

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบการตรวจสอบความเสียหายจากการก่อสร้างอุโมงค์ใต้ดินแบบ Real – Time Real – time damage monitoring system for tunnel construction

โดย
นายฉัตรพด กมลภุณาสิริ
นายนิสิต รุ่งรัตนกิจก่าจร

เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

62417
17 ส.ค. 2549

b. 11623329

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REAL – TIME DAMAGE MONITORING SYSTEM FOR TUNNEL CONSTRUCTION

BY

MR. NUTTAPOL KAMOLKUNASIRI

MR. NITIT RUNGRATTANAKITKOMJOHN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์ ระบบการตรวจสอบความเสียหายจากการก่อสร้างอุโมงค์ใต้ดินแบบ
Real – Time

TITLE Real – time damage monitoring system for tunnel construction

โดย นายณัฐพล กมลคุณาสิริ รหัสนักศึกษา 45010239
นายนิธิศ รุ่งรัตนกิจกำจร รหัสนักศึกษา 45010404

อาจารย์ที่ปรึกษา อาจารย์เกล็ดดาว สุวรรณสวัสดิ์
ผศ.ดร. สุ��วีร์ สุวรรณสวัสดิ์

ระดับการศึกษา ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศ

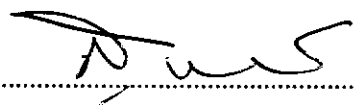
ปีการศึกษา 2548

วิทยานิพนธ์นี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรม
ศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดย
ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเรียบร้อยแล้ว



(อาจารย์เกล็ดดาว สุวรรณสวัสดิ์)

อาจารย์ที่ปรึกษา



(ผศ.ดร. สุ��วีร์ สุวรรณสวัสดิ์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ขออนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบการตรวจสอบความเสียหายจากการก่อสร้างอุโมงค์ใต้ดินแบบ Real – Time
ชื่อนักศึกษา	นายณัฐพล กมลคุณาสิริ รหัสนักศึกษา 45010239 นายนิธิศ รุ่งรัตนกิจกำจร รหัสนักศึกษา 45010404
อาจารย์ที่ปรึกษา	อาจารย์เกสรัดดาว สุวรรณสวัสดิ์ ผศ.ดร. สุชัยวีร์ สุวรรณสวัสดิ์
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

ในปัจจุบันได้มีการก่อสร้างอุโมงค์ใต้ดินภายในกรุงเทพฯ ซึ่งโครงสร้างในการก่อสร้างอาคารที่พักอาศัยต่าง ๆ ภายในกรุงเทพฯ ในอดีตนั้นไม่ได้มีการคำนึงถึงการก่อสร้างอุโมงค์ใต้ดินที่อาจก่อให้เกิดการทรุดตัวและเสียหาย จึงทำให้การก่อสร้างอุโมงค์ใต้ดินต้องเป็นไปอย่างระมัดระวังอย่างมาก เนื่องจากอาจเกิดการทรุดตัวและความเสียหายต่ออาคารที่พักอาศัยข้างบนในขณะที่ทำการก่อสร้างอุโมงค์ใต้ดินได้ ดังนั้นผู้ควบคุมการก่อสร้างจะต้องมีการตรวจเช็คการทรุดตัวของผิวดินอยู่ตลอดเวลา ซึ่งในปัจจุบันการตรวจเช็คการทรุดตัวของผิวดินเป็นไปอย่างล่าช้าไม่ทันทั่วทั้งที่ อาจเป็นผลทำให้อาคารที่พักอาศัยที่ก่อสร้างอยู่ข้างบนเกิดความเสียหายและไม่สามารถแก้ไขได้ทันทั่วทั้งที่ ซึ่งโครงการนี้ได้นำเทคโนโลยีทางสารสนเทศมาช่วยในการตรวจเช็คการทรุดตัวของผิวดินให้เป็นไปอย่างรวดเร็วมากยิ่งขึ้น โดยการทำ Application ในการตรวจเช็คการทรุดตัวของผิวดินบน Pocket pc phone โดยให้ผู้ที่ทำการวัดการทรุดตัวของผิวดินใส่ค่าการทรุดตัวลงที่ Pocket pc phone และจาก Pocket pc phone ส่งค่าไปที่เซิร์ฟเวอร์ และจากเซิร์ฟเวอร์ส่งค่านั้นเพื่อแสดงผลให้คนที่เกี่ยวข้องสามารถตรงเช็คทางระบบอินเทอร์เน็ตได้ทันทั่วทั้งที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Real – time damage monitoring system for tunnel construction
Student Mr. Nuttapol Kamolkunasiri ID. 45010239
Mr. Nitit Rungratanakitkomjohn ID. 45010404
Advisor Mrs. Kleddao Suwansawat
Asst.Prof. Suchatvee Suwansawat
Graduate Level Bachelor Degree of Information Engineering
Department Information Engineering
Academic Year 2005

ABSTRACT

For safety driving tunnel, constructors have to take into consideration about structure of surface in area that the buildings are constructed without meditating underground structure. Driving tunnel may cause subsidence of surface and destruction of building. Rechecking subsidence of surface in every step of construction make construction delay. This project applies information technology to improve the way of checking faster. Step of checking is start by messenger measuring subsidence level and entering value into pocket PC phone. After that application on pocket PC phone connects and sends value to compute server. So, people who involves in system can know and update status via an internet promptly.

กิตติกรรมประกาศ

โครงการระบบการตรวจสอบความเสียหายจากการก่อสร้างอุโมงค์ใต้ดินแบบ Real - Time ผ่านpdaนี้ จะไม่สำเร็จได้เลย ถ้าไม่ได้รับการสนับสนุนจากพ่อแม่ของพวกเราที่แม้ว่าจะไม่รู้เรื่องของโครงการนี้ซักเท่าไร แต่ก็คอยสนับสนุนด้วยแรงใจและทุนทรัพย์ที่ทำให้เรามีเงินมีใช้ในตอนทำโครงการ คอยโทรมาซักถามความก้าวหน้าของโครงการเหมือนเป็นอาจารย์ที่ปรึกษาคนที่3และ4 ในส่วนอุปกรณ์ก็มีอาจารย์เกิ้ลคิ้ว (อาจารย์ที่ปรึกษา)และอาจารย์ สุัชชวีร์(อาจารย์ที่ปรึกษา) ที่คอยให้คำปรึกษาและจัดหาอุปกรณ์มาให้เรา ทำให้ดำเนินโครงการได้อย่างเต็มที่ และอาจารย์ชงคี่ที่คอยให้คำปรึกษาเรื่องการเขียนโปรแกรม

นอกจากนี้ก็ยังมีส่วนเพื่อนๆและน้องๆมาช่วยเหลือในด้านเทคนิคต่างอันได้แก่ เพื่อนชาญที่มาแนะแนวทางสว่างให้ *เพื่อนมดที่มาช่วยเป็นที่ปรึกษา ยังมีน้องเอที่อุทิศเครื่องให้ทดลอง และเพื่อนพร น้องปึก ที่คอยมาเปิดเครื่องเอให้ทดลองเสมอ

และนอกจากพี่ๆแล้วพวกเราก็ยังมีอาจารย์ที่คอยให้คำปรึกษาในเรื่องของการเขียน โปรแกรม และหาข้อมูลความรู้ต่างๆอีก 2 สถาบัน คือ อ.ก๊วกเกิ้ล แห่งwww.google.com และอาจารย์สำนักไมโครซอฟท์แห่ง<http://msdn1.microsoft.com/en-us/default.aspx>

ถ้าไม่ได้บุคคลเหล่านี้มาช่วยเหลือและให้คำปรึกษาเรา โครงการนี้คงไม่สำเร็จไปได้จึงขอแสดงความขอบคุณผ่านกิตติกรรมประกาศมาไว้ ณ ที่นี้ ขอขอบคุณมากครับ

ปล.ขอบคุณ 7-11 ที่ทำให้เราอิ่มท้องกันทุกคืน และเราก็คงช่วยให้ 7-11 รายขึ้นอีกนิดนึง ด้วยจำนวนครั้งการกินต่อวันที่มากมาย

ณัฐพล กมลคุณาศิริ

นิธิศ รุ่งรัตนกิจกำร

สารบัญ

เรื่อง	หน้า
บทคัดย่อไทย	ก
บทคัดย่ออังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญภาพ	ช
สารบัญตาราง	ฉ
บทที่ 1 บทนำ	1
1.1 แนวคิดและที่มาของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 ผลที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนในการดำเนินงาน	3
1.6 อุปกรณ์ที่ใช้ในการพัฒนาโครงการ	3
1.6.1 ฮาร์ดแวร์	3
1.6.2 ซอฟต์แวร์	3
บทที่ 2 ทฤษฎีและหลักการที่ใช้ในโครงการ	4
2.1 การพัฒนาระบบเชิงวัตถุ	4
2.1.1 หลักการของการพัฒนาระบบเชิงวัตถุ	5
2.1.2 ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ	7
2.2 ภาษายูเอ็มแอล	10
2.2.1 ความเป็นมาของภาษายูเอ็มแอล	10
2.2.2 ข้อดีของยูเอ็มแอล	11
2.2.3 องค์ประกอบของยูเอ็มแอล	11
2.3 เทคโนโลยีคือทเน็ตเฟรมเวิร์ค	15
2.3.1 สถาปัตยกรรม .NET	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.3.2 เลเซอร์ คอมมอนแลงเกวจรัน ไทม์	15
2.3.3 เลเซอร์เบสคลาสไลบรารีแลงเกวจ	19
2.3.4 เลเซอร์คอมมอนแลงเกวจเสปคซีพีเคชัน	20
2.4 สถาปัตยกรรมของคอปเน็ตคอมแพคแฟรมเวิร์ค	21
2.5 เทคโนโลยีเอดีโอค็อทเน็ต	23
2.5.1 ออบเจกต์ที่สำคัญในเอดีโอค็อทเน็ต	24
2.6 เอสคิวแอล เซิร์ฟเวอร์ 2000	26
2.6.1 ส่วนของผู้ให้บริการ	26
2.6.2 ส่วนของผู้ใช้บริการ	27
2.6.3 ข่ายการสื่อสารหรือเน็ตเวิร์ค	27
2.6.4 สถาปัตยกรรมของเอสคิวแอล เซิร์ฟเวอร์	27
2.7 การออกแบบฐานข้อมูลโดยการใช้ในแอม โมเดล	28
2.7.1 ส่วนประกอบของในแอมโมเดล	28
2.7.2 การเขียนความสัมพันธ์ในรูปแบบของในแอม	29
2.8 Web Services	29
2.9 หลักการในการทำระดับ(Principles of Leveling)	31
บทที่ 3 การออกแบบโครงการ	36
3.1 การออกแบบระบบ	36
3.1.1 ส่วนสำหรับพนักงาน	36
3.1.2 โครงสร้างทั่วไปของระบบ	37
3.2 การออกแบบโครงการ	38
3.2.1 ยูสเคสไดอะแกรม	38
3.2.2 แอคทิวิตี้ไดอะแกรม	42
3.2.3 การออกแบบฐานข้อมูล	47
3.2.4 คาท้าดิกชันนารี	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4 ผลการทดสอบ	50
4.1 ส่วนการทำงานของการทำงานของการเปลี่ยนแปลง Site งานการขุดคู โมงค์	51
4.2 ส่วนการทำงานของการทำงานของการเปลี่ยนแปลง Section งานการขุดคู โมงค์	54
4.3 ส่วนการทำงานของการทำงานของการเปลี่ยนแปลง Instrument งานการขุดคู โมงค์	55
4.4 ส่วนการเรียกดูข้อมูลเพื่อเปรียบเทียบการพังทลายของงาน การขุดคู โมงค์ (Monitor Status)	55
4.5 ส่วนการบันทึกข้อมูลโดยตรงจากศูนย์ปฏิบัติการ (Direct Record)	57
4.6 ส่วนการบันทึกข้อมูลจาก Site งานผ่าน Pocket PC Phone (PDA Record)	59
4.7 ส่วนการบันทึกข้อมูลจาก Site งานผ่าน Pocket PC Phone (PDA Record)	61
บทที่ 5 สรุป	62
5.1 สรุปการพัฒนาโครงการ	62
5.2 ปัญหาที่พบและข้อจำกัดในการพัฒนาโครงการ	62
5.3 แนวทางแก้ไข	63
5.4 การนำโครงการไปประยุกต์ใช้ในอนาคต	63
บรรณานุกรม	
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
รูปที่ 1.1 ขั้นตอนในการทำงานของการสำรวจ	1
รูปที่ 1.2 ขั้นตอนในการดำเนินงานของการสำรวจ	3
รูปที่ 2.1 คลาสไดอะแกรมของระบบ Course	12
รูปที่ 2.2 ออบเจกต์ไดอะแกรมของระบบ User	12
รูปที่ 2.3 ยูสเคสไดอะแกรม	13
รูปที่ 2.4 แสดงตัวอย่างแอ็กทิวิตีไดอะแกรม มีดังนี้	14
รูปที่ 2.5 รูปแสดงโครงสร้างการพัฒนาแอปพลิเคชันคือทเน็ต	15
รูปที่ 2.6 แสดงสถาปัตยกรรมของคอมมอนแลงกเวจรันไทม์	16
รูปที่ 2.7 แสดงความสามารถในการติดต่อข้ามแอปพลิเคชัน เมื่อเขียนด้วยภาษาที่ต่างกัน	17
รูปที่ 2.8 เบสคลาสไลบรารีของสถาปัตยกรรมคือทเน็ต	19
รูปที่ 2.9 คอมมอนแลงกเวจสเปคซิฟิเคชันของสถาปัตยกรรมคือทเน็ต	20
รูปที่ 2.10 แสดงสถาปัตยกรรมของคือทเน็ตคอมแพ็คแฟรมเวิร์ค	21
รูปที่ 2.11 แสดงคลาสไลบรารีของคือทเน็ตคอมแพ็คแฟรมเวิร์ค	22
รูปที่ 2.12 สถาปัตยกรรมการเชื่อมต่อข้อมูลผ่านเอดีโอ	23
รูปที่ 2.13 แสดงเส้นทางในการติดต่อแลกเปลี่ยนข้อมูลและคำสั่ง ระหว่างเครื่องเซิร์ฟเวอร์และเครื่องไคลเอ็นต์	27
รูปที่ 2.14 แสดงเส้นทางในการติดต่อแลกเปลี่ยนข้อมูลและคำสั่ง ระหว่างเครื่องคาล์เบสเซิร์ฟเวอร์และเครื่องไคลเอ็นต์	27
รูปที่ 2.15 สถาปัตยกรรมเว็บเซอร์วิส	30
รูปที่ 2.16 สถาปัตยกรรม XML เว็บเซอร์วิส	30
รูปที่ 3.1 โดเมน โมเดล (Domain Model)	37
รูปที่ 3.2 ยูสเคสไดอะแกรมสำหรับการใช้ข้อมูลของ Staff (Survey Man) และ พนักงานประจำศูนย์ปฏิบัติการ	38

รูปที่	หน้า
รูปที่ 3.3 ยูสเคสไดอะแกรมสำหรับการสร้าง Site station ของ พนักงานประจำศูนย์ปฏิบัติการ	38
รูปที่ 3.4 ยูสเคสไดอะแกรมสำหรับการลบค่าภายในของ Site station ของ พนักงานประจำศูนย์ปฏิบัติการ	39
รูปที่ 3.5 ยูสเคสไดอะแกรมสำหรับการ Update ค่าภายในของ site station ของ พนักงานประจำศูนย์ปฏิบัติการ	39
รูปที่ 3.6 ยูสเคสไดอะแกรมสำหรับการเรียกดูค่า Site station ของ พนักงานประจำศูนย์ปฏิบัติการ	40
รูปที่ 3.7 ยูสเคสไดอะแกรมสำหรับการเรียกดูค่าภายในของ Site station ของ staff (Survey Man)	40
รูปที่ 3.8 ยูสเคสไดอะแกรมสำหรับการเรียกดูกราฟข้อมูล Settlement ของ staff (Survey Man)	41
รูปที่ 3.9 ยูสเคสไดอะแกรมสำหรับการใส่ข้อมูลของ Staff (Survey Man)	41
รูปที่ 3.10 แอคทีวิตีไดอะแกรมสำหรับการบันทึกข้อมูลของ Staff (Survey Man)	42
รูปที่ 3.11 แอคทีวิตีไดอะแกรมสำหรับการบันทึกข้อมูลของ Staff (Survey Man) (Form2)	43
รูปที่ 3.12 แอคทีวิตีไดอะแกรมสำหรับการบันทึกข้อมูลของ staff (Survey Man) (Form3)	43
รูปที่ 3.13 แอคทีวิตีไดอะแกรมสำหรับการบันทึกข้อมูลลง ฐานข้อมูลโดยตรงของ Officer (Form1)	44
รูปที่ 3.14 แอคทีวิตีไดอะแกรมสำหรับการบันทึกข้อมูลลง ฐานข้อมูลโดยตรงของ Officer (Form2)	44
รูปที่ 3.15 แอคทีวิตีไดอะแกรมสำหรับการสร้าง Site Section Instrument ของ Officer	45
รูปที่ 3.16 แอคทีวิตีไดอะแกรมสำหรับการปรับเปลี่ยน Site Section Instrument ของ Officer	45
รูปที่ 3.17 แอคทีวิตีไดอะแกรมสำหรับการลบ Site Section Instrument ของ Officer	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่	หน้า
รูปที่ 3.18 แอคทีวิตี้ไดอะแกรมสำหรับการดูภาพรวมของการทูลดตัว เพื่อประเมินผลของ Officer	46
รูปที่ 3.19 แอคทีวิตี้ไดอะแกรมสำหรับการสร้างรายงานประจำ การตรวจสอบของ Officer	46
รูปที่ 3.20 ไนแอม โมเดล (Niam Model) ของระบบฐานข้อมูล	47
รูปที่ 4.1 หน้าแรกโปรแกรมด้านเซอร์เวอร์	50
รูปที่ 4.2 แสดงการจ่อเมา์บริเวณ ไอคอน Site	51
รูปที่ 4.3 หน้าแรกการจัดการกับ Site	51
รูปที่ 4.4 แสดงการเลือก Site จาก Drop-down List	52
รูปที่ 4.5 แสดงการสร้าง Site งานใหม่ และ คำอธิบาย Site	52
รูปที่ 4.6 แสดง Site ใหม่ที่สร้างขึ้น	53
รูปที่ 4.7 แสดงการทำการเปลี่ยนแปลงคำชื่อ Site และคำอธิบาย	53
รูปที่ 4.8 แสดงค่าที่เปลี่ยนแปลงชื่อ Site และคำอธิบาย	54
รูปที่ 4.9 แสดงหน้าแรกของการจัดการ Section	54
รูปที่ 4.10 แสดงหน้าแรกของการจัดการ Instrument	55
รูปที่ 4.11 แสดงหน้าแรกของเรียกดู Monitor Status	55
รูปที่ 4.12 แสดงเลือกช่วงเวลา และ สถานที่ที่ต้องการดู	56
รูปที่ 4.13 แสดงผลกราฟเปรียบเทียบการเปลี่ยนแปลง	56
รูปที่ 4.14 แสดงหน้าแรกของการบันทึกค่าโดยตรง	57
รูปที่ 4.15 แสดงการเลือก Site Section และ ใส่พารามิเตอร์ต่างๆ	57
รูปที่ 4.16 แสดงกราฟจากการบันทึก และมีการเตือนภัย	58
รูปที่ 4.17 แสดงกราฟจากการบันทึก	58
รูปที่ 4.18 แสดงหน้าแรกของการบันทึกค่าผ่าน Pocket PC Phone	59
รูปที่ 4.19 แสดงการเลือก Site Section และ ใส่พารามิเตอร์ต่างๆลง Pocket PC Phone	59
รูปที่ 4.20 แสดงการใส่พารามิเตอร์ค่าความเสียหาย จากอุปกรณ์ที่ Site ลง Pocket PC Phone	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่

หน้า

รูปที่ 4.21 แสดงกราฟจากการบันทึก มี 2 แบบ 1. มีการเตือนภัย 2. บันทึกค่าปกติ	60
รูปที่ 4.22 แสดงหน้าแรกของการสร้างรายงานผลการบันทึก และการเลือก Site	61
รูปที่ 4.23 แสดงรายงานประจำการตรวจสอบ	61



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 เปรียบเทียบการเชื่อมต่อฐานข้อมูลระหว่างเทคโนโลยี ADO และ ADO.NET	26
ตารางที่ 3.1 ตารางข้อมูลเครื่องมือ	48
ตารางที่ 3.2 ตารางรายละเอียดเครื่องมือ	48
ตารางที่ 3.3 ตารางเครื่องมือ	49
ตารางที่ 3.4 ตารางส่วนย่อย	49
ตารางที่ 3.4 ตารางข้อมูล Site	49



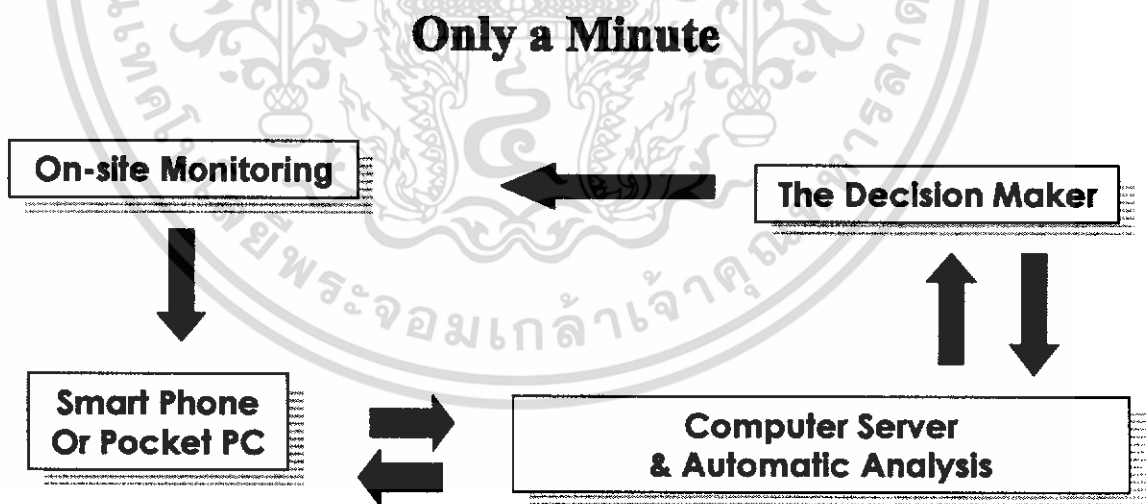
บทที่ 1

บทนำ

1.1 แนวคิดและที่มาของปัญหา

ในสังคมยุคปัจจุบัน ประเทศที่กำลังพัฒนา เช่น ประเทศไทยนั้นการก่อสร้างสาธารณูปโภคเป็นส่วนสำคัญในการพัฒนา และการคมนาคมกลายเป็นปัจจัยหลักอย่างหนึ่งของระบบเศรษฐกิจ ประเทศไทยจึงพยายามพัฒนาระบบคมนาคม เพื่อความสะดวกรวดเร็วจึงเพิ่มระบบรถไฟฟ้าใต้ดินและในอนาคตอาจจะกลายเป็นการคมนาคมหลัก ดังเช่นประเทศที่พัฒนาแล้ว ทั่วๆไป

ระบบการเช็คค่าความเสียหายเนื่องจากการขุดอุโมงค์รถไฟฟ้าใต้ดินเป็นส่วนสำคัญในการก่อสร้าง เพื่อรับรู้ผลกระทบที่เกิดจากการขุดเจาะของอุโมงค์ซึ่งในปัจจุบันใช้การตรวจสอบโดยการจดบันทึกค่าความเสียหายจาก Site งานเป็นเอกสาร จากนั้นนำกลับมาที่ศูนย์ปฏิบัติการ แล้วจึงแจ้งไปที่แท่นขุดเจาะ ซึ่งในขั้นตอนนี้จะใช้เวลาและบางครั้งไม่สามารถแก้ปัญหาได้ทันที ทางผู้จัดทำโครงการจึงเล็งเห็นความสำคัญและแนวทางการแก้ไขและพัฒนาเพื่อทำให้การสื่อสารมีประสิทธิภาพมากขึ้น โดยนำซอฟต์แวร์ และ เทคโนโลยีการสื่อสาร ไร้สาย เข้ามาประยุกต์ใช้ เพื่อก่อให้เกิดประโยชน์สูงสุด



รูปที่ 1.1 ขั้นตอนในการทำงานของการสำรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อพัฒนาระบบการขุดเจาะอุโมงค์รถไฟใต้ดิน โดยใช้เทคโนโลยีไร้สาย GPRS ใช้ในการรับส่งข้อมูลติดต่อระหว่าง Pocket pc phone
- 1.2.2 เพื่ออำนวยความสะดวกและลดอัตราการเสี่ยงในการก่อสร้างได้มากขึ้น
- 1.2.3 มีระบบจัดเก็บข้อมูลที่สามารถอ้างอิงและตรวจสอบได้อย่างรวดเร็ว
- 1.2.4 ลดระยะเวลาในการรับส่งข้อมูลให้รวดเร็วมากขึ้น
- 1.2.5 ช่วยลดระยะเวลาในกระบวนการก่อสร้างให้มีความรวดเร็วมากขึ้น

1.3 ขอบเขตของโครงการ

1.3.1 ด้านพนักงานสำรวจ

- 1.3.1.1 โครงการสามารถแสดงผลบนหน้าจอ pocket pc phone ได้
- 1.3.1.2 ผู้ใช้สามารถส่งข้อมูลผ่าน Pocket pc phone และสามารถรับข้อมูลจากเซิร์ฟเวอร์มายัง pocket pc phone
- 1.3.1.3 ผู้ใช้สามารถตรวจสอบข้อมูลได้ทันที

1.3.2 ด้านศูนย์ปฏิบัติการ

- 1.3.2.1 ผู้ใช้สามารถสร้างกราฟเปรียบเทียบผลของข้อมูล ณ ศูนย์ปฏิบัติการเป็นช่วงเวลาได้
- 1.3.2.2 ผู้ใช้สามารถสร้างรายงานผลของการสำรวจได้
- 1.3.2.3 ผู้ใช้สามารถสร้าง แก้ไข และลบรายละเอียด Site งานเพิ่มเติมได้

1.4 ผลที่คาดว่าจะได้รับ

- 1.4.1 ระยะเวลาในการรับส่งข้อมูลรวดเร็วขึ้น
- 1.4.2 สามารถแก้ไขปัญหาการทรุดตัวของดินได้ทันที
- 1.4.3 มีระบบจัดเก็บข้อมูลที่สามารถอ้างอิงและสืบค้นได้อย่างรวดเร็ว
- 1.4.4 สามารถเรียกดูผลการวิเคราะห์ได้หลายช่องทาง
- 1.4.5 เพิ่มความปลอดภัยในการก่อสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนการดำเนินงาน

ID	Task Name	2005							2006			
		Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	
1	Problem Definition & Get Requirements											
2	Analysis & Design											
3	Software Design											
4	Database Design											
5	Interface Design											
6	Implementation											
7	Testing & Debugging											
8	Documentation											

รูปที่ 1.2 ขั้นตอนในการดำเนินงานของการสำรวจ

1.6 อุปกรณ์ที่ใช้ในการพัฒนาโครงการ

1.6.1 ฮาร์ดแวร์

- Pocket pc phone
- Tablet, personal computer

1.6.2 ซอฟต์แวร์

- Microsoft Visual Studio.NET 2005 beta 2
- Microsoft SQL Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่ใช้ในโครงการงาน

2.1 การพัฒนาระบบเชิงวัตถุ(Object Oriented)

การพัฒนาระบบซอฟต์แวร์หรือระบบสารสนเทศในอดีตนั้นจะใช้วิธีการพัฒนาแบบโครงสร้าง(Structural) ซึ่งมีข้อดีเหนือกว่าการพัฒนาแบบโดยไม่ใช้โมเดล (Model) แต่ในปัจจุบันพบว่า การพัฒนาแบบโครงสร้างนั้นมีปัญหาคือ โมเดลในการวิเคราะห์และออกแบบระบบจะไม่เชื่อมต่อไปถึงรายละเอียดในการพัฒนาระบบ ตัวอย่างเช่น เมื่อผู้พัฒนาระบบวิเคราะห์และออกแบบระบบได้เป็นแผนภาพคอนเท็กซ์ (Context Diagram) หรือแผนภาพ DFD ระดับสูงสุด, แผนภาพกระแสข้อมูล (Data Flow) หรือแม้แต่แผนภาพที่แสดงความสัมพันธ์ขององค์ประกอบต่างๆ เพื่อสร้างฐานข้อมูลเชิงสัมพันธ์ (Entity Relations) เองแผนภาพเหล่านี้เพียงบอกแต่ว่าระบบจะมีลักษณะการทำงานอย่างไร แต่ไม่ได้บอกวิธีการพัฒนาระบบหรือรูปแบบในการเขียนโปรแกรม ทำให้ระบบที่พัฒนาขึ้นมีปัญหา เนื่องจากว่าการตรวจสอบความถูกต้องของระบบ (Validation) ในเชิงพฤติกรรม (Behavior) นั้นทำยากและไม่มีโมเดลใดๆ ที่รองรับจึงได้มีผู้คิดค้นและพัฒนาวิธีการหรือโมเดลในการพัฒนาระบบเชิงวัตถุ ขึ้นมา

การพัฒนาระบบเชิงวัตถุ เป็นแนวความคิดในการออกแบบระบบต่างๆ โดยมองระบบต่างๆ ว่าเกิดจากการประกอบกันของวัตถุ (Object) ซึ่งหมายถึงสิ่งต่างๆ ที่อยู่รอบๆ ตัวเรา ออบเจกต์หนึ่งอาจจะประกอบด้วยออบเจกต์อื่นๆ หลายๆ ออบเจกต์ ขนาดที่เล็กที่สุดหรือใหญ่ที่สุดของออบเจกต์จะขึ้นอยู่กับขอบเขตของระบบที่เรากำลังสนใจอยู่ ว่าต้องการรายละเอียดระดับใด

ในการกำหนดออบเจกต์จะต้องกำหนดคลาสขึ้นมาก่อน คลาสคือรูปแบบทั่วไปของออบเจกต์หรือเป็นพิมพ์เขียวของออบเจกต์ และออบเจกต์คือ อินสแตนซ์(Instant) ของคลาส เช่น Mr. John เป็นอินสแตนซ์ ของคลาส Person

ออบเจกต์จะมีโครงสร้างที่ประกอบด้วยแอททริบิวต์ และพฤติกรรม เช่นในคลาส Person จะมีแอททริบิวต์ คือ height, weight และ age และมีพฤติกรรมคือ eat, sleep, read, write เป็นต้น ซึ่งแอททริบิวต์และพฤติกรรมดังกล่าวจะมีอะไรข้างนั้น ขึ้นอยู่กับระบบที่ต้องการออกแบบ เช่น Person ในระบบ Human Resource ก็จะมีรายละเอียดอื่นๆ เกี่ยวข้องเพิ่มขึ้น

นอกจากนั้นการวิเคราะห์และออกแบบระบบเชิงวัตถุจะมีโมเดลที่รองรับ ซึ่งโมเดลที่เป็นมาตรฐานอยู่ในขณะนี้ชื่อว่า UML (Unified Modeling Language) โดยกระบวนการความคิด (Methodology) และโมเดลที่สนับสนุนการวิเคราะห์และออกแบบเชิงวัตถุที่สามารถเข้าใจได้ง่ายและสามารถใช้ประกอบในขั้นตอนการลงมือเขียนโปรแกรมได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 หลักการของการพัฒนาระบบเชิงวัตถุ

Abstraction

หมายถึงการกำหนดออบเจกต์ โดยไม่ได้ระบุรายละเอียดของแอททริบิวต์และพฤติกรรม และระบุเป็นส่วนน้อยเท่าที่สามารถทำได้ ซึ่งทำให้การออกแบบมีความยืดหยุ่นมากขึ้น และไม่เกิดการกำหนดออบเจกต์เดียวกันมากกว่า 1 ออบเจกต์ ผู้ใช้ในแต่ละฟังก์ชันสามารถกำหนดแอททริบิวต์ และ พฤติกรรมเพิ่มเติมให้กับออบเจกต์ได้ ตามความจำเป็นที่ต้องมีในฟังก์ชันงานนั้นๆ

Inheritance

การสืบทอดคุณสมบัติ คือ เป็นวิธีการในการสร้างคลาสใหม่จากคลาสเดิมที่มีอยู่ ทั้งนี้คลาสที่สร้างขึ้นใหม่จะมีวัตถุประสงค์ในการทำงานที่เฉพาะเจาะจงมากยิ่งขึ้น โดยการที่ออบเจกต์มีคุณสมบัติทั้งหมดของคลาสที่ออบเจกต์นั้นเป็นอินสแตนซ์อยู่ คลาสที่สืบทอดจะเรียกว่าซับคลาส จะมีคุณสมบัติเหมือนกับซูเปอร์คลาสทุกประการ โดยไม่ต้องกำหนดซ้ำ

Polymorphism

คุณสมบัติการเปลี่ยนรูป (Polymorphism) หมายถึงการเปลี่ยนรูปร่างของออบเจกต์หนึ่งๆ ซึ่งในเชิงโปรแกรมมิ่งจะเป็นการที่ตัวแปรออบเจกต์ของคลาสใดคลาสหนึ่งสามารถเปลี่ยนรูปแบบไปจากคลาสเดิมได้ โดยคลาสที่มีการสืบทอดแต่โอเปอเรชันในซับคลาส มีการทำงานที่แตกต่างกันกับโอเปอเรชันเดียวกันในซูเปอร์คลาส เช่น ถ้าคลาส polygon มีโอเปอเรชัน draw และ คลาส rectangle สืบทอดจากคลาส polygon ซึ่งจะทำให้มีโอเปอเรชัน draw เช่นเดียวกัน แต่ การ draw ของ polygon กับของ rectangle มีการทำงานคนละอย่างกัน ดังนั้นจึงต้องมีการกำหนดโอเปอเรชัน draw ของคลาส rectangle ใหม่

Encapsulation

คุณสมบัติ (Encapsulation) หรือเรียกอีกอย่างหนึ่งคือว่า การซ่อนข้อมูล (Information hiding) คือการจัดกลุ่มองค์ความคิดที่คล้ายคลึงกันเข้าเป็นหน่วยเดียวกันเพื่อทำการอ้างอิงถึงด้วยชื่อเดียวกัน เป็นการซ่อนข้อมูลหรือกระบวนการในการทำโอเปอเรชันจากผู้ใช้คลาสในการออกแบบเชิงออบเจกต์สามารถทำได้ 3 ระดับ คือ

- Private เป็นการกำหนดให้ข้อมูลหรือโอเปอเรชันนั้นเข้าถึงได้จากภายในคลาสเดียวกันเท่านั้น โดยสามารถทำได้ทั้งการอ่านและเขียน
- Protected เป็นการกำหนดข้อมูลหรือโอเปอเรชันนั้นเข้าถึงได้จากภายในคลาสเดียวกันและคลาสอื่น โดยถ้าเป็นคลาสเดียวกัน จะสามารถทำได้ทั้งการอ่านและเขียน แต่คลาสอื่นจะทำได้เพียงการอ่านอย่างเดียว
- Public เป็นการกำหนดให้ข้อมูลหรือโอเปอเรชันนั้น เข้าถึงได้จากภายในคลาสเดียวกันและคลาสอื่น โดยสามารถทำได้ทั้งอ่านและเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Message Sending

การส่งข้อความทำให้ออบเจกต์ในระบบสามารถทำงานร่วมกันได้ โดยออบเจกต์หนึ่งจะส่งข้อความให้กับอีกออบเจกต์หนึ่งเพื่อให้ออบเจกต์นั้นทำโอเปอเรชันอย่างใดอย่างหนึ่ง ออบเจกต์ที่ได้รับข้อความก็จะทำโอเปอเรชันตามข้อความที่ได้รับแต่ในบางครั้งพบว่าการกำหนดให้ออบเจกต์ทำโอเปอเรชันอาจจะทำได้โดยเรียกใช้อินเตอร์เฟซของออบเจกต์นั้นโดยตรง เช่น การเรียกใช้ฟังก์ชันที่ประกาศเป็น Public ซึ่งถือเป็น message sending อย่างหนึ่ง

Association

ออบเจกต์ที่อยู่ในระบบเดียวกัน จะมีความสัมพันธ์กับออบเจกต์อื่นอย่างน้อย 1 ออบเจกต์เสมอในบางครั้งอาจมีความสัมพันธ์ระหว่างออบเจกต์ คือการกำหนดจำนวน (Multiplicity) ซึ่งเป็นการกำหนดจำนวนออบเจกต์ในคลาสหนึ่ง ที่มีความสัมพันธ์กับอีกออบเจกต์ในอีกคลาสหนึ่ง

Aggregation

เป็นรูปแบบความสัมพันธ์อีกแบบหนึ่ง โดยออบเจกต์หนึ่งประกอบขึ้นจากออบเจกต์อื่นๆ ตั้งแต่ 1 ออบเจกต์ขึ้นไป เช่น ในออบเจกต์ Computer A มีออบเจกต์ Hard drive 2 ออบเจกต์เป็นต้น ออบเจกต์ Computer เรียกว่า Aggregate Object และออบเจกต์ ที่เป็นองค์ประกอบจะเรียกว่า Component Object

อีกรูปแบบหนึ่งของ Aggregation ที่เป็นการสัมพันธ์แบบความสัมพันธ์เชิงแข็ง (Strong relationship) คือ Composition โดย Component Object จะต้องอยู่ใน Composite Object เช่น ออบเจกต์คอมพิวเตอร์ จะต้องมีส่วนซีพียู (CPU) จึงเป็น Composite แต่ออบเจกต์คอมพิวเตอร์อาจจะไม่มีเครื่องพิมพ์ก็ได้ ทำให้ความสัมพันธ์ระหว่าง คอมพิวเตอร์ กับ เครื่องพิมพ์ เป็นความสัมพันธ์แบบ Aggregation แต่ไม่เป็น Composition

กล่าวโดยสรุป ข้อดีของการประยุกต์ใช้แนวคิดเชิงวัตถุในการพัฒนาระบบซอฟต์แวร์มีดังต่อไปนี้

- สนับสนุนการพัฒนาระบบที่ซับซ้อน (Problem Decomposition)

ในการพัฒนาโปรแกรมรูปแบบเดิม นักพัฒนาโปรแกรมจะทำงานกันในระดับฟังก์ชันกล่าวคือตั้งแต่การเริ่มพัฒนาจนจบ จะเป็นการสร้างปรับปรุงแก้ไขรายละเอียดของฟังก์ชันต่างๆภายในตัวโปรแกรม แต่สำหรับแนวคิดเชิงวัตถุ นักพัฒนาจะทำการวิเคราะห์ออกแบบระบบกันในระดับออบเจกต์ซึ่งประกอบด้วยทั้งข้อมูลและฟังก์ชันภายในแต่ละออบเจกต์ ดังนั้นจึงกล่าวได้ว่าการอาศัยแนวคิดเชิงวัตถุจะช่วยให้นักพัฒนาสามารถสร้างโปรแกรมได้ง่ายขึ้นมาก

- สนับสนุนการนำกลับมาใช้งานอีกครั้ง (Promotion of Reusability)

เนื่องจากแต่ละคลาสหรือออบเจกต์ที่กำหนดขึ้นนั้น จะมีความสมบูรณ์อยู่ในตัวเองบนพื้นฐานแนวคิดของแต่ละออบเจกต์เอง รวมทั้งยังเป็นอิสระจากสภาพแวดล้อมอื่น ดังนั้นแต่ละคลาสจึงง่ายต่อการนำกลับมาใช้งานปรับปรุงเพิ่มเติม การนำกลับมาใช้งานอาจอยู่ในรูปแบบของ

การสืบทอดคุณสมบัติระหว่างออบเจกต์จากคลาสปุ่นเพื่อไปใส่ไว้ในโปรแกรมต่างๆ ได้ตามต้องการ นั่นคือคลาสปุ่นสามารถถูกนำกลับมาใช้ได้

- ปรับปรุงแก้ไขและบำรุงรักษาง่าย (Easy for Change and Extensibility)

เนื่องจากข้อมูล และฟังก์ชันการทำงานที่เกี่ยวข้องกับออบเจกต์หนึ่งๆจะถูกรวบรวมอยู่ที่เดียวกันการทำงานภายในแต่ละออบเจกต์จะไม่เกี่ยวข้องฟังก์ชันกับโค้ดที่อยู่ภายนอกออบเจกต์ดังนั้น นักพัฒนาสามารถทำการแก้ไขปรับปรุงรายละเอียดภายในของแต่ละคลาสได้ โดยไม่กระทบต่อส่วนที่เรียกใช้งานภายนอกแต่อย่างใด นอกจากนี้การขยายงานระบบก็สามารถทำได้โดยง่าย โดยการสร้างออบเจกต์หรือคลาสเพิ่มเติมลงไปในตัวโปรแกรมนั่นเอง

2.1.2 ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ

วัตถุประสงค์หลักของทุกกระบวนการพัฒนาซอฟต์แวร์ คือ การแปลงของผู้ใช้ให้เป็นระบบที่มีคุณภาพและสามารถใช้งานได้จริง อีกทั้งยังช่วยลดระยะเวลาที่ต้องทำการเขียนโปรแกรม และแก้ไขข้อผิดพลาดให้น้อยลงอีกด้วย ซึ่งในระหว่างการพัฒนาจะมีการใช้เครื่องมือช่วยต่างๆ หลายอย่างในที่นี้ได้แก่ภาษามาตรฐานยูเอ็มแอล

กระบวนการที่จะอธิบายต่อไปนี้เป็นวิธีการทั่วไปในการพัฒนาซอฟต์แวร์เชิงวัตถุที่ประกอบไปด้วย 5 ขั้นตอน โดยในแต่ละตอนเรียกว่า เฟส (Phase) ซึ่งผลลัพธ์ที่ได้จากการพัฒนาในแต่ละเฟสจะต้องถูกนำไปพัฒนาต่อในเฟสถัดไป และมีเพียงเฟสแรกที่จะถูกดำเนินการในช่วงเริ่มต้นของการพัฒนาครั้งเดียวและในสี่เฟสที่เหลือจะถูกทำซ้ำ (Iteration) เพื่อทำการขยายต่อเติมฟังก์ชันของระบบจนกว่าจะได้ระบบที่สมบูรณ์ นั่นคือในแต่ละรอบของการวนซ้ำสิ่งที่ได้จะเป็นเพียงต้นแบบ (Prototype) อันประกอบไปด้วยฟังก์ชันบางอย่างสอดคล้องกับความต้องการของผู้ใช้ และสิ่งนี้เองที่ฝ่ายพัฒนาควรทำการส่งมอบให้ผู้ใช้พิจารณาเป็นระยะๆในช่วงของการพัฒนา และในความเป็นจริงสิ่งที่เกิดขึ้นตามมาซึ่งมักจะเลี่ยงไม่ได้คือความจำเป็นในการดำเนินการเฟสแรกมากกว่าหนึ่งครั้งอันเนื่องมาจากความเปลี่ยนแปลงความต้องการของผู้ใช้นั่นเอง

2.1.2.1 การวิเคราะห์ความต้องการของระบบหรือผู้ใช้ (Requirement Analysis)

โครงการที่ถือว่าประสบความสำเร็จจะต้องตอบสนองต่อความต้องการและความคาดหวังของผู้ใช้ได้อย่างครบถ้วน หรือมากกว่า อันเป็นจุดสำคัญของทุกกระบวนการของการพัฒนาซอฟต์แวร์ การวิเคราะห์ความต้องการของผู้ใช้เป็นการค้นหาขอบข่ายของระบบและเป็นการเตรียมข้อมูลของด้านความสามารถของระบบ (System Function) จากมุมมองภายนอกที่จะต้องถูกทำการพัฒนาโดยไม่คำนึงถึงรายละเอียดและกรรมวิธีทางเทคนิคต่างๆ

ในความเป็นจริงการเริ่มต้นเฟสแรกนี้จะต้องทำในรูปของการกำหนดเป็นข้อตกลง ด้วยการเจรจาระหว่างผู้ใช้งานและผู้พัฒนา ซึ่งฝ่ายผู้พัฒนาจะต้องบันทึกความต้องการของผู้ใช้อย่างละเอียดที่สุดเท่าที่จะเป็นไปได้ เรียกขั้นตอนย่อยนี้ว่า User Requirement Elicitation อันหมายถึง

เอกสารที่เขียนขึ้นเพื่อใช้ในการสื่อสารความต้องการของผู้ใช้ให้เป็นที่เข้าใจกันโดยทั้งสองฝ่าย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสัมภาษณ์ผู้ใช้งานระบบถึงความต้องการต่างๆ เฟสนี้จำเป็นอย่างมาก โดยเฉพาะการพัฒนา ระบบด้านธุรกิจ แต่มักไม่จำเป็นสำหรับการสร้างระบบเชิงวิจัยพัฒนาซึ่งมีเป้าหมายที่ชัดเจนอยู่แล้ว สำหรับส่วนที่ไม่สามารถลงรายละเอียดของความต้อการได้ก็ควรกำหนดเป้าหมายให้ชัดเจนในส่วนนั้นๆ เพื่อป้องกันความไม่เข้าใจที่ไม่ตรงกันระหว่างผู้ใช้และผู้พัฒนา

นอกจากการค้นคว้าความสามารถที่ต้องมีอยู่ในตัวระบบแล้ว ผู้พัฒนายังจำเป็นต้องทำการสำรวจความต้องการในแง่อื่นๆ ด้วย (Nonfunctional Requirements) ตัวอย่างเช่น ประสิทธิภาพ ช่วงเวลาในการตอบสนอง ส่วนอินเทอร์เน็ตเฟส การใช้งานร่วมกับระบบเดิม เป็นต้น อย่างไรก็ตาม หากมองในภาพที่กว้างขึ้น จะพบว่า การดำเนินการพัฒนาระบบจะต้องเกี่ยวข้องกับงบประมาณ ระยะเวลาในการส่งมอบผลงานแต่ละช่วง การทำข้อสัญญา และการบริหารโครงการ เป็นต้น

2.1.2.2 การวิเคราะห์ระบบ (Domain Analysis)

การวิเคราะห์ระบบ (Domain Analysis) เป็นการวิเคราะห์โครงสร้าง (Structure) และ พฤติกรรม (Behavior) ของระบบที่จะทำการพัฒนาซึ่งจะถูกนำไปกำหนดรายละเอียดเชิงเทคนิคใน เฟสการออกแบบจะถูกสร้างจริงเป็นลำดับต่อไปในอิมพลีเมนต์ระบบ กิจกรรมในเฟสนี้จะเป็นการทำความเข้าใจกับระบบกำลังจะพัฒนาในสองด้าน กล่าวคือ

- การวิเคราะห์โครงสร้าง เป็นการทำความเข้าใจกับระบบที่กำลังพัฒนาเพื่อค้นหาคลาส และความสัมพันธ์ระหว่างคลาสในระบบ ซึ่งผลที่ได้เหล่านี้จะถูกนำไปอิมพลีเมนต์สร้างเป็นระบบขึ้นจริงในเฟสถัดๆ ไป

- การวิเคราะห์พฤติกรรม เนื่องจากออบเจกต์ที่เกิดจากคลาสต่างๆ ใน โครงสร้างของระบบ จะทำงานร่วมกันเพื่อให้ระบบโดยรวมสามารถตอบสนองต่อการ ใช้งานของผู้ใช้ได้ในทุกฟังก์ชัน

โดยกิจกรรมระหว่างการทำงานร่วมกันของแต่ละออบเจกต์จำเป็นต้องถูกทำ ความเข้าใจเพื่อนำไปอิมพลีเมนต์เป็นขั้นตอนการทำงานของแต่ละฟังก์ชันของระบบได้

2.1.2.3 การออกแบบ (Design)

ซึ่งเป็นกระบวนการคิดค้นวิธีแก้ไขปัญหาหรือพิจารณารายละเอียดเชิงเทคนิค เพื่อเตรียม ที่จะอิมพลีเมนต์ระบบขึ้นจริง ซึ่งจะเป็นการนำผลการวิเคราะห์จากเฟสที่สองเพื่อนำมาแก้ไข เพิ่มเติมรายละเอียดเชิงเทคนิค เพื่อสามารถที่จะถูกนำไปสร้างขึ้นเป็นระบบซอฟต์แวร์จริงได้อย่าง สมบูรณ์ ตัวอย่างกิจกรรมที่เกิดในเฟสนี้คือ

- การเลือกใช้ภาษาโปรแกรมคอมพิวเตอร์ที่จะใช้ในการพัฒนาระบบ
- การออกแบบฐานข้อมูล โดยเฉพาะในส่วนที่เรียกว่าระดับแนวคิด ซึ่งนิยมใช้ อีอาร์โมเดลเข้ามาช่วย
- การออกแบบสถาปัตยกรรมระบบ จำนวนเครื่องคอมพิวเตอร์
- การออกแบบลักษณะการเชื่อมต่อของเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นประโยชน์ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การออกแบบหน้าตาของตัวโปรแกรม (User Interface)
- รวมถึงการจัดการกับข้อผิดพลาดที่อาจเกิดขึ้นกับการใช้งานระบบ เช่น ระบบควรมีการจัดการอย่างไรหากผู้ใช้ป้อนข้อมูลผิดประเภท เป็นต้น

ดังนั้นเฟสนี้จำเป็นต้องอาศัยความรู้ทางเทคนิคและเทคโนโลยีต่างๆ ที่มีอยู่เพื่อสามารถเลือกได้อย่างเหมาะสม

2.1.2.4 การสร้างโปรแกรมระบบ (Construction, Coding, Implementation)

หลังจากที่ระบบถูกออกแบบไว้เรียบร้อยแล้วพร้อมที่จะถูกนำไปอิมพลีเมนต์จริง โดยโปรแกรมเมอร์ ในเฟสนี้สิ่งที่เกิดขึ้นจากการออกแบบทั้งหมดจะถูกแปลงไปสู่ระบบจริงโดยส่วนใหญ่กิจกรรมในเฟสนี้จะเป็นการสร้างโปรแกรม หรือการอิมพลีเมนต์ระบบอันเป็นขั้นตอนของ OOP (Object Oriented Programming) ที่คุ้นเคยกันดี ซึ่งจะต้องอาศัยความรู้ในตัวภาษาโปรแกรมที่ใช้ในการสร้างโค้ด ในขั้นตอนนี้จะถูกดำเนินการโดยทีมโปรแกรมเมอร์ภายหลังจากได้รับข้อมูลการออกแบบที่เพียงพอจากฝ่ายวิเคราะห์ออกแบบ

2.1.2.5 การทดสอบระบบ (Testing)

เป็นการทดสอบความถูกต้องของระบบที่พัฒนาเพื่อค้นหาข้อผิดพลาดทางเทคนิค และ การตรวจสอบความสอดคล้องกับความถูกต้องที่ถูกระบุอยู่ในความต้องการของผู้ใช้งาน

ทั้งนี้การค้นพบข้อผิดพลาดจะถือว่าเป็นความสำเร็จในการทำงานในเฟสนี้ มิใช่ความล้มเหลวแต่อย่างใด นอกจากนี้ยังเป็นการประเมินความสมบูรณ์ของระบบ ว่าจำเป็นต้องทำการวิเคราะห์ออกแบบเพิ่มเติมอีกครั้งหรือไม่ โดยปกติในเฟสนี้จะต้องมีการจัดเตรียมข้อมูลที่จะใช้สำหรับการทดสอบและการผลลัพธ์ที่เรียกว่า Test Cases จะประกอบไปด้วยกิจกรรมในการทดสอบการป้อนข้อมูลและการทดสอบผลลัพธ์รวมถึงความคาดหวังจากการทำงานของระบบว่า น่าจะเป็นที่พึงพอใจแก่ผู้ใช้งานจริงหรือไม่ผลของการทดสอบจะถูกบันทึกลงรายงานการทดสอบ ซึ่งรวมถึงการบรรยายรายละเอียดข้อผิดพลาดที่ปรากฏเพื่อทำการแก้ไขต่อไปโดยโปรแกรมเมอร์

ในเฟสนี้สิ่งที่ทำให้นักพัฒนาปวดหัวอยู่เสมอคือข้อผิดพลาด (Error) ซึ่งแบ่งได้เป็นสองด้าน ได้แก่

- Functional เป็นความผิดพลาดในระรับฟังก์ชันการทำงานของระบบ ซึ่งมักเกิดขึ้นบ่อยครั้ง
- Nonfunctional เป็นข้อผิดพลาดที่ไม่เกี่ยวกับระดับฟังก์ชันการทำงานของระบบที่พัฒนาโดยตรง เช่น การที่ประสิทธิภาพของการทำงานของระบบต่ำเกินไป ช่วงเวลาในการตอบสนองนานเกินไป เป็นต้น

โดยปกติความต้องการของผู้ใช้งานมักไม่ถูกค้นพบทั้งหมดในช่วงเริ่มต้นของการสัมภาษณ์ผู้ใช้ จึงเป็นเหตุผลว่าทำไมผู้ใช้งานจึงมักร้องเรียนขอการเปลี่ยนแปลง ซึ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เกิดขึ้นกับนักพัฒนาทั้งหลายโดยตลอดมา ข้อเสนอแนะต่อไปนี้อาจช่วยป้องกันปัญหาดังกล่าวได้บ้างไม่มากนัก

- กำหนดรายละเอียดความต้องการให้ชัดเจนถึงสิ่งที่ผู้ใช้งานต้องการ และไม่ต้องกำกวมทั้งนี้ห้ามสมมติเอาเอง เพื่อใช้ในการป้องกันหรือชะลอความเสียหายจากผู้ที่มีความต้องการเปลี่ยนแปลงความต้องการบ่อยๆ

- สิ่งที่ใช้เกริ่นว่าจะต้องการแต่ไม่ใช่ตอนนี้ อาจถูกออกแบบพัฒนาเพื่อไว้โดยปกติ 10% เพื่อลดผลกระทบที่อาจเกิดขึ้นเมื่อเวลาที่ต้องพัฒนาส่วนนั้นจริงๆมาถึง

- กำหนดข้อตกลงให้ชัดเจน ในการส่งมอบงาน หรือระบบตัวอย่างแต่ละช่วงเวลาในการพัฒนา

- การร้องขอให้แก้ไขข้อผิดพลาดในระบบที่เพิ่งส่งมอบ และตรวจรับเรียบร้อย อาจถูกคิดเป็นค่าบริการพิเศษ ทั้งนี้ต้องกำหนดไว้ในสัญญาอย่างชัดเจน

เราจำเป็นต้องมองการเปลี่ยนแปลงความต้องการของผู้ใช้งานเป็นเรื่องปกติธรรมดาที่ผู้พัฒนาต้องเผชิญ หากแต่สิ่งทีควรจะทำคือการหาวิธีรองรับที่จะทำให้ได้รับผลกระทบน้อยที่สุดเมื่อความต้องการเปลี่ยนแปลงดังกล่าวเกิดขึ้น

2.2 ภาษายูเอ็มแอล

UML (Unified Modeling Language) หมายถึง ภาษาสัญลักษณ์รูปภาพมาตรฐาน (Standard Modeling Language) ที่ใช้กำหนดลักษณะคลาส การสร้างคลาสและเป็นเอกสารที่บอกถึงรายละเอียดของระบบโครงสร้าง โดยที่ยูเอ็มแอลจะแสดงความโครงสร้างของระบบเชิงวัตถุในรายละเอียดเล็กๆได้ดี ในรูปแบบของแผนภาพไดอะแกรม แผนภาพเหล่านี้จะก่อให้เกิดความเข้าใจที่ตรงกันระหว่างผู้ออกแบบระบบและ โปรแกรมเมอร์ ทำให้การปรับปรุงแก้ไข โปรแกรมทำได้ง่ายขึ้น

2.2.1 ความเป็นมาของภาษายูเอ็มแอล

ยูเอ็มแอลถูกเริ่มครั้งแรกในปี 1994 ที่บริษัท Rational Software โดย Grady Booch และ James Rumbaugh วัตถุประสงค์เบื้องต้นในการร่วมงานกันระหว่างทั้งสองคนเป็นการพัฒนากระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุที่เป็นหนึ่งเดียวกัน (Unified Method) โดยนำเอาวิธีของแต่ละคนคือวิธีของ Booch และวิธี OMT มารวมกันและปรับปรุงใหม่ ต่อมาในปี 1995 Ivar Jacobson ผู้พัฒนากระบวนการ OOSE หรือ Objectory ได้เข้าร่วมกับโครงการดังกล่าวซึ่งในครั้งนี้เป็นการสร้างภาษาโมเดลขึ้นมาใหม่เรียกว่า Unified Modeling Language (UML) และทั้งสามขนานนามตัวเองว่า Three Amigos

เนื่องจากวิธการพัฒนาซอฟต์แวร์เชิงวัตถุของทั้งสามมีชื่อเสียงอยู่แล้วในช่วงเวลานั้นดังนั้น ภาษายูเอ็มแอลที่พัฒนาขึ้นมาใหม่นี้จึงกลายเป็นที่นิยมอย่างแพร่หลายเช่นกัน ในปี 1996 มีหลาย

เอกสารที่อธิบายถึงวิธีที่หนึ่งนี้จึงกลายเป็นที่นิยมอย่างแพร่หลายเช่นกัน ในปี 1996 มีหลายไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริษัทขอเข้าร่วมในการพัฒนาภาษายูเอ็มแอล ซึ่งได้แก่ บริษัทจิตตอลอีคิวเมนต์ บริษัทเอสพี บริษัทไอบีเอ็ม รวมทั้งบริษัทออราเคิล บริษัทไมโครซอฟต์ และบริษัทอื่นๆอีกหลายบริษัท

2.2.2 ข้อดีของยูเอ็มแอล

ยูเอ็มแอลมีข้อดีหลายประการดังต่อไปนี้

- เป็นภาษารูปภาพมาตรฐานหรือภาษาสากลที่ใช้ในการพัฒนาซอฟต์แวร์เชิงวัตถุ และสามารถใช้ในการแลกเปลี่ยนโมเดลได้อย่างสื่อความหมาย รวมถึงการจัดสร้างเอกสารการวิเคราะห์ ออกแบบระบบ
- สามารถนำเสนอและสนับสนุนหลักการเชิงวัตถุได้อย่างครบถ้วนชัดเจน ทำให้นักพัฒนาระบบสามารถทำความเข้าใจกับปัญหาและค้นพบวิธีแก้ไขได้อย่างรวดเร็วและง่ายยิ่งขึ้น
- ไม่ผูกติดกับภาษาโปรแกรมใดภาษาหนึ่ง กล่าวคือโมเดลที่สร้างขึ้นจากภาษามาตรฐานซึ่งสามารถแปลงไปเป็นระบบจริงที่ถูกสร้างขึ้นด้วยภาษาโปรแกรมเชิงวัตถุ
- เป็นภาษาที่ง่ายต่อการทำความเข้าใจ ผู้ที่ทำการศึกษาหรือนำไปใช้งานไม่จำเป็นต้องมีความรู้อื่นใดนอกจากแนวคิดเชิงวัตถุ
- สามารถถูกแปลงเป็นภาษาที่ใช้ในการสร้างระบบขึ้นได้จริงได้อย่างอัตโนมัติ จึงเป็นการช่วยลดภาระเวลา และค่าใช้จ่ายการพัฒนาระบบได้อย่างมาก
- ยูเอ็มแอลใช้ในการบันทึกความคิดของนักพัฒนาในลักษณะของเอกสาร ที่พร้อมจะถูกนำมาทำความเข้าใจหรือสานต่ออีกครั้งด้วยความรวดเร็ว

2.2.3 องค์ประกอบของยูเอ็มแอล

ยูเอ็มแอลเป็นภาษาการสร้างโมเดลในแบบกราฟิกซึ่งใช้ในการประกอบเป็น ไดอะแกรม โดยมีกฎในการประกอบกันของอิตีเม้นต์ต่างๆ ไดอะแกรมจะแสดงถึงมุมมองต่างๆ (Multiple view) ของระบบ ซึ่งจะเรียกว่าโมเดล โมเดลยูเอ็มแอลจะบ่งบอกถึงรายละเอียดของระบบแต่จะไม่ระบุถึงวิธีการในการพัฒนาระบบ

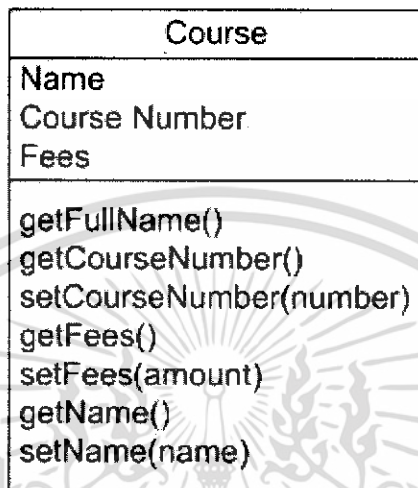
1. คลาสไดอะแกรม (Class Diagram)

คลาสไดอะแกรมเป็นแผนภาพที่ใช้แสดงคลาส และความสัมพันธ์ในแง่ต่างๆระหว่างคลาสเหล่านั้น ซึ่งความสัมพันธ์ที่กล่าวถึงในคลาสไดอะแกรมนี้ ถือเป็นความสัมพันธ์เชิงสถิตย์ (Static Relationship) หมายถึงความสำคัญที่มีอยู่แล้วในคลาสต่างๆ ซึ่งเรียกว่าความสัมพันธ์เชิงกิจกรรม (Dynamic Relationship) สิ่งปรากฏในคลาสไดอะแกรมนั้น ประกอบไปด้วยกลุ่มของคลาส และกลุ่มของความสัมพันธ์ โดยใช้สัญลักษณ์ที่ใช้ในการแสดงคลาสนั้น แทนด้วยสี่เหลี่ยมที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อผู้เอาไปเผยแพร่โดยไม่ได้รับอนุญาตให้เป็นการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบ่งออกเป็น 3 ส่วนโดยส่วนบนสุดจะเป็นชื่อคลาส ส่วนกลางจะแสดงแอททริบิวต์ ของคลาส และ ส่วนล่างจะแสดงโอเปอเรชันของคลาส

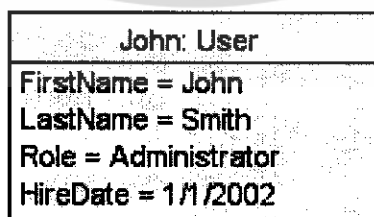
ตัวอย่างเช่น คลาสของ Course มีแอททริบิวต์คือ Name, Course Number และ Fees มีโอเปอเรชันคือ getFullName, getCourseNumber, setCourseNumber, getFees, setFees, getName, setName สามารถเขียนให้อยู่ในรูปแบบของยูเอ็มแอล คลาสไดอะแกรมได้ดังรูป



รูปที่ 2.1 คลาสไดอะแกรมของระบบ Course

2. ออบเจกต์ไดอะแกรม (Object Diagram)

ออบเจกต์ คือ อินสแตนซ์ของคลาส การกำหนดรายละเอียดของออบเจกต์ คือการกำหนดค่าให้กับแอททริบิวต์ ออบเจกต์ไดอะแกรมของยูเอ็มแอล จะแสดงเป็นรูปสี่เหลี่ยม โดยชื่อจะประกอบด้วยชื่อออบเจกต์และชื่อคลาส คั่นระหว่างชื่อด้วยเครื่องหมาย ":" และขีดเส้นใต้ที่ชื่อด้วย ตัวอย่างเช่น ออบเจกต์ User ชื่อ John มี FirstName คือ John, LastName คือ Smith, Role คือ Administrator และ HireDate คือ 1/11/2002 สามารถแสดงได้ดังรูป



รูปที่ 2.2 ออบเจกต์ไดอะแกรมของระบบ User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ยูสเคสไดอะแกรม (Use Case Diagram)

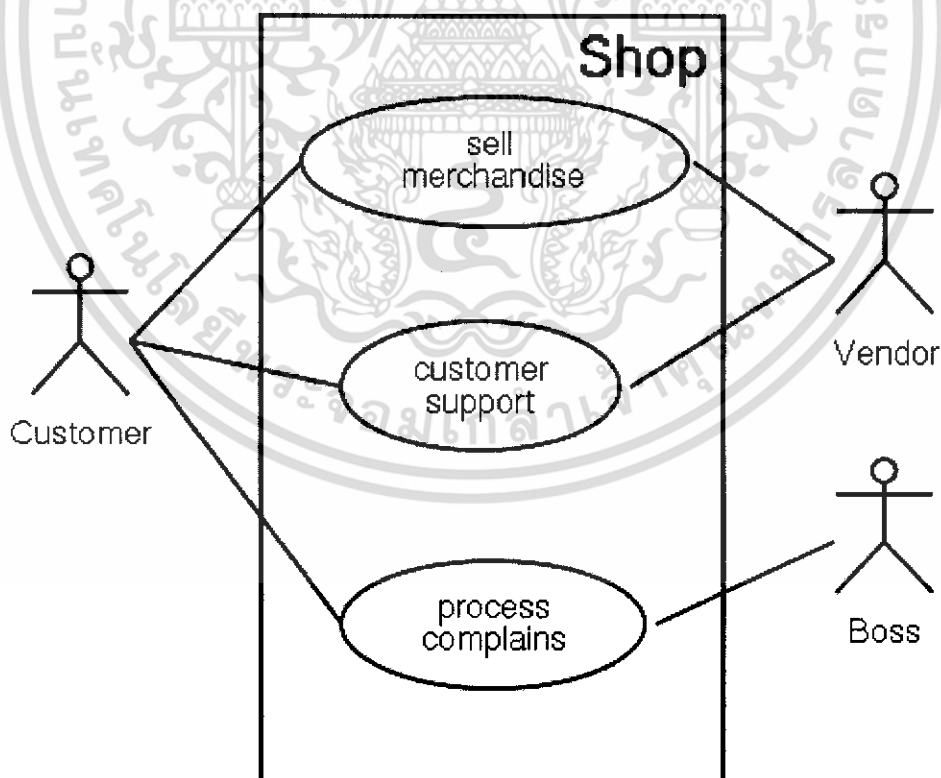
เป็นแผนภูมิที่ออกแบบมาเพื่ออธิบายถึงกิจกรรมของระบบจากมุมมองของผู้ใช้ซึ่งจะทำให้ได้ระบบที่ตรงกับความต้องการและสามารถนำไปใช้งานได้จริง ยูสเคสไดอะแกรมจะประกอบด้วย แอ็กเตอร์ (Actor), ยูสเคส (Use Case), เส้นแสดงความสัมพันธ์ (Relationship)

- แอ็กเตอร์ คือผู้ที่กระทำกับระบบ เป็นผู้ใช้ยูสเคสโดยแอ็กเตอร์ที่เริ่มต้นเหตุการณ์จะอยู่ด้านซ้ายของยูสเคส และแอ็กเตอร์ ที่ได้รับผลจากเหตุการณ์นั้นจะอยู่ทางด้านขวาของยูสเคส ชื่อของแอ็กเตอร์จะอยู่ทางด้านล่างของรูปที่ใช้แทนแอ็กเตอร์นั้น

- ยูสเคส เป็นกิจกรรมที่ทำโดย แอ็กเตอร์ซึ่งเป็นฟังก์ชันต่างๆ ที่ซอฟต์แวร์ต้องทำได้ ทั้งหมดจึงถือว่าซอฟต์แวร์ไม่มีข้อผิดพลาดหรืออีกนัยหนึ่งยูสเคส คือสิ่งที่บอกว่าระบบหรือซอฟต์แวร์ทำอะไรได้บ้าง โดย ยูสเคสจะใช้เครื่องหมายวงรีในการแสดง และชื่อของยูสเคสจะอยู่ภายในวงรีหรืออยู่ได้วงรี

Use Case Diagram คือความสัมพันธ์ระหว่าง ยูสเคส ซึ่งมี 2 ลักษณะคือแบบ inclusion และแบบ extension แบบ include เป็นการทำขึ้นตอนนี้ทุกครั้ง ส่วนแบบ extend จะเป็นการทำงานนี้เป็นบางครั้ง

- เส้นแสดงความสัมพันธ์ คือ เส้นที่บอกความสัมพันธ์ระหว่างกันทั้งของคลาส หรือแอ็กเตอร์



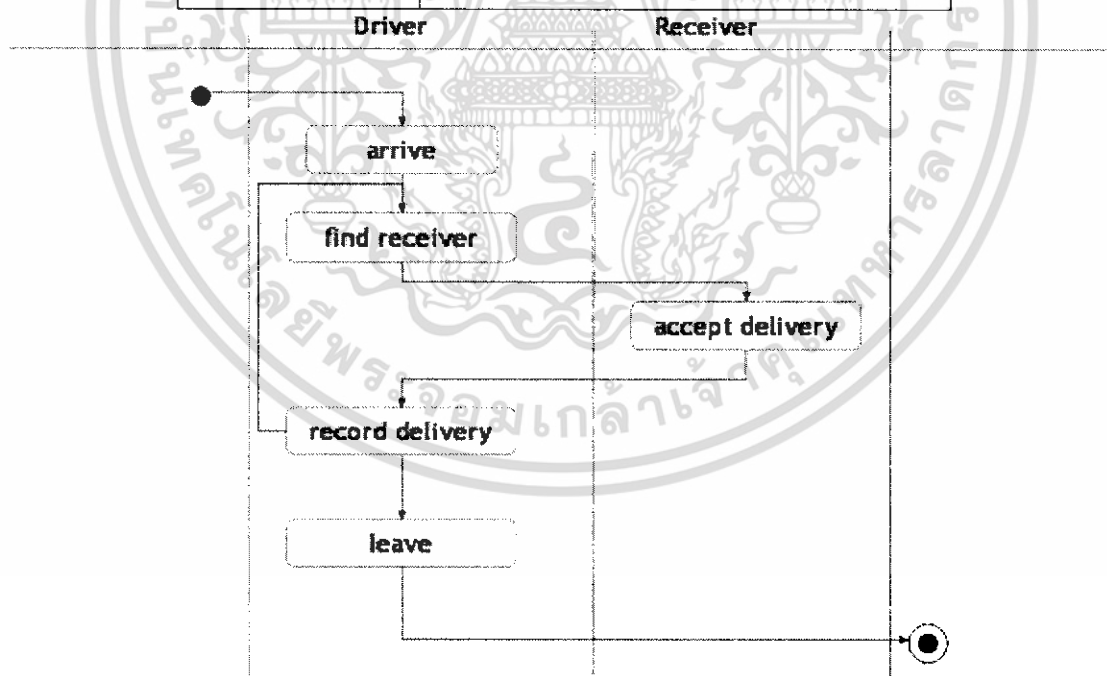
รูปที่ 2.3 ยูสเคสไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แอ็กทิวิตี้ไดอะแกรม (Activity Diagram)

แอ็กทิวิตี้ไดอะแกรมจะแสดงขั้นตอนการทำงานของยูสเคส โดยจะเน้นที่งานย่อยของวัตถุ ซึ่งการเจาะจงไปที่งานๆ หนึ่งของวัตถุนั้นจะรู้สึกเหมือนกับ สเตชาร์ตไดอะแกรมที่แสดงสถานะของวัตถุ แอ็กทิวิตี้ไดอะแกรมสามารถเปลี่ยนสถานะได้โดยไม่ต้องมีเหตุการณ์ที่กำหนดไว้ในไดอะแกรมมาก่อน แต่จะเปลี่ยนสถานะเองตามกระบวนการทำงานคล้ายกับโฟลชาร์ต สัญลักษณ์ที่ใช้ในแอ็กทิวิตี้ไดอะแกรม มีดังนี้

สัญลักษณ์	ความหมาย
●	เป็นจุดเริ่มต้นเข้าสู่แอ็กทิวิตี้ไดอะแกรม
◐	เป็นจุดสำหรับออกจากแอ็กทิวิตี้ไดอะแกรม
◇	เป็นจุดตัดสินใจเงื่อนไขว่าเป็นจริงหรือเท็จ
○ Action	เป็นกิจกรรมที่ทำ
→	แสดงขั้นตอนในการทำงานถัดไป



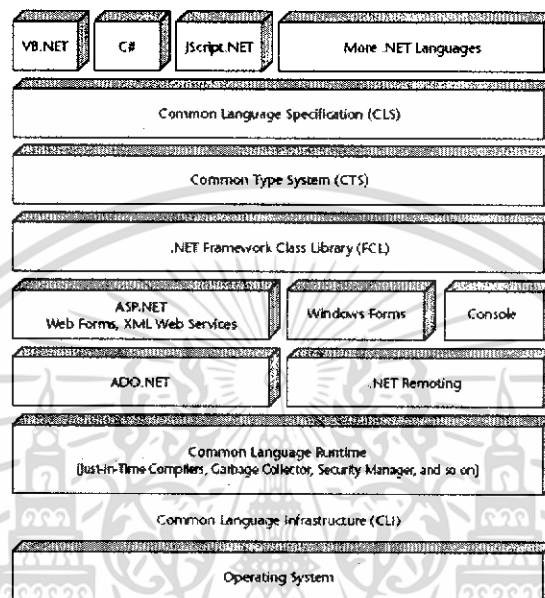
รูปที่ 2.4 แสดงตัวอย่างแอ็กทิวิตี้ไดอะแกรม มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 เทคโนโลยีคอตเน็ตเฟรมเวิร์ก (.NET Framework)

2.3.1 สถาปัตยกรรม .NET

โครงสร้างโดยรวมทั้งหมดของการสร้างแอปพลิเคชันคอตเน็ตซึ่งแสดงได้ดังโครงสร้างด้านล่าง



รูปที่ 2.5 รูปแสดง โครงสร้างการพัฒนาแอปพลิเคชันคอตเน็ต

จากรูป เป็นการแสดงถึงสถาปัตยกรรมของแอปพลิเคชัน และสถาปัตยกรรมที่ใช้โดยมีเลเยอร์ล่างสุด คือ .NET Framework SDK เปรียบเสมือน Runtime Library ที่รันอยู่คอยสนับสนุนการทำงานของแอปพลิเคชันจากนั้นจะเป็นเลเยอร์ของ Common Language Runtime ซึ่งเป็นผลลัพธ์ของการคอมไพล์แอปพลิเคชันคอตเน็ต เลเยอร์ถัดขึ้นมาเป็นเครื่องมือ (Tools) และ เทคนิคต่างๆที่สามารถใช้พัฒนาแอปพลิเคชันได้ทั้งในเรื่องของ Web service, ADO.NET, ASP.NET จนกระทั่งถึงเลเยอร์บนสุดคือ ภาษาที่ใช้ในการพัฒนาแอปพลิเคชันด้วยวิซวลสตูดิโอคอตเน็ต

2.3.2 เลเยอร์ คอมมอนแลงเวจรันไทม์ (Layer Common Language Run Time)

ก่อนที่จะมีการพัฒนาโปรแกรมเป็นเชิงวัตถุ (Object Oriented) นั้น แอปพลิเคชันแต่ละตัวเปรียบเสมือนกล่อง ภายในแอปพลิเคชันก็จะมีโค้ด (Code) และมีโครงสร้างข้อมูล (Data Structure) ต่างๆของตัวเอง มีฟังก์ชันต่างๆของตัวแอปพลิเคชันนั้น

การที่แอปพลิเคชันต่างๆ จะมีการเรียกใช้การทำงานร่วมกันและกัน หรือมีการส่งผ่านข้อมูลถึงกันและกัน เป็นสิ่งที่ทำได้ยาก ซึ่งอาจต้องมีการกำหนดว่าจะไร้ขึ้นมาเองระหว่าง 2 แอปพลิเคชันนั้นๆ จนกระทั่งในยุคถัดมา ไมโครซอฟท์ได้คิดค้นเทคโนโลยี COM (Component

Object Model) เป็นวิธีที่ทำให้เราเขียนโปรแกรมเป็นแบบเชิงวัตถุ และเรียกใช้การทำงานที่มาจากต่างแอปพลิเคชันได้

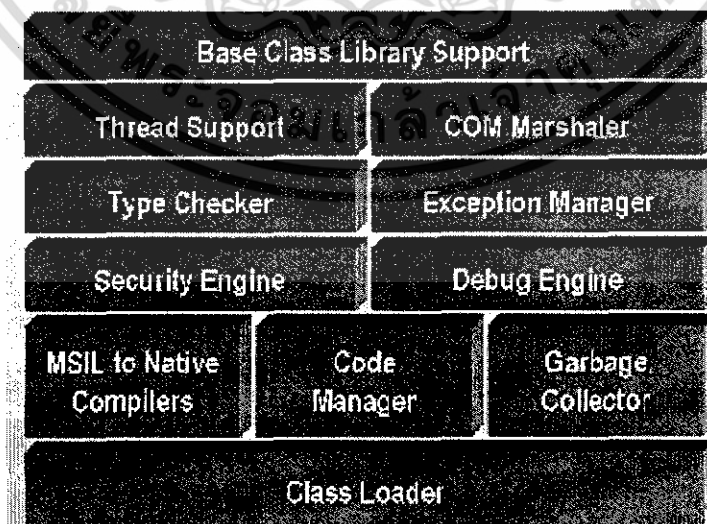
หากเราจะอธิบายให้ง่ายขึ้น ก็คือเปรียบเทียบเราเอาแพ็คเกจอันหนึ่งห่อแอปพลิเคชันของเราไว้ โดยการห่อหุ้มหรือสื่อสารกันของแอปพลิเคชันก็ห่อหุ้มผ่านแพ็คเกจที่เราห่อไว้ จนมาถึงตัววิศวกรสตูดิโอคือทเน็ต ที่ได้รับการออกแบบใหม่ จะเห็นว่าจากรูปเดิม กล่องแพ็คเกจหายไปถ้าเราพัฒนาด้วยรูปแบบเทคโนโลยีคือทเน็ตนั้น คลาสต่างๆ สามารถติดต่อกันได้โดยตรง

การพัฒนาแอปพลิเคชันด้วย วิศวกรสตูดิโอคือทเน็ตนั้นเมื่อเรากอมไพล์สิ่งที่เราจะได้ จะไม่ใช่ไค้ดไบนารี (Binary code) เลยทีเดียว แต่ได้เป็นภาษากลางอันหนึ่งเรียกว่า MSE (Microsoft Intermediate Language) ซึ่งเป็นภาษาในระดับเลเยอร์ต่างๆ โครงสร้างจะเหมือนภาษา Assembly ภายในสิ่งที่เกิดขึ้นก็จะเป็น Microsoft Intermediate Language ตัวนี้ ภายในตัวมันจะประกอบด้วยสองส่วนคือ ไค้ดกับตัวแอททริบิวต์หรือพรีอเพอร์ตีต่างๆที่ใช้อธิบายตัวไค้ดนั้นซึ่งเรียกว่า เมต้าดาต้า (Meta Data)

จากนั้น เมื่อไค้ดซึ่งเป็นภาษากลาง (Intermediate Language) ถูกเรียกใช้งานจริงๆ จะมีตัวแปลภาษาตัวหนึ่งมาทำการคอมไพล์ไค้ดตัวนั้นให้เป็นไค้ดไบนารี ซึ่งตัวแปลภาษานั้นจะเรียกว่า JIT Compiler (Just In Time Compiler) เพราะว่าจะมีการคอมไพล์เมื่อมีการใช้งาน ฉะนั้นคลาสหรือไค้ดต่างๆ ที่เราพัฒนาขึ้นแล้วจะถูกคอมไพล์มาเป็นภาษากลาง ที่มีโครงสร้างภาษาแบบเดียวกัน เพราะฉะนั้นคลาสต่างๆ ในแอปพลิเคชันจึงสามารถทำงานได้อย่างกลมกลืนกันและไม่มีข้อผิดพลาดใดๆ

2.3.2.1 การทำงานของ คอมมอนแลงเควจรันไทม์ (Common Language Runtime)

ภายในตัวคอมมอนแลงเควจรันไทม์ จะมีโมดูล (Module) ย่อยๆซึ่งเป็นสถาปัตยกรรมภายในดังรูป



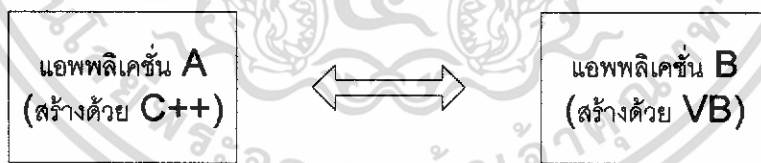
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ Microsoft. การใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย. กรุณาแจ้งผู้ดูแลระบบหากพบข้อผิดพลาดใดๆ. **รูปที่ 2.6** แสดงสถาปัตยกรรมของคอมมอนแลงเควจรันไทม์

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านล่างสุดจะมี Class Loader ซึ่งเอาไว้โหลดโปรแกรมของเราขึ้นมาทำงาน นอกจากนี้ก็จะมีคอมไพเลอร์ ซึ่งจะทำการคอมไพล์ภาษากลาง ให้เป็นภาษาไบনারีโดยจะมีโค้ดแมนเนเจอร์ (Code Manager) และ การ์เบจคอลเลคเตอร์ (Garbage Collector) คอยจัดการกับหน่วยความจำที่เราจองเวลาเรียกใช้งาน นอกจากนี้ก็จะมีเรื่องความปลอดภัย (Security) ในการทำงาน รวมทั้ง Debug Engine ในการดักจับรันไทม์เออเรอร์ (Runtime Error) และ แอ็กเซ็ชชันแมนเนเจอร์ การตรวจเช็คชนิดของตัวแปรต่างๆ และด้านบนสุดจะเป็นการใช้งานร่วมกับไลบรารีคลาสต่างๆซึ่งจัดเตรียมมาให้ เพราะฉะนั้นส่วนของเลเยอร์ของคอมมอนแลงเกจจรันไทม์จะสามารถคอมไพล์แอปพลิเคชันใดก็ตาม โดยไม่ว่าจะเป็น ASP.NET หรือเขียนแอปพลิเคชันบนวินโดวส์ธรรมดา หรือจะเป็นการเขียนเว็บเซอร์วิสก็ตาม สิ่งที่ได้จากการคอมไพล์จะเป็น คอมมอนแลงเกจจรันไทม์ ตามแผนภาพนี้

นั่นคือ จากการออกแบบเพื่อสนับสนุนการทำงานร่วมกันใน คอมมอนแลงเกจจรันไทม์ จึงจัดข้อเสียของ COM ออกไปได้ เนื่องจากข้อเสียของ COM คือเป็นเพียงการเอาอะไรบางอย่างมาห่อคลาสเอาไว้เท่านั้น ดังนั้นเวลาเอา COM ไปใช้งานจึงค่อนข้างยุ่งยาก รวมทั้งถ้ามีการเปลี่ยนแปลง COM นั้นๆก็จะทำให้เกิดปัญหาเข้ากันไม่ได้ (compatibility) ระหว่างเวอร์ชันเดิมกับเวอร์ชันปัจจุบัน แต่ถ้าเราพัฒนาด้วยวิซวลสตูดิโอคอตเน็ต ข้อเสียของ COM ก็จะถูกกำจัดเสีย

ความจริงคอมมอนแลงเกจจรันไทม์ เป็นหลักการวิวัฒนาการมาจาก COM อีกทีหนึ่ง เป็นเชิงวัตถุที่แกนของภาษาเลย โดยวิซวลสตูดิโอคอตเน็ตนั้น ถูกออกแบบเพื่อสนับสนุนเชิงวัตถุ โดยเฉพาะคลาสต่างๆที่อยู่ในแต่ละแอปพลิเคชันสามารถติดต่อกันได้โดยตรงใน แอปพลิเคชัน A เราอาจเขียนคลาสด้วย ภาษา C++ และแอปพลิเคชัน A อาจจะติดต่อกับ แอปพลิเคชัน B ซึ่งเขียนด้วยภาษา Visual Basic ได้คือการสืบทอดข้ามภาษานั้นสามารถทำได้



รูปที่ 2.7 แสดงความสามารถในการติดต่อข้ามแอปพลิเคชัน เมื่อเขียนด้วยภาษาที่ต่างกัน

ในวิซวลสตูดิโอคอตเน็ตจะคอมไพล์เป็นภาษาเดียวกันคือ ภาษากลาง ตามรูป นอกจากนี้ยังสามารถทำงานด้วยกันกับ COM แบบเดิมที่เราเคยเขียนมาแล้วได้ด้วยซึ่งใช้วิซวลสตูดิโอคอตเน็ตเหล่านี้ก็ไม่จำเป็นต้องโยนทิ้ง ในวิซวลสตูดิโอคอตเน็ต เราสามารถเรียกใช้งาน คอมโพเนนต์ที่เขียนด้วยวิซวลสตูดิโอคอตเน็ตได้เช่นกัน คือเป็นการเข้ากันได้ทั้งสองทาง (Backward Forward Compatibility) นี่ก็คือข้อดีมากๆ ของ คอตเน็ต ทำให้เราไม่ต้องพัฒนาโค้ดใหม่

2.3.2.2 การจัดการเกี่ยวกับหน่วยความจำเมื่อทำการประมวลผล

เนื่องจากการทำงานทั้งหมดของตัวคือทนี่จะมีการดูแลเรื่องคอมมอนแลงเกจจันไท้มี ปัญหาหนึ่งที่มีมักจะพบเมื่อเราพัฒนาแอปพลิเคชัน คือเรื่องของการ์เบจคอลเลคชันในการจัดการกับ หน่วยความจำ โดยเฉพาะบางภาษาอย่างเช่น ภาษาซีพลัสพลัสที่ต้องการมีการเรียกใช้งานพอยเตอร์ (Pointer) ค่อนข้างมาก ในการใช้งานพอยเตอร์นั้น ถ้าเราทำผิดไปนิดเดียวก็อาจทำให้แอปพลิเคชัน ของเราหยุดการทำงานในระหว่างรันหรือประมวลผลก็ได้ทำให้ถ้าแอปพลิเคชันของเรารันไปเรื่อยๆ หน่วยความจำเหล่านี้ก็จะขยายไปเรื่อยๆ และสักพักหนึ่งก็จะเกิดการหยุดไป

การแก้ไขข้อผิดพลาด (Debug) ตรงนี้ทำได้ยากมาก เพราะกว่าจะรู้ว่าเกิดความผิดพลาดขึ้น เราก็กี่เวลาในการรันไปแล้ว ซึ่งการจัดการกับหน่วยความจำนี้ ถ้าเราพัฒนาด้วยแพลตฟอร์ม ของคือทนี่จะมี เครื่องมือหนึ่งเรียกว่า การ์เบจคอลเลคเตอร์ เป็นตัวคอยจัดการเกี่ยวกับ หน่วยความจำให้เรา ความจริงตัวการ์เบจคอลเลคเตอร์ จะเป็นตัวทำการกำหนดค่าหน่วยความจำให้ ว่างให้เราเองในส่วนที่เราไม่ใช้งาน โดยอัตโนมัติ นี่คืข้อดีอันหนึ่งในการพัฒนาด้วย แพลตฟอร์มของคือทนี่

2.3.2.3 ระบบตรวจจับความผิดพลาด

นอกจากนี้ในเรื่องของเอ็กเซ็ปชันแมนเนเจอร์ (Exception Manager) หรือการดักรันไทม์เออ เรอร์ (Runtime Error) ในการสร้างแอปพลิเคชันที่มีการรันได้อย่างดีนั้น การดักรันไทม์เออเรอร์ที่ จะเกิดขึ้นในระหว่างการทำงาน ก็เป็นปัจจัยหนึ่งที่เราต้องคำนึงถึง ซึ่งในตัวคือทนี่ก็จะมี โครงสร้างภาษาในการดักรันไทม์เออเรอร์ ที่มีลักษณะเป็น โครงสร้างทำให้หาได้ง่ายขึ้น เรา เรียกว่าสตักเจอร์เอ็กเซ็ปชันแฮนดิง (Structure Exception Handling)

2.3.2.4 รูปแบบการคอมไพล์แอปพลิเคชัน

นอกจากนี้การคอมไพล์แอปพลิเคชันคือทนี่มีให้เลือกหลายแบบ บางคนอาจจะคอมไพล์ ให้เป็นแบบ โค้ดไบนารีเลย ไม่ต้องคอมไพล์ให้เป็นอินเทอร์พรีเตอร์มีเดียคท์แลงเกจก็สามารด จะพัฒนา ได้โดยใช้วิซวลซีพลัสพลัสค่อทนี่ (Visual C++ .NET) เรียกว่าเมเนจซีพลัสพลัส (Manage C++) หรือเราเลือกใช้เครื่องมือภาษาอื่นๆที่มีความง่ายในการพัฒนาอีกว่า คือ วิซวลเบสิก หรือ ซีชาร์ป ก็สามารดคอมไพล์เป็นอินเทอร์พรีเตอร์มีเดียคท์แลงเกจได้

ในการคอมไพล์จะไม่มีตัวอินเทอร์พรีรันไทม์ไลบรารี (Interpreter Runtime Library) ต่างๆ เช่น วิซวลเบสิก ที่ต้องมีรันไทม์ไลบรารีของตัวเองเวลาพัฒนาด้วย วิซวลสตูดิโออันนี้ก็จะไม่ จำเป็นต้องที่อีกต่อไป เราใช้ตัวรันไทม์อันเดียวกัน ก็คือตัว ค่อทนี่เฟรมเวิร์ค เป็นตัวรันไทม์ ไลบรารี ที่กล่าวมาข้างต้น

2.3.2.5 รูปแบบการทำงานร่วมกับภาษาอื่นๆ

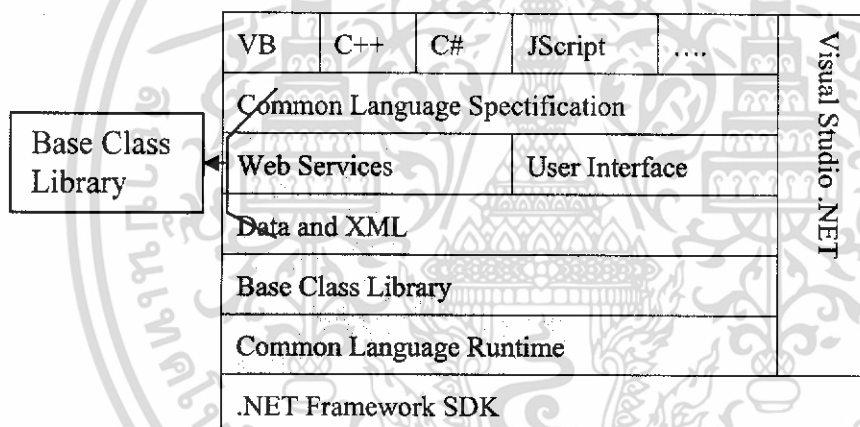
การพัฒนาแอปพลิเคชันในค่อทนี่ รวมทั้งเทคนิคต่างๆ ที่กล่าวมาแล้วนั้น ไม่จำเป็นจะใช้ เฉพาะภาษาตระกูลในวิซวลสตูดิโอไม่ว่าจะเป็นวิซวลเบสิกค่อทนี่ (VB.NET), ซีชาร์ป (C#) หรือ

ซีพลัสพลัส (C++) ถ้าภาษาใดสามารถคอมไพล์มาให้เป็นคอมมอนแลงเกจจันใหม่ได้ภาษานั้นก็จะสามารถใช้คุณลักษณะทั้งหมดของคอตเน็ตได้ซึ่งตอนนี้มีภาษาอื่นๆที่สามารถทำงานร่วมกับแอปพลิเคชันคอตเน็ต มากกว่า 20 ภาษาที่พัฒนาตัวคอมไพล์ขึ้นมาเพื่อคอมไพล์ภาษาต่างๆ ให้อยู่ในรูปของแพลตฟอร์ม คอตเน็ต ไม่ว่าจะเป็นภาษาโคบอล (COBOL), ปาสคาล(PASCAL) และอื่นๆ

ในตัววิซวลสตูดิโอคอตเน็ต ก็มีเครื่องมือระดับสูง(High Level Tools) ที่จะทำให้เราพัฒนาแอปพลิเคชันได้สะดวกและง่ายมากขึ้น นี่คือเรื่องของคอมมอนแลงเกจจันใหม่ ซึ่งถือได้ว่าเป็นเลย์เออร์ล่างสุดของการคอมไพล์แอปพลิเคชันที่เขียนด้วยวิซวลสตูดิโอคอตเน็ต

2.3.3 เลเยอร์เบสคลาสไลบรารีแลงเกจ (Base Class Library Language)

เลเยอร์ถัดมาเป็นของโครงสร้างการพัฒนาแอปพลิเคชันคอตเน็ตต่อจากคอมมอนจันใหม่แลงเกจ ก็คือเบสคลาสไลบรารี



รูปที่ 2.8 เบสคลาสไลบรารีของสถาปัตยกรรมคอตเน็ต

เทคนิคต่างๆที่วิซวลสตูดิโอคอตเน็ตจัดเตรียมให้เราในการใช้งานนั้นเบสคลาสไลบรารีเปรียบเสมือนการรวบรวมเอาฟังก์ชันของเอพีไอ (API: Application Programming Interface) ทั้งสมมติว่าตอนที่เรพัฒนาแอปพลิเคชันด้วยวิซวลสตูดิโอ เวลาเราเรียกใช้งานฟังก์ชันบางอย่างในระดับสูง หรือต้องการทำงานแบบลึกๆกับระบบเรามักจะเรียกใช้เอพีไอ ภาษาในวิซวลสตูดิโอมีความสามารถในการเรียกฟังก์ชันมีความสามารถในการเรียกฟังก์ชันเอพีไอได้ต่างกัน

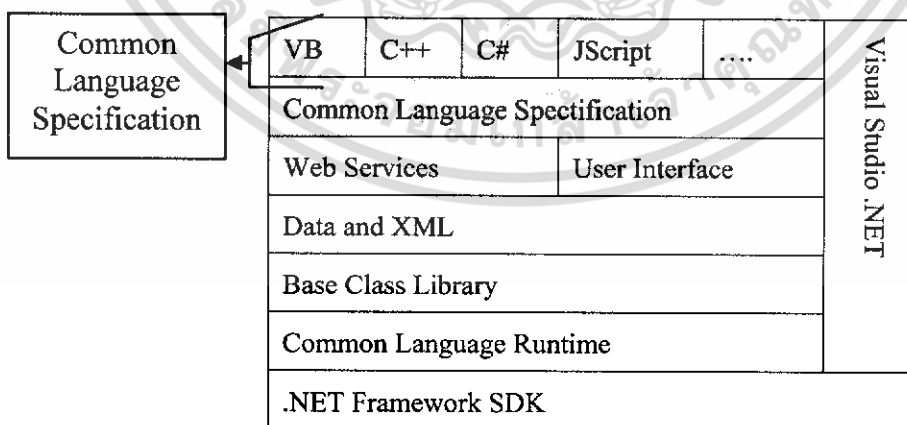
ตัวเบสคลาสไลบรารี ก็คือการที่เรารวบรวมฟังก์ชันเอพีไอซึ่งกระจัดกระจายอยู่เวลาจะเรียกใช้เราต้องไปค้นหาในเมนูช่วยเหลือ (Help) นั่นคือเบสคลาสไลบรารี พยายามที่จะรวบรวมเอพีไอและฟังก์ชันทั้งหมดเกี่ยวกับระบบเข้ามาไว้ในลักษณะของเชิงวัตถุทำเป็นคลาสอันหนึ่งซึ่งเป็นมาตรฐาน เป็นคลาสที่สร้างมาในตัวระบบเรียบร้อยแล้ว ซึ่งคลาสทั้งหมดจะอยู่ภายใต้คลาสหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อันหนึ่งที่เรียกว่า ซิสเต็ม (System) ทุกอย่างที่พัฒนาด้วยวิซวลสตูดิโอคือทเนท จะเป็นเชิงวัตถุทั้งหมด โดยมีคลาสที่ใหญ่ที่สุดเรียกว่าคลาสซิสเต็ม ซึ่งภายในคลาสซิสเต็ม จะมีคลาสย่อยๆ มากมาย ซึ่งแต่ละอันจะสนับสนุนฟังก์ชันเอพีไอหรือสนับสนุนการทำงานที่เราต้องการได้ไม่ว่าจะเป็นเรื่องของการทำกราฟิก การทำเกี่ยวกับโครงสร้างข้อมูลการทำเกี่ยวกับเรื่องเครือข่าย ฟังก์ชันเอพีไอเหล่านี้จะถูกจัดกลุ่มให้เป็นเชิงวัตถุ อยู่ภายใต้คลาสซิสเต็ม การเรียกชานคลาสซิสเต็ม ถ้าเป็นวิซวลเบสิกกับซีชาร์ปก็ใช้งานได้ทั้ง 2 อย่าง

2.3.4 เลเยอร์คอมมอนแลงเกจสเปคซิฟิเคชัน (Common Language Specification Layer)

เลเยอร์สุดท้ายในสถาปัตยกรรมคือทเนทที่เราจะพูดถึงคือเลเยอร์คอมมอนแลงเกจสเปคซิฟิเคชัน เรื่องมาตรฐานบนพื้นฐานคือทเนท ซึ่งคอมไพเลอร์จะต้องทำงานตามมาตรฐานดังกล่าว เพื่อให้สามารถทำงานร่วมกับภาษาบนพื้นฐานคือทเนท และภาษาอื่นๆ ได้ ไมโครซอฟท์ได้ทำการปรับภาษาต่างๆ เช่น ซีชาร์ป, วิซวลเบสิก ให้เข้ากับมาตรฐานคือทเนท นอกจากนั้นผู้ผลิตรายอื่นสามารถพัฒนาตามข้อกำหนดนี้เพื่อสามารถทำงานบนพื้นฐานคือทเนทได้ เลเยอร์ที่สามของสถาปัตยกรรมคือทเนท ก็คือเครื่องมือในการสร้างแอปพลิเคชัน แต่สิ่งที่อยู่เหนือกว่าทุกอย่างก็คือภาษาที่เราใช้งานคือทเนท นั้นมีข้อดีคือ ต้องสนับสนุนมาตรฐานเดียวกัน เรียกว่า คอมมอนแลงเกจสเปคซิฟิเคชัน ซึ่งไมโครซอฟท์ได้จดทะเบียนมาตรฐานนี้เข้ากับองค์กร ECMA แล้วซึ่งเป็นองค์กรดูแลเรื่องโครงสร้างภาษาต่างๆ และก็เป็นแบบเปิดด้วยเพราะฉะนั้นเจ้าของภาษาอื่นก็สามารถสร้างตัวแปลภาษา หรือคอมไพเลอร์ เพื่อคอมไพล์ภาษาของเขาให้มาเป็นคอมมอนแลงเกจสเปคซิฟิเคชัน อันนี้



รูปที่ 2.9 คอมมอนแลงเกจสเปคซิฟิเคชันของสถาปัตยกรรมคือทเนท

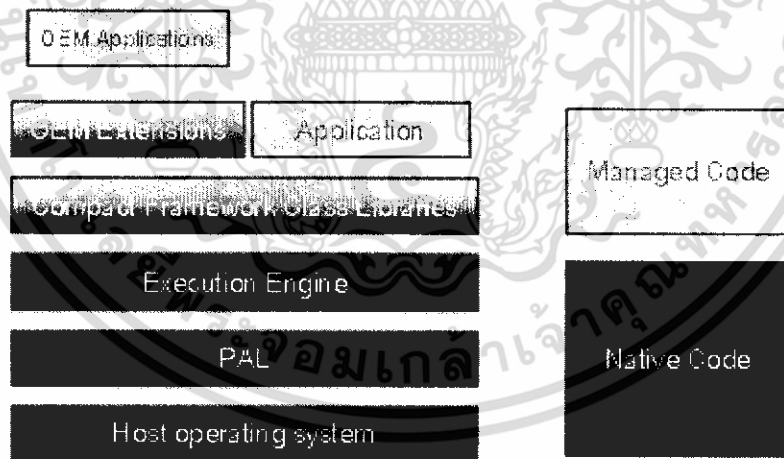
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 สถาปัตยกรรมของคอตเน็ตคอมแพคเฟรมเวิร์ก (.NET Compact Framework)

ส่วนหนึ่งของคอตเน็ตคอมแพคเฟรมเวิร์ก ถูกสร้างขึ้นโดย เป็นส่วนย่อยของ เดสทอปเฟรมเวิร์ก (Desktop Framework) โดยมีเป้าหมายหลักคือ เข้ากันได้กับทุกแพลตฟอร์ม (Platform)

.NET Framework คือ รูปแบบโครงสร้างพื้นฐานของคอมพิวเตอร์แบบใหม่ ออกแบบมาเพื่อรองรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการวินโดวส์ ในรูปแบบขั้นสูงของสิ่งแวดล้อมบนอินเทอร์เน็ต โดยมี Common Language Runtime (CLR) คือ หัวใจของ .NET Framework โดยมีหน้าที่สำหรับ compiled โปรแกรม แล้วยังจัดการหน่วยความจำผ่าน garbage collection โปรแกรมใดๆที่ทำงานใช้ CLR จะเรียกว่า managed code ส่วนที่ไม่ใช่จะเรียกว่า unmanaged code

Namespaces คือ กลุ่มเทคนิคที่เกี่ยวข้องกันรวมกันอยู่ในรูปของ Namespace เพื่อใช้ค้นหาหรืออ้างอิงได้ง่ายขึ้น กล่าวคือทุกคลาสที่เขียนใน ภาษา C# จะเป็นของ Namespace โดยทั่วไปคลาสจะเป็นของ Namespace ที่มีชื่อเดียวกับ โปรเจกต์นั้นๆ (ทั้งนี้รวมถึงชื่อที่ตั้งค่าเริ่มต้นไว้) อย่างไรก็ตามยังสามารถเปลี่ยนชื่อโปรเจกต์ได้ โดยที่ Namespace จะใช้สำหรับคลาสทั้งหมดโดยใช้โปรเจกต์ properties Namespaces จะจัดการเส้นทางกรสืบทอดคุณสมบัติ ของชื่อคลาสเพื่อหลีกเลี่ยง 'Name pollution'

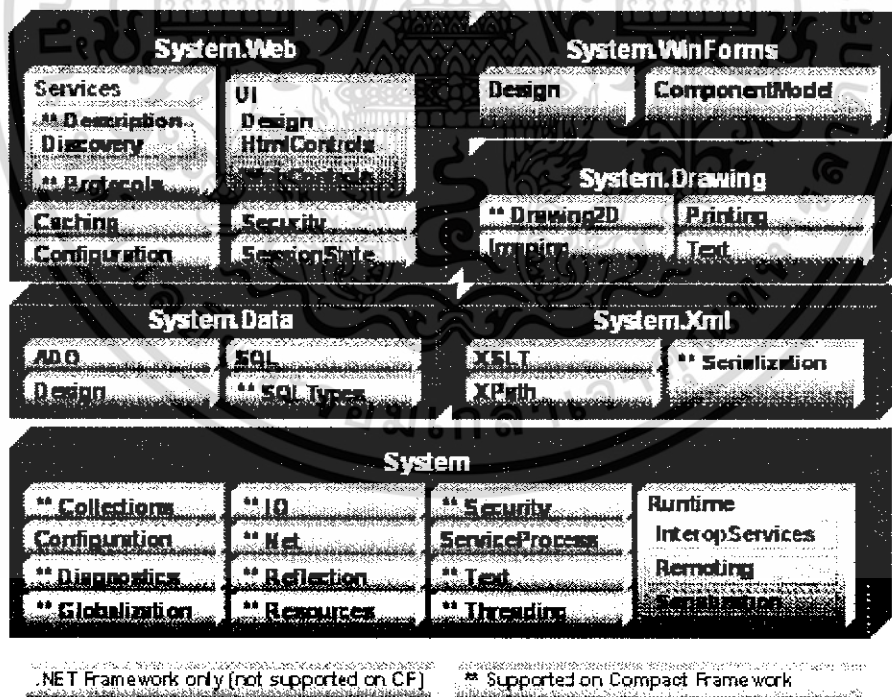


รูปที่ 2.10 แสดงสถาปัตยกรรมของคอตเน็ตคอมแพคเฟรมเวิร์ก

ขนาดของคอมแพคเฟรมเวิร์ก จะถูกติดตั้งบนอุปกรณ์ต่างๆ แต่ละอุปกรณ์ โดยปกติจะมีขนาด 1.7 MB - 2.6 MB และสามารถติดตั้งใน RAM, ROM หรือ FlashROM โดยในอนาคต คาดว่าถ้าเราเริ่มด้วย RAM เราจะสามารถให้ใช้งานได้ทุกอุปกรณ์ทันที OEMs จะทำให้ FlashROM พัฒนาขึ้น และสามารถรวมอยู่ในคอมแพคเฟรมเวิร์ก สำหรับอุปกรณ์พวก Pocket PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Host Operating System เป็นชั้นล่างสุดของคอมแพคแฟรมเวิร์ค โดยส่วนที่เป็นโค้ด ที่เขียนสำหรับคอมแพคแฟรมเวิร์ค จะถูกประมวลผลบน Host Operating System เช่น Windows CE โดยคอมแพคแฟรมเวิร์คจะถูกรันบน Windows CE 3.0 และ Windows CE.NET 4.1
- PAL คือส่วนประกอบที่สำคัญ ที่จะทำให้ฟอร์ทของแพลตฟอร์มสามารถทำงานได้ ซึ่งจะบรรจุระบบย่อยๆ ที่เป็นฟังก์ชันการทำงานของระบบปฏิบัติการ (OS) และระบบฮาร์ดแวร์ (Hardware) โดยที่ฮาร์ดแวร์จะอยู่ในส่วนของ API, NSL, Execution Engine ตัวอย่างเช่น PAL จะประกอบด้วย อินเตอร์เฟส (Interface) สำหรับอุปกรณ์ต่างๆ, ระบบจัดการหน่วยความจำ และพอร์ตติดต่ออุปกรณ์ภายนอก (I/O Port)
- Execution Engine จะมีกลุ่มของบริการที่สำคัญๆ ซึ่งแต่ละส่วนจะทำได้เหมือน CLR โดยให้บริการที่เหมือนกันสำหรับเดสทอปและเซิร์ฟเวอร์ แอปพลิเคชัน ซึ่งต้องมีการประมวลผลบนคือทเน็ตแอปพลิเคชัน แต่อาจจะใช้ทรัพยากรมาก แต่ในความเป็นจริงแล้ว ตัวหน่วยความจำมันจะมีข้อจำกัด ซึ่งในตัวของ Execution Engine จะถูกสร้างมาโดยคำนึงถึงข้อจำกัดนี้



รูปที่ 2.11 แสดงคลาสไลบรารีของค็อตเน็ตคอมแพคแฟรมเวิร์ค

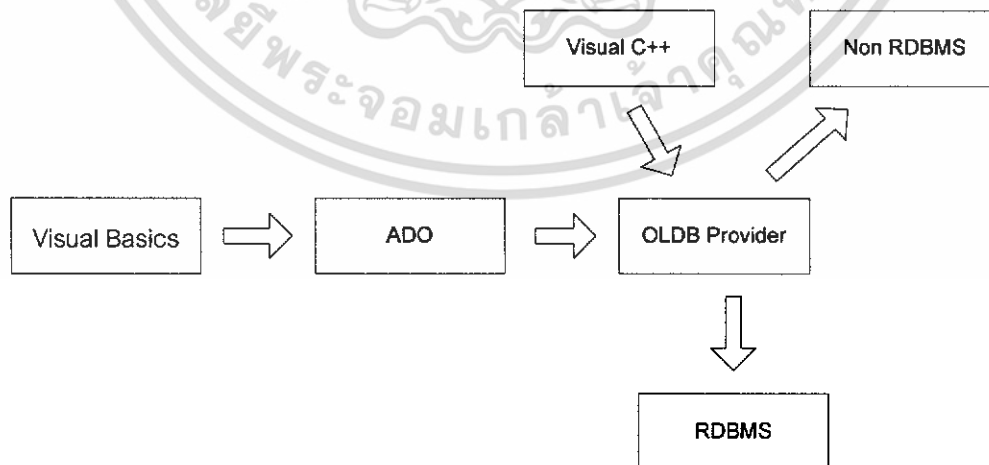
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Class Library เพื่อที่จะสร้าง ไรบัสโปรแกรมคอมแพคเฟรมเวิร์ค ก็มีกลุ่มของ Class Library เป็นภาษาในการสร้างไฮราคีคอลเนมสเปซ (Hierarchical namespace) ซึ่งก็จะเหมือนกับที่เราเจอในเดสท็อปเฟรมเวิร์ค แต่ก็จะมีส่วนที่ต่างกันอยู่ 4 อย่าง คือ
 1. อาจจะมองได้ว่า คอมแพคเฟรมเวิร์ค เป็นซัพเซตของ Desktop Framework
 2. ส่วนที่สองที่ทำให้ คอมแพคเฟรมเวิร์ค ต่างจาก Desktop Class Library โดยที่มีการพัฒนาต่างจากเดิม
 3. คอมแพคเฟรมเวิร์คจะสามารถรองรับ Name space ขึ้นอีกสองชนิด โดยแต่ละชนิดมีวิธีการที่จะแอสเซมบลีของตัวเอง ซึ่งจะทำให้มีการประมวลผลที่ดีขึ้นคล้ายกับว่าทำให้อุปกรณ์ฉลาดขึ้น
 4. คอมแพคเฟรมเวิร์คจะสามารถรองรับการทำงานได้น้อยกว่าของ เดสท็อปเฟรมเวิร์ค ซึ่งส่วนที่ คอมแพคเฟรมเวิร์คสามารถรองรับได้ มักจะถูกอ้างอิงในนามของ Common Type System (CTS) เพราะว่าเป็นส่วนสำคัญที่สามารถรองรับได้

2.5 เทคโนโลยีเอดีโอดีเอทเน็ต (ADO.NET)

เทคโนโลยีเอดีโอดีเอทเน็ต (ActiveX Data Object Dot Net) เป็น เอพีไอ ที่ใช้ในการติดต่อกับฐานข้อมูลต่างๆ ซึ่งตัวเอดีโอดีเอทเน็ต นี้ถูกสร้างมาเพื่อทดแทนอินเทอร์เน็ตเฟสแบบเก่าที่มีความจำกัดในหลายๆอย่างคือ DAO (Data Access Objects) และ RDO (Remote Data Objects: R)

ตัวเอดีโอดีเอทเน็ตเป็นเทคโนโลยีใหม่อีกหนึ่งที่ช่วยในเรื่องการติดต่อกับแหล่งข้อมูลต่างๆ และเอดีโอดีเอทเน็ต ยังมีโครงสร้างเหมือนเอดีโอ เพียงแต่การออกแบบนั้นมีความแตกต่างกันบ้าง



รูปที่ 2.12 สถาปัตยกรรมการเชื่อมต่อข้อมูลผ่านเอดีโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเก็บข้อมูลในเอดีโอ เมื่อมีการเลือกข้อมูลนั้น ผลลัพธ์ที่ออกมาจะเก็บข้อมูลในโครงสร้างของ COM (Common Object Model) แต่ตัวเอดีโอคือทเน็ต เมื่อเราเลือกข้อมูลออกมาแล้วต่างๆที่เลือกออกมาจะเก็บไว้ในรูปข้อมูลของ เอ็กซ์เอ็มแอล (XML) เพราะฉะนั้น เอดีโอคือทเน็ตจึงเหมาะกับการทำงานที่มี เป็นแอปพลิเคชันบนอินเทอร์เน็ต

ถ้าเป็นเอดีโอปกติ จะเหมาะกับการทำงานที่เป็น ไคลเอนต์/เซิร์ฟเวอร์ ที่อยู่ในเครือข่ายแลนหรือเครือข่ายอินเทอร์เน็ต (Internet) ในองค์กรของเรามากกว่า

การเชื่อมต่อฐานข้อมูลเครือข่ายภายในองค์กรนั้น เราสามารถกำหนดการเชื่อมต่อได้คือเมื่อเปิดแอปพลิเคชัน เราก็ค่อยมาเปิดการเชื่อมต่อข้อมูล ซึ่งเรียกว่าเป็น Connection Oriented แต่เมื่อเอาไปใช้กับการทำงานที่เป็นแอปพลิเคชันบนอินเทอร์เน็ต ซึ่งเป็นระบบที่จำนวนผู้ใช้ไม่แน่นอน และสูงกว่าระบบเครือข่ายในบริษัทแน่นอน เพราะฉะนั้นการที่เราเชื่อมต่อกับฐานข้อมูลทั้งไว้จึงไม่เหมาะสมกับการทำงานในเครือข่ายอินเทอร์เน็ตนัก เอดีโอคือทเน็ตจึงมีคุณสมบัติที่สนับสนุน Disconnected Connection ด้วย

นอกจากนั้นยังสามารถใช้กับแหล่งข้อมูลหลายๆแบบเหมือนเดิม เช่น Access กับ OLE DB Provider ของหลายๆแบบ หรือกระทั่งข้อมูลใน Outlook Exchange เราก็สามารถใช้ เอดีโอคือทเน็ตไปเลือกออกมาได้เช่นเดียวกัน

2.5.1 ออบเจกต์ที่สำคัญในเอดีโอคือทเน็ต

ออบเจกต์ที่ใช้อยู่บ่อยๆใน เอดีโอคือทเน็ตซึ่งจะกล่าวถึงในส่วนนี้คือ Connection, Dataset และ Data Adapter

2.5.1.1 ออบเจกต์คอนเนกชัน (Connection Object)

ออบเจกต์ตัวแรกก็คือ Connection เป็นออบเจกต์ที่อยู่ใน เอดีโอคือทเน็ตหน้าที่คือ การเปิดการเชื่อมต่อ (Connection) ระหว่างไคลเอนต์ของเรากับแหล่งข้อมูล โดยที่เราสามารถกำหนดรูปแบบเชื่อมต่อไปยังฐานข้อมูลที่ต้องการ พร้อมค่ารายละเอียดต่างๆ เช่น รหัสผ่านเมื่อจะเชื่อมต่อกับฐานข้อมูล ซึ่งยังคงคล้ายกับ เอดีโอแบบเดิมนั่นเอง

2.5.1.2 ออบเจกต์คอมมานด์ (Command Object)

ออบเจกต์ตัวนี้มีหน้าที่เอาไว้เขียนคำสั่ง เอสคิวแอล (SQL) จากไคลเอนต์ไปทำงานบนแหล่งข้อมูล เช่นเดียวกับออบเจกต์ Connection คือใน เอดีโอปกติก็ยังมีออบเจกต์นี้อยู่

การใช้งานออบเจกต์ Command ในเอดีโอคือทเน็ตก็เหมือนกับการทำงานใน เอดีโอเดิมคือในเวลาที่เราจะส่งคำสั่งเอสคิวแอลที่ไม่ใช่การเลือก (Select) เช่น Insert, Update หรือ Delete ที่ไม่มีการคืน (Return) ค่ากลับมานั้น ซึ่งส่วนใหญ่ในทุกฐานข้อมูลจะมีโครงสร้างภาษาเอสคิวแอลให้เราสามารถสร้างฟังก์ชันฝังไว้ในตัวฐานข้อมูลได้

2.5.1.3 ออบเจกต์ค้ำค่าเซต (Dataset Object)

ออบเจกต์ค้ำค่าเซตนี้มีเฉพาะในเอทีโอคือทเน็ตเท่านั้น ค้ำค่าเซตเปรียบเสมือนฐานข้อมูล ทั้งก่อนเลย คือเมื่อเรา เลือก (Select) ออกมา เราไม่ได้เฉพาะแถวข้อมูลออกมา แต่ได้ข้อมูลอันใหม่ อันหนึ่ง โดยเป็นฐานข้อมูลที่อยู่ในหน่วยความจำของเรา

ภายในค้ำค่าเซตนั้น จะมองว่าเป็นฐานข้อมูลเหมือนกัน ในนั้นมีตาราง (Table), คอลัมน์ (Columns) และแถว (rows) ต่างๆอยู่ภายใน มีความสัมพันธ์ระหว่างตารางได้มีการกำหนดข้อบังคับ ความถูกต้อง (Constrain), ครอบครอง (Index key), คีย์ร่วม (Public key) ได้โดยการเก็บข้อมูลของค้ำค่าเซตจะเก็บโครงสร้าง (schema) ข้อมูลของตัวมันด้วย เราสามารถเลือกข้อมูลออกมาเก็บไว้ในค้ำค่าเซต จากนั้นเราก็สามารถออกจากเซตต่อได้โดยเราก็จะมีค้ำค่าเซตเอาไว้ใช้งานจากนั้นภายในโปรแกรมเราก็สามารถใช้คำสั่ง Insert, Update, Delete ลงไปใน Dataset ได้ จนเสร็จเรียบร้อยแล้ว เราสามารถกลับมาเชื่อมต่อกับฐานข้อมูลอีกทีหนึ่ง แล้วทำการซิงค์ (Synch) ข้อมูลกลับขึ้นไปได้ โดยตัวเอทีโอคือทเน็ตจะคอยจัดการความแตกต่างระหว่างข้อมูลด้วยตัวเอง

การซิงค์ข้อมูลกลับขึ้นไปในลักษณะนี้ คือการเขียนโปรแกรมแบบ Disconnected Programming โดยไม่จำเป็นต้องเชื่อมต่อกับฐานข้อมูลตลอดเวลา เพราะแอปพลิเคชันบน อินเทอร์เน็ตนั้น ถ้าเราเชื่อมต่อกับฐานข้อมูลไว้ตลอดเวลาจะเป็นการสิ้นเปลืองทรัพยากรของ เซิร์ฟเวอร์มาก

2.5.1.4 ออบเจกต์ค้ำค่าอะแด็ปเตอร์ (Data Adapter Object)

ออบเจกต์ ค้ำค่าอะแด็ปเตอร์ มีเฉพาะในเอทีโอคือทเน็ตเช่นกัน ออบเจกต์ค้ำค่าอะแด็ปเตอร์ ก็เหมาะสำหรับการเลือก (Select) ทั่วๆไป ซึ่งเปรียบเสมือนเป็นสะพานเชื่อมระหว่างแหล่งข้อมูลที่มีข้อมูลกับตัวค้ำค่าเซต ของเราโดยลำเลียงข้อมูลจากแหล่งข้อมูลเอามาเก็บในตัวค้ำค่าเซต จากนั้นจึงสามารถใช้คำสั่งเอสคิวแอลในค้ำค่าเซต ทำการเปลี่ยนแปลงข้อมูลในแถวต่างๆได้ และหากต้องการซิงค์ข้อมูลจากค้ำค่าเซต กลับไปในแหล่งข้อมูลใหม่ เราก็ใช้ค้ำค่าอะแด็ปเตอร์ ทำการอัปเดตข้อมูลกลับขึ้นไปในแหล่งข้อมูล

หากเปรียบเทียบภาพของค้ำค่าอะแด็ปเตอร์ ก็เหมือนสะพานเชื่อมโยงระหว่างฐานข้อมูล ของเรากับค้ำค่าเซต นั่นเอง การที่จะเอาข้อมูลมาเก็บไว้ในค้ำค่าเซต เราต้องสร้างออบเจกต์ขึ้นมา 2 ตัว ตัวแรกเป็นค้ำค่าเซต ตัวที่ 2 เป็นค้ำค่าอะแด็ปเตอร์ หลังสร้างเสร็จแล้วให้ใช้คำสั่งเอสคิวแอลเพื่อดึงข้อมูลมา จากนั้นใช้คำสั่ง Fill เพื่อนำข้อมูลที่ได้จากการอ่านเข้ามาใส่ในค้ำค่าเซต

คุณลักษณะ	ADO	ADO.NET
สถาปัตยกรรมที่ใช้	บนพื้นฐานคอม(COM) การเชื่อมโยง Connection Oriented	XML กับ HTTP การเชื่อมโยงแบบ Connectionless
การเข้าถึงระบบป้องกัน	ไม่ผ่านระบบ Firewall	ใช้ HTTP และสามารถผ่าน Firewall ได้เป็นอย่างดี
ออบเจกต์ที่ใช้	Connection, Command, Record set	Connection, Command, Dataset, Dataset view, Data view

ตารางที่ 2.1 เปรียบเทียบการเชื่อมต่อฐานข้อมูลระหว่างเทคโนโลยี ADO และ ADO.NET

2.6 เอสคิวแอล เซิร์ฟเวอร์ 2000

เอสคิวแอลเซิร์ฟเวอร์ 2000 (SQL Sever 2000 : Sequential Query Language Server 2000) จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDMBS : Relation Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมกันมากในปัจจุบัน สามารถจัดการฐานข้อมูลอย่างมีประสิทธิภาพ และสนับสนุนการใช้งานบนระบบปฏิบัติการวินโดวส์ (Window) ได้เป็นอย่างดีโดยเอสคิวแอล เซิร์ฟเวอร์ 2000 (SQL Sever 2000 : Sequential Query Language Server 2000) ถูกออกแบบมาให้ทำงานในลักษณะที่เป็นระบบฐานข้อมูลระหว่างเครื่องเซิร์ฟเวอร์ กับเครื่องลูกข่าย (Client Server Database) ในระบบปฏิบัติการ (Windows NT) และระบบปฏิบัติการวินโดวส์ 2000 (Windows 2000) นอกจากนี้ เอสคิวแอลเซิร์ฟเวอร์ 2000 ยังสามารถใช้งานร่วมกับเว็บดีวีลีออปเม้นท์แพลตฟอร์ม (Web Development Platform) ทั้งหมดไม่ว่าจะเป็นภาษาซี (C) ซีพลัสพลัส (C++) จาวา (Java) เพิร์ล (Perl) ทีซีแอล (Tcl) หรือ เอเอสพี (ASP) ก็ตาม

เอสคิว แอลเซิร์ฟเวอร์ 2000 เป็นระบบจัดการฐานข้อมูลแบบ Client Server Relation Database ทำให้ช่วยเพิ่มประสิทธิภาพในการทำงาน และระบบการจัดการแบบควบคุมจากศูนย์กลาง (Centralize Management) โดยระบบรักษาความปลอดภัย ระบบจัดสรรการทำงาน และใช้ข้อมูลจากหลายๆงานพร้อมกันได้ ซึ่งประกอบด้วย 3 ส่วนหลักคือส่วนของผู้ให้บริการ (Sever) ส่วนของผู้ใช้บริการ (Client) และข่ายการสื่อสารหรือเน็ตเวิร์ค โดในแต่ละส่วนก็จะมีโปรแกรมสำหรับการทำงานตามหน้าที่ของตน

2.6.1 ส่วนของผู้ให้บริการ (Server)

เป็นคอมพิวเตอร์หลักในระบบเครือข่าย (Network) ที่ทำการติดตั้งโปรแกรมการทำงานเอสคิวแอลเซิร์ฟเวอร์ โดยทำหน้าที่จัดเก็บรวบรวม ค้นหา เรียกดู จัดการข้อมูลในรูปของ ฐานข้อมูล

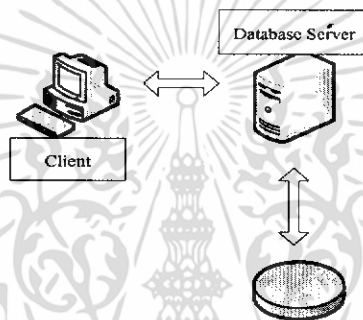
และจัดเก็บข้อมูลที่เกิดจากการที่ผู้ใช้แต่ละคนสร้างขึ้นมา เพื่อที่คอมพิวเตอร์ในเครือข่ายจะมาเรียกไปใช้ได้

2.6.2 ส่วนของผู้ใช้บริการ (Client)

เป็นเครื่องคอมพิวเตอร์ที่ใช้ทำการติดตั้ง เว็บไซต์ลิโอบเมนท์แพลตฟอร์มต่างๆ เช่น จาวาเพิร์ล เอเอสพี เป็นต้น และ เชื่อมต่อกับเอสคิวแอลเซิร์ฟเวอร์ ฟังเซิร์ฟเวอร์ โดยโปรแกรมฝั่งไคลเอนต์ จะทำหน้าที่ส่งต่อข้อมูลจากค้ำเบส และมีโปรแกรมสำหรับเรียกดู และจัดการข้อมูล โดยมีสิทธิ์เรียกดูข้อมูลบนเครื่องเซิร์ฟเวอร์ได้

2.6.3 ข่ายการสื่อสารหรือเน็ตเวิร์ค (Communication Network)

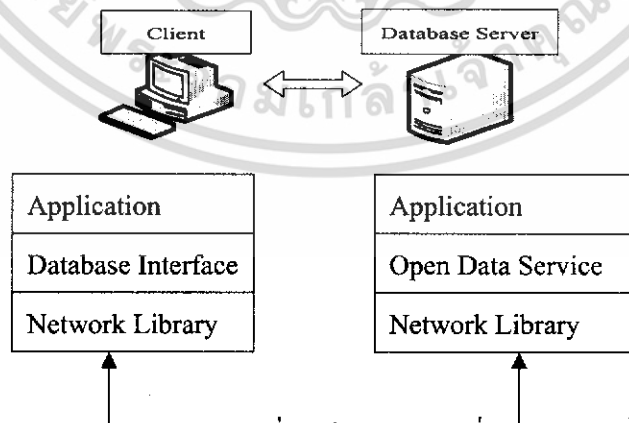
เป็นเส้นทางในการติดต่อแลกเปลี่ยนข้อมูลและคำสั่ง ระหว่างเครื่องเซิร์ฟเวอร์และเครื่องไคลเอนต์ ดังนี้



รูปที่ 2.13 แสดงเส้นทางในการติดต่อแลกเปลี่ยนข้อมูลและคำสั่ง ระหว่างเครื่องเซิร์ฟเวอร์และเครื่องไคลเอนต์

2.6.4 สถาปัตยกรรมของเอสคิวแอล เซิร์ฟเวอร์ (SQL Sever)

สถาปัตยกรรมหรือโครงสร้างภายในของเอสคิวแอลเซิร์ฟเวอร์ จะมีส่วนที่ทำหน้าที่ติดต่อสื่อสารกับทางฝั่งไคลเอนต์อยู่ด้วย แต่ก็แยกส่วนจัดการเน็ตเวิร์คและโปรโตคอลออกจากส่วนที่เป็นแอปพลิเคชัน ทำให้แอปพลิเคชันสามารถทำงานอยู่บนเน็ตเวิร์คแบบใดก็ได้



รูปที่ 2.14 แสดงเส้นทาง ในการติดต่อแลกเปลี่ยนข้อมูลและคำสั่ง ระหว่างเครื่องค้ำเบสเซิร์ฟเวอร์และเครื่องไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.14 แอปพลิเคชัน (Application) หมายถึง โปรแกรมที่พัฒนาขึ้นเพื่อใช้ข้อมูลจากคำดาเบสอินเทอร์เฟซของโปรแกรมที่เรียกว่า เอพีไอ (API : Application Program Interface) เพื่อใช้งานกับดีวีดีออปเม้นท์แพลตฟอร์มต่างๆมากมาย ไม่ว่าจะเป็นภาษาซี ซีพลัสพลัส จาวา ไอเฟล เพิร์ล หรือ พีซีแอล เป็นต้น และนอกจากนี้ยังสามารถใช้งานร่วมกับโอดีบีซี (ODBC : Open Database Connectivity) ซึ่งทำให้เราสามารถใช้งานได้กับเครื่องมืออื่นๆ บนวินโดวส์แพลตฟอร์ม เช่น Access เป็นต้น รวมทั้ง สามารถนำมาประยุกต์เพื่อใช้งานร่วมกับ ASP (Active Server Page)

คำดาเบสอินเทอร์เฟซ (Database Interface) หมายถึง อินเทอร์เฟซที่ใช้โดยแอปพลิเคชันที่ติดต่อไปยังเอสคิวแอลเซิร์ฟเวอร์ เช่น ODBC โดยสร้างการติดต่อระหว่าง แอปพลิเคชัน และ OLE DB (Object Linking and Embedding Database)

เน็ตเวิร์คไลบรารี (Network Library) หมายถึง โปรแกรมที่ทำหน้าที่จัดส่งข้อมูลผ่านทางเน็ตเวิร์ค โดย เอสคิวแอล เซิร์ฟเวอร์ ซึ่งเซิร์ฟเวอร์ มีเน็ตเวิร์คไลบรารี ที่ทำงานร่วมกับโปรโตคอลได้หลายประเภท เช่น ทีซีพีไอพี (TCP/IP) เนมไปป์ (Named Pipe หรือ NT) เป็นต้น

โอเพนคำดาเซอว์วิส (Open Data Service) เป็นอินเทอร์เฟซระหว่างเน็ตเวิร์คไลบรารีกับแอปพลิเคชันฝั่งเซิร์ฟเวอร์

2.7 การออกแบบฐานข้อมูลโดยการใช้ในแอมโมเดล (NIAM Model)

NIAM(Nijssen's Information Analysis Methodology) คิดค้นโดย ศาสตราจารย์ G.M.Nijssen ละทีมงาน เป็นวิธีการในการออกแบบฐานข้อมูลโดยใช้สัญลักษณ์ต่างๆในการแทนการจำลองข้อมูล ความสำคัญและข้อกำหนดต่าง ของข้อมูล เป็นวิธีที่มีอัลกอริทึมที่สามารถออกแบบฐานข้อมูลแบบความสัมพันธ์(Relational)ให้อยู่ในรูปแบบบรรทัดฐานที่ 5 (5 NF) โดยไม่ต้องทำการนอร์มาไลซ์(Normalization) และมีอินเทอร์เฟซที่คนสามารถเข้าใจได้ง่ายเนื่องจากเป็นภาษาธรรมชาติ(Natural Language Interface)

2.7.1 ส่วนประกอบของในแอมโมเดล

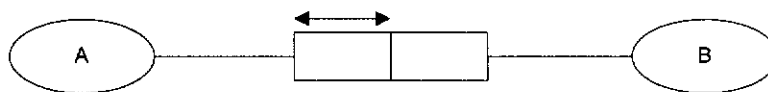
- Entity type คือ เซตของสิ่งที่สนใจซึ่งอาจจะเป็นสิ่งที่จับต้องได้หรือไม่ได้
- Label (Value type) คือเซตของสิ่งของที่ใช้บอกความแตกต่างหรือชื่อของแต่ละ Entity โดย Entity type ทุกตัวจะต้องมีlabel type
- Role ใช้ในการแสดงความสัมพันธ์ที่เกี่ยวข้องกับ Entity type ที่เชื่อมต่ออยู่
- Fact type คือ เซตของความสัมพันธ์ระหว่างสมาชิกของ Entity type ตั้งแต่2 Entity ขึ้นไป
- Reference type คือ เซตของความสัมพันธ์ระหว่างสมาชิกของ Entity type กับ สมาชิกของ Label type

-Nested Fact type เป็น Entity type ชนิดหนึ่งที่แสดงความสัมพันธ์ในการกำหนดกลุ่มของ

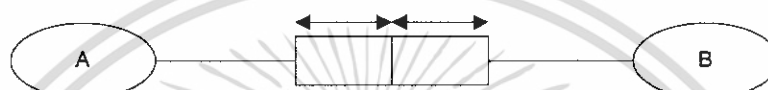
เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 การเขียนความสัมพันธ์ในรูปแบบของโนแอม

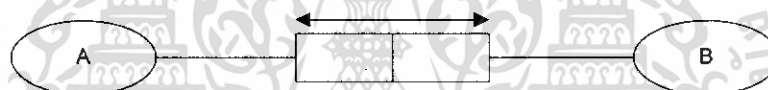
2.7.2.1 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to many relationship 1:M)



2.7.2.2 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to one relationship 1:1)



2.7.2.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to many relationship M:M)



2.8 Web Services

คือ แอปพลิเคชันบนเว็บ นั่นหมายถึงเราสามารถร้องขอแอปพลิเคชันนี้แบบโปรแกรมบนเว็บโดยใช้โปรโตคอล SOAP

XML + HTTP = SOAP

เว็บเซอร์วิสอนุญาตให้แอปพลิเคชันใช้ข้อมูลร่วมกันได้ และสามารถถูกเรียกใช้งานข้ามรูปแบบพื้นฐาน(Platform) และระบบปฏิบัติการได้โดยอ้างเป็นภาษาโปรแกรม โดย .NET เป็นโครงสร้างพื้นฐานของ Microsoft สำหรับ XML เว็บเซอร์วิส ใช้ SOAP เพื่อร้องขอเว็บเซอร์วิส (HTTP-GET,HTTP-POST,SOAP)

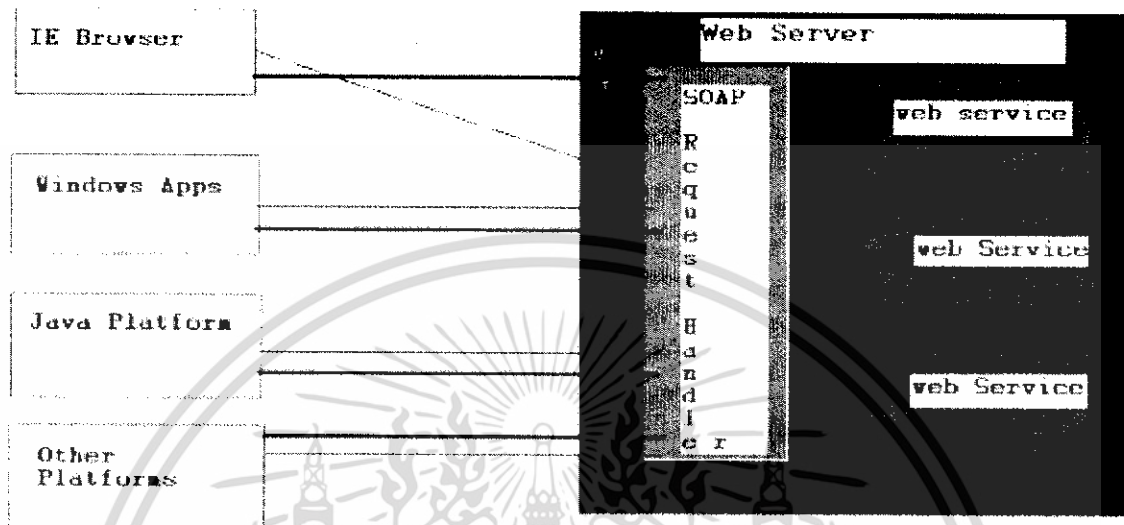
SOAP - Simple Object Access Protocol เป็นโปรโตคอลแบบข้อความพื้นฐาน (Message-Base) ขึ้นกับการร้องขอการตอบสนอง

ข้อได้เปรียบของการใช้เว็บเซอร์วิส อยู่ตรงที่การถ่ายเมื่อใช้ ตัวอย่าง เช่น การผลิตแอปพลิเคชันออกมาเป็นซอร์ฟแวร์แล้วจะใช้ต้องเสียค่าลิขสิทธิ์เพื่อซื้อมาใช้ ซึ่งอาจจะแพงเกินไป แต่ถ้าเป็นเว็บเซอร์วิสแล้วละก็เมื่อใช้ก็ถ่ายจ่ายค่าทำเนียมตามครั้งที่ใช้ก็พอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web Services Architecture

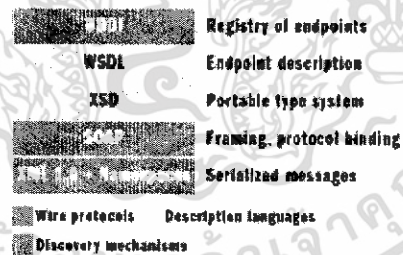
WebService Architecture



SOAP Request
SOAP Response

รูปที่ 2.15 สถาปัตยกรรมเว็บเซอร์วิส

XML Web Service



รูปที่ 2.16 สถาปัตยกรรม XML เว็บเซอร์วิส

คือ โครงสร้างพื้นฐานในการเคลื่อนย้าย เพื่อกระจายข้อมูลบนอินเทอร์เน็ต เป็นส่วนประกอบของโปรแกรมที่อนุญาตให้สร้าง Application ที่ขยายได้, ยืดหยุ่น, ไม่ขึ้นกับ platform ใด เว็บเซอร์วิสมีการให้ใช้ แอปพลิเคชันเพื่อแลกเปลี่ยนข้อความโดยใช้โปรโตคอลพื้นฐาน เช่น HTTP, XML, XSD, SOAP และ WSDL

เว็บบริการ XML สามารถประกอบด้วยการเจาะจงการเขียนโปรแกรมเหตุผล เพื่อ จัดทำคุณสมบัติการใช้งาน เช่น การคำนวณภาษีเงินได้ XML เว็บเซอร์วิสที่คำนวณภาษีเงินได้ต้องการเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอปพลิเคชันทางลูกค้าเพื่อขอข้อมูลของลูกค้า เช่น รายได้, เงินออม และ การลงทุนระหว่างปี แอปพลิเคชันทางลูกค้าสามารถเรียกวิธีการบนบริการเพื่อเรียกข้อมูลที่จำเป็น ในการทำงาน call (Method Call) และข้อมูลที่เกี่ยวข้องกับการทำงานนั้นจะถูกดึงมาที่เว็บเซอร์วิสในรูปแบบ XML โดยใช้ SOAP บน HTTP

XML เว็บเซอร์วิส มีส่วนประกอบ 3 ส่วน

-Directories เป็นส่วนกลางที่ใช้เก็บข้อมูลสาธารณะ เกี่ยวกับ XML เว็บเซอร์วิส UDDI จะระบุเฉพาะเจาะจงเกี่ยวกับแนวทางสำหรับข้อมูลสาธารณะ เกี่ยวกับ XML เซอร์วิส

-Discovery เป็นกระบวนการอนุญาตให้ลูกค้าใช้ลูกค้าได้ทราบเกี่ยวกับรูปปลักษณ์ของ XML เว็บเซอร์วิสและเกี่ยวกับที่ตั้งเฉพาะเจาะจง XML เซอร์วิส

-Description เป็นเอกสาร XML ที่เจาะจงรูปแบบของข้อความที่ XML เว็บเซอร์วิส สามารถเข้าใจได้ เช่นเอกสารบรรยายเกี่ยวกับรูปแบบข้อความ SOAP ที่ใช้ร้องขอวิธีการ บน XML เว็บเซอร์วิส

-Communication Wire Formats การติดต่อแบบใช้สายสามารถเข้าใจได้ทุกระบบที่สนับสนุนการใช้ เว็บธรรมดา เช่น HTTP, SOAP โดย HTTP POST เป็นโปรโตคอล ที่อนุญาตให้ส่งพารามิเตอร์ และ โปรโตคอล HTTP GET อนุญาตให้ส่ง URL-encode พารามิเตอร์ไปยัง XML เว็บเซอร์วิส โปรโตคอล SOAP อนุญาตให้แลกเปลี่ยนโครงสร้าง และรูปแบบข้อมูลระหว่าง แอปพลิเคชัน บนอินเทอร์เน็ต

2.9 หลักการในการทำระดับ(Principles of Leveling)

หลักการเบื้องต้นเกี่ยวกับเครื่องมือคือ เมื่อตั้งกล้องและตั้งระดับของกล้องแล้ว แนวตั้งกล้อง (Collimation line หรือ Line of sight) จะเป็นแนวเดียวกันกับแนวราบ (Horizontal line)

สำหรับไม้วัดระดับ เมื่อตั้งฟองกลมของไม้วัดระดับ (Staff) เราจะถือว่าไม้วัดระดับนั้นอยู่ในแนวตั้งและไม้วัดระดับหน้า-หลัง (FS/BS) จะขนานกันในแนวตั้ง Staff บางที่เรียกว่า Leveling rod

พื้นหลักฐานการระดับ (Datum) จะอ้างอิงจากระดับน้ำทะเลปานกลาง (Mean sea Level = MSI หรือ รทก.) ที่ระดับ รทก. จะถือว่ามีความระดับเท่ากับศูนย์เสมอ ระดับน้ำทะเลปานกลางจะทำการวัดประมาณ 18.6ปี (Mutation period) จึงจะให้ค่าที่ดี ทั้งนี้เพราะขั้วโลกจะหมุนมาอยู่ตำแหน่งเดิมพอดี

ในประเทศไทยสถานีวัดระดับน้ำทะเลจะอยู่ที่เกาะหลัก จังหวัดประจวบคีรีขันธ์ ในแต่ละประเทศก็จะมีจุดระดับเป็นของตัวเอง เพราะฉะนั้นค่าระดับจะเปรียบเทียบกันไม่ได้ ถ้าทำการก่อสร้างสะพานหรืออุโมงค์ข้ามประเทศ จึงต้องให้ค่าระดับมาจากจุดจุดเดียวจึงจะใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{ค่าความต่างระดับ (}\Delta h\text{)} = BS - FS$$

$$\text{ค่าระดับของจุด B} = \text{ค่าระดับที่ A} + BS - FS$$

ความหมายของค่าต่างๆที่ใช้ในการถ่ายระดับ มีดังนี้

1. ความสูงของแกนกล้อง (Collimation line) ความสูงนี้จะนับจากระดับน้ำทะเลปานกลางจนถึงแกนกล้อง ทุกครั้งที่มีการทำระดับ จะคำนวณหาค่าระดับของจุดตั้ง Staff ต่อเนื่องกันไป บางทีเรียกว่า วิธีความสูงแกนกล้อง (Height of Instrument) HI หรือ HPC (Height of Plane of Collimation) เพราะส่องรอบตัว

2. หมุดหลักฐานการระดับ (มฐ.) หรือ Bench Mark (BM) จะเป็นหมุดอ้างอิงค่าระดับจริงที่นับเนื่องจากระดับน้ำทะเลปานกลาง กรมแผนที่ทหารจะทำการถ่ายระดับและสร้างหมุดหลักฐานกระจายไปทั่วประเทศ

3. หมุดถ่ายระดับ (Turning Point (TP)) หรือหมุดเปลี่ยนระดับ หรือหมุดตั้ง Staff (Change Point (CP)) หมายถึง จุดตั้ง Staff เพื่อทำการถ่ายระดับ ซึ่งจะต้องวาง TP หรือ Ground Plate เพื่อตั้ง Staff หัวหมุดจะต้องเป็นทรงกลมเสมอ เพราะจะต้องมีจุดสูงสุดเพียงจุดเดียว เราจะใช้ถ่ายระดับต่อเนื่องกันไป โดยการส่อง Back Sight (BS) หรือ Staff หลังและส่อง Fore Sight (FS) หรือ Staff หน้า

4. ค่า Back Sight (BS) หรือไม้หลังหรือ Staff หลัง หมายถึงค่า Staff ที่อ่านได้ครั้งแรกหลังจากตั้งกล้องตั้งระดับแล้วจึงส่องอ่านค่าดังกล่าว

5. ค่า Fore Sight (FS) หรือไม้หน้า หมายถึง ค่า Staff ที่อ่านได้ครั้งสุดท้ายก่อนใช้ถ่ายระดับในจุดนั้น

6. ค่า Intermediate Fore Sight (IFS) หรือค่าไม้กลาง บางทีเรียกว่า Intermediate Sight (IS) หมายถึงค่า Staff ที่อ่านได้หลังจากการส่อง Back Sight (BS) แล้ว จะส่องก็ค่าก็ได้ ใช้ในการทำระดับเพื่อหาระดับดินเดิมของพื้นดิน (NGL = Natural Ground Level) ซึ่งเราเรียกว่า การทำระดับพื้นที่ (Area leveling) และการทำระดับตามแนวศูนย์กลาง (Profile leveling) เพื่อหาระดับดินเดิม

7. ตอนการระดับ (Section) หมายถึงการถ่ายระดับระหว่างหมุดระดับ (Bench Mark) สองหมุดปกติความยาวจะประมาณ 2 ถึง 3 กิโลเมตร

8. ช่วงการระดับ (Subsection) ถ้าตอนการระดับยาว 2 กิโลเมตร เราอาจจะแบ่งย่อยลงอีกเป็นระยะ 1 กิโลเมตรเพราะถ้าถ่ายระดับไป 2 กิโลเมตร และกลับอีก 2 กิโลเมตร ถ้าทำผิดจะต้องถ่ายระดับใหม่อีก 2 กิโลเมตร ถ้าแบ่งย่อย 1 กิโลเมตร จะทำให้ทราบว่าช่วงไหนทำผิด สามารถแก้ไขโดยถ่ายเพียง 1 กิโลเมตร

9. สายการระดับ (Leveling line) คือ ระยะทางรวมของการถ่ายระดับแต่ละตอนต่อเนื่องกันไป เช่น สมมติถ่ายระดับจากกรุงเทพมหานคร ไปสระบุรี เราจะเรียกว่าสายการระดับ ถ้าถ่ายไป-กลับทีละสองกิโลเมตร เราเรียกว่าตอนการระดับ ดังกล่าวแล้ว

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. หมุดหลักฐานการระดับชั่วคราว (Temporary Bench Mark = TBM) เป็นหมุด BM (มฐ.) ชั่วคราวจะทำโดยใช้ตะปูตอกบนรากต้นไม้ และจะกำหนดหมุดขึ้นเมื่อสิ้นสุดการทำงานแต่ละวัน

หลักการปฏิบัติของกล้องระบบ Mechanics

1. จะต้องตั้งกล้องอยู่กึ่งกลางระหว่าง Staff ทาง BS และ FS ฉะนั้นการตั้งกล้องผู้ส่องกล้องจะต้องประมาณระยะให้ดี ให้จำเป็นหลักการง่ายๆว่า ระยะหน้าหลังต้องเท่ากัน
2. การตั้งกล้องอยู่กึ่งกลางจะกำจัดความผิดที่เกิดจากความโค้งของโลก การหักเหของแสง และความผิดที่เกิดจากสายใยเอียง (Collimation error)
3. การตั้งหรือตอก TP จะต้องให้แน่น เพื่อป้องกันไม่ให้เกิดการทรุดตัว จะทำให้ค่า Staff ผิด
4. ปรับภาพ Staff ภาพของสายใยให้ชัดเท่ากัน โดยการเลื่อยตาขึ้นลงถ้าภาพของสายใยและภาพของ Staff เคลื่อนที่ไปด้วยกัน แสดงว่าชัดเท่ากัน ถ้าเคลื่อนไปคนละทาง แสดงว่ายังไม่ชัดเท่ากัน
5. กางร่มให้กล้องทุกครั้งและตลอดเวลาที่ส่องอ่าน
6. จดค่า Staff ให้ถูกต้อง คนจกจะต้องอ่านทวนให้คนกล้องฟังเสมอ
7. ถ้า Ground plate มี 2 ปุ่มควรจะต้องตั้ง staff 2 ครั้ง จะแทนการถ่ายไป-กลับได้เลย จะได้ไม่เสียเวลาถ่ายระดับกลับ

กฎการจดสมุดสนาม(Field book rule)

1. จะต้องไม่ลืมวิธีการจด เช่น ค่า BS FS ใ้ให้ถูกช่อง
2. บันทึก วันที่ เวลา และอากาศ
3. เขียนคำอธิบายเพื่อความเข้าใจ
4. ใช้สัญลักษณ์ทั่วไปที่คนอื่นเข้าใจได้
5. จะต้องเขียนรูปมาด้วย เช่น ตำแหน่ง BM, ตำแหน่งของสิ่งก่อสร้างที่อยู่ใกล้
6. อย่าเขียนค่าต่างๆ ปนกัน หรือมากจนล้นหน้ากระดาษ
7. ห้ามลบขูดตัวเลขที่ผิด ข้อมูลที่บันทึกผิดให้ขีดฆ่าไว้
8. บันทึกทุกสิ่งที่เกี่ยวข้อง เช่น คนส่องกล้อง คนถือ Staff คนทาสี
9. ใช้ตัวหนังสือมาตรฐาน เช่น ถ้าภาษาอังกฤษควรจะเป็นตัวพิมพ์ใหญ่ ควรใช้คนที่มียามมือสวยงาม
10. ถ้านำค่าไปเขียนแบบ เช่น Profile ควรจะใช้อักษร โตตามมาตรฐานของกรม กอง นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ Cross section คือการทำระดับของดินเดิม (Existing ground) ไปในแนวตั้งฉากกับเส้นฐานของแนวสำรวจ (Base line) ค่าระดับของดินเดิมที่ได้มานำมาเขียนลงในกระดาษเขียนแบบ จุดประสงค์ก็เพื่อที่จะหาจำนวนปริมาณงานดินเป็นประการสำคัญ และเพื่อให้เป็นแนวทางพิจารณาการออกแบบทางด้านกระบวนน้ำบ้างในบางแห่ง และกำหนดแนวทาง และอื่นๆ

เมื่อทราบจุดประสงค์ที่ต้องทำ Cross section แล้วช่างสำรวจก็ต้องทำ Cross section ให้ได้ค่าที่จะนำไปหาปริมาณงานดิน ที่ใกล้เคียงความจริงมากที่สุด การเก็บค่าระดับดินเดิมในบางจุดบน Station ที่จะมีผลทำให้ได้ระดับดินเดิม (NGL) เมื่อนำไปคำนวณหาปริมาณงานดินแล้วจะทำให้ผิดจากความจริงไปมาก ก็ให้เสี่ยงไปหาระดับของจุดที่ใกล้เคียง ซึ่งเมื่อพิจารณาแล้วเห็นว่าได้ค่าระดับดินเดิมนั้นถูกต้อง เมื่อนำไปหาปริมาณงานดินแล้วจะได้ค่าถูกต้องกว่า

การทำ Cross section ด้วยกล้องระดับ

วิธีปฏิบัติในสนาม

หมุดระดับที่จะต้องใช้ในการทำ Cross Section ก็คือ BM หรือ TBM หรือ Station ที่ได้ทำการส่อง Profile ดังนั้นจะสามารถนำมาใช้ได้ เพราะทุกจุดทราบค่า Elevation บน HUB แล้ว การปฏิบัติทำได้ดังนี้

1. คนกล้องตั้งกล้องส่อง BS ไปยัง BM หรือ HUB หรือ TBM
2. หมุนกล้องส่องค่า IFS บน Staff ที่ตั้งบน Cross Line หรือ Spur Line ในขณะเดียวกัน Chain Man ก็วัดระยะจาก Base Line ไปยังจุดตั้ง Staff นั้น
3. ข้อมูลต่างๆผู้จุดบนที่ก็จะต้องจด และเขียนภาพ Sketch ต่างๆ อย่างละเอียด
4. การจับฉากถ้าระยะใกล้จะใช้ Optical square ถ้าไกลมากจะใช้กล้อง Theodolite วางแนว

จุดที่ต้องทำ Cross Section

1. ทุก Station 25 เมตร (Sta. เดียวกับ Profile)
2. ทุกจุดที่มีท่อ
3. ทุกจุดที่มีทางน้ำ
4. คอสะพานทั้งสองข้าง
5. ทุกจุดของทางแยก ที่ไม่ได้วาง Spur Line

ขอบเขตของการเก็บ Cross Section

การเก็บระดับจะต้องเก็บกว้างข้างละเท่ากับ ROW หรือมากกว่าถ้าจำเป็น ในกรณี Spur Line ที่เป็นถนนอาจเก็บออกไปไม่เท่ากับ ROW จาก Center line เช่น ประมาณ 15-20 เมตร ดูข้อกำหนดทางหลวง ในกรณี Spur Line ที่เป็นคลองให้เก็บคลุมขอบตลิ่งไปข้างละไม่น้อยกว่า 10 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดที่ต้องเก็บระดับในแต่ละ Cross Section

1. ในกรณีที่มีถนนเดิมอยู่ให้เก็บที่ Base line กลางถนน ไหล่ทาง ขอบร่องน้ำข้างถนน กันร่องน้ำข้างถนน และบนพื้นดินทุกจุดที่เปลี่ยนความลาด (Slope)
2. ในกรณีที่ไม่มีคันทาง เก็บ Base line และตามจุดที่ดินเปลี่ยนความลาด
3. ในกรณีที่ไม่มีคันทางและพื้นดินราบเสมอกัน สังเกตการณ์เปลี่ยนความลาดของพื้นดินได้ยาก เช่นทุ่งนา หรือ ป่า ที่พื้นดินเสมอกัน ก็ให้เก็บที่ Base line และทุก 5, 10, 20 เมตรและที่ ROW
4. หลีกเลี่ยงการเก็บระดับบนพื้นที่ที่เปลี่ยนความลาด ซึ่งจะทำให้ทำงานดินผิดไปมาก เช่น จอมปลวก เนินดิน ที่มีความยาวไปตาม Station ไม่ถึง 10 เมตร และคันนา หลุมหรือบ่อที่มีความยาวไปตาม Station ไม่ถึง 10 เมตร
5. ใน Cross Section บนท่อ นอกจากเก็บที่ Base line กลางถนนและไหล่ทางแล้ว ต้องตั้ง Staff เก็บที่ปลายปากท่อ (หลังท่อ) และที่ Inlet และ Outlet และบนพื้นดินที่เปลี่ยนความลาดจนถึง ROW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบโครงการ

3.1 การออกแบบระบบ

การออกแบบระบบการตรวจสอบความเสียหายจากการก่อสร้างอุโมงค์ใต้ดินแบบ Real – Time จะมีส่วนหลักๆคือ

3.1.1 ส่วนสำหรับพนักงานภาคสนาม โดยจะมีรายละเอียดการทำงานดังนี้

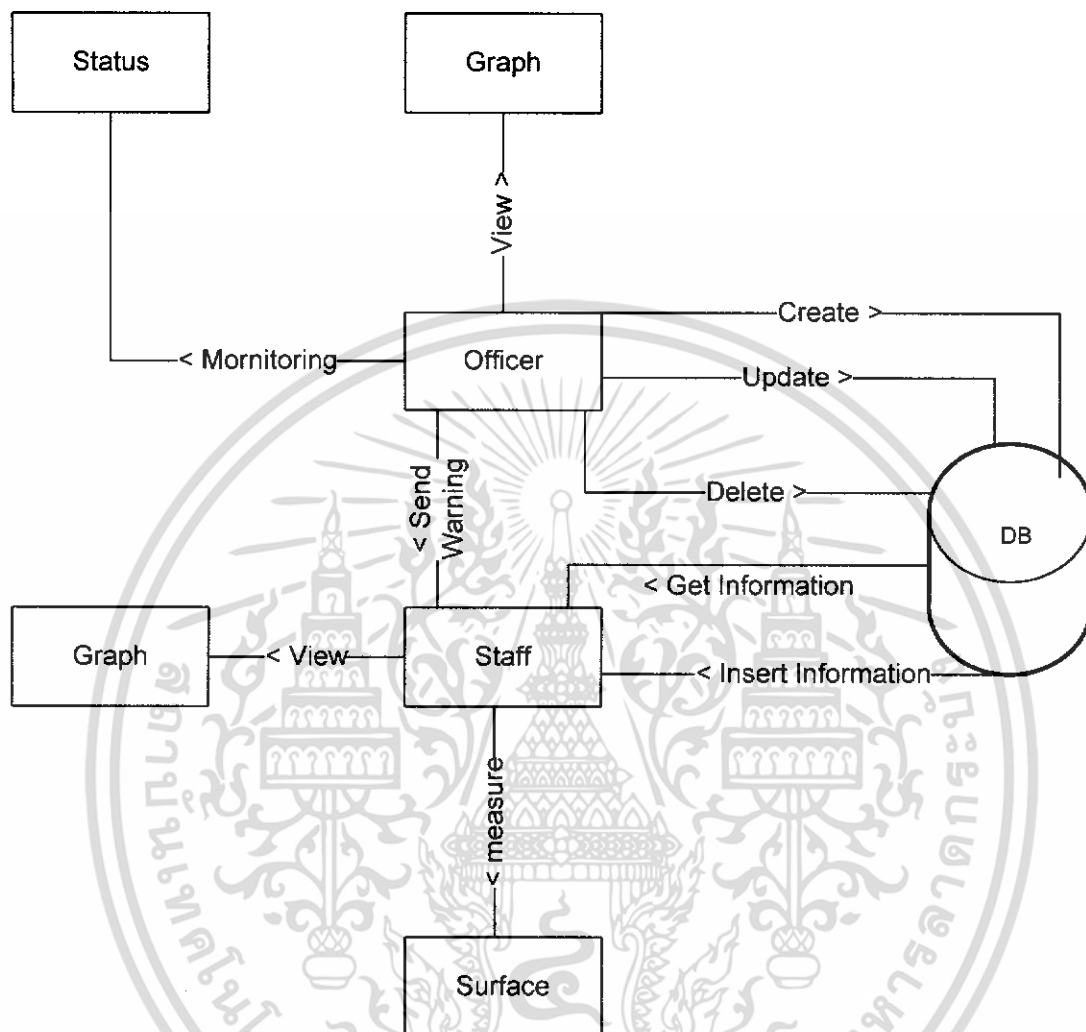
- สามารถเลือก site และ section จากหน้าจอของ pocket pc phone
- สามารถกรอกรายละเอียดเกี่ยวกับการวัดผ่านหน้าจอ pocket pc phone
- สามารถประมวลผลการทรุดตัวของดินผ่านหน้าจอ pocket pc phone
- สามารถวาดกราฟการทรุดตัวของดินผ่านหน้าจอ pocket pc phone

3.1.2 ส่วนสำหรับพนักงานศูนย์ปฏิบัติการ โดยจะมีรายละเอียดการทำงานดังนี้

- สามารถสร้าง, เลือก และลบข้อมูลของ site, section และ Instrument จากหน้าจอของ PC server
- สามารถกรอกรายละเอียดเกี่ยวกับ site, section และ Instrument จากหน้าจอของ PC server
- สามารถประมวลผลการทรุดตัวของดินผ่านหน้าจอจากหน้าจอของ PC server
- สามารถสร้างรายงานการทรุดตัวของดินผ่านหน้าจอ PC server
- สามารถวาดกราฟเปรียบเทียบการทรุดตัวของดินจากการวัดหลายครั้งในผ่านหน้าจอ PC server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 โครงสร้างทั่วไปของระบบ

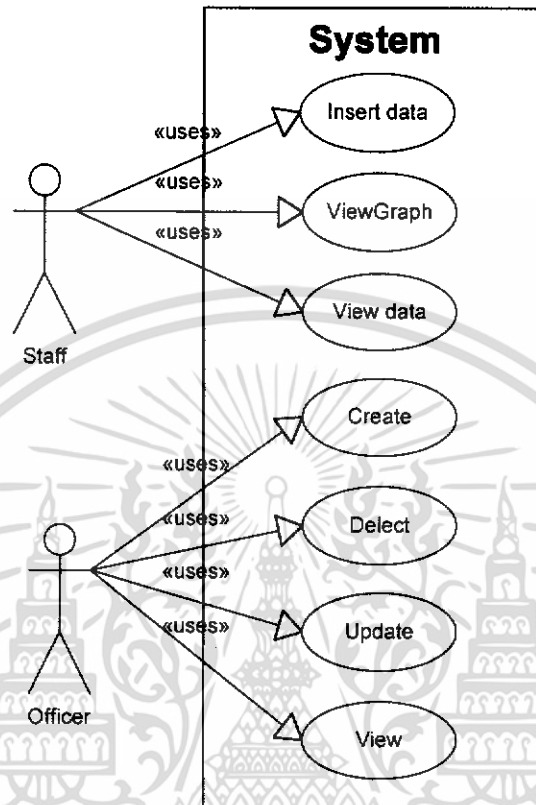


รูปที่ 3.1 โดเมน โมเดล (Domain Model)

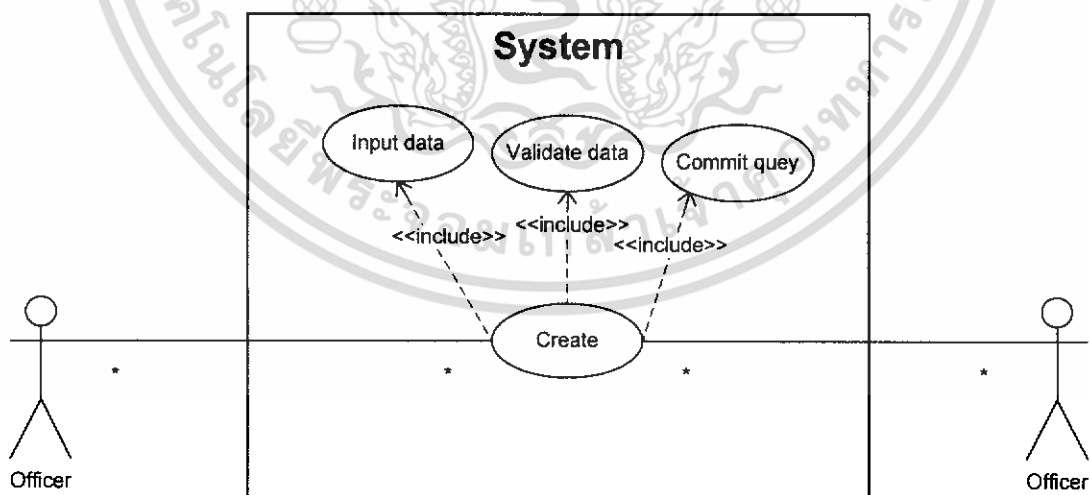
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบโครงการงาน

3.2.1 ยูสเคสไดอะแกรม (Use case diagram)

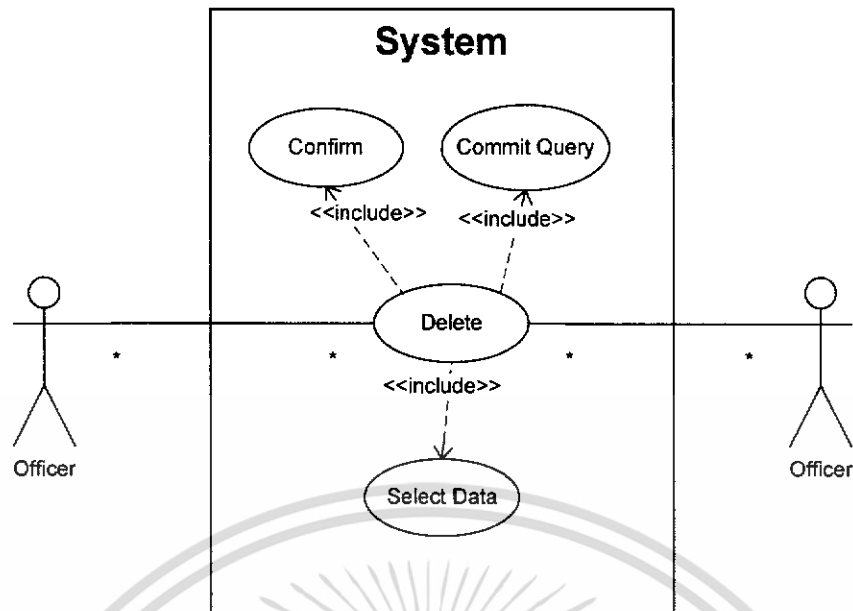


รูปที่ 3.2 ยูสเคสไดอะแกรมสำหรับการใช้ข้อมูลของ Staff (Survey Man) และ พนักงานประจำศูนย์ปฏิบัติการ

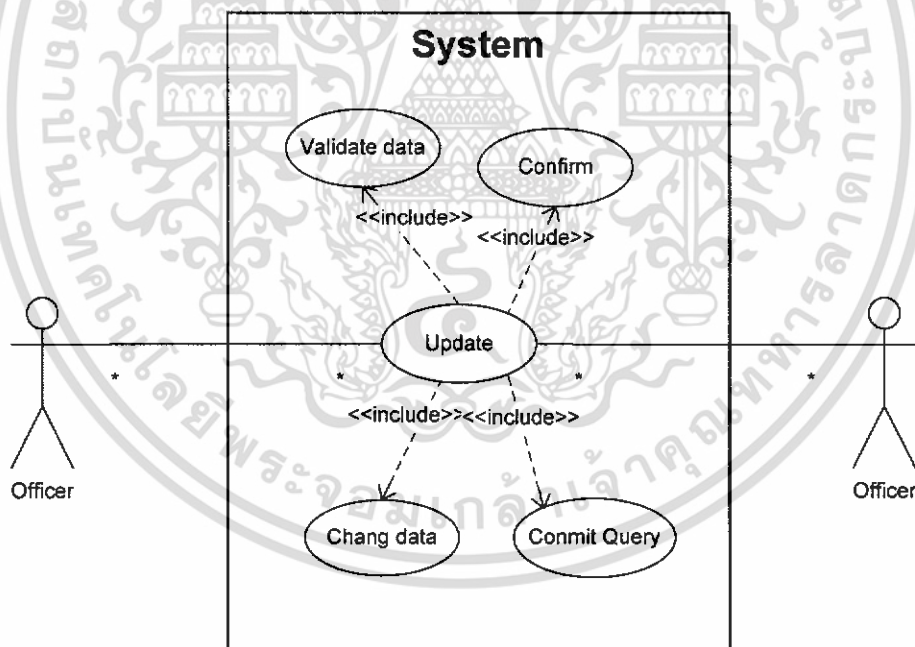


รูปที่ 3.3 ยูสเคสไดอะแกรมสำหรับการสร้าง site station ของ พนักงานประจำศูนย์ปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

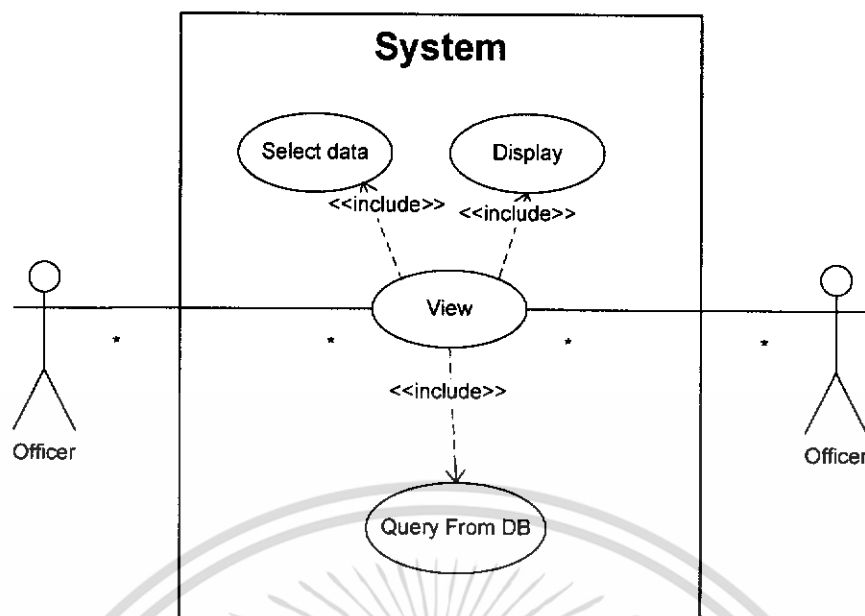


รูปที่ 3.4 ยูสเคสไดอะแกรมสำหรับการลบค่าภายในของ Site station ของ พนักงานประจำศูนย์ปฏิบัติการ

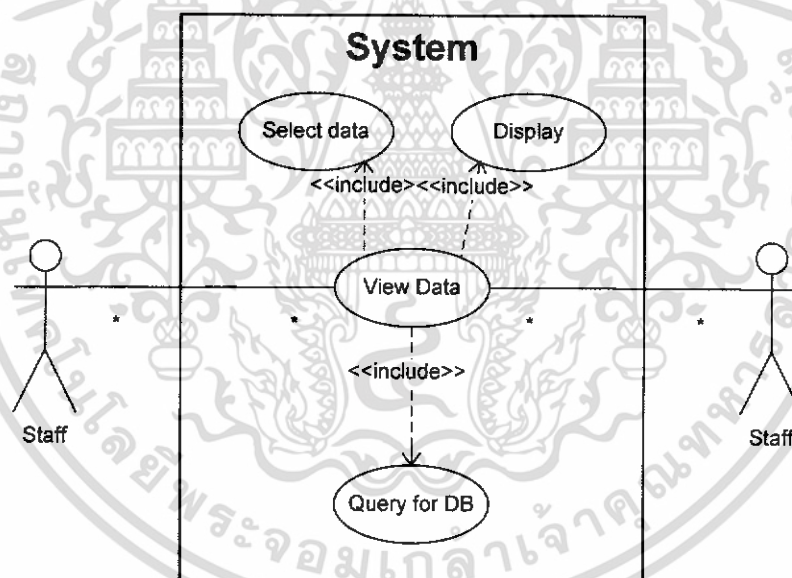


รูปที่ 3.5 ยูสเคสไดอะแกรมสำหรับการUpdate ค่าภายในของ Site station ของ พนักงานประจำศูนย์ปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

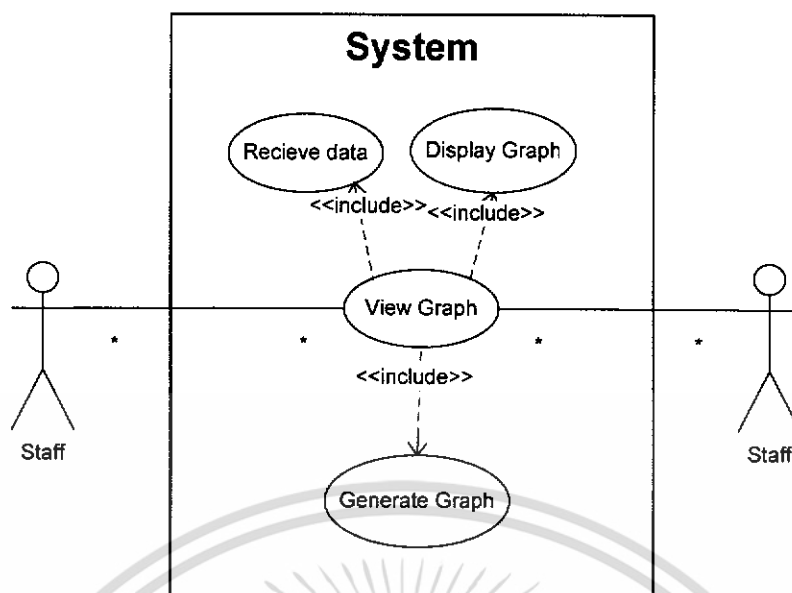


รูปที่ 3.6 ยูสเคสไดอะแกรมสำหรับการเรียกดูค่า Site station ของ พนักงานประจำศูนย์ปฏิบัติการ

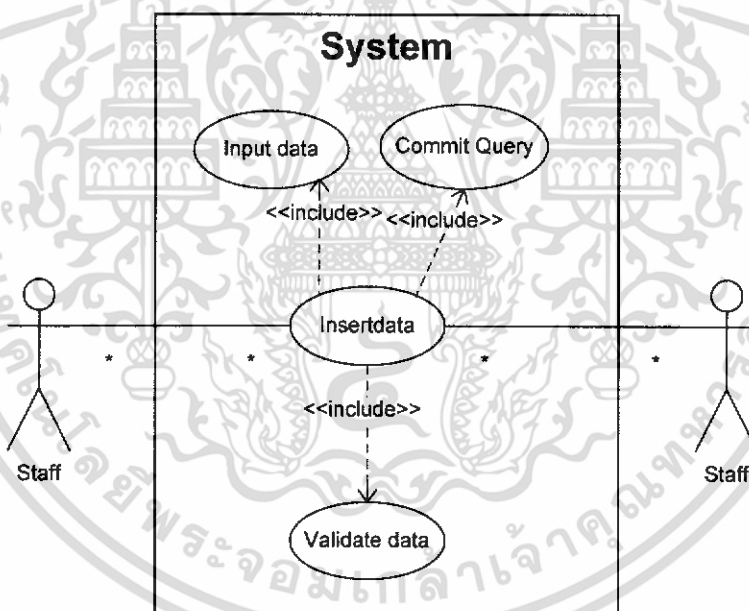


รูปที่ 3.7 ยูสเคสไดอะแกรมสำหรับการเรียกดูค่าภายในของ Site station ของ staff (Survey Man)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



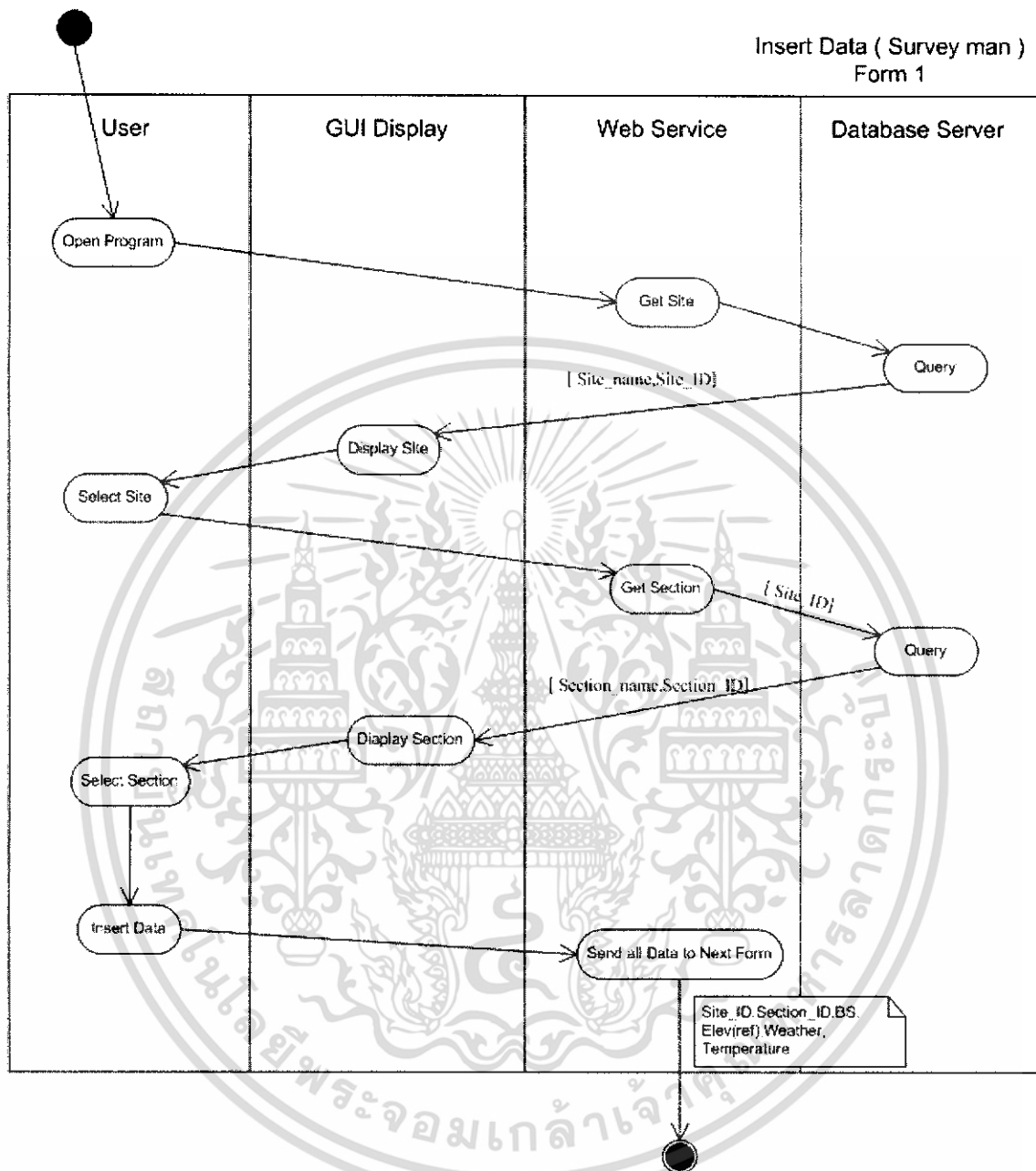
รูปที่ 3.8 ยูสเคสไดอะแกรมสำหรับการเรียกดูกราฟข้อมูล Settlement ของ staff (Survey Man)



รูปที่ 3.9 ยูสเคสไดอะแกรมสำหรับการใส่ข้อมูลของ Staff (Survey Man)

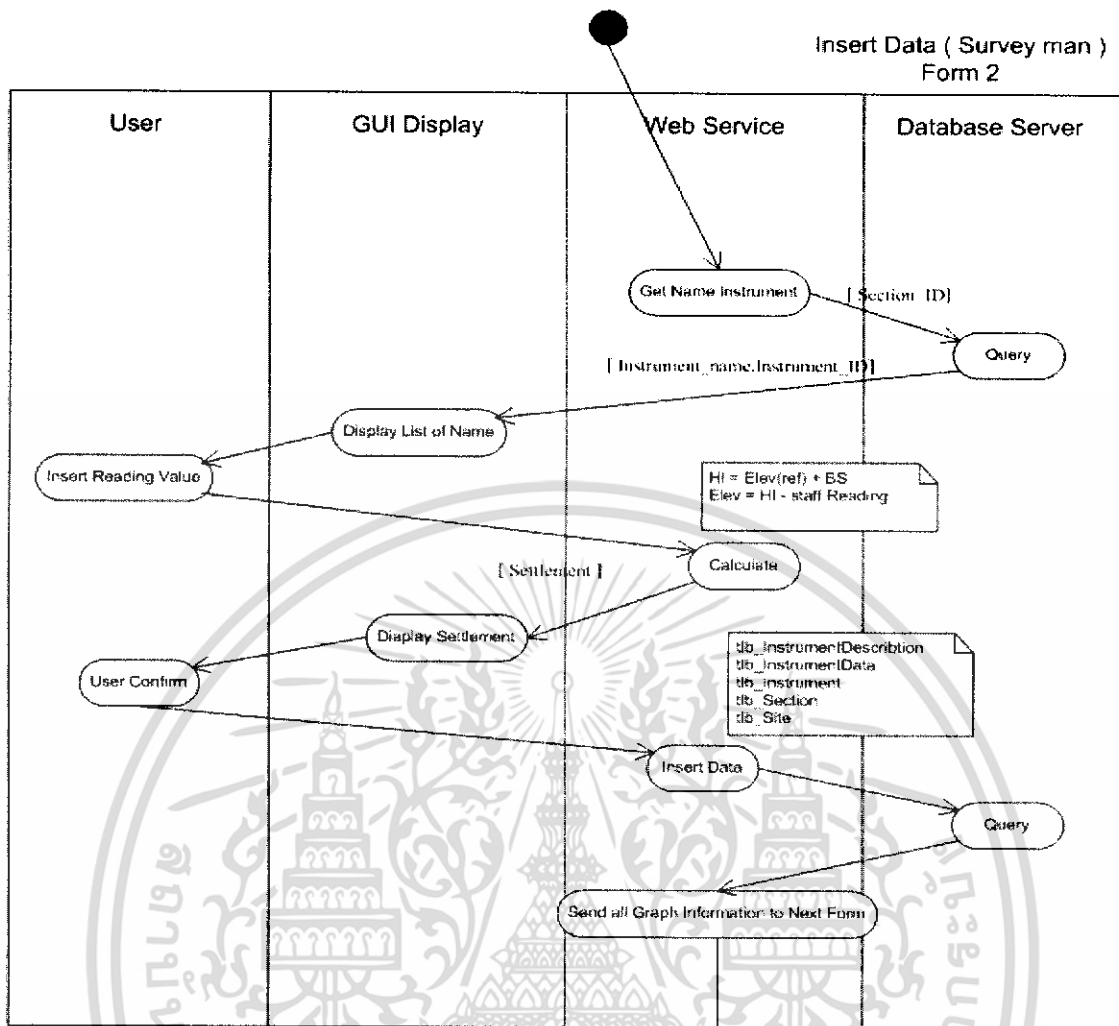
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 แอกทิวตี้ไดอะแกรม (Activity diagram)

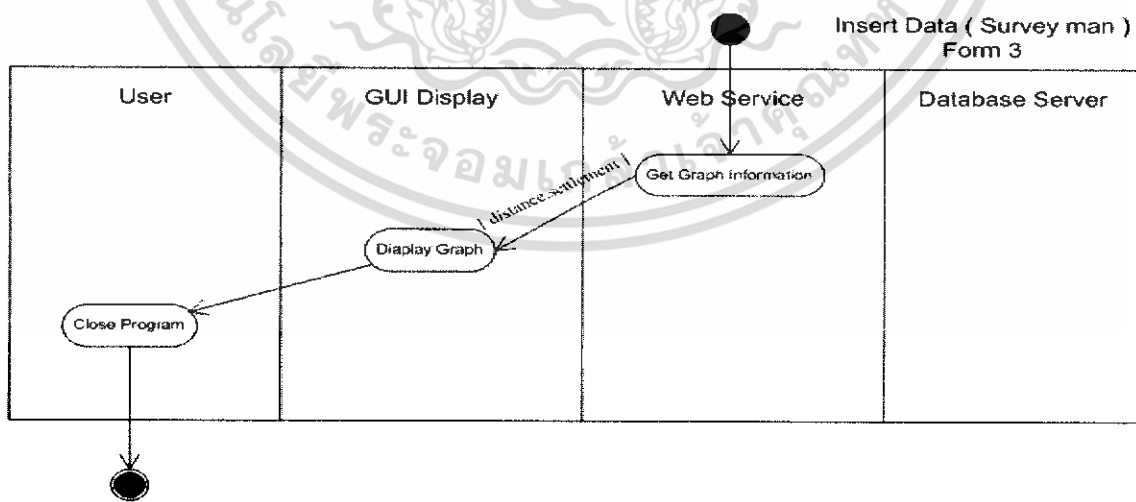


รูปที่ 3.10 แอกทิวตี้ไดอะแกรมสำหรับการบันทึกข้อมูลของ Staff (Survey Man)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

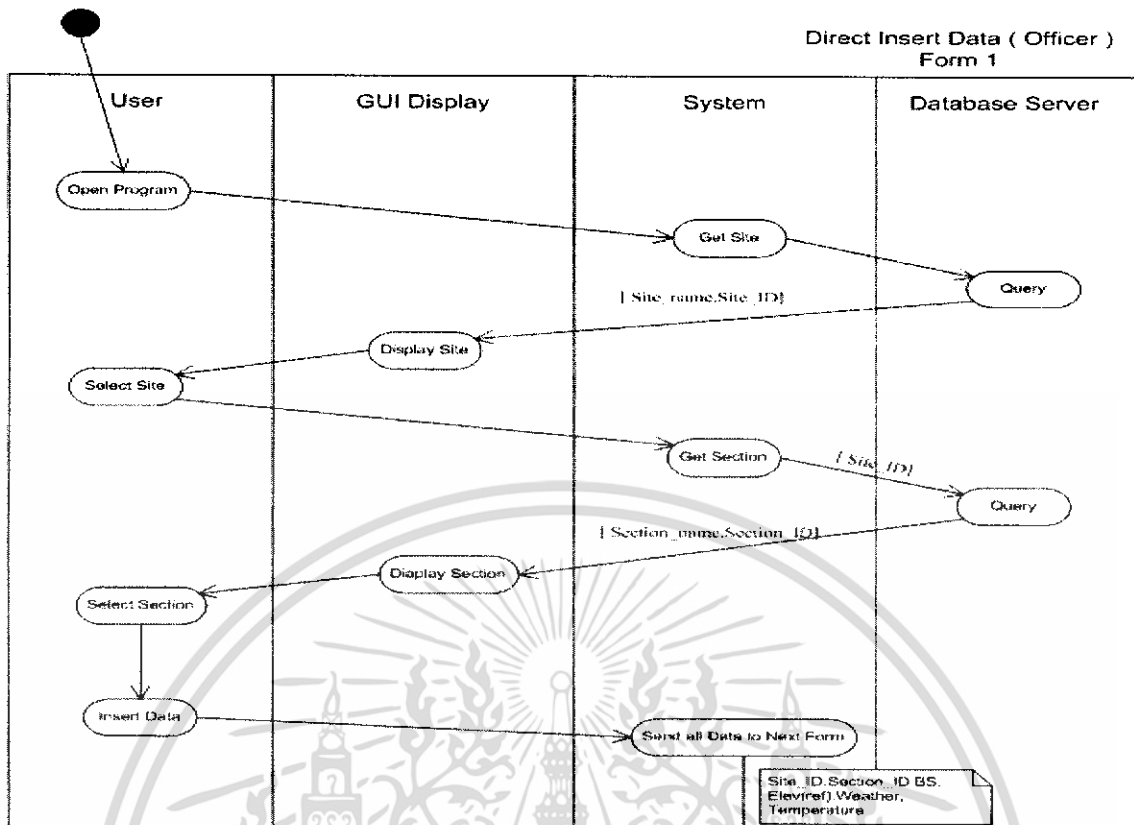


รูปที่ 3.11 แอคทิวิตี้ไดอะแกรมสำหรับการบันทึกข้อมูลของ Staff (Survey Man) (Form2)

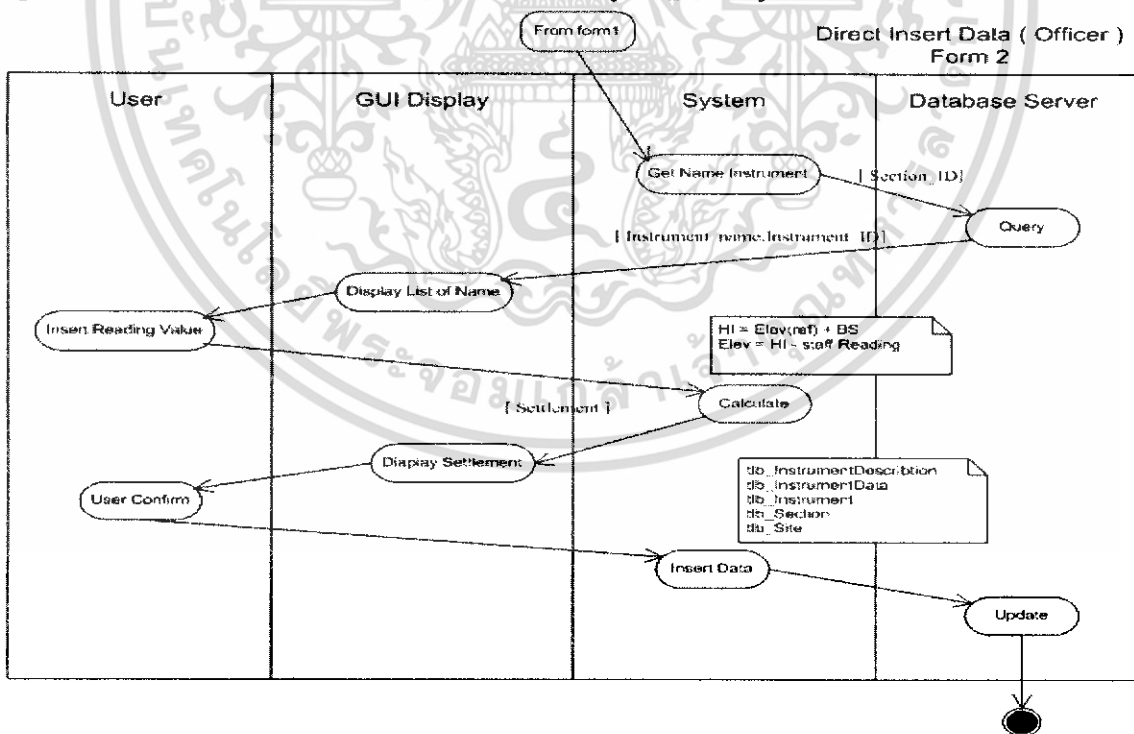


รูปที่ 3.12 แอคทิวิตี้ไดอะแกรมสำหรับการบันทึกข้อมูลของ staff (Survey Man) (Form3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

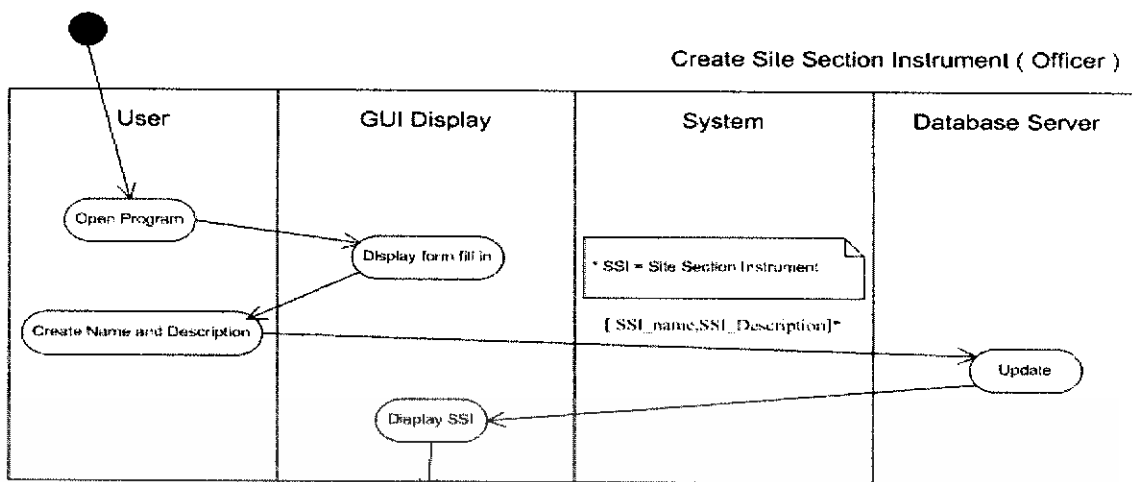


รูปที่ 3.13 แอคทีวิตี้ไดอะแกรมสำหรับการบันทึกข้อมูลลงฐานข้อมูลโดยตรงของ Officer (Form1)

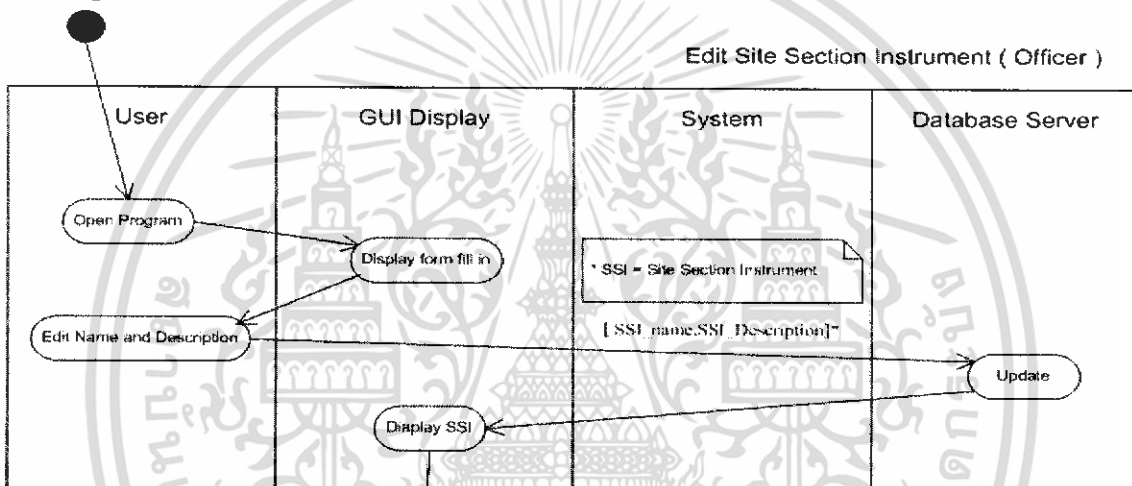


รูปที่ 3.14 แอคทีวิตี้ไดอะแกรมสำหรับการบันทึกข้อมูลลงฐานข้อมูลโดยตรงของ Officer (Form2)

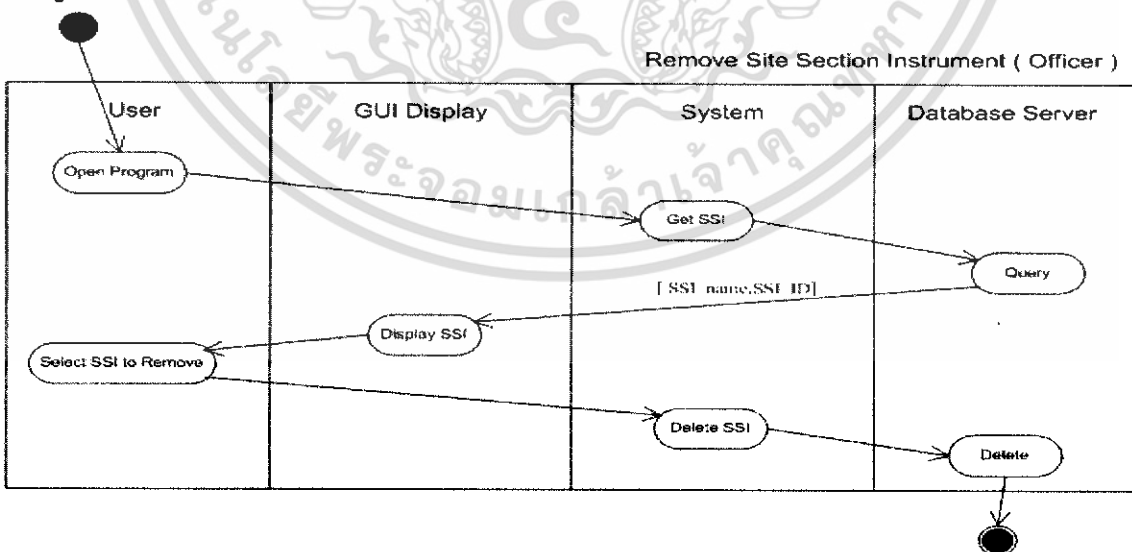
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แอคทีวิตี้ไดอะแกรมสำหรับการสร้าง Site Section Instrument ของ Officer

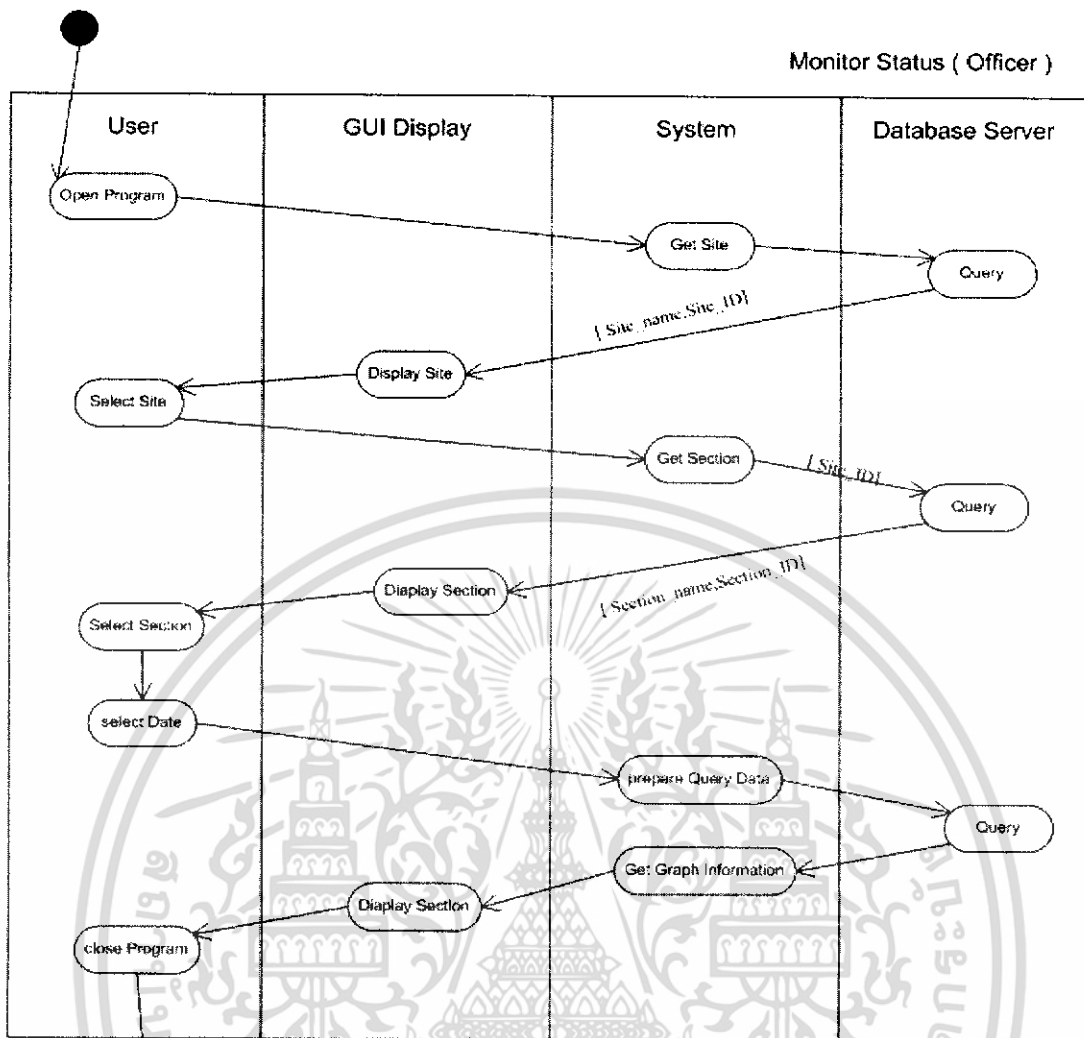


รูปที่ 3.16 แอคทีวิตี้ไดอะแกรมสำหรับการปรับเปลี่ยน Site Section Instrument ของ Officer

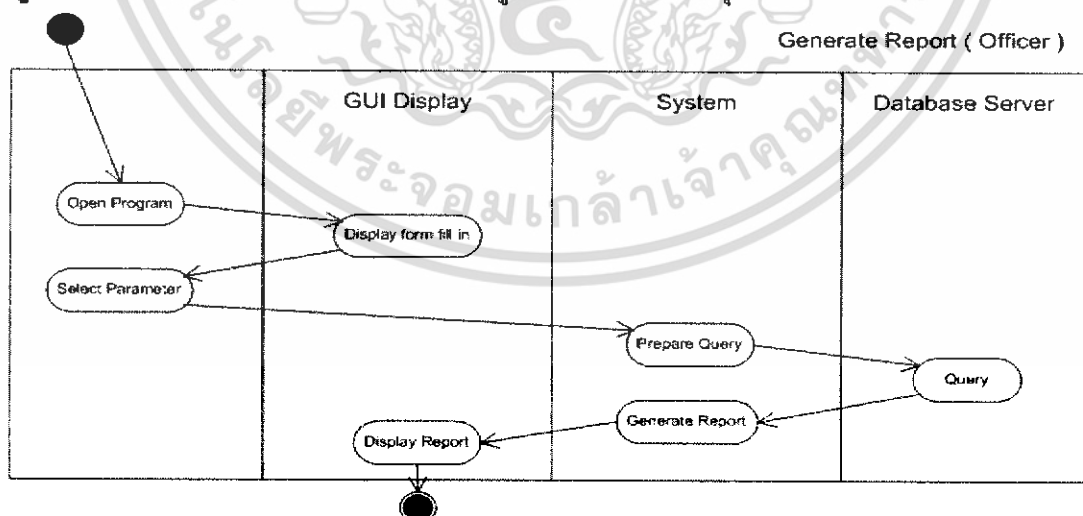


รูปที่ 3.17 แอคทีวิตี้ไดอะแกรมสำหรับการลบ Site Section Instrument ของ Officer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



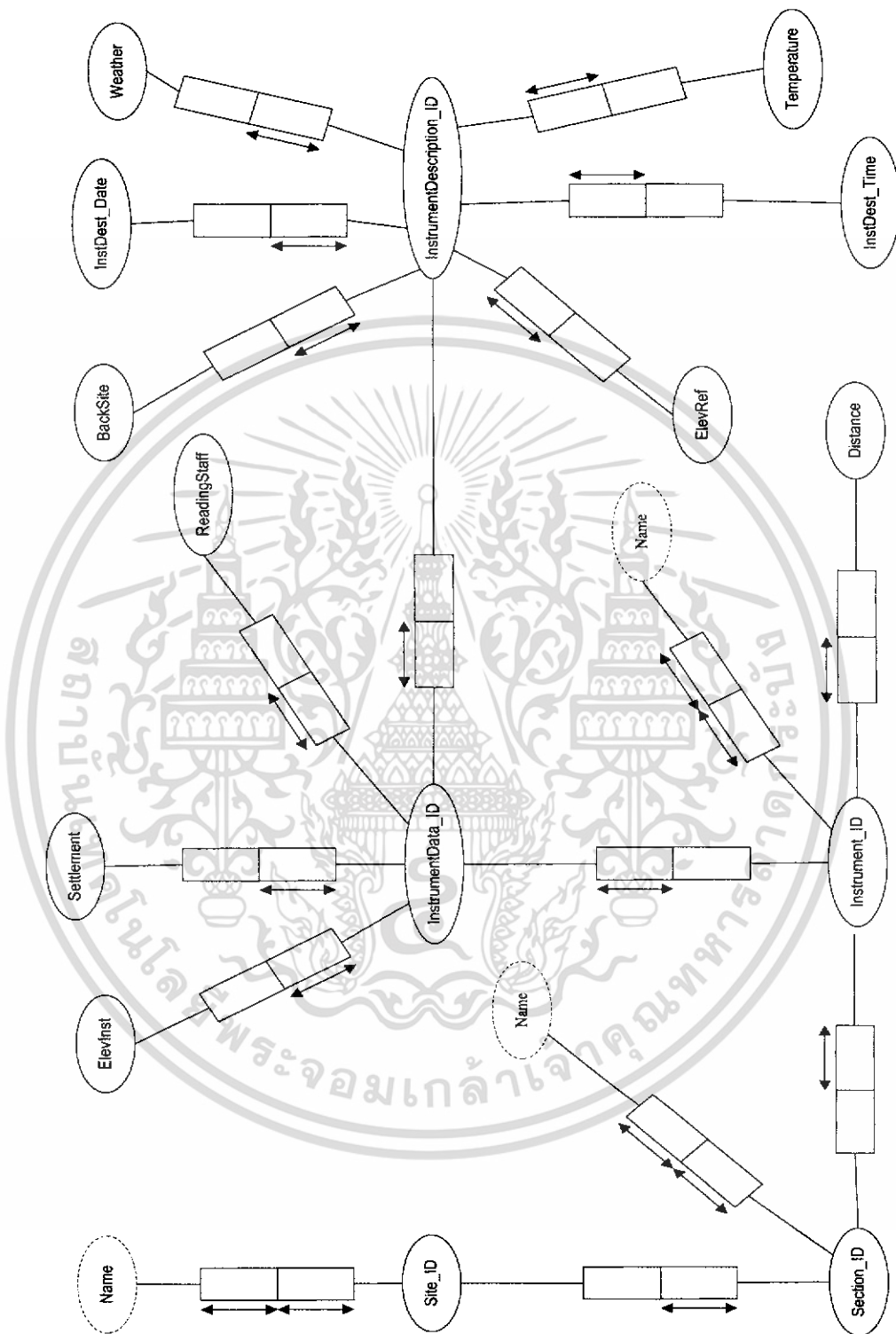
รูปที่ 3.18 แอคทิวิตี้ไดอะแกรมสำหรับการดูภาพรวมของการทุจริตเพื่อประเมินผลของ Officer



รูปที่ 3.19 แอคทิวิตี้ไดอะแกรมสำหรับการสร้างรายงานประจำการตรวจสอบของ Officer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การออกแบบฐานข้อมูล (Database Design)



รูปที่ 3.20 ไนแอม โมเดล (Niam Model) ของระบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 คาด้าคิกชันนารี

ตารางที่ 3.1 ตารางข้อมูลเครื่องมือ(Instrument Data)

Attribute	Data type	Null	Key	Meaning
InstrumentData_ID	Numeric[9]	ไม่	PK	รหัสของข้อมูลจุดวัด
ReadingStaff	Float[8]	ไม่		ค่าที่ staff อ่านมาจากจุดที่วัด
ElevInst	Float[8]	ไม่		ค่า ELE
Settlement	Float[8]	ไม่		ค่าทรุดตัวของดิน
InstrumentDescription_ID	Varchar[14]	ไม่	FK	รหัสที่อ้างอิงถึงรายละเอียดการวัดในครั้งนั้นๆ
Instrument_ID	Numeric[9]	ไม่	FK	รหัสอ้างอิงถึงที่จุดวัด

ตารางที่ 3.2 ตารางรายละเอียดเครื่องมือ (InstrumentDescription)

Attribute	Data type	Null	Key	Meaning
InstrumentDescription_ID	Varchar[14]	ไม่	PK	รหัสรายละเอียดการวัด
InstDest_Date	Varchar[10]	ไม่		วันที่
InstDest_Time	Varchar[8]	ไม่		เวลา
BackSite	Float[8]	ไม่		ค่า BS
ElevRef	Float[8]	ไม่		ค่า ELE อ้างอิง
Temperature	Float[8]			ค่าอุณหภูมิ
Weather	Nvarchar[50]			สภาพภูมิอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ตารางเครื่องมือ (Instrument)

Attribute	Data type	Null	Key	Meaning
Instrument_ID	Numeric[9]	ไม่	PK	รหัสประจำจุดวัด
Name	Nvarchar[50]	ไม่		ชื่อจุดวัด
Section_ID	Numeric[9]	ไม่	FK	รหัสอ้างอิงถึง section ที่วัด
Distance	Float[8]	ไม่		ระยะห่างระหว่างจุดวัด

ตารางที่ 3.4 ตารางส่วนย่อย (Section)

Attribute	Data type	Null	Key	Meaning
Section_ID	Numeric[9]	ไม่	PK	รหัสของ section
Name	Nvarchar[50]	ไม่		ชื่อของ section
Site_ID	Numeric[9]	ไม่	FK	รหัสอ้างอิงของ site

ตารางที่ 3.5 ตารางข้อมูล Site (Site)

Attribute	Data type	Null	Key	Meaning
Site_ID	Numeric[9]	ไม่	PK	รหัสของ site
Name	Nvarchar[50]	ไม่		ชื่อของ site

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดสอบ

ในบทนี้เป็นผลการทดสอบโปรแกรมที่ได้ทำการพัฒนาแล้ว ผลการทดสอบนั้นได้มาจากการรันโปรแกรมเพื่อสร้างSiteงานในการขุดคูโมงค์ การบันทึกค่าลงในฐานข้อมูล และการแสดงผลเป็นกราฟเพื่อดูค่าการทรุดตัวของดินเนื่องมาจากการขุดคูโมงค์

โดยมีหน้าต่างของแอปพลิเคชันด้าน Server ดังนี้



รูปที่ 4.1 หน้าแรกโปรแกรมด้านเซิร์ฟเวอร์

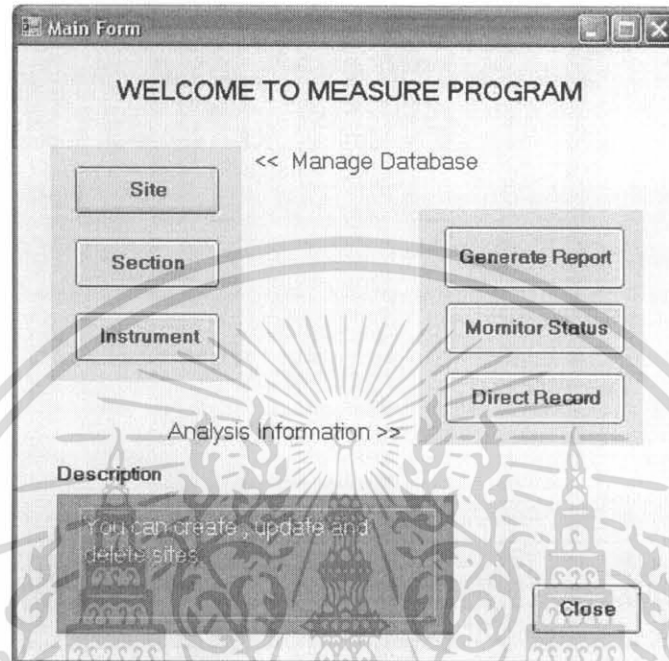
จากหน้าแรกจะแบ่งการทำงานด้าน Server ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ส่วนการทำงานของ การเปลี่ยนแปลง Site งานการชุดอุโมงค์

การเปลี่ยนแปลง Site งานนั้นจะทำโดยพนักงานที่ศูนย์ปฏิบัติการเท่านั้นไม่สามารถสร้างจากที่ pocket PC phone ได้โดยมีขั้นตอนดังนี้

-เมื่อเลื่อนเมาส์ไปที่ปุ่ม จะมีคำอธิบายขึ้นด้านล่าง



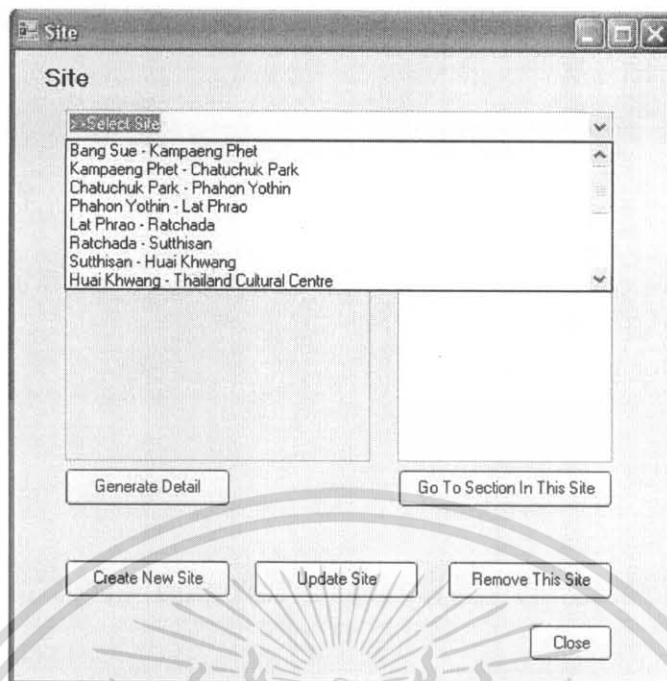
รูปที่ 4.2 แสดงการจ่อเมาส์บริเวณ ไอคอน Site

-และเมื่อคลิกเข้าไป จะเป็นหน้าต่างเพื่อเลือกรายละเอียด



รูปที่ 4.3 หน้าแรกการจัดการกับ Site

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงการเลือก Site จาก Drop down List

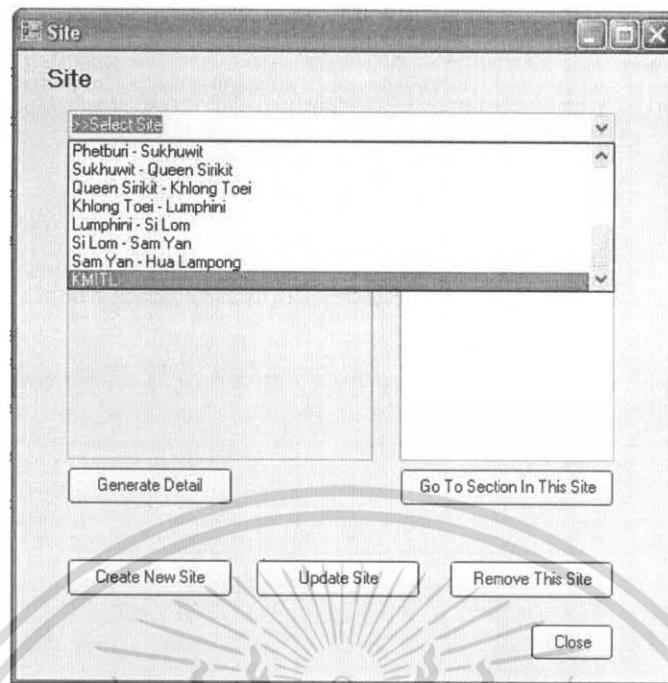
จะมีฟังก์ชันการใช้งานดังนี้

- Generate Detail คือ การเพิ่มคำอธิบายรายละเอียดของ Site งานนั้น
- Go To Section In This Site คือ การเลือกที่จะไปยังรายละเอียดของ section ใน site งานนั้น
- Create New Site คือ การสร้าง site งานใหม่ที่เกิดขึ้นเพื่อวัดการทุจริตตัวเมื่อชุดข้อมูล



รูปที่ 4.5 แสดงการสร้าง Site งานใหม่ และ คำอธิบาย Site

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



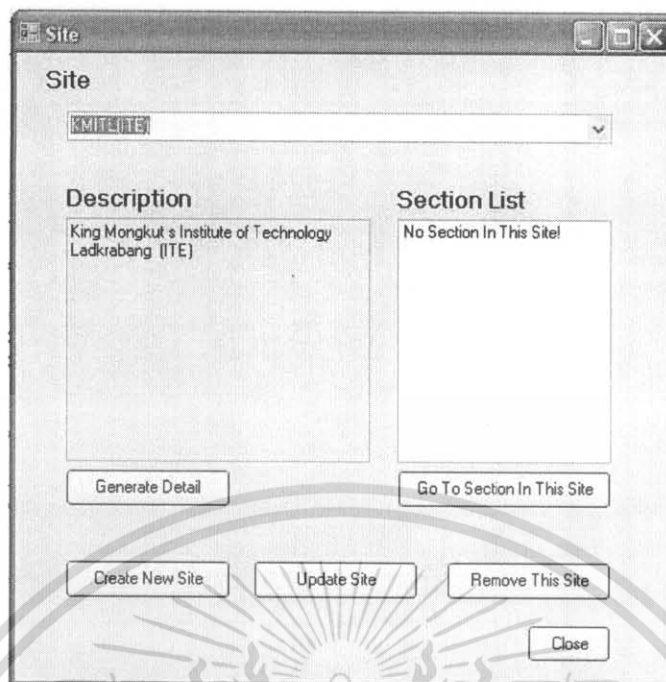
รูปที่ 4.6 แสดง Site ใหม่ที่สร้างขึ้น

- Update Site คือ การปรับเปลี่ยนรายละเอียดของ site งาน



รูปที่ 4.7 แสดงการทำการเปลี่ยนแปลงค่าชื่อ Site และคำอธิบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

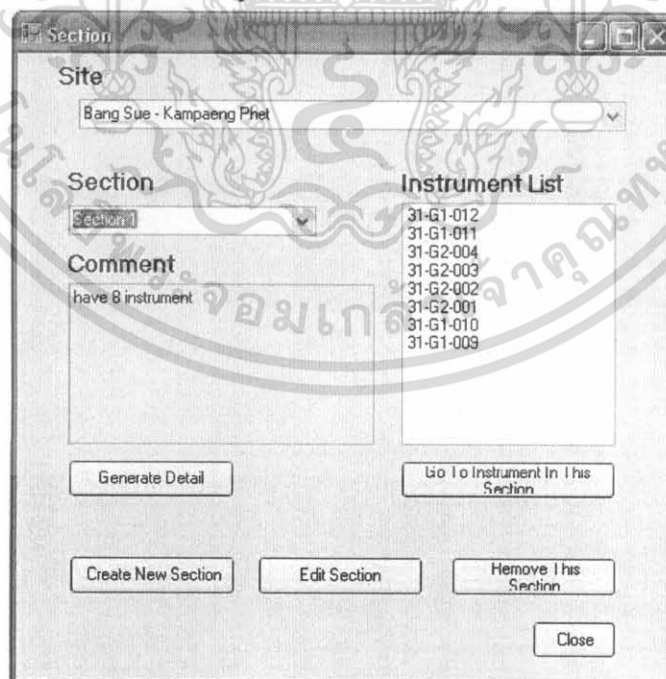


รูปที่ 4.8 แสดงค่าที่เปลี่ยนแปลงชื่อ Site และคำอธิบาย

- Remove This Site คือ การลบรายละเอียดของ site งานนั้น

4.2 ส่วนการทำงานของ การเปลี่ยนแปลง Section งานการชุดคู่มือ

จะเป็นส่วนที่เกี่ยวข้องกับการเปลี่ยนแปลงรายละเอียดของ section ซึ่งจะมีฟังก์ชันการทำงานคล้ายคลึงกับการทำงานของ site ดังรูป



รูปที่ 4.9 แสดงหน้าแรกของการจัดการ Section

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ส่วนการทำงานของการทำงานของการเปลี่ยนแปลง Instrument งานการชุดอุโมงค์

จะเป็นส่วนที่เกี่ยวข้องกับการเปลี่ยนแปลงรายละเอียดของ section ซึ่งจะมีฟังก์ชันการทำงานคล้ายคลึงกับการทำงานของ site ดังรูป

รูปที่ 4.10 แสดงหน้าแรกของการจัดการ Instrument

4.4 ส่วนการเรียกดูข้อมูลเพื่อเปรียบเทียบการพังทลายของงานการชุดอุโมงค์ (Monitor Status)

การเรียกดูผลการวัดการพังทลาย เพื่อทำการเปรียบเทียบระดับการทรุดตัวของอุโมงค์จากฐานข้อมูล แล้วจึงทำการประเมินผลความเสียหายของการทำงานของแต่ละจุด โดยมีรูปแบบการทำงานดังนี้

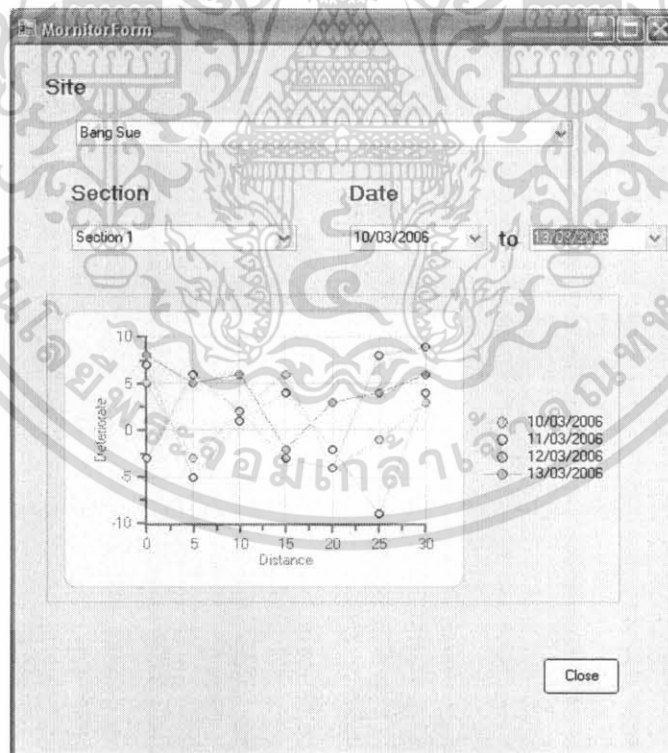
รูปที่ 4.11 แสดงหน้าแรกของการเรียกดู Monitor Status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากนั้นทำการเลือก ข้อมูลสถานที่ช่วงเวลา ที่ต้องการดูข้อมูล

รูปที่ 4.12 แสดงเลือกช่วงเวลา และ สถานที่ที่ต้องการดู

- แสดงผลออกเป็นกราฟข้อมูลของค่าจากการวัดในช่วงเวลาที่เลือก



รูปที่ 4.13 แสดงผลกราฟเปรียบเทียบการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ส่วนการบันทึกข้อมูลโดยตรงจากศูนย์ปฏิบัติการ (Direct Record)

เป็นการบันทึกค่าลงฐานข้อมูลจากศูนย์ปฏิบัติการ โดยตรง

- ใส่รายละเอียดต่างๆของ Site

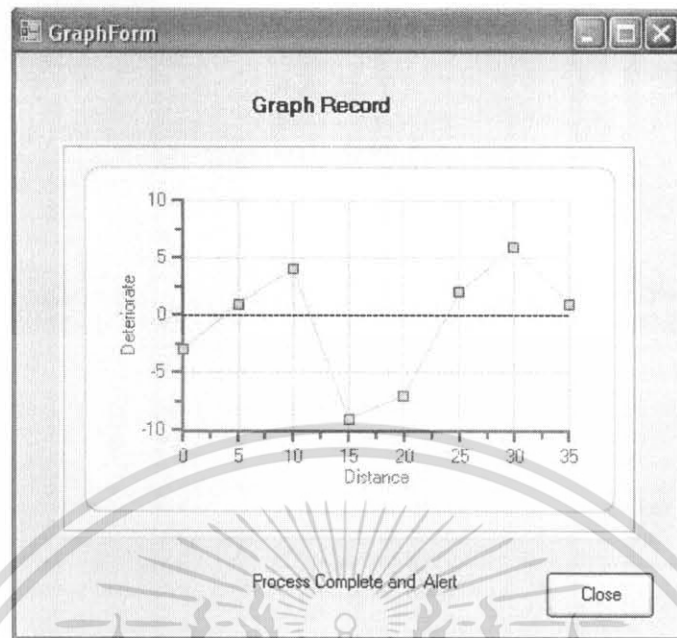
รูปที่ 4.14 แสดงหน้าแรกของการบันทึกค่าโดยตรง

- ใส่ค่าความเสียหายจากการทรุดตัว

รูปที่ 4.15 แสดงการเลือก Site Section และ ใส่พารามิเตอร์ต่างๆ

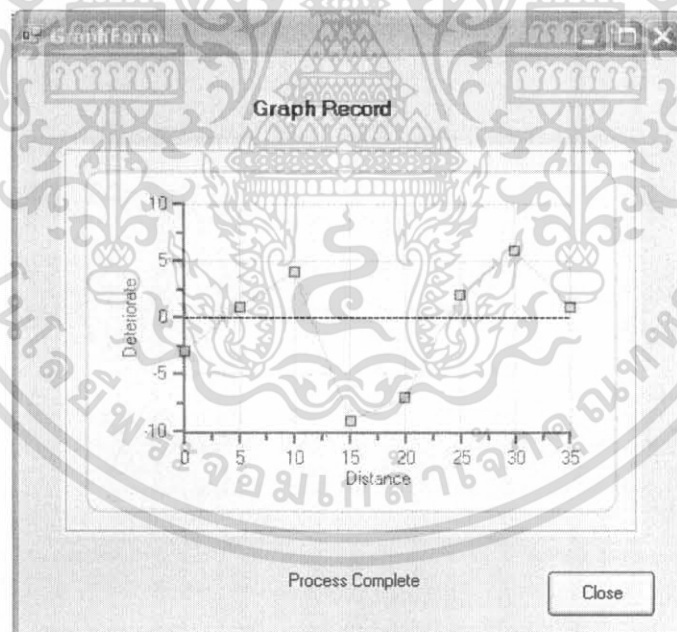
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บันทึกค่าแล้วมีการส่งข้อความเตือนภัย



รูปที่ 4.16 แสดงกราฟจากการบันทึก และมีการเตือนภัย

- บันทึกค่าแล้วไม่มีการส่งข้อความเตือนภัย



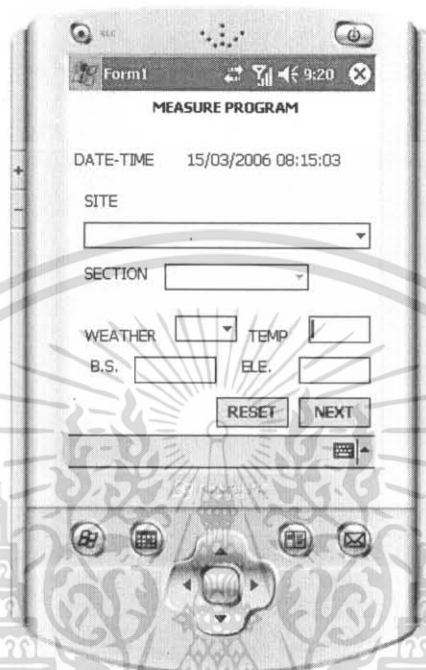
รูปที่ 4.17 แสดงกราฟจากการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ส่วนการบันทึกข้อมูลจาก Site งานผ่าน Pocket PC Phone (PDA Record)

การจดบันทึกค่าจาก Site งาน โดย Survey Man ลง Pocket PC Phone ส่งผ่านเว็บเซอร์วิส
แล้วบันทึกลงในฐานข้อมูลที่ศูนย์ปฏิบัติการ

- หน้าหลัก



รูปที่ 4.18 แสดงหน้าแรกของการบันทึกค่าผ่าน Pocket PC Phone

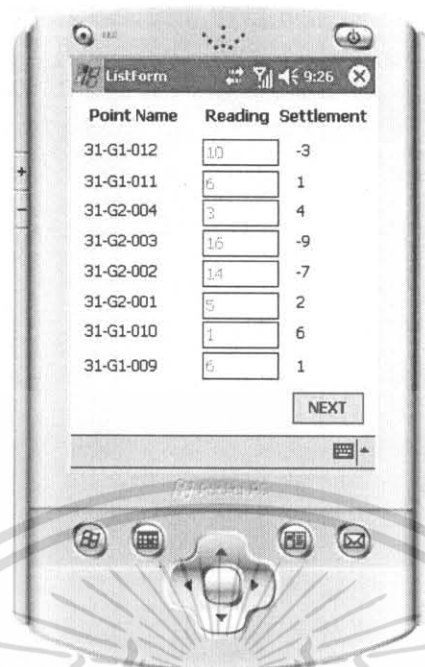
- กรอกค่าพารามิเตอร์ต่างๆ ลงในแบบฟอร์ม



รูปที่ 4.19 แสดงการเลือก Site Section และ ใส่พารามิเตอร์ต่างๆลง Pocket PC Phone

- กรอกค่าของความเสียหายจากอุปกรณ์บริเวณ Site งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดงการใส่พารามิเตอร์ค่าความเสียหายจากอุปกรณ์ที่ Site ลง Pocket PC Phone

- แสดงผลกราฟมี 2 ชนิดมีการเตือนภัยและแบบปกติ



รูปที่ 4.21 แสดงกราฟจากการบันทึก มี 2 แบบ 1. มีการเตือนภัย 2. บันทึกค่าปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ส่วนการบันทึกข้อมูลจาก Site งานผ่าน Pocket PC Phone (PDA Record)

การจดบันทึกค่าจาก Site งาน โดย Survey Man ลง Pocket PC Phone ส่งผ่านเว็บเซอร์วิส แล้วบันทึกลงในฐานข้อมูลที่ศูนย์ปฏิบัติการ

- หน้าแรก และการเลือกค่าวันที่ต้องการทำรายงาน

รูปที่ 4.22 แสดงหน้าแรกของการสร้างรายงานผลการบันทึก และการเลือก Site

- หน้ารายงานผลการตรวจสอบ

Name	Staff Read	E.L.E.	Settlement
AJ-CCY-1	2	-5	4
AJ-CCY-2	4	13	2
AJ-CCY-3	4	7	4
AJ-CCY-4	1	6	7
AJ-CCY-5	3	9	3
AJ-CCY-6	6	11	5
AJ-CCY-7	9	3	5
AJ-CCY-8	5	2	1

รูปที่ 4.23 แสดงรายงานระแงการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุป

5.1 สรุปการพัฒนาโครงการ

ในโครงการนี้ ระบบสามารถทำงานได้ตามวัตถุประสงค์ของโครงการ โดยในแต่ละส่วนสามารถบรรลุวัตถุประสงค์ต่างๆ ดังนี้คือ

5.1.1 ด้านพนักงานสำรวจ

5.1.1.1 สามารถเปิดบันทึกค่าความเสียหายของการขุดเจาะอุโมงค์ผ่านทาง Pocket PC Phone ได้

5.1.1.2 สามารถส่งข้อมูลจาก Site งานมายังเซิร์ฟเวอร์ที่ศูนย์ปฏิบัติงานแบบ Real-Time ได้

5.1.1.3 สามารถประมวลผลเป็นกราฟที่ Site งานเพื่อแสดงผลได้ทันที

5.1.2 ด้านเซิร์ฟเวอร์

สามารถรองรับไฟล์ที่อัปโหลดขึ้นมา และสามารถให้ส่วนต่างๆ ภายในระบบเข้ามาดึงข้อมูลไปได้

5.1.3 ด้านศูนย์ปฏิบัติการ

5.1.3.1 สามารถสร้าง ปรับเปลี่ยน และ ลบ ข้อมูลของ Site งานได้

5.1.3.2 สามารถสร้างรายงานของการตรวจสอบในแต่ละครั้งได้

5.1.3.3 สามารถส่งการเตือนภัยไปยัง Site งานได้

5.1.3.4 สามารถเปรียบเทียบการตรวจสอบของการขุดเจาะอุโมงค์จากฐานข้อมูลเป็น

ช่วงเวลา

เพื่อทำการประเมินแนวโน้มและออกนโยบายเกี่ยวกับการขุดได้

5.1.3.5 สามารถทำการกรอกข้อมูลลงไปโดยตรงจากทางศูนย์ปฏิบัติการได้

5.2 ปัญหาที่พบและข้อจำกัดในการพัฒนาโครงการ

5.2.1 การทำงานของโปรแกรมทางด้าน Pocket PC Phone จะต้องอาศัยการร้องขอข้อมูลผ่านเว็บเซิร์ฟวิสนั้นถ้าเกินขีดพลาตของเว็บกล่าวคือ Pocket PC Phone ไม่สามารถติดต่อเว็บเซิร์ฟวิสนี้ได้ โปรแกรมก็จะไม่สามารถทำงานได้

5.2.2 การทำงานของโปรแกรมทาง Pocket PC Phone นั้นจะต้องทำการต่อกับอินเทอร์เน็ต โปรแกรมจึงจะสามารถทำงานได้

5.2.3 โปรแกรมนี้จะใช้ได้เฉพาะ Platform Pocket PC Phone เท่านั้น platform อื่นไม่สามารถใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางแก้ไข

- 5.3.1 ทำการเปิด GPRS ค้างไว้ตั้งแต่เปิดระบบครั้งแรก
- 5.3.2 ต้องเขียน โปรแกรมให้เซิร์ฟเวอร์ทำงานส่วนการเปลี่ยน Format ภาพแทน
- 5.3.3 อาจจะเปลี่ยนไปใช้ EDGE ซึ่งก็ยังมีหายากในประเทศไทย
- 5.3.4 อาจจะเปลี่ยนไปใช้ คลาส A ซึ่งก็ยังมีหายากในประเทศไทย หรือเลือกการส่งอย่างใดอย่างหนึ่ง โดยเลือกจากเทคโนโลยีของมือถือของผู้ใช้ว่ารองรับอะไรบ้าง

5.4 การนำโครงการไปประยุกต์ใช้ในอนาคต

โครงการนี้เป็นการพัฒนาจากระบบเก่าที่จะต้องใช้การส่งข้อมูลโดยตัวบุคคล ซึ่งเป็นการเสียเวลาและไม่ทันการ การทำให้เป็นเรียไทม์จึงมีข้อได้เปรียบด้านเวลา แต่ระบบยังคงต้องอาศัยคนสำรวจที่ไปยัง Site งานอยู่คั่นในอนาคตจึงมีแนวทางพัฒนา ดังเช่นตัวอย่างต่อไปนี้

- 5.4.1 นำไปพัฒนาเป็นระบบที่ดึงข้อมูลจากอุปกรณ์ทางด้าน โยธาโดยตรง โดยที่ไม่ต้องใช้การจดบันทึก กลายเป็นระบบ หนึ่งสัมผัสแทน
- 5.4.2 นำไปพัฒนาเป็นระบบนี้มาสร้างเป็นแผนที่ของการสำรวจการทุจริต โดยทำเป็นวงกว้างก็อาจสร้างเป็นระบบมอนิเตอร์การทุจริตของแผ่นดินประจำเมืองสำคัญๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] จาก <http://www.codeproject.com/Purgatory/XMLWebservices.asp>
- [2] จาก <http://www.codeproject.com/useritems/WebServices.asp>
- [3] <http://www.csharpcorner.com>
- [4]
- <http://msdn.microsoft.com/webservices/webservices/understanding/webservicebasics/default.aspx>
- [5] <http://www.codeproject.com/Purgatory/XMLWebservices.asp>
- [6] <http://www.codeproject.com/useritems/WebServices.asp>
- [7] <http://msdn.microsoft.com/webservices/webservices/understanding/webservicebasics/default.aspx?pull=/library/en-us/dnwebsrv/html/webservbasics.asp>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

การติดตั้งโปรแกรม

ก่อนที่จะสามารถเขียนโปรแกรม visual c# ได้จะต้องทำการลงโปรแกรมที่จำเป็นสำหรับใช้งานก่อน เพื่อให้สามารถรันชุดคำสั่ง ซึ่งโปรแกรมห่วงล้อมก็คือ visual studio.net 2005 beta 2 และ sql server 2000 ซึ่งมีวิธีการติดตั้งดังต่อไปนี้

visual studio.net 2005 Professional ซึ่งเป็นตัวคอมไพเลอร์ของภาษาvisual c#ที่ใช้ในการคอมไพล์โค้ดของ c#

1. เริ่มต้นการติดตั้ง โดยทำการใส่แผ่นติดตั้งvisual studio 2005 professional
2. จากนั้นแผ่นจะทำการauto run เพื่อทำการติดตั้ง
3. จากนั้นจะเข้าสู่หน้าจอการติดตั้ง โปรแกรม
4. เลือก

sql server 2000 เป็นค่าแบสเวิร์ทเวอร์ที่สนับสนุนการทำงานของ

1. เริ่มต้นการติดตั้ง โดยการใส่แผ่นติดตั้ง sql server 2000
2. จากนั้นก็เลือก SQL Server 2000 Component



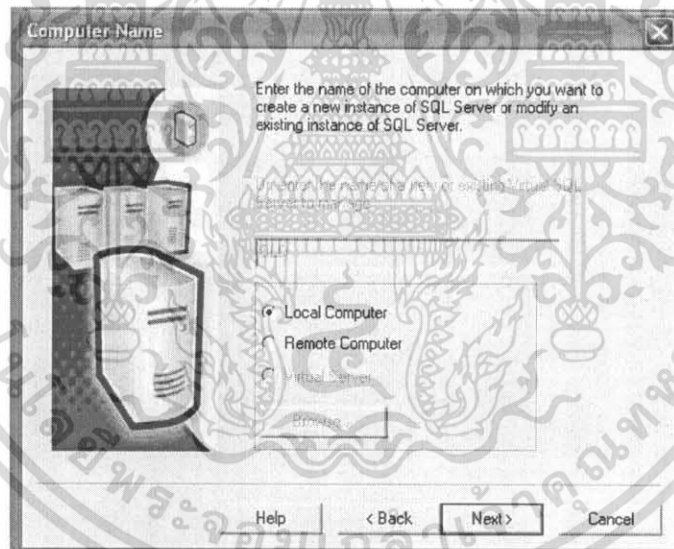
สถาปัตยกรรมการเชื่อมต่อข้อมูลผ่านเอดีโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.เลือก Install Database Server



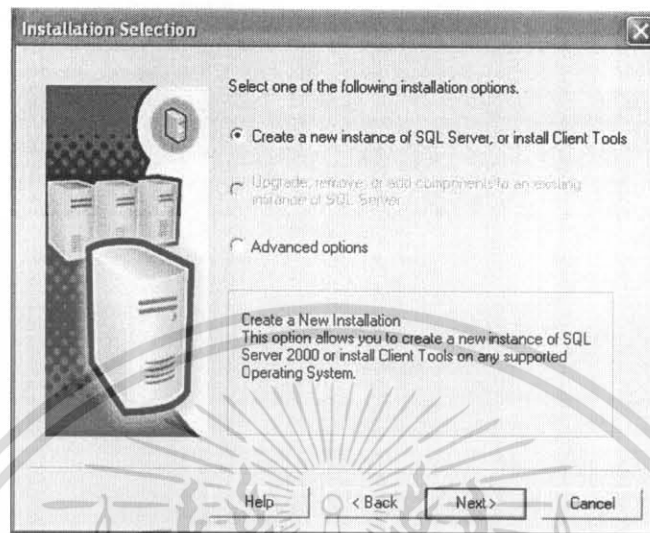
4.เลือก Local Computer จากนั้นกด Next



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จากนั้นเลือกที่ Create a new instance of SQL Server, or install Client Tools แล้ว

กด Next

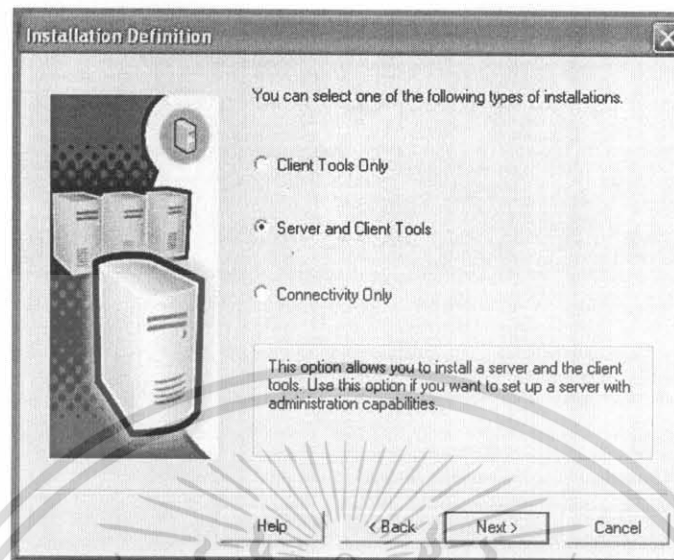


6. กรอกชื่อผู้ใช้

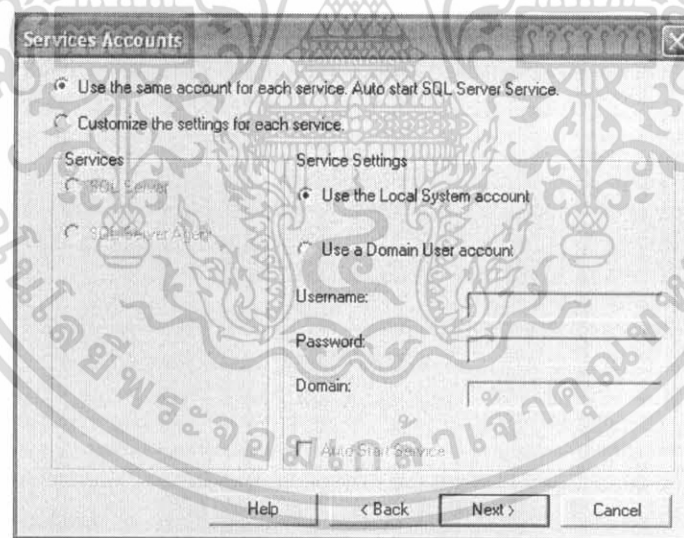


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เลือกที่ Server and Client Tools จากนั้นกด Next

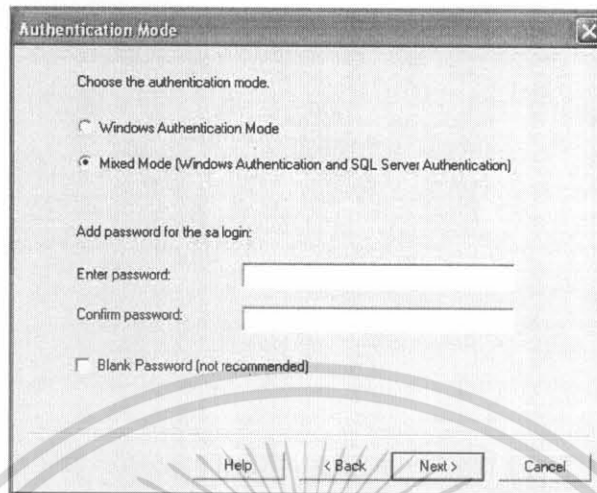


8. จากนั้นเลือกที่ Use the same account for each service. Auto start SQL Server Service และ ที่ Use the Local System account



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

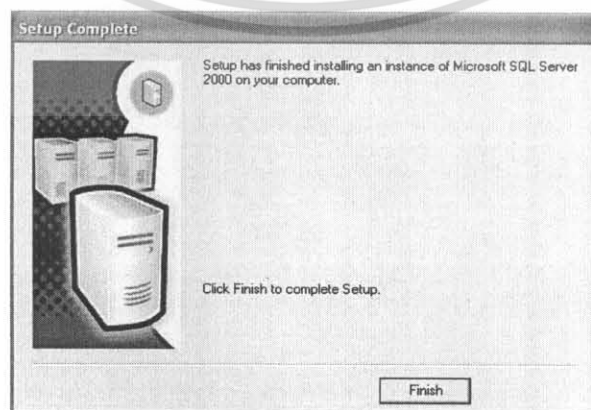
9. เลือกที่ Mixed Mode (Windows Authentication and SQL Server Authentication)
และกำหนดรหัสผ่าน พร้อมด้วยการยืนยันรหัสผ่านอีกครั้ง



10. จากนั้นจะเป็นหน้าที่แสดงถึงความคืบหน้าของการติดตั้ง



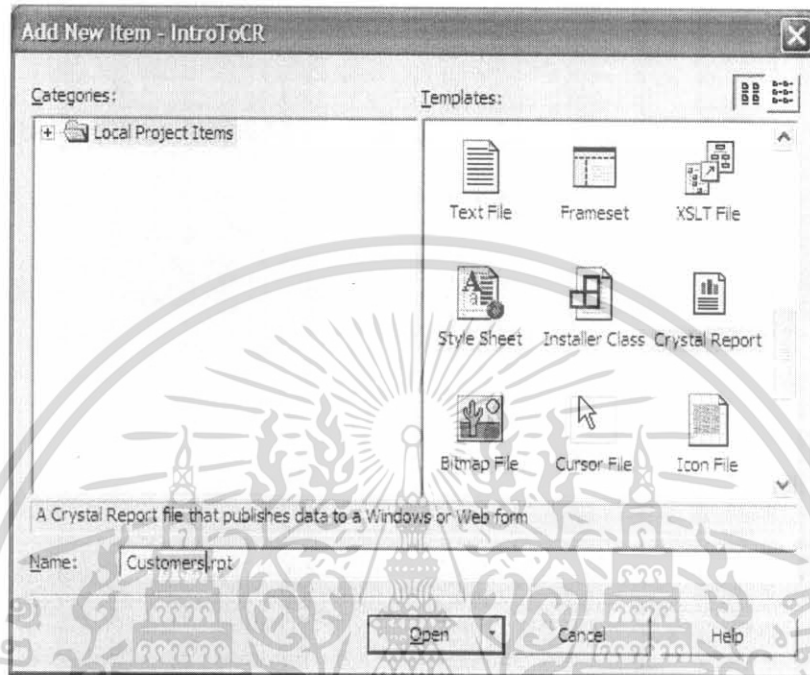
11. ถ้าโปรแกรมติดตั้งสมบูรณ์แล้วสามารถกดปุ่ม Finish



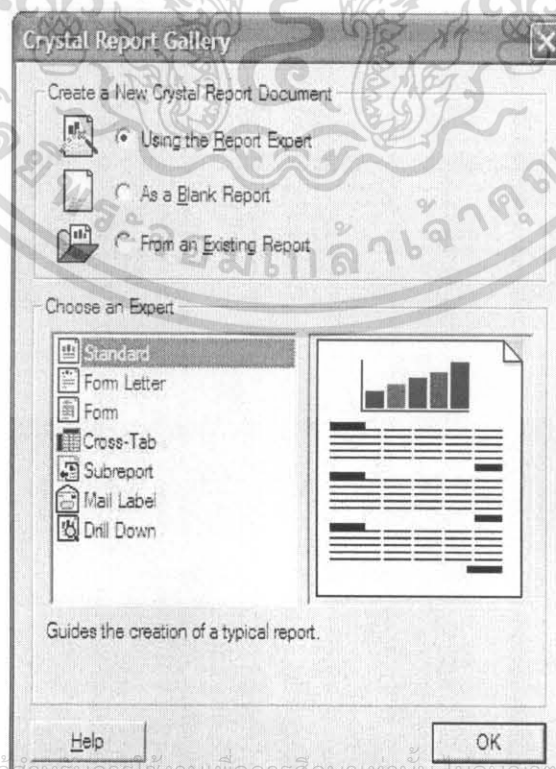
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้ Crystal Report

Crystal Report เป็นส่วนที่ต้องเพิ่มเข้าไปใหม่ใน visual studio เพื่อทำรายงานจากโปรเจ็ค โดยคลิกขวาจาก Solution Explorer เลือก add New Item -> Crystal Report ดังภาพที่ 1 โดยตั้งชื่อรูปแบบที่แสดงออกมา

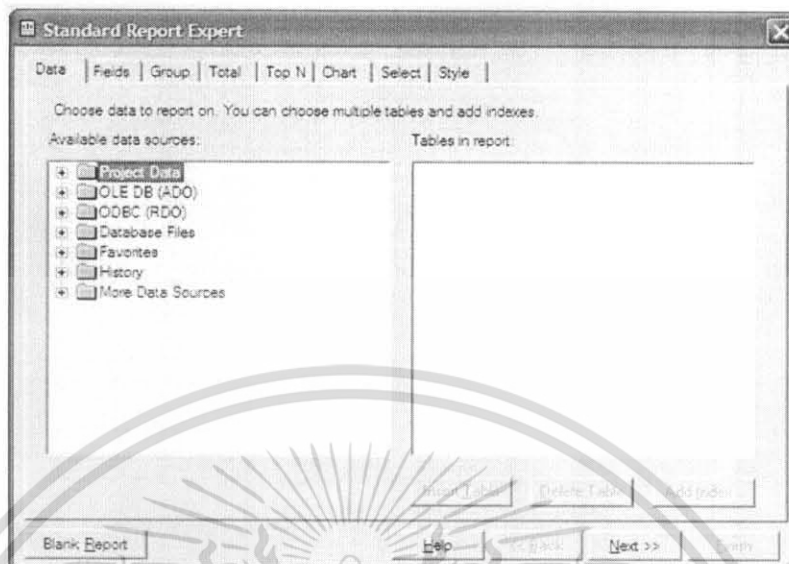


จากนั้นเลือกรูปแบบของเอกสารซึ่งมีรูปแบบ จาก Crystal Report Gallery คือ 1. Report Expert 2. Blank Report 3. Existing Report



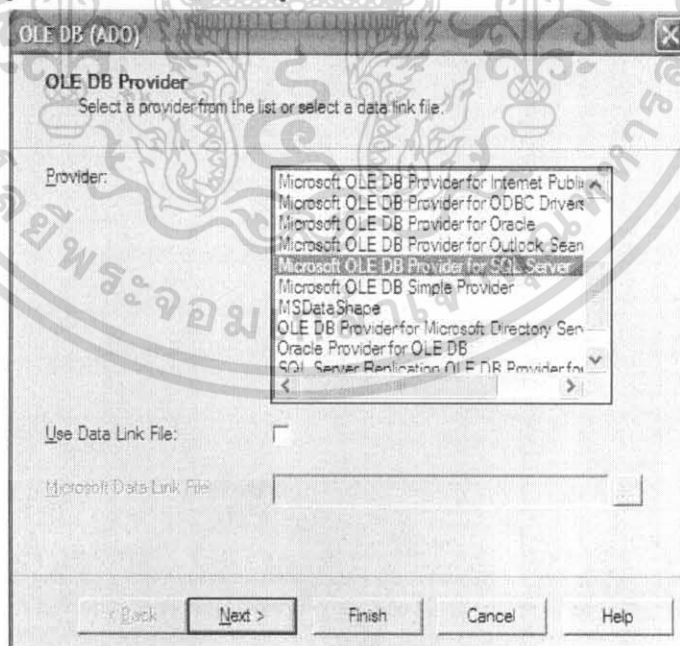
เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้เฉพาะบุคคลซึ่งจะบ่งชี้ถึงคุณลักษณะของเอกสารนี้ให้ดูที่หน้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้ว ok จะสร้างไฟล์รายงานขึ้นมาชื่อ (ชื่อไฟล์).rpt ขึ้นกับ โปรเจกต์จะสร้างรูปแบบรายงาน ขึ้นมาดังที่เห็นในรูป 3 และสามารถเลือกแหล่งข้อมูล โดยมีให้เลือก 8 tab



ต่อไปคือวิธีเชื่อมเข้ากับแหล่งข้อมูล โดยมีหลายวิธีแต่ทางที่ง่ายที่สุดคือการใช้ OLE DB(ADO)

เลือก OLE DB(ADO) แล้วจะทำงานแสดงแหล่งข้อมูลโดยเราสามารถเลือกได้จากตรง ส่วนนี้จากภาพ คือการเลือกจาก SQL Server หรือเลือกแหล่งต่างๆ ได้ เช่น Access อื่นๆ เมื่อเลือกได้แล้วจะเห็นว่าข้อมูลถูกเชื่อมต่อไปยังฐานข้อมูลที่ต้องการ



กด finish ถ้าเราเพิ่มอะไรลงไป ใน ฐานข้อมูลเราจะเห็นได้จากตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

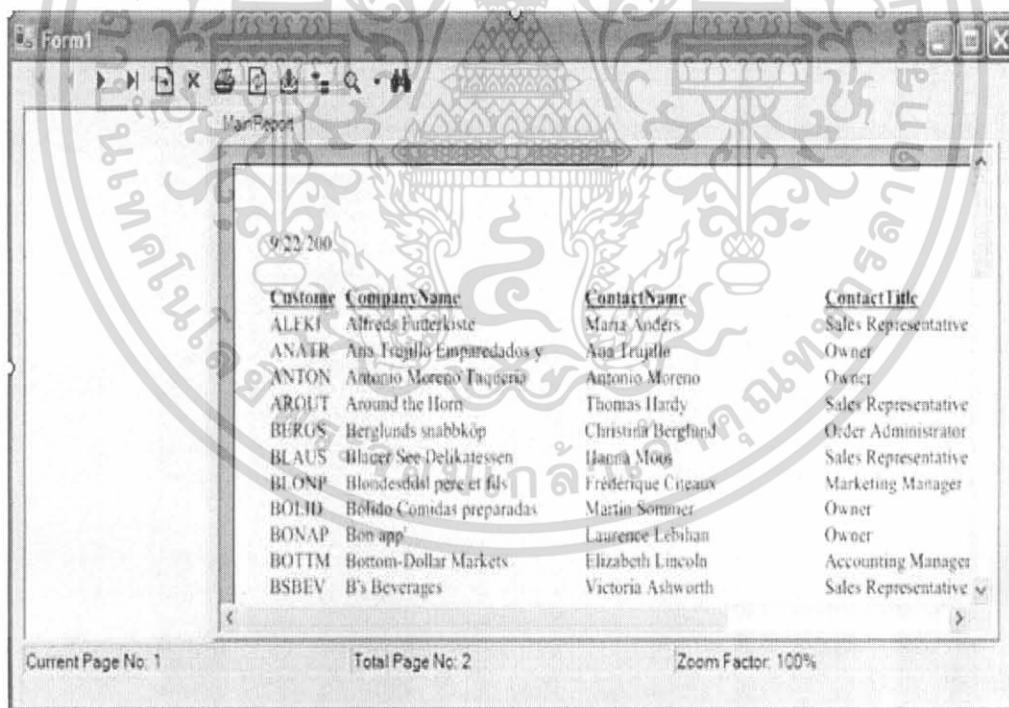
การลง Crystal Report ใน มุมมองควบคุม (Crystal Report Viewer Controller)

ขั้นต่อไปคือการทำรายงานลง form โดยการลากจากกล่องเครื่องมือ ไปที่ form และปรับขนาดให้พอดีกับ form การลงไฟล์.rpt ไปยัง Crystal Report มุมมองควบคุม เราต้องตั้งค่า property ของ Crystal Report มุมมองควบคุมบน form load event handler เราสร้าง Instance ของ class ลูกค้า (Customer) และตั้งค่าให้เป็น ของ Crystal Report มุมมองควบคุม โดยโปรแกรมดังนี้

```
Private void Form1_Load(object sender, System.EventArgs e)
```

```
{
Customers custReport = new Customer();
crystalReportViewer1.ReportSource = custReport;
}
```

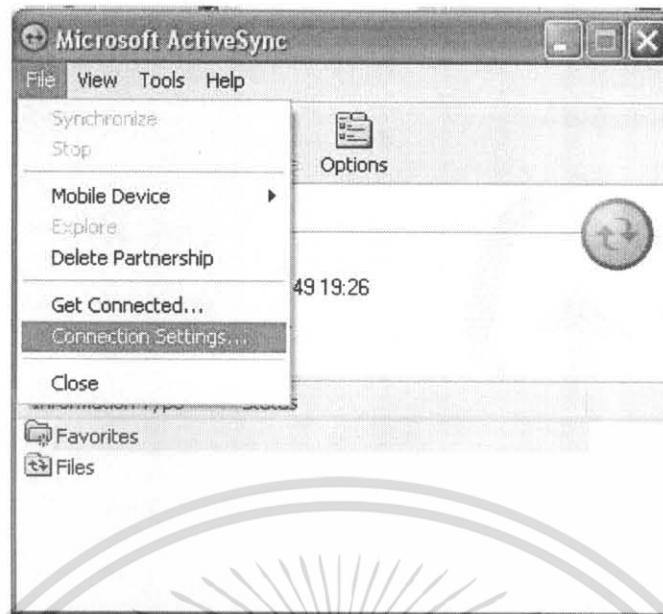
ตอนนี้เราทำการเปิดโปรแกรมแล้วจะมีผลแสดงออกมาดังรูป และสามารถทำการ Zoom in, Zoom out, print, move ได้



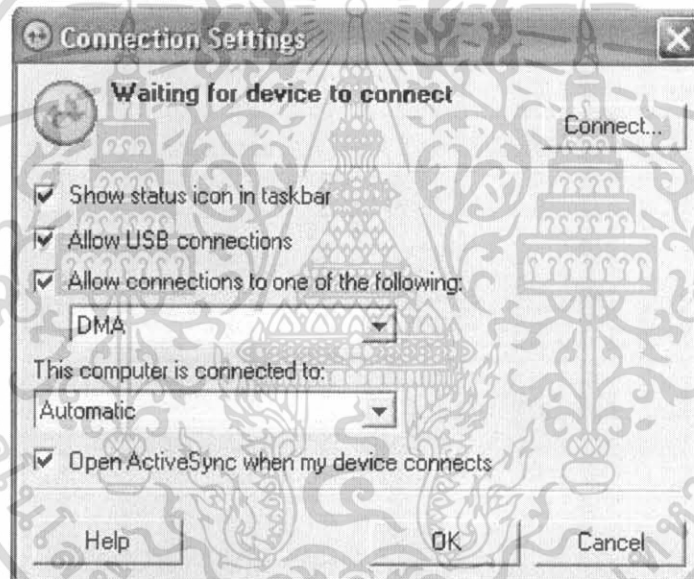
การตั้งค่าให้ Emulator ของ **visual studio.net 2005 Professional** ให้สามารถต่ออินเทอร์เน็ตได้

1. ทำการลงโปรแกรม Microsoft ActiveSync เวอร์ชัน 4.0 ขึ้นไป
2. ทำการเปิดโปรแกรม Microsoft ActiveSync แล้วเลือกที่ File->Connection Setting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

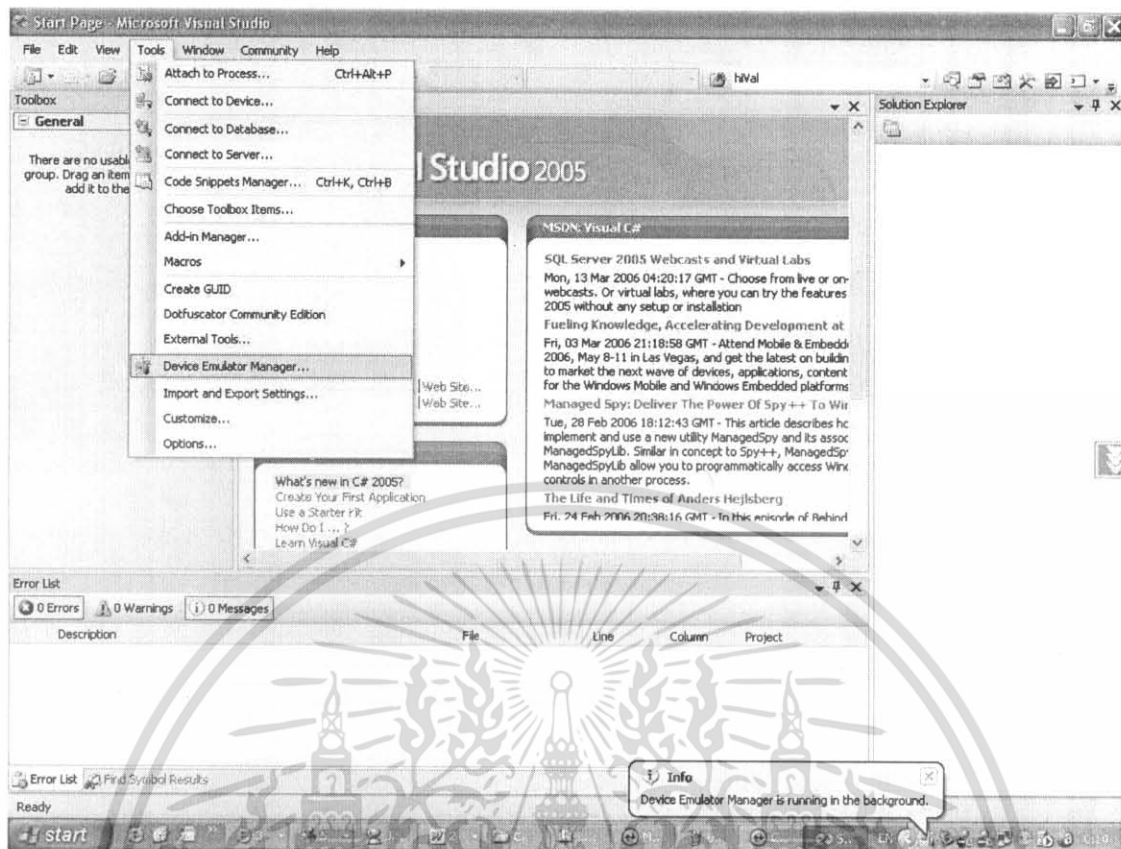


3. จากนั้นเลือกตรงช่อง Allow connections to one of the following: แล้วเลือกให้เป็น DMA ตามรูป

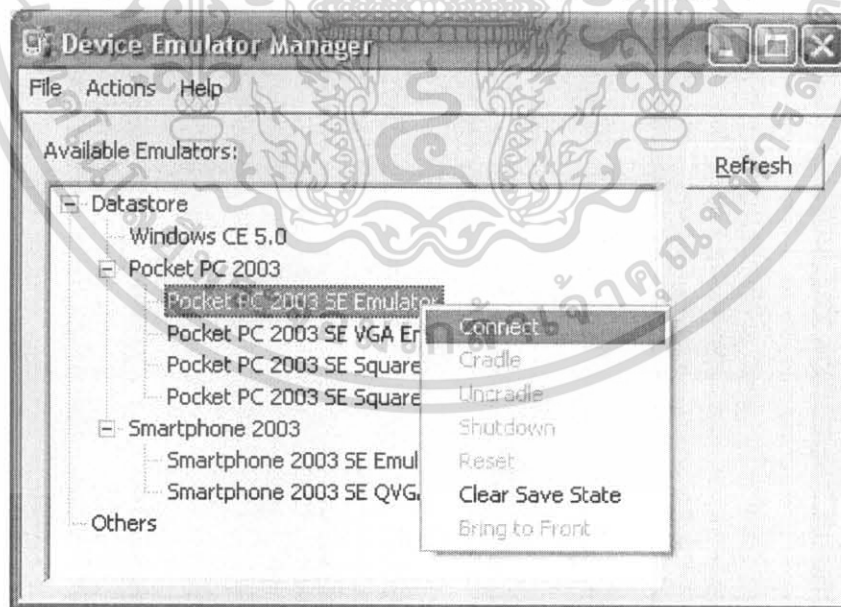


4. จากนั้นเปิดโปรแกรม **visual studio.net 2005 Professional** ขึ้นมาเลือกไปที่ **tool->Device Emulator Manager**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

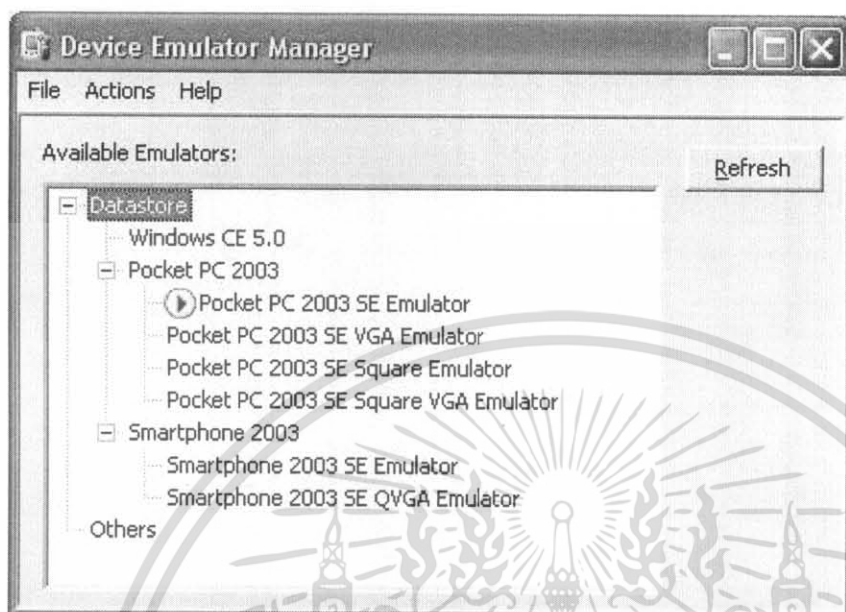


5. จากนั้นเลือก Emulator ที่ต้องการจะประมวลผลแล้วคลิกขวาเลือก connect

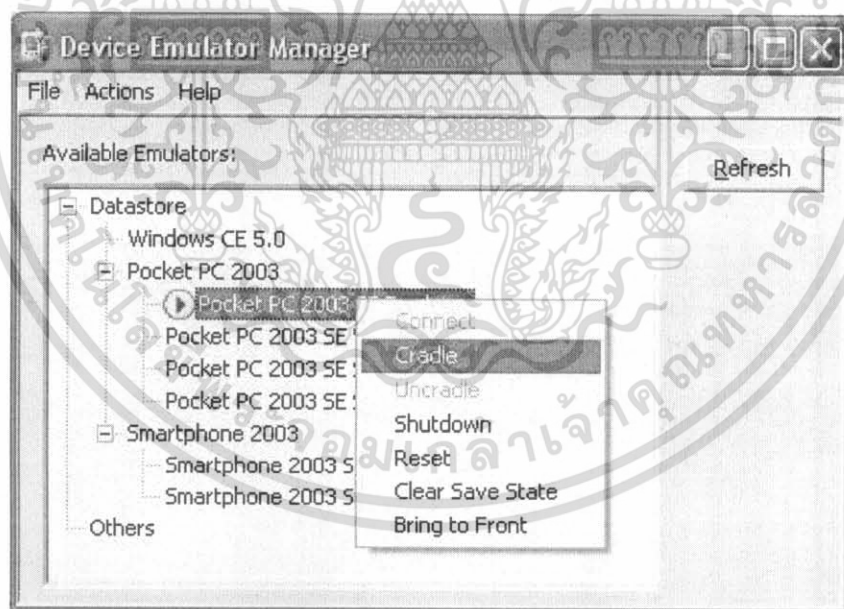


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.เมื่อตัวEmulatorนั้น ได้ถูกเรียกขึ้นมาแล้ว **Device Emulator Manager** จะแสดงสถานะการติดต่อ
ดังรูป



7.จากนั้นทำการคลิกขวาที่ตัวEmulatorที่เราเลือกแล้วเลือกไปที่Cradle



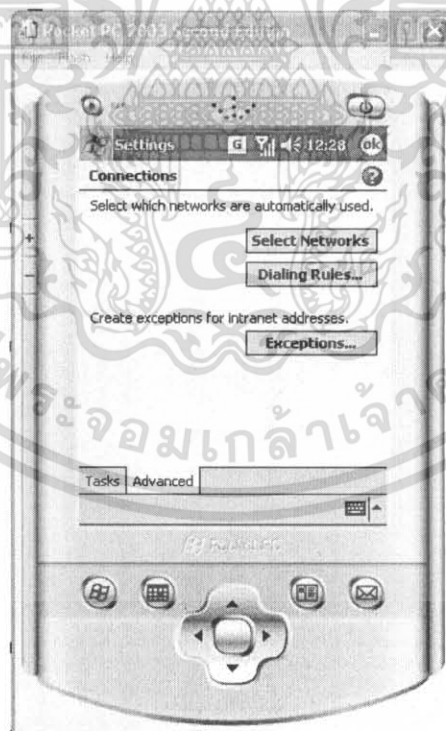
8.ตัวEmulatorจะทำการติดต่อกับMicrosoft ActiveSync

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. เลือกไปที่ Emulator กิด star->settings->Connections แล้วเลือกที่ Connections

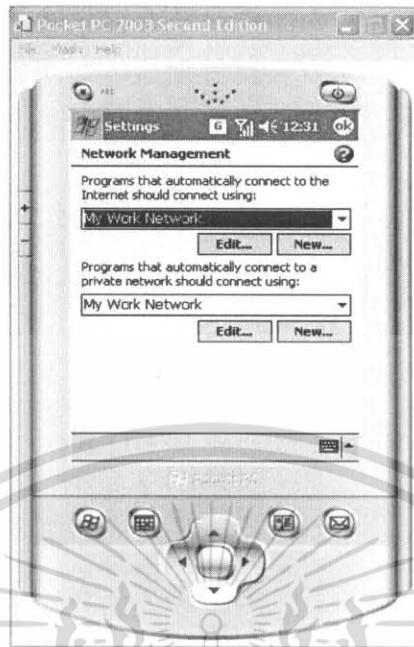


10. จากนั้นเลือกที่ Advanced->Select Networks



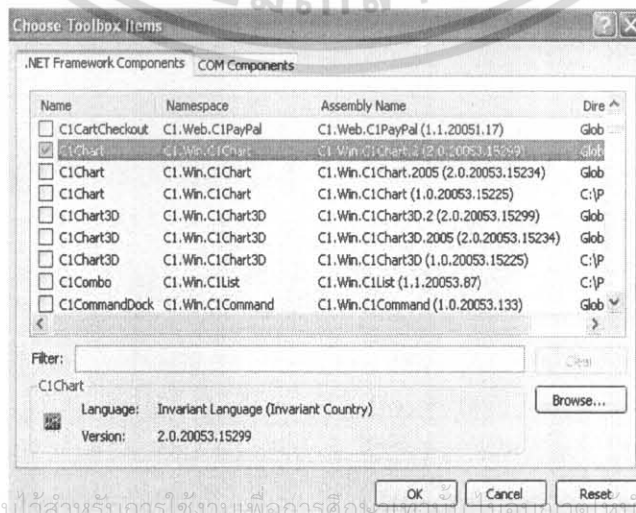
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. เลือกช่องแรกให้เป็น My Work Network แล้วกด ok



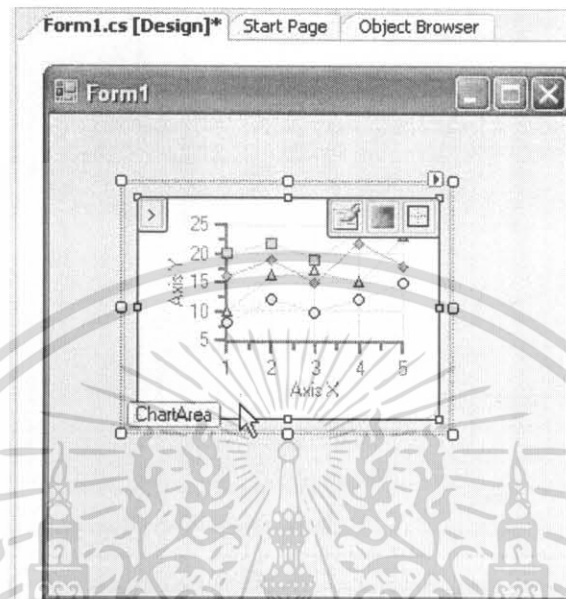
การตั้งค่าให้ Component one ใช้งานร่วมกับ **visual studio.net 2005 Professional**

1. เปิดโปรแกรม Visual Studio 2005 โดยต้องเป็นหน้าที่ tool box นั้น visible อยู่ จากนั้นทำการคลิกขวาที่ toolbox แล้วเลือกเปิดเมนู context ถ้าต้องการให้ component ของ C1Chart นั้นปรากฏให้คลิกขวาแล้วเลือก Add Tab จากนั้นตั้งชื่อ
2. คลิกขวา tab ที่เราสร้างไว้แล้วเลือกที่ **Choose Items**
3. ใน **Choose Toolbox Items** ให้เลือกไปที่ tab ของ **.NET Framework Components** เลือกตามภาพข้างล่าง



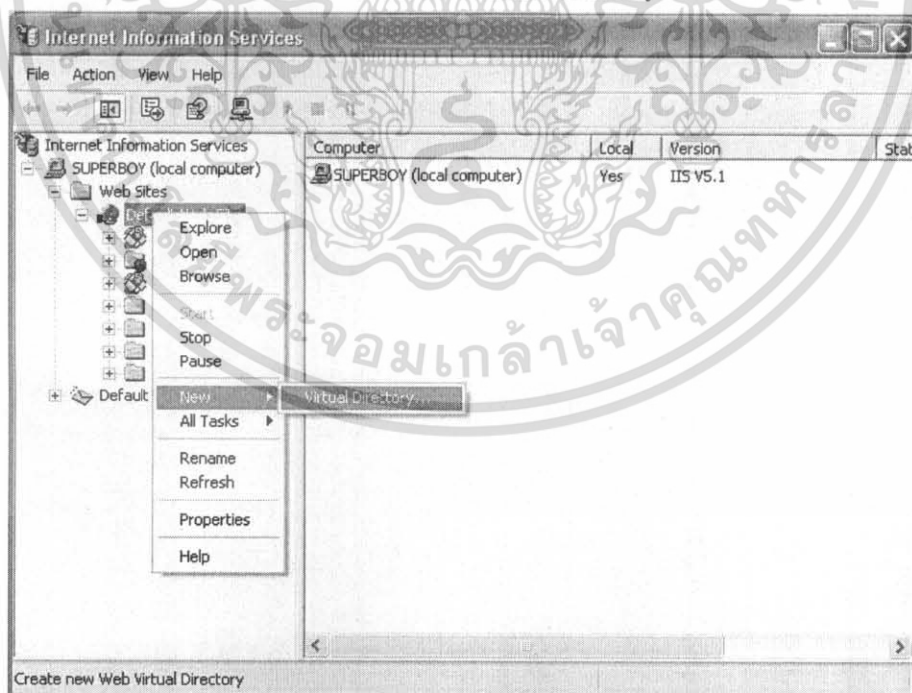
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษามาก่อน ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือกปุ่ม **Browse** buttonแล้วหาไฟล์ **C1.Win.C1Chart.2.dll** จากนั้นเลือก open
5. ดับเบิลคลิกที่ตัว **desired** ของ**C1Chart** componentsจาก Toolbox เพื่อที่จะใส่ลงในform



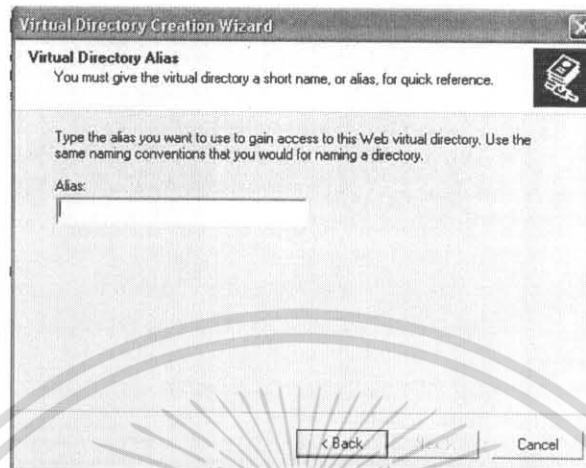
การตั้งค่า Virtual Directory ในIIS

- 1.เลือกcontrol panel-> Administrative Tools-> Internet Information Service
- 2.จากนั้นคลิกขวาที่Default Web SiteเลือกNew->Virtual Directory

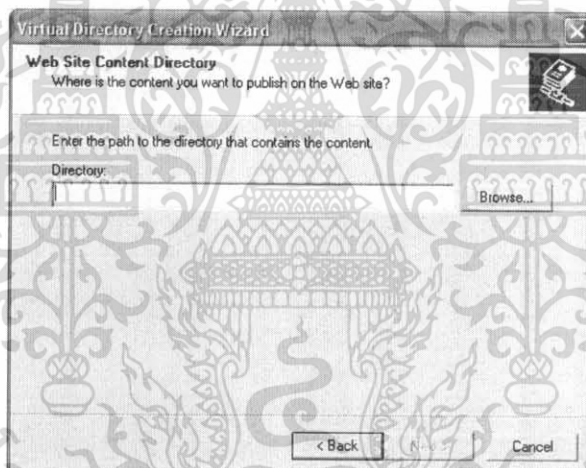


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

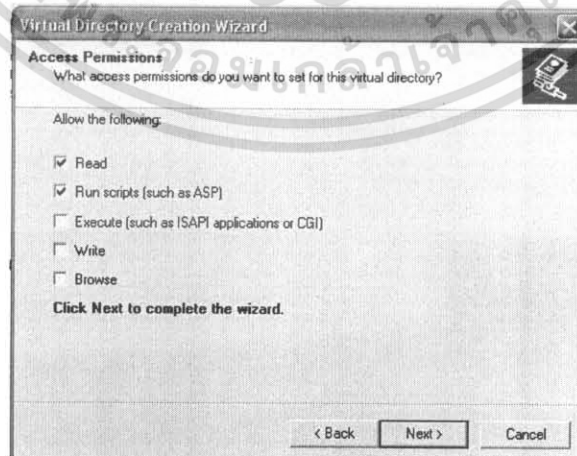
3. จะเข้าสู่ wizard ของ Virtual Directory โดยอย่างแรกทำการตั้งชื่อ



4. เลือก path ที่เราจะทำ Virtual Directory



5. กำหนดสิทธิในการจัดการเกี่ยวกับข้อมูล



6. เสร็จสิ้นการตั้งค่า Virtual Directory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้