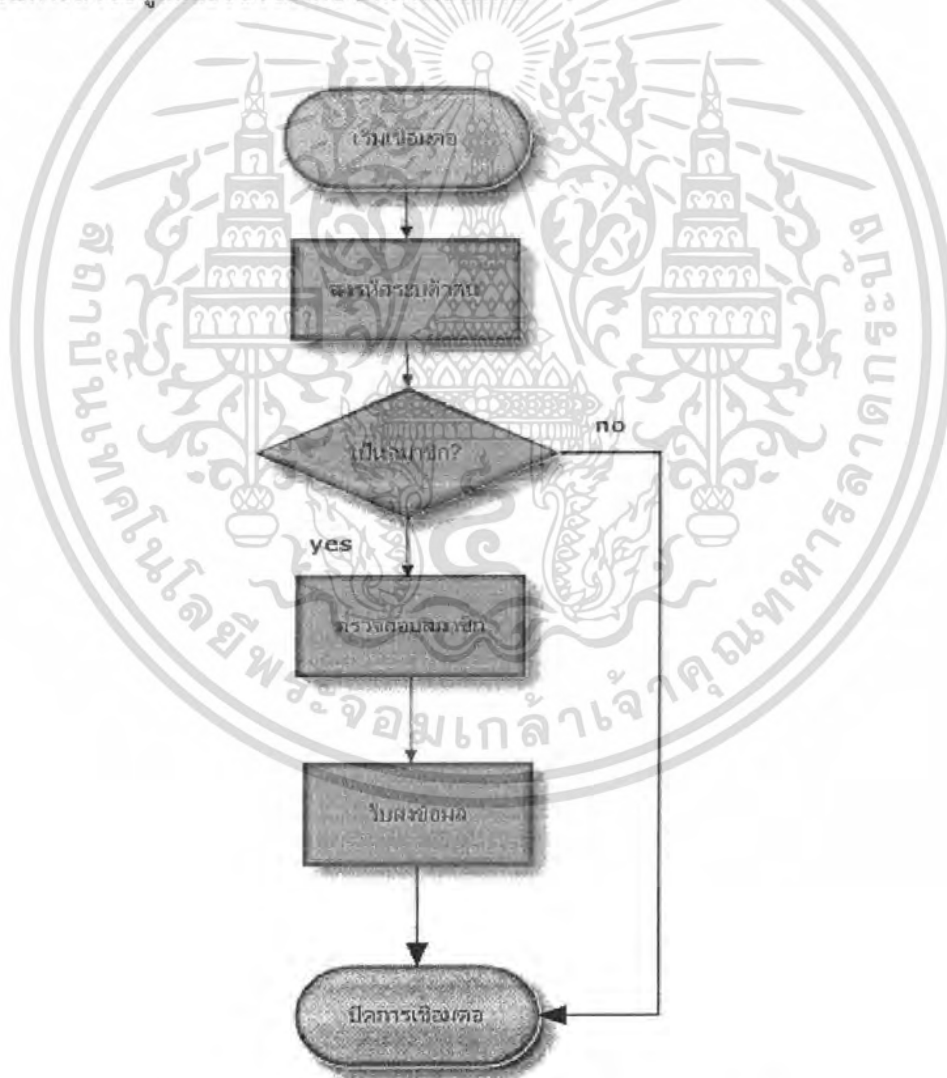
1. ข้อมูลต้องไม่สูญหายระหว่างการถ่ายโอนข้อมูล
2. ข้อมูลที่ส่งต้องรับประกันความถูกต้อง
3. ไม่มีเงื่อนไขของความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงแก้ไข หรือเผยแพร่ข้อมูลใดๆ อันอาจก่อให้เกิดความเสียหายแก่เจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 เส้นทางการเชื่อมต่อ

จากเงื่อนไขขั้นต้นเราจึงใช้ TCP/IP เป็น Protocol ในการทำงาน และจะทำการเพิ่มขั้นตอนในการที่มิเตอร์ต้องแจ้งต่อ Server ว่าเริ่มการส่งข้อมูล เมื่อทำการส่งข้อมูลเสร็จแล้ว มิเตอร์จะต้องแจ้งว่าจบการส่งข้อมูลและร้องขอเพื่อ ปิดการเชื่อมต่อ



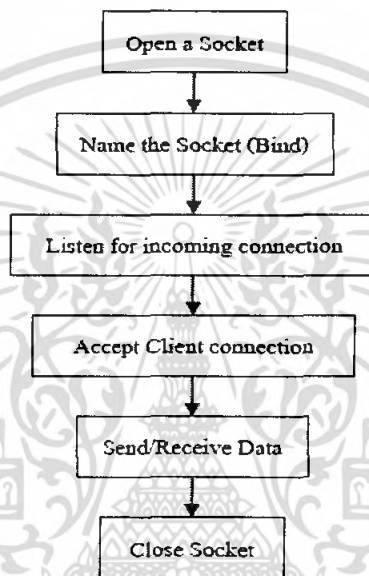
รูปที่ 3.3 การทำงานของระบบมิเตอร์เมื่อจะทำการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของระบบเริ่มที่การเชื่อมต่อ เมื่อเชื่อมต่อสำเร็จก็จะทำการส่งข้อมูลไปทันทีแล้วจึงทำการระบบตัวตน โดยเช็คที่ตัวข้อมูล แล้วถ้าหากเป็นข้อมูลที่ยืนยันแล้วว่าเป็นข้อมูลที่ต้องการถูกต้องก็จะถูกบันทึกลงฐานข้อมูล แต่หากไม่ใช่ก็จะลบข้อมูลที่ได้มาทิ้งทั้งหมด

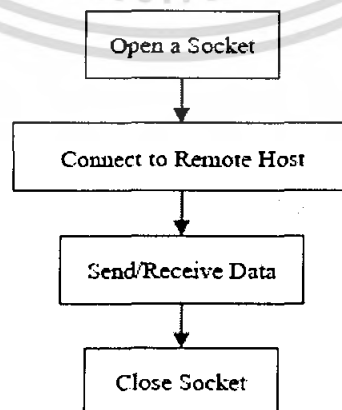
### 3.3 ขั้นตอนการทำงานของระบบการอินเทอร์เน็ตเฟสของมิเตอร์คุณภาพไฟฟ้า

การทำงานของระบบอินเทอร์เน็ตเฟสของมิเตอร์ไฟฟ้านี้จะประกอบด้วยสองส่วนนั้นคือ Server และส่วนของ Meter หรือ Client โดยระบบจะมีขั้นตอนการทำงานดังนี้



รูปที่ 3.4 ขั้นตอนการสร้าง Passive Sockets

จากรูป 3.4 นี้คือขั้นตอนการทำงานทั้งหมดของ Server ในการทำงานของ Socket เพื่อรอรับการเชื่อมต่อจาก มิเตอร์วัดคุณภาพไฟฟ้า



รูปที่ 3.5 ขั้นตอนการสร้าง Active Sockets

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 แสดงขั้นตอนการทำงานของ Client เริ่มตั้งแต่เปิดการเชื่อมต่อจนถึงปิดการเชื่อมต่อ

#### 1.Initial ขั้นตอนการเริ่มต้นของระบบ

เป็นการเริ่มทำงานของ Server จะทำหน้าที่รอรับการเชื่อมต่อจากมิเตอร์ลูกค้า การเริ่มทำงานนี้จะต้องมีค่าเริ่มต้นซึ่ง ผู้ดูแลระบบได้ทำการตั้งค่าไว้

#### 2.Connection ขั้นตอนการเชื่อมต่อ

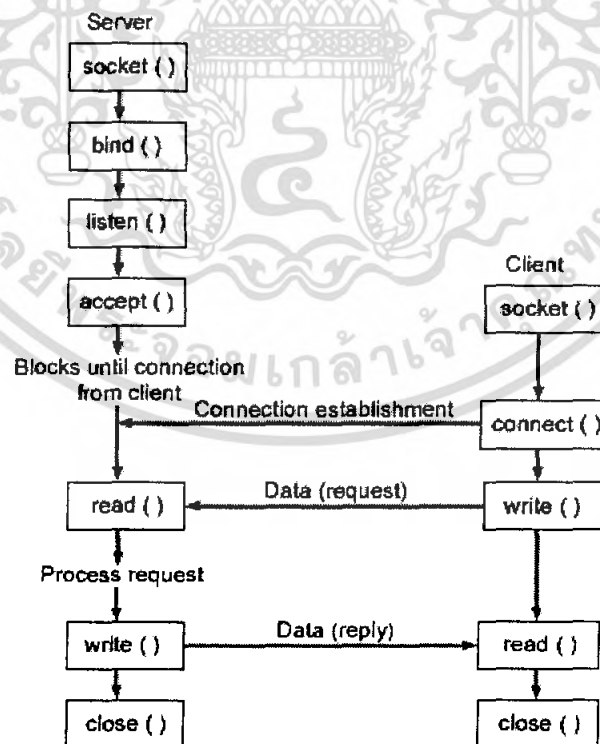
ขั้นตอนการเชื่อมต่อนี้จะเป็นการทำงานของมิเตอร์เพื่อร้องขอส่งข้อมูลไปยัง Server โดยจะต้องผ่านการตรวจสอบก่อนว่ามีเตอร์ที่ร้องขอนั้นมีการลงทะเบียนไว้โดยผู้ดูแลระบบหรือไม่ หากไม่ผ่านขั้นตอนนี้ก็จะถูกตัดการเชื่อมต่อไป

#### 3.Transfer ถ่ายโอนข้อมูล

เมื่อผ่านระบบการพิสูจน์ตัวตนของระบบแล้วก็จะเป็นการเริ่มการถ่ายโอนข้อมูลของมิเตอร์เข้ามาที่ Server ในการส่งข้อมูลนี้จะส่งในลักษณะ Stream ซึ่งการถ่ายโอนข้อมูลนี้จะทำอยู่บนโปรโตคอลในชั้น Application ที่เราได้ทำการออกแบบขึ้นมาโดยรายละเอียดได้อยู่ในหัวข้อของ Protocol แล้ว เมื่อถ่ายโอนข้อมูลเสร็จสิ้นก็จะยืนยันการตัดการเชื่อมต่อ

#### 4.Close Connection จบการเชื่อมต่อ

เมื่อได้รับการยืนยันว่าจบการถ่ายโอนข้อมูลเสร็จสิ้นแล้วเราจะต้องทำการตัดการเชื่อมต่อ



รูปที่ 3.6 การทำงานของระบบการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 ระบบมิเตอร์จำลอง

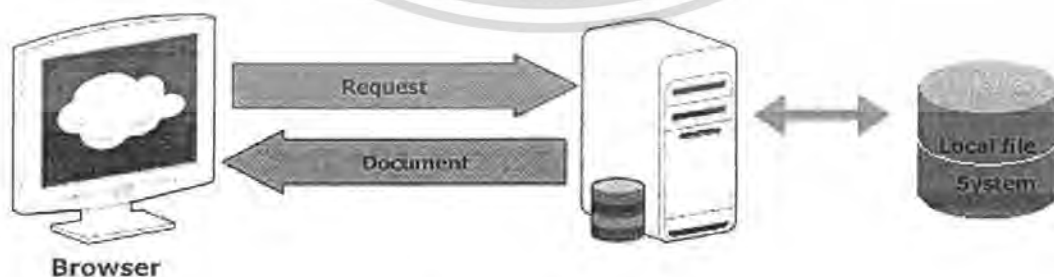
จากที่ได้ทราบไปแล้วในเบื้องต้นว่าระบบของเรานั้นไม่มี มิเตอร์วัดคุณภาพไฟฟ้าจริงๆเพื่อใช้ทดลอง จึงจำเป็นต้องสร้าง โปรแกรมจำลองมิเตอร์เพื่อทำงานในการสร้างข้อมูลที่เราต้องการส่งมาที่ Server การทำงานของระบบการอินเทอร์เน็ตเฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้ามีข้อกำหนดที่ต้องเพิ่มขึ้นมาก็คือ

1. มิเตอร์จำลองนี้จะต้องมีหมายเลขประจำเครื่อง หรือ MAC Address
2. มิเตอร์จำลองต้องสามารถสร้างข้อมูลจำลองสถานการณ์เช่น ไฟตก ไฟเกิน ไฟดับ
3. มิเตอร์จำลองต้องกำหนดได้ว่าจะส่งข้อมูลเมื่อใด ในที่นี้ คือกำหนดให้ส่งเมื่อมีเหตุการณ์เกิดขึ้น หรือกำหนดให้ส่งทุกๆเวลาเท่าใด
4. มิเตอร์จำลองต้องสามารถสร้างข้อความชนิดต่างๆได้ เช่น ข้อความสำหรับระบุตัวตน ข้อความสำหรับเริ่มส่งข้อมูล
5. มิเตอร์จำลองต้องสามารถ Random สร้างข้อมูลที่ส่งออกมาได้
6. รูปแบบของข้อมูลที่มิเตอร์จำลองสร้าง ต้องเป็นข้อมูลที่ Server สามารถเข้าใจได้

### 3.5 ระบบแสดงผลมิเตอร์ที่อัปเดตข้อมูลล่าสุดตามจำนวนที่กำหนดผ่านเว็บ

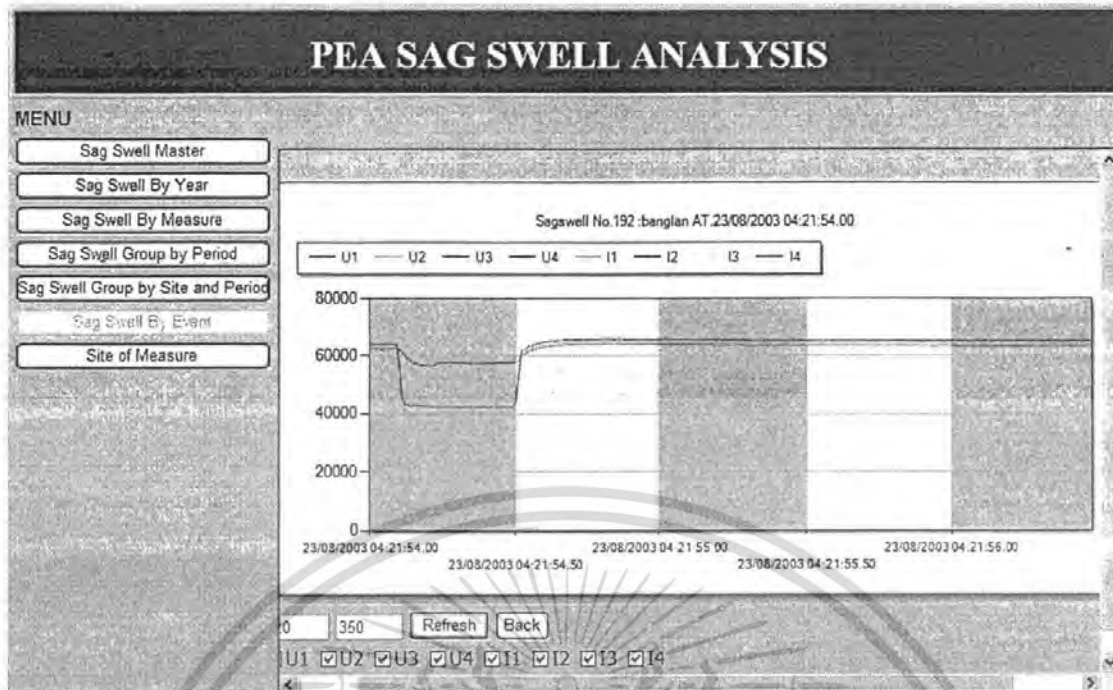
ในระบบอินเทอร์เน็ตเฟสกับมิเตอร์วัดคุณภาพไฟฟ้านี้ ได้สร้างส่วนที่ช่วยอำนวยความสะดวกให้กับผู้ใช้คือส่วนในการแสดงข้อมูลแจ้งเหตุของมิเตอร์ที่ส่งข้อมูลมาที่ Server ล่าสุด เพื่อสามารถทราบได้ทันทีว่าตอนนี้เกิดเหตุการณ์ไฟตกที่ไหน หรือไฟเกินที่ไหน ซึ่งเหตุการณ์เหล่านี้ก็จะช่วยในการทราบได้ว่าอาจจะมีเหตุไฟดับตามมา ส่วนนี้จึงมีความจำเป็นมาก

ระบบนี้มีการทำงานที่ง่ายก็คือเมื่อผู้ใช้เข้ามาที่หน้าเว็บเข้าไปในส่วนของหน้านี้ก็จะมียรายชื่อมิเตอร์ที่มีการส่งข้อมูลเข้ามาล่าสุดขึ้นมาแสดง สำหรับจำนวนในการแสดงจำนวนมิเตอร์ของแต่ละหน้าจะเป็น คุณสมบัติที่มีในระบบให้ผู้ใช้เลือกได้เช่นกัน



รูปที่ 3.7 การทำงานของเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงรูปเว็บแสดงข้อมูลมิเตอร์ที่อัพเดทล่าสุด

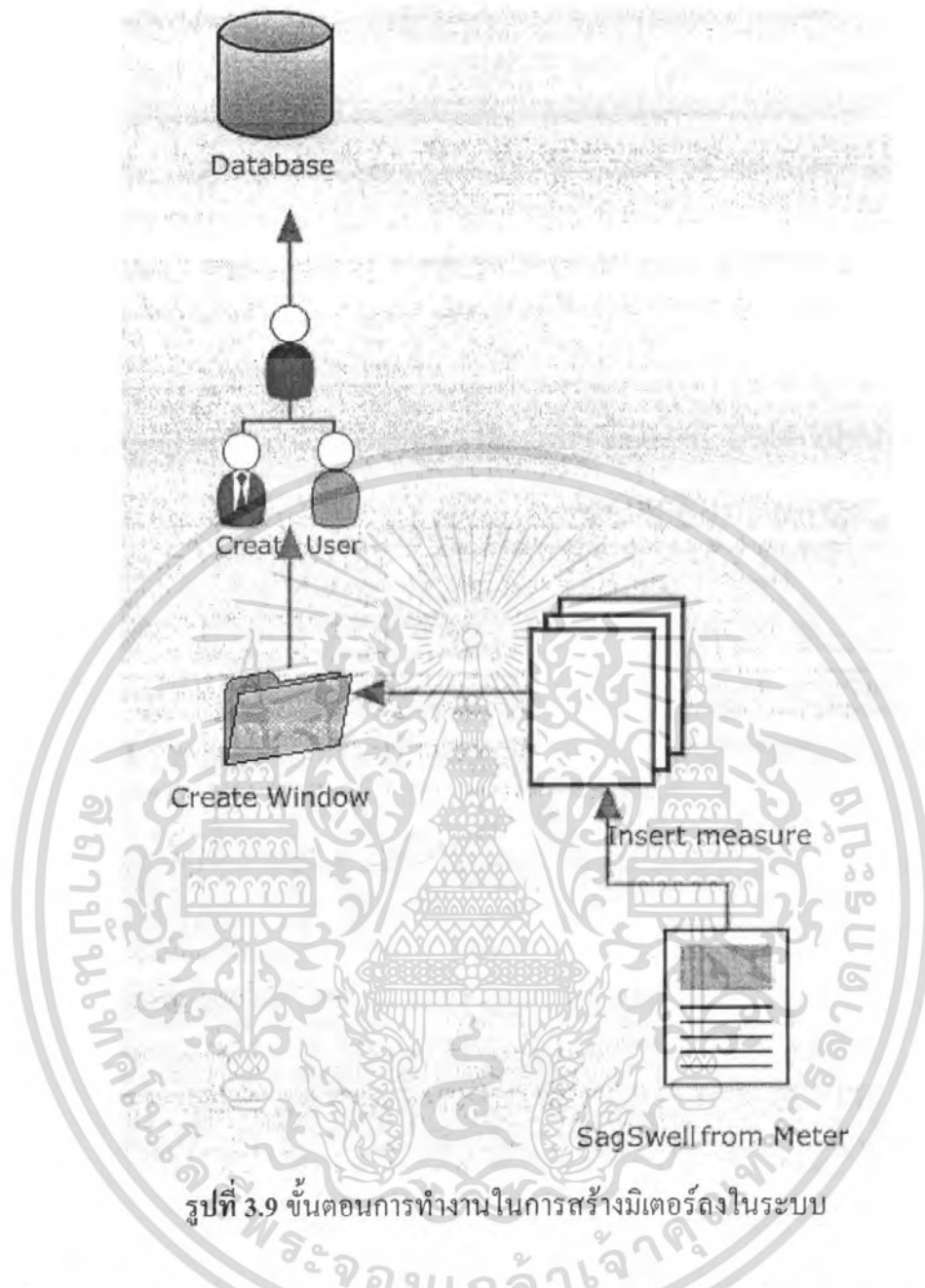
### 3.6 การสร้างและการลบมิเตอร์ออกจากระบบ

การทำงานของมิเตอร์วัดคุณภาพ ไฟฟ้าที่นั่นแต่ละครั้งก่อนที่จะส่งข้อมูลต้องทำการระบบตัวตนเพื่อทำการแยกแยะ และยังมีความปลอดภัยของระบบด้วยคั้งนั้น ต้องมีการสร้างมิเตอร์ในระบบก่อนนั้นก็คือการลงทะเบียนนั่นเอง

การลงทะเบียนนี้เราจะต้องมีการกรอกรายละเอียดของมิเตอร์เพื่อเป็นการเก็บข้อมูลเฉพาะของมิเตอร์ซึ่งใช้ในการระบุตัวตนแต่ละครั้งนั่นเอง ข้อมูลเบื้องต้นนี้นั้นอาจจะได้แก่

- รหัสของมิเตอร์ ซึ่งกำหนดมาจากโรงงานผู้ผลิตซึ่งไม่ซ้ำ คล้ายกับแมคแอดเดรสของมิเตอร์ตัวนั้น
- รหัสของสถานที่ซึ่งเราต้องกำหนดให้ว่ามีเตอร์ตัวนั้นตั้งอยู่ที่ใดใช้รหัสใด ซึ่งต้องเป็นมาตรฐานเพื่อไม่ให้เกิดการซ้ำซ้อนกับมิเตอร์ตัวอื่นๆ
- ข้อมูลอื่นๆซึ่งส่วนนี้จะถูกเก็บเป็นข้อมูลสำหรับUSER เพื่อใช้ในการสร้างรายงาน เช่น มิเตอร์นี้ รุ่น ไหน เริ่มใช้งานเมื่อใด อยู่ในสถานที่ไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ขั้นตอนการสร้างมิเตอร์เข้าสู่ระบบนั้นอ้างอิงจาก ตัวฐานข้อมูล เริ่มที่ ตารางUserTable ซึ่งเป็น ขั้นตอนการระบุว่าจะใช้ user ใดเพื่อสร้าง Window ลงในระบบ ซึ่ง window ที่ระบบสร้างเปรียบเสมือน พื้นที่ในการบรรจุ มิเตอร์ที่เราจะสร้างลงในระบบ ซึ่งwindow เหล่านี้จะถูกเก็บลงใน ตาราง WindowTable

เมื่อเราได้สร้าง Window ลงในฐานข้อมูลแล้วเราจึงสามารถสร้าง Measure ซึ่งก็คือมิเตอร์ วัตถุประสงค์ไฟฟ้า มิเตอร์เหล่านี้จะถูกบันทึกลงในตาราง MeasureTable เมื่อเราสร้างมิเตอร์ลงในระบบแล้วค่าต่างๆที่เราส่งให้ก็จะถูกบันทึกลงใน ตาราง SagSwellTable นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.7 ฟอรัมไฟล์ที่ใช้ในการสื่อสารของระบบ

การทำงานของระบบอินเทอร์เน็ตเฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้าด้วย .NET Framework นี้มีจำเป็นต้องมีการกำหนดมาตรฐานของไฟล์ที่เป็นกลาง ในโครงการของเราไม่สามารถอ้างอิงกับไฟล์ตัวอย่างที่มีอยู่ได้จึงได้ทำการกำหนดโดยใช้มาตรฐานของ XML ในการบรรจุข้อมูลซึ่งเป็นคุณลักษณะทางไฟฟ้าเราจึงจะมาศึกษา ภาษานี้เพื่อความเข้าใจพื้นฐานดังนี้

#### ภาษา XML (Extensible Markup Language)

XML ถูกกำหนดโดยกลุ่มทำงาน XML ของสถาบัน World Wide Web Consortium(W3C) กลุ่มทำงานกลุ่มนี้ได้บรรยายถึงภาษา XML ไว้ว่า Extensible Markup Language เป็นฟอรัมไฟล์ที่อธิบายถึงรายละเอียดของ โครงสร้างและแบบของข้อมูลเป็นภาษาหรือชุดคำสั่งเกี่ยวกับข้อมูลบนเว็บที่ให้การพัฒนาและมีศักยภาพในส่วนของ โครงสร้างข้อมูลจากหลากหลายแอปพลิเคชันมา นำเสนอบนเครื่องเดสก์ทอปด้วย XML จะทำให้การจัดการข้อมูลหรือเรียกใช้ข้อมูลจากแอปพลิเคชันต่างๆ จะเข้าสู่มาตรฐานเดียวกัน

#### ลักษณะ โครงสร้างของ XML

XML เป็นการใช้อ็ความเพื่อบ่งบอกโครงสร้างของเอกสาร พิจารณาตัวอย่างรูปแบบ โครงสร้างของหนังสือ เมื่อหนังสือประกอบด้วยจำนวนบท 2 บท ในแต่ละบทประกอบด้วย เนื้อความ

Begin Book

Begin Chapter 1

Text for Chapter 1

End Chapter 1

Begin Chapter 2

Text for Chapter 2

End Chapter 2

End Book

หนังสือที่มีอยู่ในปัจจุบันจะมีโครงสร้างที่มีรายละเอียดที่ซับซ้อนมากกว่านี้ เช่น บทนำ, สารบัญ เป็นต้น เช่นเดียวกัน ภายในส่วนเนื้อความยังประกอบด้วยโครงสร้างย่อยคือ ย่อหน้า (Paragraph) แต่ละย่อหน้ายังประกอบขึ้นจากประโยค คำและตัวอักษรด้วย

ลักษณะของเอกสาร XML นั้น สามารถอธิบายโดยใช้ตัวอย่างที่ 1 ได้ ดังนี้  
ตัวอย่างที่ 1

```
<?xml version="1.0" encoding="windows-874"?>
```

```
<mali>
```

```
<malisom>ขึ้นต้นด้วยมะลิซ้อน</malisom>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในท้องถิ่นเท่านั้น เมื่อผู้ดูแลให้หน้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<malila>พอแตกใบอ่อนเป็นมะลิลา</malila>
```

```
</mali>
```

```
<?xml-stylesheet type="text/xsl" href=""?>
```

บรรทัดที่ 1 นั้นหมายความว่าเราประกาศเอกสารนี้เป็นเอกสาร XML และมีการเข้ารหัสอักขระเป็น windows-874 เพื่อให้ใช้ภาษาไทยได้ จากนั้นเราจะเห็นสิ่งที่เราคุ้นเคยที่เราเรียกว่า tag แต่จริงๆแล้วภาษา XML จะแบ่งโครงสร้างเป็น 2 ส่วนใหญ่ๆ คือ Tag และ Element สามารถอธิบายเพิ่มเติมได้จากตัวอย่างที่ 2

ตัวอย่างที่ 2

```
<root>
```

```
<element>
```

```
<tag></tag>
```

```
</element>
```

```
</root>
```

ความหมายของ Tag กับ Element

Tag

สำหรับใน XML แล้ว tag มีความหมายในลักษณะเดียวกับที่ใช้ใน HTML Tag คือข้อความที่อยู่ระหว่างสัญลักษณ์ < และ >

Tagเปิด(Start

```
<book>
```

จากตัวอย่างที่แสดง ด้านบนถูกเรียกว่า Tag เปิดดังนั้น Tag เปิดจึงมีสัญลักษณ์คือ <... >

Tagปิด (End

```
</book>
```

Tag ที่ถูกเรียกว่า Tag ปิด ต่อเมื่อใน Tag มีเครื่องหมาย / อยู่หลังสัญลักษณ์ < ดังนั้นลักษณะของ Tag ปิดจึงมีรูปแบบคือ </... > หากพิจารณาระหว่าง Tag เปิดกับ Tag ปิดแล้ว ข้อแตกต่างอีกข้อหนึ่งคือ Tag เปิดเป็น Tag ที่สามารถใส่ข้อมูล attribute ลงไปภายใน tag ได้ แต่ tag ปิดจะไม่ทำกัน

Element

ในที่นี้คือโครงสร้างหลักของ XML ซึ่งอยู่ในรูปของ Tag เช่นเดียวกันตามตัวอย่างข้างบน

Element คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<root>
  <element>
    </element>
</root>

```

จะมีลักษณะซ้อนกันเป็นชั้นๆ โดย root element เป็น element แรกสุดของเอกสาร XML

Element ใช้เป็นส่วนประกอบของเนื้อหาของเอกสาร XML และ Attribute เพื่อให้ผู้อ่านได้เข้าใจความหมายของศัพท์หลายๆ คำศัพท์ก่อนที่จะนำไปใช้ในการสร้าง XML ซึ่งจำเป็นอย่างมากพิจารณาคำจำกัดความของ element

```
<chap number="1">Text for chapter 1</chap>
```

สังเกตจากลักษณะรูปร่างด้านบนทั้งหมดตั้งแต่ <chap... จนถึง </chap> ถูกเรียกว่า element หรือถ้ามองง่ายๆ คือ element เริ่มต้นที่ Tag เปิด และสิ้นสุดที่ Tag ปิดใน Tag คำสั่งเดียวกัน

Content

เนื้อความหรือ Content ถือได้ว่าเป็นข้อมูลเพื่อใช้ในการแสดงให้ผู้อ่านเอกสารได้เห็น หรือกล่าวอีกนัยหนึ่งคือ Content อยู่หลัง Tag เปิด และจบก่อนที่จะถึง Tag ปิดนั่นเอง

Attribute

Attribute คือข้อมูลความหมายเพิ่มเติม แน่นอนถ้ากล่าวถึงสิ่งของหากเราไม่ได้ให้ความหมายเพิ่มเติมเราก็จะไม่ทางรู้หรือแยกแยะสิ่งที่คล้ายๆ กันออกจากกันได้ เช่น ถ้าพูดถึง บทในหนังสือ แต่นี่เป็นความหมายโดยรวม แต่ถ้าบอกว่าบทที่ 1 ในหนังสือ เลข 1 ในที่นี้คือความหมายเพิ่มเติมให้กับบท

Attribute จะถูกบรรจุอยู่ภายใน Tag เปิด และ Attribute นี้ในบางครั้งอาจจะมีหรือไม่มีก็ได้ รวมถึงถ้ามีอาจมีได้มากกว่าหนึ่งตัวใน Tag เปิด

Dom(Document Object Model)

การใช้งาน XML นั้นเราสามารถที่จะทำอะไรกับข้อมูลในแอปพลิเคชันที่เราเขียนขึ้นได้ ไม่ใช่เพียงการเข้าถึงเท่านั้นแต่ยังรวมถึงการแก้ไขและเพิ่มเติมเอกสาร XML ที่เรามีอยู่อีกด้วย จึงเกิดมี Document Object Model(DOM) ขึ้นเพื่อช่วยในการทำงานกับเอกสาร XML ทั้งหมด

DOM มักจะถูกใส่เข้าไปเป็นชั้นที่คั่นกลางระหว่าง XML Parser กับแอปพลิเคชันที่ต้องการใช้ข้อมูลในเอกสาร XML ซึ่งหมายความว่า parser จะอ่านข้อมูลจากเอกสาร XML ให้กับ DOM จากนั้น DOM จะถูกใช้โดยแอปพลิเคชันระดับสูงกว่า แอปพลิเคชันสามารถทำงานกับเอกสาร XML ได้ตามต้องการ รวมถึงการใส่ DOM เข้าไปใน DOM ตัวอื่น ซึ่งในแต่ละภาษาก็จะมีแบบจำลอง DOM เป็นของตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ไฟล์ XML ที่ถูกนำมาใช้ในระบบมีส่วนประกอบดังนี้

เนื่องจากไฟล์ XML ที่ถูกนำมาใช้จะต้องบรรจุข้อมูลที่ตรงกับ ค่าคุณลักษณะทางไฟฟ้าที่ ถูกบันทึกอยู่ในฐานข้อมูล ซึ่งจะต้องมีชนิดของข้อมูลที่ตรงกัน มีความหมาย ดังนี้

MeasureNo คือ หมายเลขของมิเตอร์ที่ถูกบันทึกลงฐานข้อมูลใช้สำหรับระบุตัวคนได้

StartTime คือ เวลาเริ่มต้นที่ทำการบันทึกเหตุการณ์นั้นๆ

Resolution คือ ค่าความละเอียดของการบันทึกเทียบกับเวลา

Method ไม่ถูกนำมาใช้งานมีค่าเป็นศูนย์เสมอ

TrigPhase คือ จำนวนเฟสไฟฟ้ามิเตอร์ทำการเก็บข้อมูลอยู่

Duration คือ ค่าเฉลี่ยของข้อมูลทางไฟฟ้าที่เก็บมา

Depth คือ ระดับความรุนแรง

Uref คือ โวลเตจอ้างอิงของมิเตอร์วัดคุณภาพไฟฟ้าตัวนั้น

Deviation คือ ค่าความคลาดเคลื่อนของมิเตอร์วัดคุณภาพไฟฟ้า

PreTrig ไม่ถูกนำมาใช้งานมีค่าเป็น สิบเสมอ

UW2R คือ โวลเตจอ้างอิงของไฟ U1

IW2R คือ กระแสอ้างอิงของไฟ I1

U1 คือ โวลเตจของไฟเฟสที่หนึ่ง

U2 คือ โวลเตจของไฟเฟสที่สอง

U3 คือ โวลเตจของไฟเฟสที่สาม

U4 คือ โวลเตจของไฟเฟสที่สี่

I1 คือ กระแสของไฟเฟสที่หนึ่ง

I2 คือ กระแสของไฟเฟสที่สอง

I3 คือ กระแสของไฟเฟสที่สาม

I4 คือ กระแสของไฟเฟสที่สี่

UW2R2 คือ โวลเตจอ้างอิงของไฟ U2

UW2R3 คือ โวลเตจอ้างอิงของไฟ U3

UW2R4 คือ โวลเตจอ้างอิงของไฟ U4

IW2R2 คือ กระแสอ้างอิงของไฟ I2

IW2R3 คือ กระแสอ้างอิงของไฟ I3

IW2R4 คือ กระแสอ้างอิงของไฟ I4

Length คือ ความยาวของการบันทึกข้อมูลเทียบเวลามาตรคือ 250 ตำแหน่ง

UW2R5 คือ โวลเตจอ้างอิงของไฟ U5

IW2R5 คือ กระแสอ้างอิงของไฟ I5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

U5 คือ โวลเตจของไฟเฟสที่ห้า

I5 คือ กระแสของไฟเฟสที่ห้า

ชนิดของข้อมูลในแต่ละหัวข้อเรียงลำดับดังนี้

MeasureNo ชนิดเลขจำนวนเต็ม

StartTime ชนิด DateTime

Resolution ชนิด เลขทศนิยม double

Method ชนิดเลขจำนวนเต็ม

TrigPhase ชนิดเลขจำนวนเต็ม

Duration ชนิด เลขทศนิยม double

Depth ชนิด เลขทศนิยม double

Uref ชนิด เลขทศนิยม double

Deviation ชนิด เลขทศนิยม double

PreTrig ชนิดเลขจำนวนเต็ม

UW2R ชนิด เลขทศนิยม double

IW2R ชนิด เลขทศนิยม double

U1 ชนิดอาร์เรย์ของ binary

U2 ชนิดอาร์เรย์ของ binary

U3 ชนิดอาร์เรย์ของ binary

U4 ชนิดอาร์เรย์ของ binary

I1 ชนิดอาร์เรย์ของ binary

I2 ชนิดอาร์เรย์ของ binary

I3 ชนิดอาร์เรย์ของ binary

I4 ชนิดอาร์เรย์ของ binary

UW2R2 ชนิด เลขทศนิยม double

UW2R3 ชนิด เลขทศนิยม double

UW2R4 ชนิด เลขทศนิยม double

IW2R2 ชนิด เลขทศนิยม double

IW2R3 ชนิด เลขทศนิยม double

IW2R4 ชนิด เลขทศนิยม double I4

Length ชนิดเลขจำนวนเต็ม

UW2R5 ชนิด เลขทศนิยม double

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IW2R5 ชนิด เลขทศนิยม double

U5 ชนิดอาเรย์ของ binary

I5 ชนิดอาเรย์ของ binary

โครงสร้างของไฟล์มีรูปแบบดังนี้

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<ArrayOfBaseData xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<BaseData>
```

```
<MeasureNo></MeasureNo>
```

```
<StartTime></StartTime>
```

```
<Resolution></Resolution>
```

```
<Method></Method>
```

```
<TrigPhase></TrigPhase>
```

```
<Duration></Duration>
```

```
<Depth></Depth>
```

```
<Uref></Uref>
```

```
<Deviation></Deviation>
```

```
<PreTrig></PreTrig>
```

```
<UW2R></UW2R>
```

```
<IW2R></IW2R>
```

```
<U1> </U1>
```

```
<U2> </U2>
```

```
<U3> </U3>
```

```
<U4> </U4>
```

```
<I1> </I1>
```

```
<I2> </I2>
```

```
<I3> </I3>
```

```
<I4> </I4>
```

```
<UW2R2></UW2R2>
```

```
<UW2R3></UW2R3>
```

```
<UW2R4></UW2R4>
```

```
<IW2R2></IW2R2>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





B3gHeAd4B3gHeAd4B3gHeAd4B3gHeAd4B3gHeAd4B3gHeAd4B48HjwePB48HjwePB48HjwePB48H

jwePB48HjwePB48HjwePB48HjwePB48HjwePB48HjwePB48HjwePB48HjwePB48HjwePB48HjwePB48HjweP

B2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gH

aAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdo

B2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHaAdoB2gHcwdzB3MHcwdzB3MHcwdzB3MH

cwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdz

B3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MHcwdzB3MH

cwdzB3MHcwdzB3MHcwdzB3MHfgd+B34Hfgd+B34Hfgd+B34Hfgd+B34Hfgd+B34Hfgd+B34Hfgc=</U3>

<U4>DAA

AA

AA

AA

AA

AA

AA

AA

AA

AA

AA

AA

AA

AA

AA

AA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AA  
AA=</U4>

<I1>jQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0Cj  
QKNAo0CjQKNAo0CjQKN  
Ao0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAroCugK6Ar  
oCugK6AroCugK6AroC  
ugK6AroCugK6AroCugK6AroCugK6AroCugK6AroCugK6AroCugK6AroCugK6Ar  
oCugK6AroCugK6  
AmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJs  
AmwCbAJsAmwCbAJsAmwC  
bAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCb  
AJsAmwCbAJsAmwCbAJs  
AmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCggKC  
AoICggKCAoICggKCAoIC  
ggKCAoICggKCAoICggKCAoICggKCAoICggKCAoICggKCAoICggKCAoICggKC  
AoICggKCAoICggKC  
AoICggKCAoICggKCAoICggKCAoICggKCAoICggKCAoICggKCAoICggKCAoIC  
ggKCAoICggKCAoIC  
ggKCAoICggKCAoICggKCAoICmQKZApkCmQKZApkCmQKZApkCmQKZApkC  
mQKZApkCmQKZApkCmQI=</I1>

<I2>jQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0Cj  
QKNAo0CjQKNAo0CjQKN  
Ao0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAo0CjQKNAroCugK6Ar  
oCugK6AroCugK6AroC  
ugK6AroCugK6AroCugK6AroCugK6AroCugK6AroCugK6AroCugK6AroCugK6Ar  
oCugK6AroCugK6  
AmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJs  
AmwCbAJsAmwCbAJsAmwC  
bAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCb  
AJsAmwCbAJsAmwCbAJs  
AmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAmwCbAJsAoICggKCA  
oICggKCAoICggKCAoIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=
```

```
<UW2R2>6.5996618270874</UW2R2>
```

```
<UW2R3>6.5996618270874</UW2R3>
```

```
<UW2R4>0.0329983085393906</UW2R4>
```

```
<IW2R2>0.452548235654831</IW2R2>
```

```
<IW2R3>0.452548235654831</IW2R3>
```

```
<IW2R4>0.00188561796676368</IW2R4>
```

```
<Length>250</Length>
```

```
<UW2R5>6.5996618270874</UW2R5>
```

```
<IW2R5>0.452548235654831</IW2R5>
```

```
<U5>AA==</U5>
```

```
<I5>AA==</I5>
```

```
</BaseData>
```

กระบวนการสร้าง XML ไฟล์

กระบวนการสร้าง XML ไฟล์ นี้ได้มาจากการรวมอ็อบเจกต์(Object)ของคลาสพื้นฐานข้อมูล ที่ระบบได้สร้างขึ้น และภายในนั้นได้บรรจุข้อมูลตามฟิลด์ที่เรียงกันตามลำดับ โดยได้นำมาผ่าน กระบวนการแปลงเป็น XML ที่เรียกว่า XML Serialization โดยในคลาสที่จะทำการ Serialize ได้ ต้องมีคุณสมบัติดังนี้

1. ต้องเป็น public class
2. member ที่ต้องการให้ serialize ได้ต้องเป็น public
3. constructor ของคลาสต้อง ไม่มี parameter

จะแสดงตัวอย่างโค้ดกับไฟล์ XML ที่ได้จากการ serialize เพื่อให้ง่ายต่อการเข้าใจ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[XmlAttribute("CartItem")]
public class ShoppingCartItem
{
    [XmlAttribute] public Int32 productId;
    public decimal price;
    public Int32 quantity;
    [XmlIgnore] public decimal total;
    public ShoppingCartItem()
    {
    }
}
```

ผลลัพธ์ที่ได้จากการ Serialize ได้ดังนี้

```
<?xml version="1.0" ?>
<CartItem productId="100">
  <price>10.25</price>
  <quantity>2</quantity>
</CartItem>
```

ส่วนขั้นตอนการถอดข้อมูลที่เป็น XML ไฟล์นี้คืนกลับเป็น ออปเจ็ทเช่นเดิมเรียกว่า  
Deserialize โดยมีขั้นตอนดังนี้

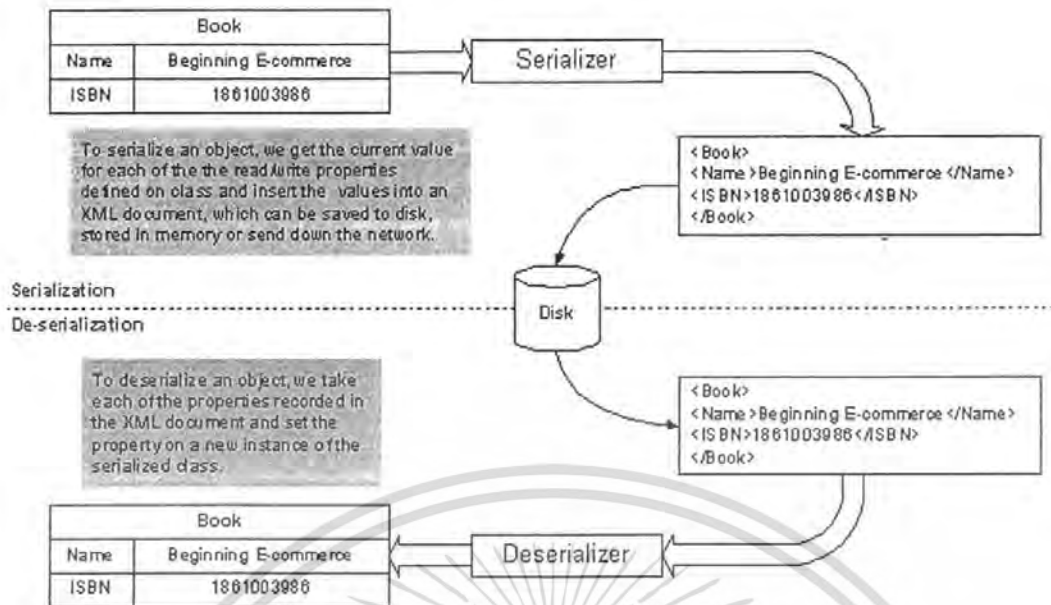
1. สร้าง stream, TextReader, or XmlReader object เพื่ออ่านข้อมูลจาก serialized input  

```
FileStream fs = new FileStream("SerializedDate.XML", FileMode.Open);
```
2. สร้าง XmlSerializer object (in the System.Xml.Serialization namespace) โดยใส่ชนิดข้อมูลให้ถูกต้อง

```
XmlSerializer xs = new XmlSerializer(typeof(DateTime));
```

3. เรียก method XmlSerializer.Deserialize เพื่อ deserialize object และ casting ให้ตรงกับชนิด object

```
DateTime previousTime = (DateTime)xs.Deserialize(fs);
```



รูปที่ 3.10 แสดงกระบวนการ Serialize และการ De-serialize

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

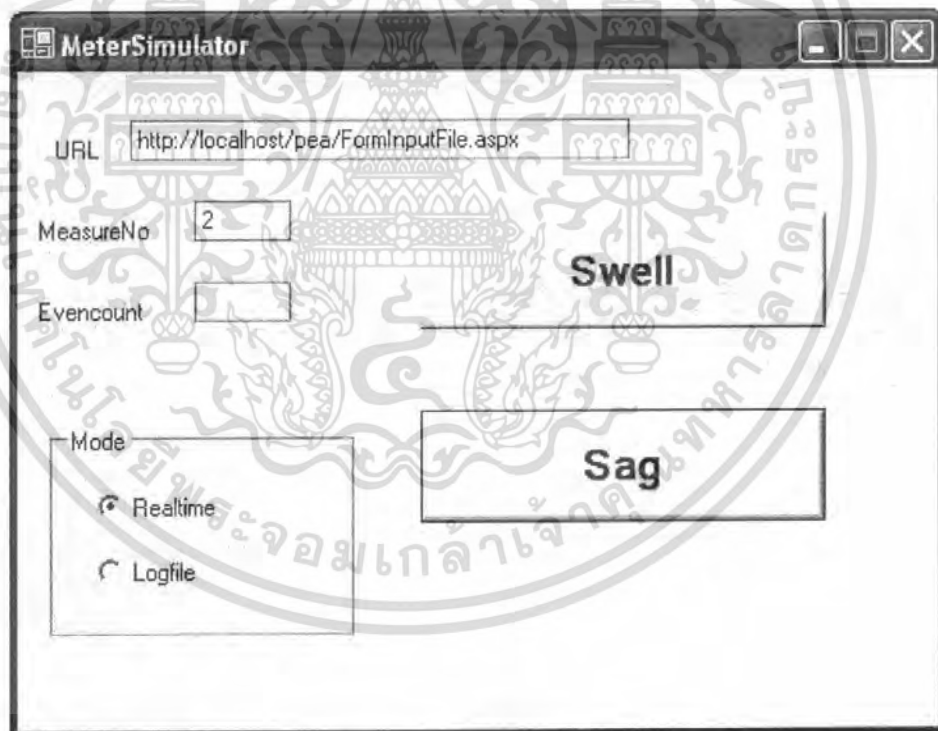
### ผลการทดลอง

#### 4.1 การใช้งาน

ในการใช้งานระบบอินเทอร์เน็ตเฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้านั้นเราจะทำการใช้งานโดยผ่านโปรแกรม มิเตอร์ซิมูเลเตอร์(Meter Simulator) ที่เราได้สร้างขึ้น แล้วเรียกไปที่ URL ที่เรากำหนดไว้เป็นฟอร์มรับข้อมูล เช่น <http://localhost/pea/FormInputFile.aspx> ซึ่งในโปรแกรมซิมูเลเตอร์ของมิเตอร์นี้จะมีคุณสมบัติในการทำงานดังนี้

##### 1. Realtime Mode

- กรอก URL ของอินเทอร์เน็ตเฟส Server
- กรอกหมายเลขของมิเตอร์ที่ต้องการจำลอง
- ในโหมดนี้จะไม่สามารถเลือกจำนวนเหตุการณ์ซึ่งจะถูกสร้างขึ้น



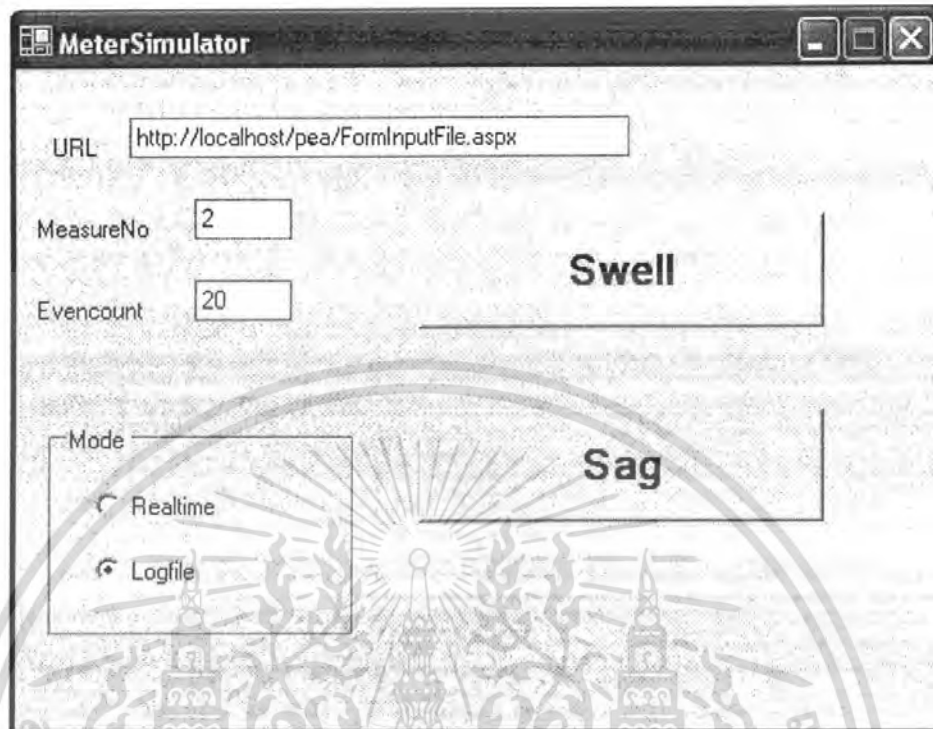
รูปที่ 4.1 ภาพแสดง Meter Simulator ในโหมด Real time

##### 2. Logfile Mode

- กรอก URL ของอินเทอร์เน็ตเซิร์ฟเวอร์
- กรอกหมายเลขมิเตอร์ที่ต้องการจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กรอกจำนวน เหตุการณ์ในการสร้าง ล็อกไฟล์(logfile)



รูปที่ 4.2 ภาพแสดง Meter Simulator ในโหมด Logfile

#### 4.1.1 การทำงานในปุ่มของไฟตกและไฟเกิน

ไฟเกิน(Swell) จากความหมายของคำว่าไฟเกินก็คือ ค่าคุณลักษณะทางไฟฟ้าต้องเกินกว่าค่ามาตรฐานของระบบที่ทำงานได้ทั่วไป จากการศึกษาพบว่า ค่าไฟฟ้ามาตรฐานของระบบจะอยู่ในช่วง 7000 KVA ซึ่งหากเพิ่มเกินกว่านี้มากก็จะถือว่าเป็นไฟล้นเกิน

ไฟตก(Sag) จากความหมายของคำว่าไฟตกก็คือค่าคุณลักษณะทางไฟฟ้าต้องมีค่าน้อยกว่าค่ามาตรฐานของระบบที่ทำงานได้ทั่วไป ซึ่งก็คือค่าเดียวกัน ช่วง 7000 KVA น้อยกว่านี้พ้นหน่วยก็ถือว่าเป็นไฟตกแล้ว

การสร้างข้อมูลที่มีลักษณะดังนี้ โดย เราจะระบุว่า ข้อมูลของไฟล์ในแต่ละช่วงถูกเก็บไว้ในอาร์เรย์ จำนวน 500 ตำแหน่ง แบ่งได้ดังนี้คือ ถ้าหากเป็นตำแหน่งที่เริ่มที่ 0 ถูกนำไปแปลงได้เป็นเลขหลักพันของกราฟดังตัวอย่างเช่น ที่ U1 จะถูกเก็บข้อมูลด้วย อาร์เรย์ 500 ลำดับดังนี้

U1[0] = 94    ค่าย่อย

U1[1] = 7    ค่าหลัก

U1[2] = 93    ค่าย่อย

U1[3] = 7    ค่าหลัก

U1[4] = 94    ค่าย่อย

เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

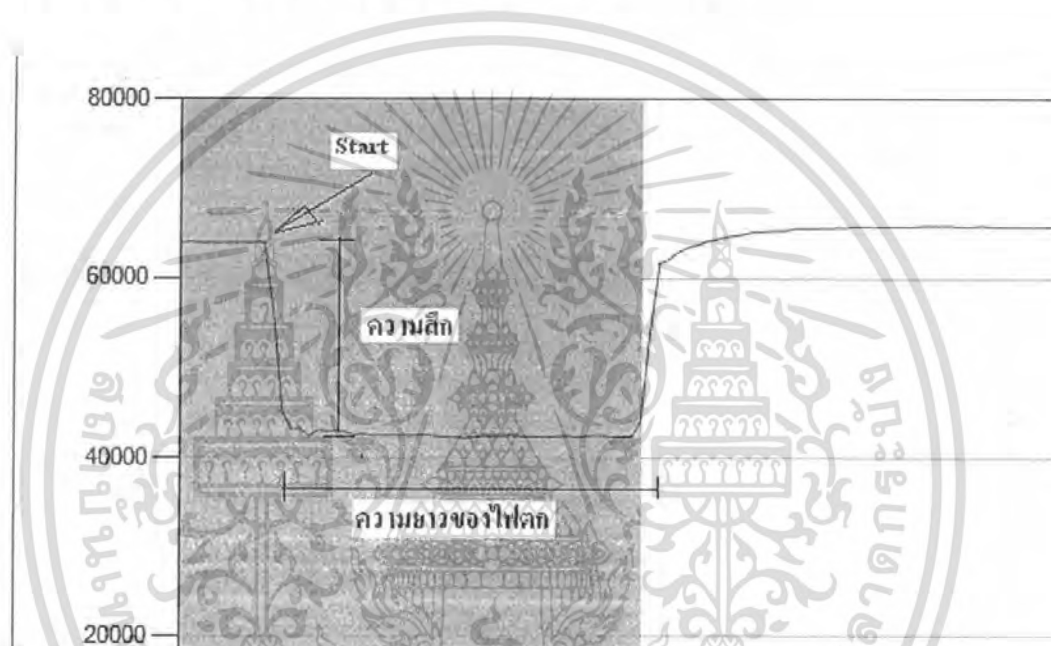
$U1[\dots] = \dots$  ค่าหลัก, ค่าย่อย

$U1[499] = 7$  ค่าหลัก

ดังนั้นเมื่อคิดตามช่วงเวลาในการเก็บค่าของมิเตอร์คือ จุดละ 0.01 วินาที ซึ่งก็คือ จุดหนึ่งจะใช้ อาร์เรย์ 2 ช่องในการคำนวณ เมื่อนำ  $0.01 * 256$  ก็จะได้ความยาวของกราฟที่ถูกวาด ก็คือ 2.5 วินาทีนั่นเอง สมการที่ใช้ในการวาดกราฟมีดังนี้

$$U1[j-1] + (U1[j]*256)*Uref \quad (4.1)$$

เมื่อ  $j$  คือตำแหน่งของเลขที่ภายในอาร์เรย์  $Uref$  คือค่าอ้างอิงที่ใช้ในการคำนวณ



รูปที่ 4.3 แสดงลักษณะกราฟไฟตก

จากรูปค่าพื้นฐานสามค่านี้ในกรณีไฟตกนั้นคือรู้ระยะเวลาเริ่มต้น ความลึกหรือช่วงในการตกว่าจะตกมากหรือน้อยเท่าใด และ ความยาวในการตกหรือระยะเวลาในการตกนั่นเอง ค่าทั้งสามนั้นจะถูกสร้างด้วยวิธีการ แรนดอมช่วงซึ่งเราทราบค่าอยู่แล้วจากทฤษฎีขั้นต้นทำให้เราได้ข้อมูลที่เป็นไฟตกได้ดังนี้

$$U1[0] = 94$$

$$U1[1] = 7$$

$$U1[2] = 93$$

$$U1[3] = 7$$

$$U1[4] = 97$$

$$U1[5] = 6$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$U1[6] = 115$$

$$U1[7] = 5$$

$$U1[8] = 113$$

$$U1[9] = 4$$

$$U1[10] = 97$$

$$U1[11] = 4$$

$$U1[12] = 93$$

$$U1[13] = 4$$

$$U1[14] = 97$$

$$U1[15] = 6$$

$$U1[16] = 117$$

$$U1[17] = 7$$

$$U1[18] = 115$$

$$U1[19] = 7$$

$$U1[...499] = 7$$

กรณีไฟเกิน ได้ดังนี้

$$U1[0] = 94$$

$$U1[1] = 7$$

$$U1[2] = 91$$

$$U1[3] = 7$$

$$U1[4] = 148$$

$$U1[5] = 9$$

$$U1[6] = 132$$

$$U1[7] = 11$$

$$U1[8] = 152$$

$$U1[9] = 11$$

$$U1[10] = 137$$

$$U1[11] = 11$$

$$U1[12] = 146$$

$$U1[13] = 11$$

$$U1[14] = 176$$

$$U1[15] = 9$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

U1[16] = 54

U1[17] = 7

U1[18] = 45

U1[19] = 7

U1[...499] = 7

## 4.2 ผลการทดลองใช้งาน Meter Simulator

### 4.2.1 เมื่อเปิดเข้าสู่ระบบ

ระบบมิเตอร์ซิมูเลเตอร์ได้กำหนดค่าเริ่มต้นพื้นฐานไว้ดังนี้

- URL คือตำแหน่งของเซิร์ฟเวอร์เป้าหมายในการส่งข้อมูล ในที่นี้ได้กำหนดค่าเริ่มต้นไว้ดังรูปที่ 4-2 ซึ่งสามารถเปลี่ยนได้ในภายหลัง
- MeasureNo คือหมายเลขของมิเตอร์ที่ใช้ในการแยกมิเตอร์ซึ่งถูกติดตั้งไว้ในระบบแล้ว ซึ่งหากหมายเลขที่กรอกไปไม่ตรงกับมิเตอร์ซึ่งถูกสร้างไว้แล้วนั้นก็จะไม่ถูกบันทึกลงในฐานข้อมูล
- Mode ที่เป็นค่าเริ่มต้นคือ Realtime Mode ซึ่งจะทำให้การส่งเหตุการณ์ปัจจุบันของ มิเตอร์ไปเพียงเหตุการณ์เดียว และหากเปลี่ยนเป็น Logfile Mode นั้นจะต้องมีการกำหนดค่าเหตุการณ์ที่จะส่งไปในแต่ละครั้ง

### 4.2.2 เมื่อทำการส่งข้อมูล

ระบบมิเตอร์ซิมูเลเตอร์จะทำการแรนคอมสร้างค่าที่เป็นคุณลักษณะทางไฟฟ้าขึ้นมาตามจำนวนเหตุการณ์ที่กำหนดโดย ฟอर्मแมตที่นำมาใช้ คือ XML ซึ่งในกระบวนการ Serialization ของ .NET Framework ในการทำ XML Serialization แล้วจะได้ไฟล์ดังนี้





## บทที่ 5

# บทวิจารณ์และสรุป

### 5.1 บทวิจารณ์และสรุป

โปรแกรมอินเทอร์เน็ตเฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้า ที่พัฒนาขึ้นมาสามารถทำงานบนเว็บแอปพลิเคชันได้เป็นอย่างดี โดยสามารถแปลงข้อมูลและป้องกันข้อมูลได้อย่างถูกต้อง สามารถคัดแยกข้อมูลของมิเตอร์ที่เข้ามาว่ามีการแอดไว์ในระบบได้อย่างถูกต้อง สามารถแยกประเภทไฟล์ที่เข้ามาได้อย่างถูกต้องและหากมีข้อผิดพลาด มิเตอร์ซิมูเลเตอร์ก็สามารถแสดงออกมาได้อย่างถูกต้อง สำหรับโปรแกรมมีการใช้งานที่เรียบง่าย การทำงานร่วมกับฐานข้อมูลทำได้เป็นอย่างดี

เมื่อเปรียบเทียบกับระบบเดิมซึ่งใช้เจ้าหน้าที่เดินทางไปติดตั้งมิเตอร์ในสถานที่หลายๆแห่ง ซึ่งมีความห่างไกลกันนั้น ระบบที่พัฒนาขึ้นมาได้ช่วยให้การไฟฟ้าลดค่าใช้จ่ายในการจ้างเจ้าหน้าที่ในการขับรถเพื่อไปเก็บข้อมูล ลดการใช้น้ำมันเพราะไม่ต้องขับรถไป ข้อมูลของมิเตอร์ตามต่างๆสามารถส่งเข้ามาที่ศูนย์ข้อมูลได้ทันที ทำให้ทางการไฟฟ้าสามารถแก้ไขข้อขัดข้องได้อย่างรวดเร็ว ช่วยเพิ่มศักยภาพของการให้บริการมากขึ้น

เทคโนโลยีที่นำมาใช้ในการพัฒนาระบบอินเทอร์เน็ตเฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้า ในโครงการนี้ เป็นเทคโนโลยี .NET Framework และ XML เพื่ออำนวยความสะดวกให้ระบบสามารถทำงานได้

### 5.2 ปัญหาที่เกิดขึ้นในการพัฒนาและแนวทางการแก้ปัญหา

1. ปัญหาเรื่องฟอร์มเมตของข้อมูลซึ่งไม่สามารถถอดฟอร์มเมตของข้อมูลตัวอย่างได้ ทำให้ทำได้เพียงเรียนแบบข้อมูลที่ถูกบันทึกอยู่ในฐานข้อมูลแล้ว แล้วสร้างเป็นฟอร์มเมต XML ซึ่งใช้เป็นตัวอย่างของมาตรฐานกลางในการสื่อสารข้อมูล
2. ปัญหาที่เกิดจากไม่มีมิเตอร์จริงเพื่อใช้ในการทดสอบระบบทำให้ไม่รู้ว่ามีปัญหาจริงในการเชื่อมต่อกับระบบของมิเตอร์เป็นอย่างไร

### 5.3 ข้อจำกัดของระบบ

ระบบอินเทอร์เน็ตเฟสรับข้อมูลจากมิเตอร์วัดคุณภาพไฟฟ้านี้ทำงานบนระบบอินเทอร์เน็ต ดังนั้นมิเตอร์ที่สามารถเชื่อมต่อกับระบบได้จึงต้องมีสายโทรศัพท์ไปถึงหรือสามารถต่ออินเทอร์เน็ตได้ในทางใดทางหนึ่ง และข้อจำกัดอีกด้านคือด้านความเร็วในการส่งข้อมูลของอินเทอร์เน็ตก็เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อจำกัดของระบบเพราะเราไม่สามารถคาดการณ์ระยะเวลาในการส่งข้อมูลจากมิเตอร์มาถึงระบบได้ ทำให้เกิดการรอที่นานเกินไปก็จะเกิดข้อมูลที่สูญหายได้เพราะ เกิด time out เกิดขึ้นนั่นเอง

#### 5.4 แนวทางการพัฒนา

1. สามารถเพิ่มเติมฟังก์ชันในการถอดรหัสข้อมูลตามฟอร์แมตต่างๆ ของมิเตอร์แต่ละแบบ เพื่อสามารถทำงานได้ครอบคลุมมิเตอร์ที่จะเพิ่มเข้ามาในระบบศูนย์กลางข้อมูลได้มากขึ้น
2. หากมีมิเตอร์ตัวอย่างที่ใช้จริงก็จะสามารถพัฒนาให้รองรับกับการทำงานจริงๆ ได้มากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

UniPower. 2006. การติดตั้งและใช้งานฐานข้อมูลPQSecure.

บัญชา ปะทีเตตัง. 2546. คู่มือการเขียนโปรแกรมด้วย Microsoft Visual C# .net ฉบับสมบูรณ์.  
บริษัทซีเอ็ดจำกัดมหาชน.

Institute of Electrical and Electronic Engineers, Inc. 2002. IEEE P1159.3D9 Draft:

Recommended Practice for the Transfer of Power Quality Data. [Online].

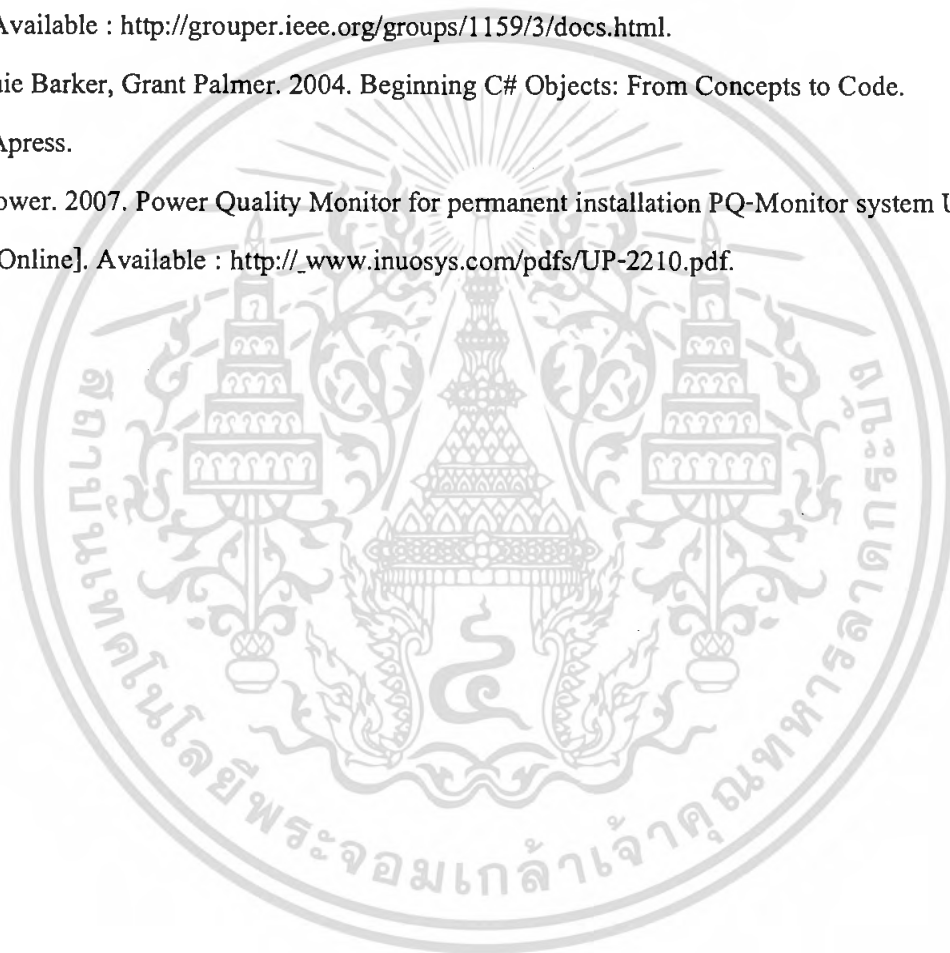
Available : <http://grouper.ieee.org/groups/1159/3/docs.html>.

Jacque Barker, Grant Palmer. 2004. Beginning C# Objects: From Concepts to Code.

Apress.

Unipower. 2007. Power Quality Monitor for permanent installation PQ-Monitor system UP-2210.

[Online]. Available : [http://\\_www.inuosys.com/pdfs/UP-2210.pdf](http://_www.inuosys.com/pdfs/UP-2210.pdf).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

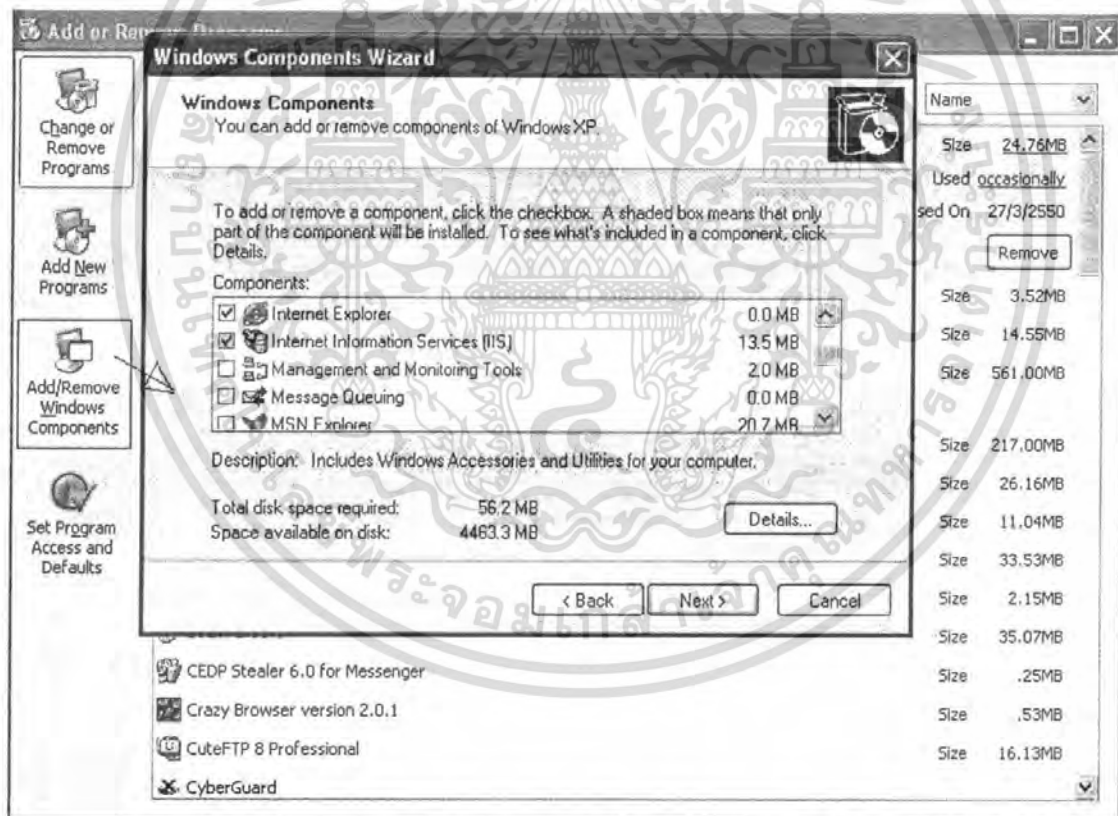
## ภาคผนวก ก

# การติดตั้ง IIS (Internet Information Server)

ขั้นตอนการติดตั้ง IIS มีดังนี้

### 1. ติดตั้ง IIS

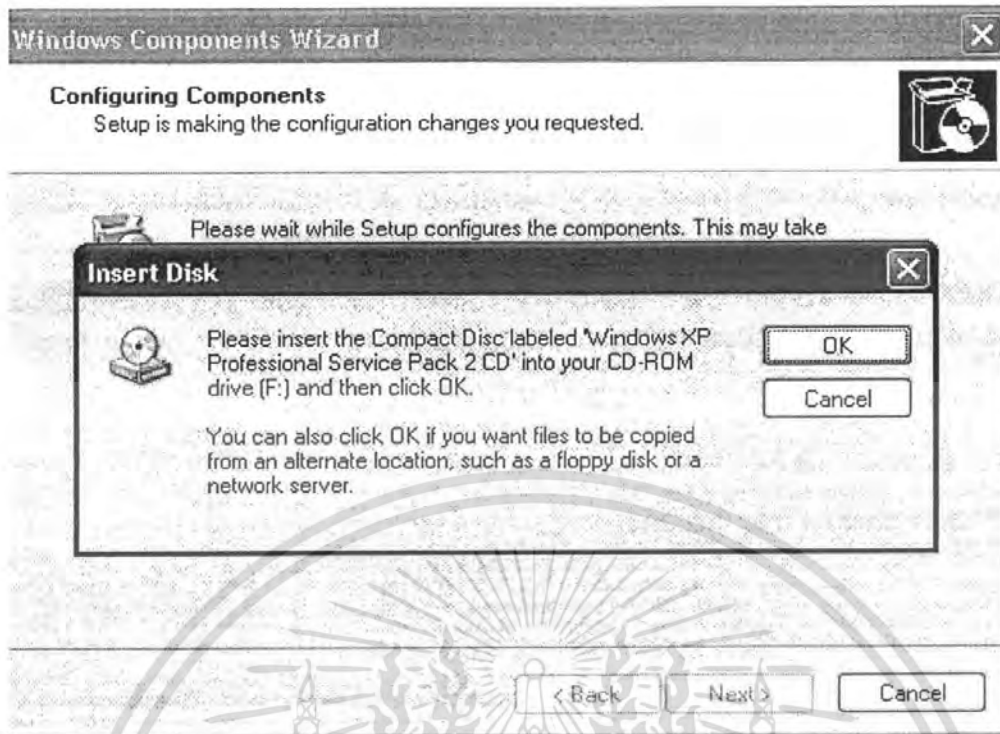
- 1.1 คลิกที่ Start > Settings > Control Panel
- 1.2 ดับเบิลคลิกที่ Add or Remove Programs
- 1.3 คลิกที่ Add/Remove Windows Components
- 1.4 คลิกเลือกที่ Internet Information Services (IIS) กรณีต้องการดูรายละเอียดเพิ่มเติม ให้คลิกเลือกที่ Details...



รูปที่ ก.1 แสดงเมนูสำหรับการติดตั้ง IIS ลงบนวินโดวส์ XP

- 1.5 คลิกปุ่ม Next> เพื่อเริ่มติดตั้ง
- 1.6 กรณีระบบถามหาตัวติดตั้ง Windows XP ให้คลิกปุ่ม OK แล้วทำการ Browse หาแผ่น CD ติดตั้ง Windows XP หีอง i386

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 แสดงขั้นตอนการติดตั้งเมื่อระบบตามหาตัวติดตั้ง Windows XP

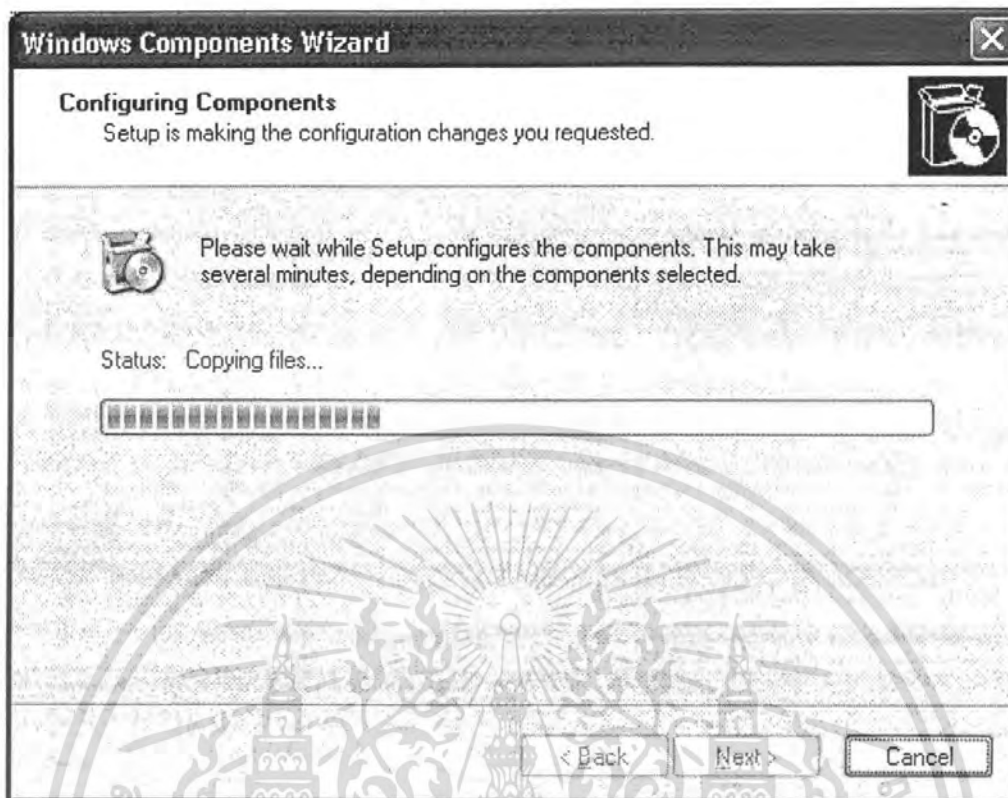


รูปที่ ก.3 แสดงการเลือกไฟล์ที่จะติดตั้ง

### 1.7 คลิกปุ่ม Open, OK

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้และจะทำการติดตั้ง IIS ดังรูป

• ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 แสดงช่วงเวลาในการติดตั้ง

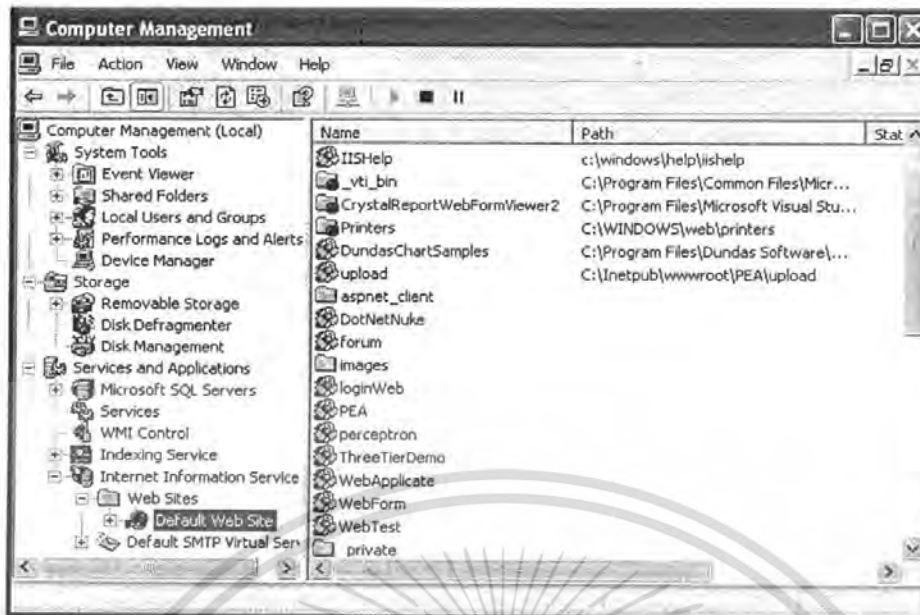
### 1.9 คลิกปุ่ม Finish

## 2. การปรับแต่ง IIS

2.1 ทำการคลิกขวาที่ไอคอน My Computer > Manage

2.2 คลิกที่ Services and Applications > Internet Information Services > WebSites > Default Web Site

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 แสดงโครงสร้างโฟลเดอร์ภายใน IIS

2.3 เริ่มปรับแต่งโดยการคลิกขวาที่ Default Web Site > Properties

2.4 แท็บ Web Site

- Description : ระบุชื่อเว็บไซต์ที่ต้องการเช่น www.kmitl.com
- IP Address : เลือก IP Address ที่ต้องการใช้เรียกเว็บไซต์ (หากต้องการให้เรียกได้ทุก IP ก็เลือกเป็น All Unassigned)
- TCP Port : พอร์ตในการเรียกใช้เว็บไซต์ เลือกเป็น 80

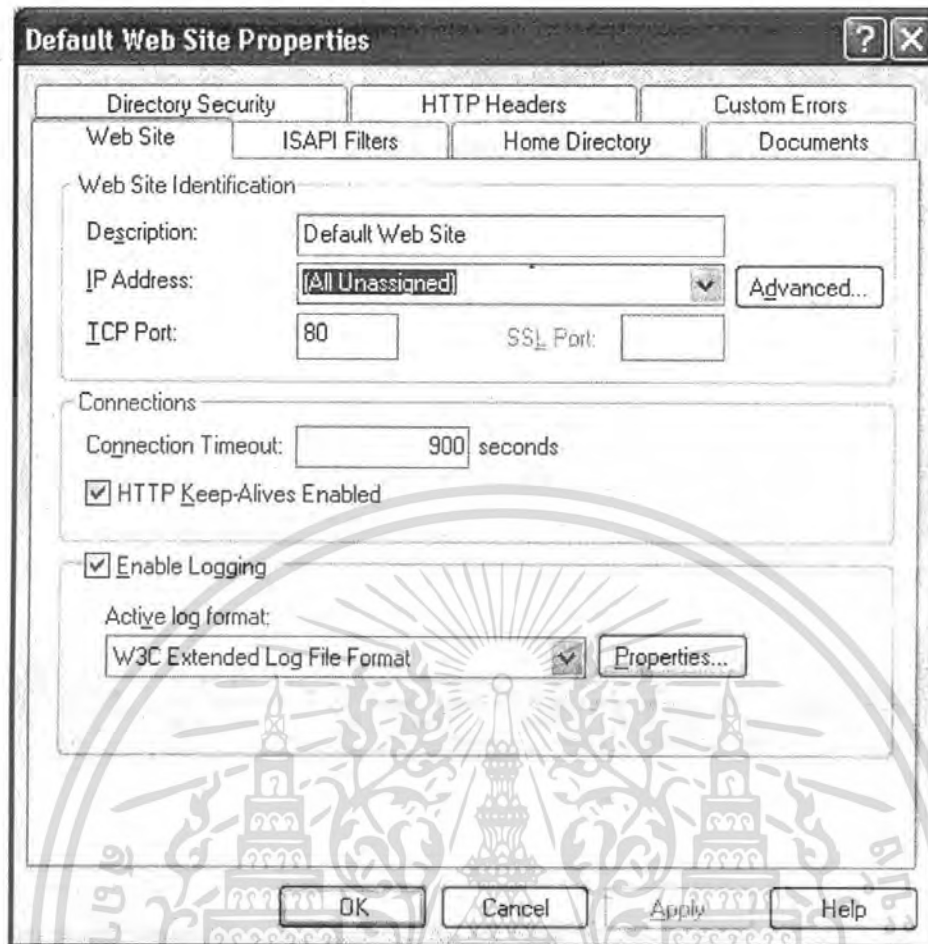
กรณีใช้ port 80 การเรียกเว็บไซต์ไม่จำเป็นต้องอ้างอิงพอร์ต เช่น

http://192.168.1.5 (เบื้องหลังเป็น http://192.168.1.5:80)

กรณีต้องการเปลี่ยนพอร์ต อาทิ เปลี่ยนเป็นพอร์ต 81 ตอนเรียกเว็บก็เรียกเป็น

<http://192.168.1.5:81>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ๓.๖ แสดงการปรับแต่งโดยการคลิกขวาที่ Default Web Site > Properties

2.5 แท็บ Home Directory Local Path : พาทที่ต้องการเก็บเว็บไซต์ ปกติจะเป็น

<drive:\>Inetpub\wwwroot

[ ] Script source access : ต้องการให้มองเห็นสคริปต์ไฟล์

[/] Read : อ่านไฟล์ได้

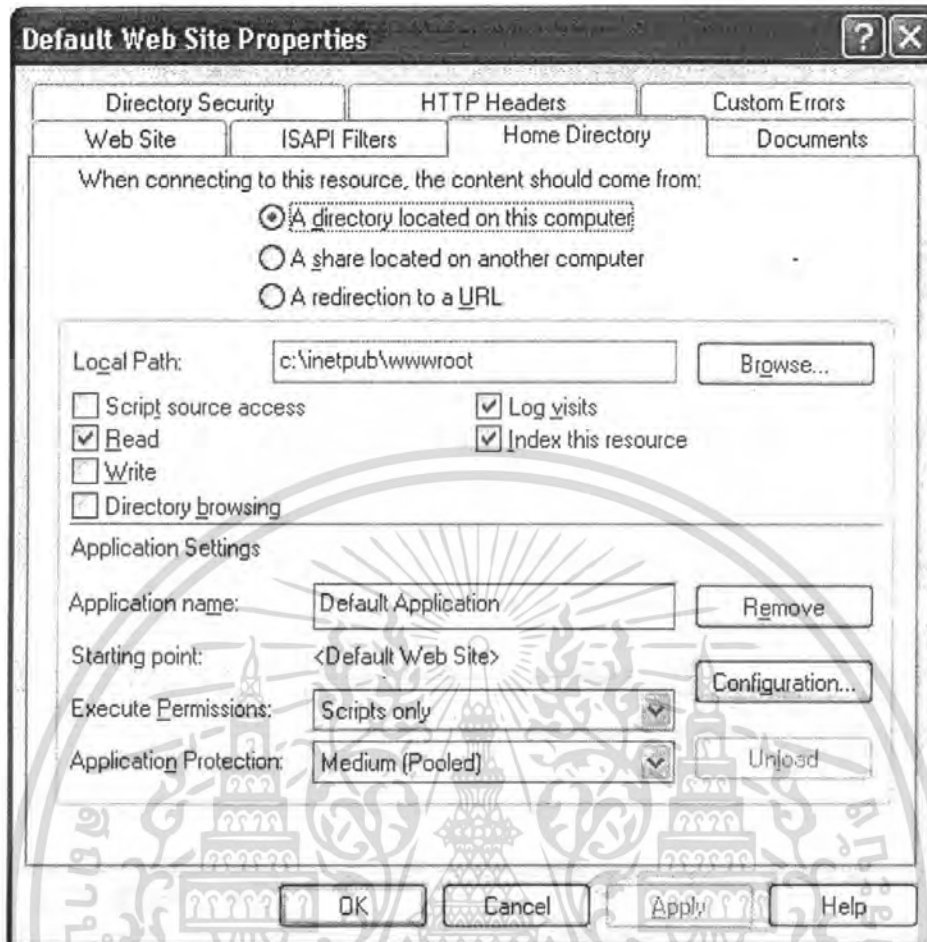
[ ] Write : เขียนทับไฟล์ได้

[ ] Directory browsing : ต้องการให้สามารถมองเห็นข้อมูลในไดเรกทอรีได้

[/] Log visits : มีการเก็บล็อกไฟล์ไว้

[/] Index this resource : มีการเรียกหาไฟล์ index

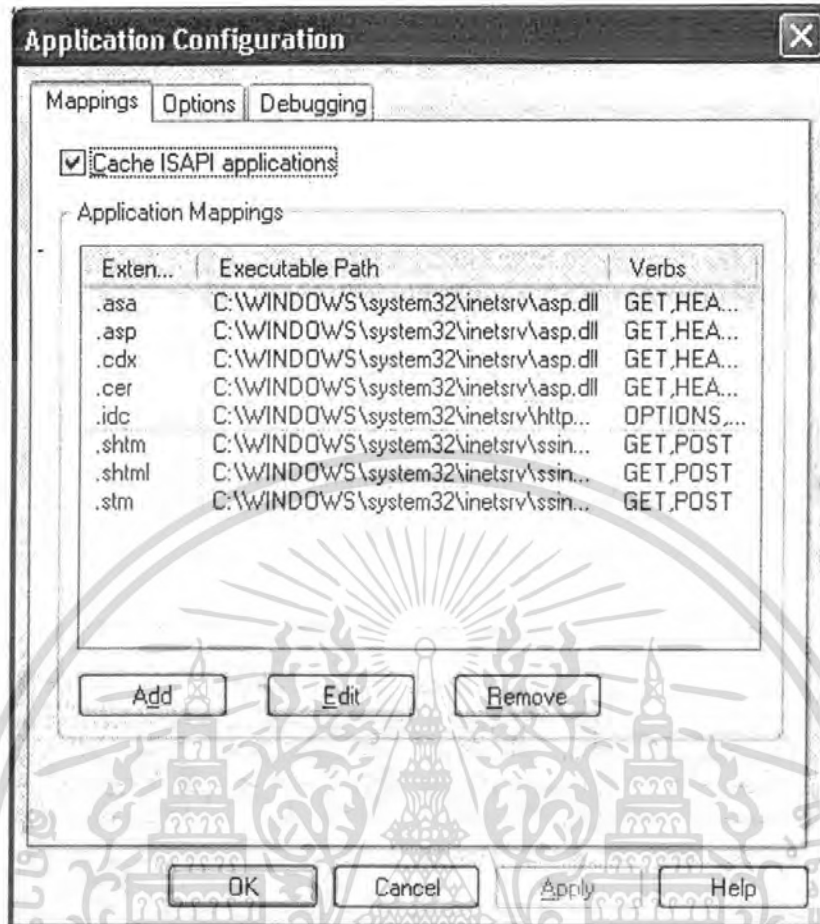
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.7 แสดงการปรับแต่ง โดยการคลิกขวาที่ Default Web Site > Properties

กรณีต้องการตรวจสอบภาษาที่สามารถใช้งานได้ ก็คลิกที่ ปุ่ม Configuration...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.8 แสดงการเลือกภาษา

2.6 แท็บ Document เพิ่ม ไฟล์ Home Page หรือไฟล์หน้าแรกที่ต้องการเรียกใช้งาน โดยการคลิกที่ปุ่ม Add

- index.html > ภาษา HTML
- index.shtml > ภาษา SHTML
- index.asp > ภาษา ASP
- index.aspx > ภาษา ASP.NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้