สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้งานระบบไคลเอนต์ – เซิร์ฟเวอร์ผ่านระบบเครือข่าย WLAN

CLIENT – SERVER APPLICATION BY USING WLAN

โดย

นาย สุนทร พิทักษ์สมบูรณ์

นาย สมชาย ภาคกลาง

ปพ.
ส7897ก
2549

เลขหมู่.....
เลขทะเบียน 73016
วัน,เดือน,ปี 27 ส.ค. 2550

b 11776377
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งานระบบไคลเอนต์ – เซิร์ฟเวอร์ผ่านระบบเครือข่าย WLAN

CLIENT – SERVER APPLICATION BY USING WLAN

โดย

นายสุนทร พิทักษ์สมบูรณ์ 47015191

นายสมชาย ภาคกลาง 47015225

อาจารย์ที่ปรึกษา

รศ.ดร.มนัส สังวรศิลป์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษาที่ 2549

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประยุกต์ใช้งานระบบ ไคลเอนต์ – เซิร์ฟเวอร์ ผ่านระบบเครือข่าย WLAN

Client – Server Application by using WLAN

ผู้จัดทำ

นายสุนทร พิทักษ์สมบูรณ์ รหัส 47015191

นายสมชาย ภาคกลาง รหัส 47015225

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สังวรศิลป์

ลงชื่อ.....อาจารย์ที่ปรึกษา

(รศ.ดร. มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งานระบบ ไคลเอนต์ – เซิร์ฟเวอร์ ผ่านระบบเครือข่าย WLAN

นาย สุนทร พิทักษ์สมบูรณ์ รหัส 47015191
นาย สมชาย ภาคกลาง รหัส 47015225
รศ.ดร. มนัส สังวรศิลป์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

ในปัจจุบันระบบการสื่อสารเครือข่ายไร้สายมีความต้องการมากขึ้นกว่าในอดีต อีกทั้งได้เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น ทำให้เกิดแนวคิดในการพัฒนาโครงงานที่เป็นการประยุกต์ใช้งานระบบไคลเอนต์ – เซิร์ฟเวอร์ ผ่านระบบเครือข่ายไร้สาย โดยมีการนำอุปกรณ์ไร้สายมาร่วมใช้ในการพัฒนาทำให้เกิดความสะดวก และรวดเร็วในการใช้งาน

สำหรับระบบที่พัฒนาขึ้นนี้ได้มีการนำเทคโนโลยีดอทเน็ตมาประยุกต์ใช้กับระบบร้านอาหาร เพื่อใช้แสดงรายการอาหาร สั่งอาหารและบริการต่างๆภายในร้าน เพื่อให้การบริการเป็นไปได้อย่างรวดเร็วและเป็นที่ประทับใจแก่ลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLIENT – SERVER APPLICATION BY USING WLAN

Mr. Sunthon Pitaksomboon ID.47015191
Mr. Somchai Pakklang ID.47015225
Assoc.Prof.Dr. Manas Sangworasil (Advisor)
Education Year 2006

Abstract

Nowadays, Wireless Network Communication is required much more than in the past and it's also participate in our modern life style. Therefore, Client–Server Application by using Wireless Lan is established. In addition wireless device is used in this project for more convinces and can access in any time.

This system is developed based on Dot Net Technology that able to apply for system restaurant in use to show food menu send order and many service for able to service in fast and impressing for the customer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การสร้างโครงงานเรื่อง การประยุกต์ใช้งานระบบ ไคลเอนต์ – เซิร์ฟเวอร์ ผ่านระบบเครือข่าย WLAN นี้ใช้เวลาในการศึกษาหาข้อมูล รวบรวมข้อมูลและเก็บรายละเอียดต่างๆ โดยใช้เวลาประมาณ 4 เดือน ซึ่งแต่ละขั้นตอนของการทำงานได้พบอุปสรรคต่างๆ มากมาย แต่ก็ผ่านไปด้วยดี ทั้งนี้เพราะอาจารย์ที่ปรึกษา คือ รศ.ดร.มนัส สังวรศิลป์ คอยให้ความรู้คำแนะนำต่างๆ ซึ่งมีประโยชน์อย่างมาก และยังคอยกระตุ้นให้มีการทำงานอย่างสม่ำเสมอ และผู้จัดทำหวังว่าความรู้และประสบการณ์ต่างๆ ที่ได้รับจากการทำงานครั้งนี้คงจะมีประโยชน์อย่างมากในการทำงานในอนาคต

ผู้จัดทำขอขอบคุณอาจารย์ที่ปรึกษา อาจารย์ท่านอื่นๆ และที่ขาดไม่ได้คือพี่โอ๋ ที่ให้คำปรึกษาและให้ความช่วยเหลือในการทำโครงงานนี้เป็นอย่างดี จนงานนี้สำเร็จไปได้ด้วยดีตามที่คาดหวังไว้

สุนทร พิทักษ์สมบูรณ์

(นายสุนทร พิทักษ์สมบูรณ์)

สมชาย ภาคกลาง.

(นายสมชาย ภาคกลาง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 แนวคิดและที่มาของปัญหา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 ผลที่คาดว่าจะได้รับ	2
1.5 อุปกรณ์ที่ต้องใช้ในการพัฒนาโครงงาน	2
บทที่ 2 ทฤษฎีพื้นฐาน	3
2.1 การพัฒนาระบบเชิงวัตถุ(Object Oriented)	3
2.2 การออกแบบระบบเชิงวัตถุโดยใช้ยูเอ็มแอล (UML)	4
2.2.1 ความหมายของวัตถุ	4
2.2.2 การใช้ยูเอ็มแอลในการวิเคราะห์ขั้นตอนการทำงานของระบบ	4
2.2.2.1 คลาสไดอะแกรม (Class Diagram)	5
2.2.2.2 ยูสเคสไดอะแกรม (Use Case Diagram)	9
2.3 ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ	11
2.3.1 การวิเคราะห์ความต้องการของระบบหรือผู้ใช้	12
2.3.2 การวิเคราะห์ระบบ (Domain Analysis)	12
2.3.3 การออกแบบ(Design)	13
2.3.4 การสร้างโปรแกรมระบบ (Construction, Coding, Implementation)	13
2.3.5 การทดสอบระบบ (Testing)	14
2.2 ภาษายูเอ็มแอล	15
2.4.1 ความเป็นมาของภาษายูเอ็มแอล	15

เอกสารนี้เป็นการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
2.4.2 ข้อดีของยูเอ็มแอล	15
2.4.3 องค์ประกอบของยูเอ็มแอล	16
2.5 เทคโนโลยีดอทเน็ตเฟรมเวิร์ค (.NET Framework)	20
2.5.1 ประโยชน์และข้อดีของ .NET Framework	21
2.5.2 สถาปัตยกรรม .NET	22
2.5.3 เลเยอร์คอมมอนแลงเกวจรันไทม์(Common Language Runtime)	23
2.5.4 เลเยอร์เบสคลาสไลบรารีแลงเกวจ(Base Class Library Language)	28
2.5.5 เลเยอร์คอมมอนแลงเกวจสเป็คซิฟิเคชั่น (Common Language Specification Layer)	29
2.6 ระบบฐานข้อมูล(Database System)	29
2.6.1 ระบบฐานข้อมูลเชิงสัมพันธ์(Relational Database)	30
2.6.2 คำศัพท์ต่างๆในระบบฐานข้อมูล	30
2.7 เทคโนโลยีเอดีโอดอทเน็ต(ADO.NET)	31
2.7.1 องค์ประกอบภายในของเอดีโอดอทเน็ต	32
2.7.2 ออปเจ็กต์ที่สำคัญในเอดีโอดอทเน็ต	33
2.8 โครงสร้างของเอกสาร ASP.NET	34
2.9 ลักษณะการทำงานของเว็บ ASP.NET	40
2.9.1 การทำงานที่เกิดขึ้นเมื่อเรียกใช้งานเว็บเพจ ASP.NET	40
2.9.2 การทำงานในรูปแบบ View State	41
บทที่ 3 การออกแบบ	43
3.1 ข้อมูลเบื้องต้น	43
3.1.2 โครงสร้างโดยทั่วไปของระบบ	45
3.2 การออกแบบระบบ	46
3.2.1 ไฮเลเวลยูสเคสไดอะแกรมของระบบ(Hight level Use Case Diagram)	46
3.2.2 ยูสเคสไดอะแกรม (Use Case Diagram)	46
3.2.3 แอคทิวิตี้ไดอะแกรม(Activity Diagram)	50
3.2.4 ดาต้าดิกชันนารี (Data Dictionary)	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 ผลการทดลอง 56

4.1 ส่วนของพนักงาน 56

4.1.1 ส่วนของหน้าจอสำหรับการเข้าสู่ระบบ (Login) 56

4.1.2 ส่วนของหน้าจอหลักสำหรับพนักงาน 57

4.1.3 ส่วนของรายการอาหาร 58

4.1.4 ส่วนของรายการอาหารชนิดต่างๆ 59

4.1.5 ส่วนของการเลือกเครื่องดื่ม 60

4.1.6 ส่วนของการคิดเงิน 61

4.1.7 ส่วนของโต๊ะต่างๆ 62

4.1.8 ส่วนของการตรวจสอบที่ห้องครัว 63

4.1.9 ส่วนของการตรวจสอบที่บาร์เครื่องดื่ม 64

4.2 ส่วนของห้องครัว 65

4.3 ส่วนของบาร์เครื่องดื่ม 66

4.4 ส่วนการจัดการระบบ 67

4.4.1 ส่วนของการดูรายการอาหาร 67

4.1.2 ส่วนของการแก้ไขรายการอาหาร 68

4.1.3 ส่วนของการเพิ่มรายการอาหาร 69

4.1.4 ส่วนของการลบรายการอาหาร 70

4.1.5 ส่วนของสถิติการขายอาหาร 71

4.1.6 ส่วนของรายการเครื่องดื่มคงเหลือ 72

บทที่ 5 สรุปและวิเคราะห์ผลการทดลอง 73

บรรณานุกรม

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 สัญลักษณ์ของคลาสและคลาสไอคอน	6
รูปที่ 2.2 ความสัมพันธ์แบบดีเพนเดนซี	7
รูปที่ 2.3 ตัวอย่างความสัมพันธ์แบบอะกรีเกชัน	8
รูปที่ 2.4 ตัวอย่างความสัมพันธ์แบบคอมโพสิท	8
รูปที่ 2.5 สัญลักษณ์การเขียนยูสเคสไดอะแกรม	9
รูปที่ 2.6 คลาสไดอะแกรมของระบบ Washing Machine	16
รูปที่ 2.7 ออบเจ็กต์ไดอะแกรมของระบบ washing machine	16
รูปที่ 2.8 แสดงตัวอย่างยูสเคสไดอะแกรม	17
รูปที่ 2.9 แสดงตัวอย่างของแอ็กทิวิตี้ไดอะแกรม	18
รูปที่ 2.10 โครงสร้างของ .NET Framework	20
รูปที่ 2.11 รูปแสดงถึงสถาปัตยกรรมการพัฒนาแอพพลิเคชั่นด็อทเน็ต	21
รูปที่ 2.12 โครงสร้างแอพพลิเคชั่นแรกๆ และการติดต่อระหว่างแอพพลิเคชั่น	22
รูปที่ 2.13 การพัฒนาแอพพลิเคชั่นบนเทคโนโลยีด็อทเน็ตคลาสภายในแต่ละแอพพลิเคชั่นสามารถติดต่อกันได้	23
รูปที่ 2.14 โครงสร้างการคอมไพล์จากวิชวลสตูดิโอด็อทเน็ตและเมื่อนำไปใช้งานจริง	23
รูปที่ 2.15 แสดงสถาปัตยกรรมของคอมมอนแลงแกวจรันไทม์	24
รูปที่ 2.16 แสดงความสามารถในการติดต่อข้ามแอพพลิเคชั่น	25
รูปที่ 2.17 เบสคลาสไลบรารีของสถาปัตยกรรมด็อทเน็ต	27
รูปที่ 2.18 คอมมอนแลงเกวจสเป็คซิฟิเคชั่นของสถาปัตยกรรมด็อเน็ต	28
รูปที่ 2.19 แสดงโครงสร้างภายในของเอดีโอด็อทเน็ต	31
รูปที่ 2.20 แสดงโครงสร้างหลักของเอกสาร ASP.NET	33
รูปที่ 2.21 แสดงขั้นตอนการทำงานที่เกิดขึ้นเมื่อมีการเรียกวิ่บเพจ ASP.NET	39
รูปที่ 3.1 โดเมนโมเดล (Domain Model)	45
รูปที่ 3.2 แสดงไฮเลเวลยูสเคส (Hight level Use Case)	46
รูปที่ 3.3 ยูสเคสไดอะแกรมสำหรับการจัดการเกี่ยวกับรายการสั่งสินค้า (Order product)	46
รูปที่ 3.4 ยูสเคสไดอะแกรมการจัดการเกี่ยวกับการสร้างรายการสั่งสินค้า (Create Order)	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 3.4 ยูสเคสไดอะแกรมการจัดการเกี่ยวกับการยกเลิกรายการสั่งสินค้า (Cancel Product) 44
รูปที่ 3.5 ยูสเคสไดอะแกรมสำหรับการเรียกเก็บเงิน(Check Bill) 45
รูปที่ 3.6 ยูสเคสไดอะแกรมสำหรับแจ้งการทำอาหารเสร็จ(Notify Finish Food) 45
รูปที่ 3.7 ยูสเคสไดอะแกรมสำหรับการตรวจสอบสถานะของโต๊ะ(Check Table) 46
รูปที่ 3.8 ยูสเคสไดอะแกรมสำหรับการส่งรายการอาหารจากบริกรไปห้องครัว (Send Order) 49
รูปที่ 3.9 ยูสเคสไดอะแกรมการส่งรายการอาหารจากบริกรไปยังพนักงานคิดเงิน (Send Order) 49
รูปที่ 3.10 แอคทิวิตี้ไดอะแกรมสำหรับการล็อกอินเข้าใช้งานระบบ 50
รูปที่ 3.11 แอคทิวิตี้ไดอะแกรมสำหรับการสั่งอาหาร 51
รูปที่ 3.12 แอคทิวิตี้ไดอะแกรมสำหรับการเรียกเก็บเงิน 52
รูปที่ 3.13 แอคทิวิตี้ไดอะแกรมสำหรับการยกเลิกรายการอาหาร 53
รูปที่ 4.1 หน้าจอสำหรับการล็อกอินเข้าใช้ระบบ 56
รูปที่ 4.2 หน้าจอหลักสำหรับพนักงาน 57
รูปที่ 4.3 หน้าจอแสดงเมนูเลือกชนิดอาหาร 58
รูปที่ 4.4 หน้าจอแสดงรายการอาหารชนิดต่างๆ 59
รูปที่ 4.5 หน้าจอแสดงรายการเครื่องดื่ม 60
รูปที่ 4.6 หน้าจอสำหรับทำการคิดเงินโต๊ะต่างๆ 61
รูปที่ 4.7 หน้าจอแสดงส่วนของโต๊ะต่างๆ 62
รูปที่ 4.8 หน้าจอแสดงส่วนของการตรวจสอบที่ห้องครัว 63
รูปที่ 4.9 หน้าจอแสดงส่วนของการตรวจสอบที่บาร์เครื่องดื่ม 64
รูปที่ 4.10 แสดงส่วนของห้องครัว 65
รูปที่ 4.11 แสดงส่วนของบาร์เครื่องดื่ม 66
รูปที่ 4.12 แสดงส่วนของการดูรายการอาหารทั้งหมด 67
รูปที่ 4.13 แสดงส่วนของการแก้ไขรายการอาหาร 68
รูปที่ 4.14 แสดงส่วนของการเพิ่มรายการอาหาร 69
รูปที่ 4.15 แสดงส่วนของการลบรายการอาหาร 70
รูปที่ 4.16 แสดงส่วนของสถิติการขายอาหาร 71
รูปที่ 4.17 แสดงจำนวนเครื่องดื่มคงเหลือ 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 3.1 ตารางข้อมูลชนิดอาหาร	54
ตารางที่ 3.2 ตารางข้อมูลโต๊ะ (Table)	54
ตารางที่ 3.2 ตารางข้อมูลพนักงาน (Employee)	55
ตารางที่ 3.4 ตารางข้อมูลห้องครัว	55
ตารางที่ 3.5 ตารางข้อมูลวัตถุดิบ (InStock)	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 แนวคิดและที่มาของปัญหา

ในยุคปัจจุบันความเจริญก้าวหน้าทางด้านการสื่อสารได้พัฒนาขึ้นอย่างรวดเร็ว ดังจะเห็นได้ว่า การติดต่อสื่อสารแบบไร้สายได้ก้าวเข้ามามีบทบาทและมีอิทธิพลต่อชีวิตประจำมากขึ้น โดยเฉพาะในทางธุรกิจ โดยมีการนำเทคโนโลยีการสื่อสารแบบไร้สายมาเป็นสื่อกลางในการติดต่อและให้บริการแก่ลูกค้า เพื่อให้การดำเนินธุรกิจมีความสะดวกและรวดเร็ว รวมถึงเป็นการลดค่าใช้จ่ายในระยะยาวในการดำเนินธุรกิจนั้น ซึ่งเราจะเห็นได้ว่า ราคาของอุปกรณ์ไร้สายมีราคาที่ถูกลงจึงเป็นการดีที่จะนำเทคโนโลยีไร้สายมาประยุกต์ใช้กับการดำเนินงานต่างๆ

โครงงานนี้ผู้จัดทำได้เล็งเห็นถึงแนวทางและประโยชน์อย่างมากในการพัฒนาการใช้ประโยชน์จากการผสมผสานของความสามารถในระบบซอฟแวร์(Software) และอุปกรณ์ไร้สาย (Wireless Device)เพื่อเป็นการพัฒนาและสร้างสรรค์เพื่อก่อให้เกิดประโยชน์สูงสุด

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อพัฒนาระบบจัดการสารสนเทศร้านอาหารให้มีมาตรฐาน

1.2.2 เพื่ออำนวยความสะดวกและเพิ่มความรวดเร็วในการให้บริการ

1.2.3 เพื่อเป็นการศึกษาระบบงานธุรกิจร้านอาหาร

1.2.4 เพื่อพัฒนาและนำเทคโนโลยีไร้สาย มาใช้งานกับระบบต่างๆให้มากขึ้น

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถที่จะทำการสั่งอาหารผ่านอุปกรณ์ไร้สายได้

1.3.2 สามารถรองรับการทำงานแบบการใช้ข้อมูลพร้อมกัน(Multi-User)ได้

1.3.3 สามารถทำการพิมพ์รายการชำระเงินได้

1.3.4 ระบบฐานข้อมูลถูกออกแบบในลักษณะแบบจำลองข้อมูลเชิงสัมพันธ์

1.3.5 สามารถดูส่วนประกอบของอาหารและภาพอาหารได้

1.3.6 สามารถส่งคำสั่งรายการอาหารไปยังห้องครัวและบาร์เครื่องดื่มได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ผลที่คาดว่าจะได้รับ

1.4.1 สามารถนำระบบที่พัฒนาขึ้นมาประยุกต์ใช้กับร้านอาหารได้

1.4.2 สามารถเพิ่มประสิทธิภาพในการให้บริการได้

1.4.3 สามารถนำธุรกิจมาประยุกต์ใช้ร่วมกับเทคโนโลยีใหม่ๆในปัจจุบันได้

1.4.4 มีการพัฒนาและนำเทคโนโลยีไร้สายมาประยุกต์ใช้มากขึ้นในอนาคต

1.5 อุปกรณ์ที่ใช้ในการพัฒนาโครงงาน

1.5.1 ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์สำหรับพัฒนาโปรแกรม 2 เครื่อง
- เครื่องคอมพิวเตอร์สำหรับเป็นเซิร์ฟเวอร์
- Pocket PC
- Wireless LAN
- Wireless Access Point

1.5.2 ซอฟต์แวร์

- Active Server Page (ASP) .Net
- Microsoft Access 2003
- Microsoft Windows Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐาน

2.1 การพัฒนาระบบเชิงวัตถุ(Object Oriented)

การพัฒนาระบบซอฟต์แวร์หรือระบบสารสนเทศในอดีตนั้นจะใช้วิธีการพัฒนาแบบโครงสร้าง(Structural) ซึ่งมีข้อดีเหนือกว่าการพัฒนาระบบโดยไม่ใช้โมเดล(Model) แต่ในปัจจุบันพบว่า การพัฒนาโครงสร้างนั้นมีปัญหาคือ โมเดลในการวิเคราะห์และออกแบบระบบจะไม่เชื่อมต่อไปถึงรายละเอียดในการพัฒนาระบบ ตัวอย่างเช่น เมื่อผู้พัฒนาระบบวิเคราะห์และออกแบบระบบได้เป็นแผนภาพคอนเท็กซ์(Context Diagram) แผนภาพกระแสข้อมูล(Data Flow) หรือแม้แต่แผนภาพแสดงความสัมพันธ์ขององค์ประกอบต่างๆเพื่อสร้างฐานข้อมูลเชิงสัมพันธ์(Entity Relationship) เองแผนภาพเหล่านี้เพียงบอกแต่ระบบจะมีลักษณะการทำงานอย่างไร แต่ไม่ได้บอกวิธีในการพัฒนาระบบหรือรูปแบบในการเขียนโปรแกรมเลย ทำให้ระบบที่พัฒนาขึ้นมีปัญหาเนื่องจากว่าการตรวจสอบความถูกต้องของระบบ(Validation) ในเชิงพฤติกรรม(Behavior) นั้นทำได้ยากและไม่มีโมเดลใดๆที่รองรับ จึงได้มีผู้คิดค้นและพัฒนาวิธีการหรือโมเดลในการพัฒนาระบบเชิงวัตถุขึ้นมา

การพัฒนาระบบเชิงวัตถุ เป็นแนวคิดในการออกแบบระบบต่างๆ โดยมองระบบต่างๆว่าเกิดจากการประกอบกันของวัตถุ(Object) ซึ่งหมายถึงสิ่งต่างๆที่อยู่รอบตัวเรา ออบเจ็กต์หนึ่งอาจจะประกอบด้วยออบเจ็กต์อื่นๆหลายออบเจ็กต์ ขนาดที่เล็กที่สุดหรือใหญ่ที่สุดของออบเจ็กต์จะขึ้นอยู่กับขอบเขตของระบบที่เรากำลังสนใจอยู่ ว่าต้องการรายละเอียดในระดับใด

ในการกำหนดออบเจ็กต์จะต้องกำหนดคลาสขึ้นมาก่อน คลาสคือรูปแบบทั่วไปของออบเจ็กต์หรือเป็นพิมพ์เขียวของออบเจ็กต์ และออบเจ็กต์คืออินแสตนซ์(Instant) ของคลาส เช่น Mr.John เป็นอินแสตนซ์ของคลาส Person

ออบเจ็กต์จะมีโครงสร้างที่ประกอบด้วยแอททริบิวท์ และพฤติกรรม เช่นในคลาส Person จะมีแอททริบิวท์ คือ height, weight และ age และมีพฤติกรรมคือ eat, sleep, read, write เป็นต้น ซึ่งแอททริบิวท์และพฤติกรรมดังกล่าวจะมีอะไรบ้างนั้น ขึ้นอยู่กับระบบที่ต้องการออกแบบ เช่น Person ในระบบ Human Resource ก็จะมีรายละเอียดอื่นๆที่เกี่ยวข้องเพิ่มขึ้น

นอกจากนั้นการวิเคราะห์และออกแบบระบบเชิงวัตถุจะมีโมเดลที่รองรับ ซึ่งโมเดลที่เป็นมาตรฐานอยู่ในขณะนี้มีชื่อว่า UML (Unified Modeling Language) โดยเป็นกระบวนความคิด (Methodology) และโมเดลที่สนับสนุนการการวิเคราะห์และออกแบบเชิงวัตถุที่สามารถเข้าใจได้ง่ายและสามารถใช้ประกอบในขั้นตอนการลงมือเขียนโปรแกรมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การออกแบบระบบเชิงวัตถุโดยใช้ยูเอ็มแอล (UML)

เนื่องจากการวิเคราะห์และออกแบบระบบ คือ การทำให้นักพัฒนาและผู้ใช้มีความเข้าใจตรงกันในระบบที่จะสร้าง UML (Unified Modeling Language) ซึ่งเป็นภาษากลางในการวิเคราะห์และออกแบบระบบเชิงวัตถุ โดยเตรียมแผนภาพ (Diagram) กฎเกณฑ์และวิธีการ ที่ช่วยให้นักพัฒนาสามารถวิเคราะห์ข้อมูลต่างๆ คือ

1. วิเคราะห์ความต้องการของผู้ใช้
2. วิเคราะห์ขั้นตอนการทำงานของระบบ
3. วิเคราะห์วัตถุและความสัมพันธ์ของวัตถุที่มีอยู่ในระบบ

2.2.1 ความหมายของวัตถุ

วัตถุ หมายถึง สิ่งต่างๆที่เป็นทั้งรูปธรรมและนามธรรมที่รู้ว่ามีอยู่ การทำงานของระบบใดๆ ที่มีอยู่ในโลกเกิดจากการกระทำร่วมกันของวัตถุ

จากความหมายดังกล่าว ถือเป็นหลักการของการออกแบบระบบเชิงวัตถุ ที่พยายามจำลองระบบต่างๆในโลกของความจริงให้อยู่ในระบบคอมพิวเตอร์ วัตถุต่างๆจะถูกแทนที่ด้วยคุณสมบัติ (Attribute) และการกระทำ (Operation) ที่อยู่ร่วมกัน ระบบจะถูกแทนที่ด้วยความสัมพันธ์ระหว่างวัตถุที่ได้สร้างขึ้นในคอมพิวเตอร์ จากมุมมองดังกล่าวทำให้นักพัฒนามีมุมมองในการพัฒนาซอฟต์แวร์และระบบสารสนเทศได้ง่ายขึ้นและสะท้อนภาพเดียวกับระบบในโลกของความจริง

เนื่องจากเราไม่สามารถนำวัตถุมาใส่ในซอฟต์แวร์ได้ดังนั้นจึงต้องมีการสร้างแบบจำลองของวัตถุ ซึ่งวัตถุใดๆสามารถแทนได้ด้วยคุณสมบัติและการกระทำ

- คุณสมบัติ คือ ข้อมูลของวัตถุที่จำเป็นสำหรับการทำงานของระบบและสามารถทำให้วัตถุทำงานได้ถูกต้อง
- การกระทำ คือ สิ่งที่วัตถุสามารถกระทำได้และถูกกระทำได้

2.2.2 การใช้ยูเอ็มแอลในการวิเคราะห์ขั้นตอนการทำงานของระบบ

หลังจากที่นักพัฒนาได้มีการเก็บข้อมูลจากผู้ใช้ก็จำเป็นต้องมีการบันทึกเพื่อใช้ในการอ้างอิงซึ่งยูเอ็มแอลได้เตรียมวิธีไว้สำหรับช่วยให้นักพัฒนาบันทึกขั้นตอนการทำงานของระบบได้อย่างเป็นมาตรฐาน สามารถเข้าใจได้ง่ายและมีประสิทธิภาพ วิธีการที่ใช้อธิบายขั้นตอนการทำงานของระบบยูเอ็มแอลมีด้วยกันหลายวิธี ได้แก่

- คลาสไดอะแกรม(Class Diagram)
- ยูสเคสไดอะแกรม(Use Case Diagram)
- แอคติวิตี้ไดอะแกรม(Activity Diagram)
- สเตทไดอะแกรม(State Diagram)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ซีเควนซ์ไดอะแกรม(Sequence Diagram)

- คอลเลบอเรชันไดอะแกรม(Collaboration Diagram)

ในที่นี้จะขอกล่าวถึงรายละเอียดขั้นตอนการวิเคราะห์ระบบโดยใช้ คลาสไดอะแกรม, ยูสเคสไดอะแกรม และแอคติวิตี้ไดอะแกรม

2.2.2.1 คลาสไดอะแกรม (Class Diagram)

คลาส (Class) ย่อมาจาก คลาสซิฟิเคชัน (Classification) หมายถึงการจำแนกแยกแยะการจัดกลุ่มจัดประเภท หน้าที่ของคลาสเปรียบเสมือนเป็นพิมพ์เขียวของวัตถุหรืออาจกล่าวได้อีกนัยหนึ่งว่า คลาส คือ ชนิดของวัตถุ

การสร้างวัตถุในโปรแกรมจำเป็นต้องสร้างมาจากคลาสทุกครั้ง ซึ่งคลาสนั้นจะประกอบไปด้วย คุณสมบัติและการกระทำของวัตถุ วัตถุที่สร้างมาจากคลาสใด ก็จะมีคุณสมบัติและการกระทำของคลาสนั้นทุกประการ

คุณสมบัติของคลาส

การกำหนดออบเจ็กต์ (Abstraction)

หมายถึงการกำหนดออบเจ็กต์ โดยไม่ได้ระบุรายละเอียดของแอททริบิวท์และพฤติกรรมหรือระบุเป็นส่วนน้อยเท่าที่สามารถทำได้ ซึ่งทำให้การออกแบบมีความยืดหยุ่นมากขึ้น และไม่เกิดการกำหนดออบเจ็กต์เดียวกันมากกว่า 1 ออบเจ็กต์ ผู้ใช้ในแต่ละฟังชั่นสามารถกำหนดแอททริบิวท์ และ พฤติกรรมเพิ่มเติมให้ออบเจ็กต์ได้ ตามความจำเป็นที่ต้องมีในฟังชั่นงานนั้น

การสืบทอด (Inheritance)

เป็นวิธีการสร้างคลาสใหม่จากคลาสเดิมที่มีอยู่ หรือนำคลาสที่มีอยู่แล้วมาเป็นต้นแบบหรือพูดอีกอย่างหนึ่งได้ว่าสืบทอดจากคลาสเดิม ซึ่งทำให้เราไม่ต้องเสียเวลาสร้างคลาสใหม่ทั้งหมด การทำการสืบทอด จะทำให้คลาสที่ได้รับการสืบทอดนี้มีคุณสมบัติต่างๆเหมือนกับคลาสที่ถูกทำการสืบทอดทุกประการ ทั้งนี้คลาสที่สร้างขึ้นใหม่จะมีวัตถุประสงค์การทำงานที่เฉพาะเจาะจงมากยิ่งขึ้น โดยการที่ออบเจ็กต์มีคุณสมบัติทั้งหมดของคลาสที่ออบเจ็กต์นั้นอินสแตนซ์อยู่ คลาสที่สืบทอดจะเรียกว่า ซับคลาส(Subclass) ส่วนคลาสที่ถูกสืบทอดจะเรียกว่า ซุปเปอร์คลาส (Superclass) ซึ่งซับคลาสจะมีคุณสมบัติเหมือนกับซุปเปอร์คลาสทุกประการโดยไม่ต้องกำหนดซ้ำ

การพ้องรูป (Polymorphism)

คุณสมบัติการเปลี่ยนรูป หมายถึงการเปลี่ยนรูปร่างของออบเจ็กต์หนึ่งๆซึ่งในเชิงโปรแกรมมิ่งจะเป็นการที่ตัวแปรออบเจ็กต์ของคลาสใดคลาสหนึ่งสามารถเปลี่ยนรูปไปจากคลาสเดิมได้ โดยคลาสที่มีการสืบทอดแต่โอเปอร์เรชั่นในซับคลาส มีการทำงานที่แตกต่างกันกับโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปอร์เรชันเดียวกันในซุปเปอร์คลาส เช่น ถ้าคลาส Polygon มีโอเปอร์เรชัน draw และ คลาสสืบทอดจากคลาส Polygon มีโอเปอร์เรชัน draw เช่นเดียวกัน แต่การ draw ของ Polygon กับของ Rectangle มีการทำงานคนละอย่างกัน ดังนั้นจึงต้องมีการกำหนดโอเปอร์ชัน draw ของคลาส Rectangle ใหม่

การปกป้อง (Encapsulation)

คุณสมบัติการปกป้อง หรือเรียกอีกอย่างหนึ่งว่า การซ่อนข้อมูล (Information hiding) คือ การจัดกลุ่มองค์ความคิดที่คล้ายคลึงกันเข้าเป็นหน่วยเดียวกันเพื่อทำการอ้างถึงด้วยชื่อเดียวกัน เป็นการซ่อนข้อมูลหรือกระบวนการในการทำโอเปอร์เรชันจากผู้ใช้คลาสในการออกแบบเชิงออบเจ็กต์ สามารถทำได้ใน 3 ระดับคือ

- Private เป็นการกำหนดให้ข้อมูลหรือโอเปอร์ชันนั้น เข้าถึงได้จากภายในคลาสเดียวกันเท่านั้นโดยสามารถทำได้ทั้งการอ่านและเขียน

- Protected เป็นการกำหนดข้อมูลโดยโอเปอร์เรชันนั้นเข้าถึงได้จากภายในคลาสเดียวกันหรือคลาสอื่น โดยถ้าเป็นคลาสเดียวกันจะสามารถทำได้ทั้งการอ่านและเขียน แต่ถ้าเป็นคลาสอื่นจะทำได้เพียงการอ่านอย่างเดียว

- Public เป็นการกำหนดให้ข้อมูลหรือโอเปอร์เรชันนั้น เข้าถึงได้จากภายในคลาสเดียวกันและคลาสอื่นโดยสามารถทำได้ทั้งการอ่านและเขียน

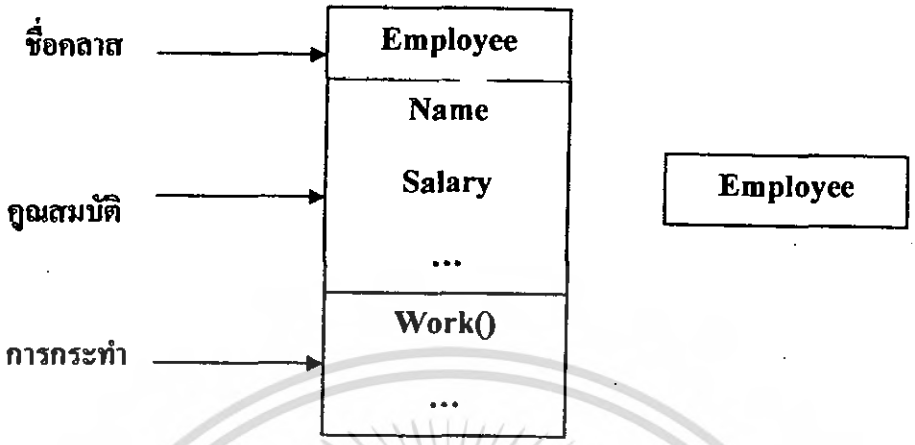
สัญลักษณ์ที่ใช้ในคลาสไดอะแกรม

สัญลักษณ์ที่ใช้ในคลาสไดอะแกรมมีอยู่ 2 ประเภท คือ สัญลักษณ์แทนคลาสและสัญลักษณ์แสดงความสัมพันธ์ระหว่างคลาส

สัญลักษณ์ของคลาส

คลาสในยูเอ็มแอลจะแสดงเป็นรูปสี่เหลี่ยม แบ่งออกเป็น 3 ส่วนคือ ชื่อคลาส, คุณสมบัติ, และการกระทำ ในกรณีที่ต้องการคลาสไดอะแกรมในโมเดลต่างๆ เช่น โดเมน (Domain Model) โดยไม่ระบุรายละเอียดของคลาส สามารถทำได้โดยการเขียนเฉพาะชื่อคลาสเท่านั้น เรียกว่า คลาสไอคอน (Class icon) ดังแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



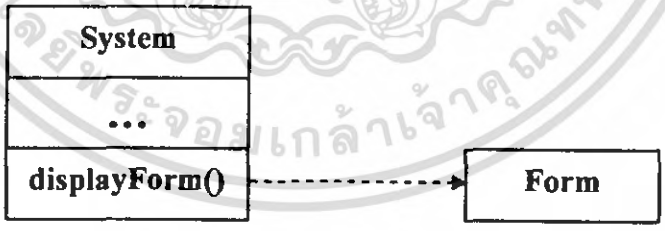
รูปที่ 2.1 สัญลักษณ์ของคลาสและคลาสไอคอน

สัญลักษณ์แสดงความสัมพันธ์ระหว่างคลาส

คลาสไดอะแกรม ประกอบด้วย คลาสและความสัมพันธ์ระหว่างคลาส ซึ่งความสัมพันธ์เหล่านี้สามารถแบ่งได้เป็น 4 รูปแบบใหญ่ๆ ดังนี้ คือ

ดีเพนเดนซี่ (Dependency)

ดีเพนเดนซี่มีลักษณะที่การกระทำของคลาสหนึ่ง ใช้อีกคลาสหนึ่งเป็นพารามิเตอร์ เช่น ในการแสดงผลคลาส ระบบทำหน้าที่ในการแสดงฟอร์ม โดยใช้ฟอร์มแสดงการกระทำ (Operation Display Form) โดยฟอร์มที่แสดงจะขึ้นอยู่กับฟอร์มที่ผู้ใช้เลือกซึ่งสามารถเขียนในคลาสไดอะแกรมได้ดังนี้



รูปที่ 2.2 ความสัมพันธ์แบบดีเพนเดนซี

แอสโซซิเอชัน (Associations)

เป็นความสัมพันธ์แบบสองทิศทางซึ่งถูกวาดด้วยเส้นตรงทึบเชื่อมระหว่างสองคลาสและมีชื่อความสัมพันธ์กำกับอยู่ โดยมักจะเป็นคำกริยา นอกจากนี้อาจมีการกำหนดทิศทางของชื่อความสัมพันธ์ โดยการวาดสามเหลี่ยมทึบไว้ด้านซ้ายหรือด้านขวาของชื่อความสัมพันธ์ โดยขึ้นอยู่กับทิศทางของความสัมพันธ์ ซึ่งลูกศรจะช่วยให้การอ่านความสัมพันธ์เป็นไปอย่างถูกต้องนอกจากนี้

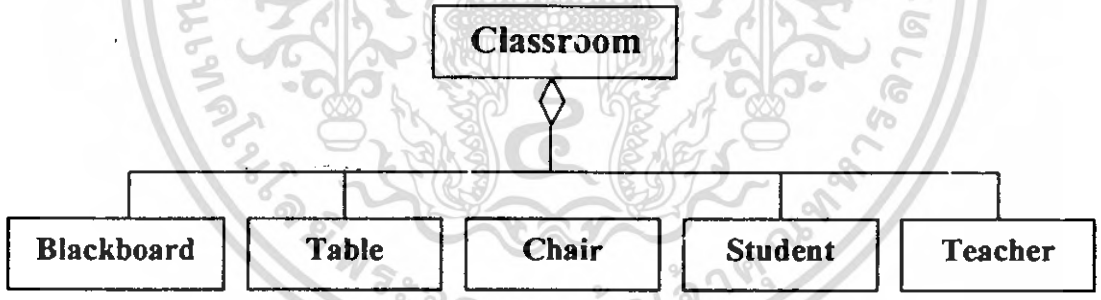
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังมีการกำหนดปริมาณของคลาสหรือวัตถุที่สัมพันธ์กันอยู่ เรียกว่า มัลติพลิซิตี้ (Multiplcity) ซึ่งสามารถทำการกำหนดได้หลายรูปแบบคลาสหนึ่งสามารถมีความสัมพันธ์กับคลาสอื่นแบบหนึ่งต่อหนึ่ง, หนึ่งต่อกลุ่ม, หนึ่งต่อหนึ่งหรือมากกว่า เป็นต้น โดยเครื่องหมาย (*) จะหมายถึง หลายๆ หรือถ้าต้องการกำหนดเป็นช่วงคือ 1..10

อะกรีเกชันและคอมโพสิท (Aggregation and Composites)

เป็นรูปแบบความสัมพันธ์อีกแบบหนึ่ง โดยออปเจ็กต์หนึ่งประกอบขึ้นจากออปเจ็กต์อื่นๆ ตั้งแต่ 1 ออปเจ็กต์ขึ้นไป เช่นในออปเจ็กต์ computer A มีออปเจ็กต์ Hard drive 2 ออปเจ็กต์เป็นต้น ออปเจ็กต์ Computer เรียกว่า Aggregate Object และออปเจ็กต์ที่เป็นองค์ประกอบจะเรียกว่า Component Object

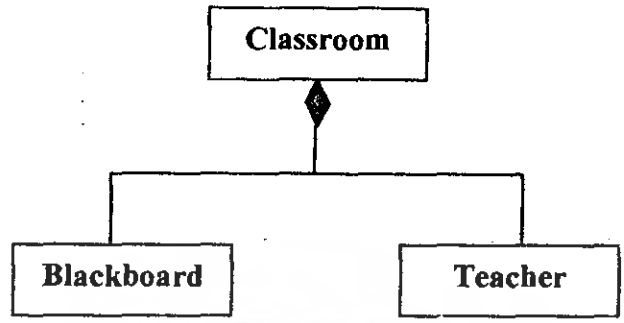
อีกรูปแบบหนึ่งของ Aggregation ที่เป็นความสัมพันธ์แบบความสัมพันธ์แบบแข็ง (Strong relationship) คือ Composition โดย Componant Object จะต้องมีใน Composite Object เช่นออปเจ็กต์คอมพิวเตอร์จะต้องมี ซีพียู (CPU) จึงเป็น Composition แต่ออปเจ็กต์คอมพิวเตอร์อาจจะไม่มีเครื่องพิมพ์ก็ได้ ทำให้ความสัมพันธ์ระหว่าง คอมพิวเตอร์ กับ เครื่องพิมพ์เป็นความสัมพันธ์แบบ Aggregation แต่ไม่เป็น Composition



รูปที่ 2.3 ตัวอย่างความสัมพันธ์แบบอะกรีเกชัน

คอมโพสิท คือ สตรองไทพ์ (Strong type) ของอะกรีเกชัน แต่ละส่วนประกอบของคลาสอื่นๆจะต้องมีในคลาส (ในขณะที่อะกรีเกชันคลาสอาจจะมีหรือไม่มีส่วนประกอบของคลาสอื่นๆก็ได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ตัวอย่างความสัมพันธ์แบบคอมโพสิท

2.2.2.2 ยูสเคสไดอะแกรม (Use Case Diagram)

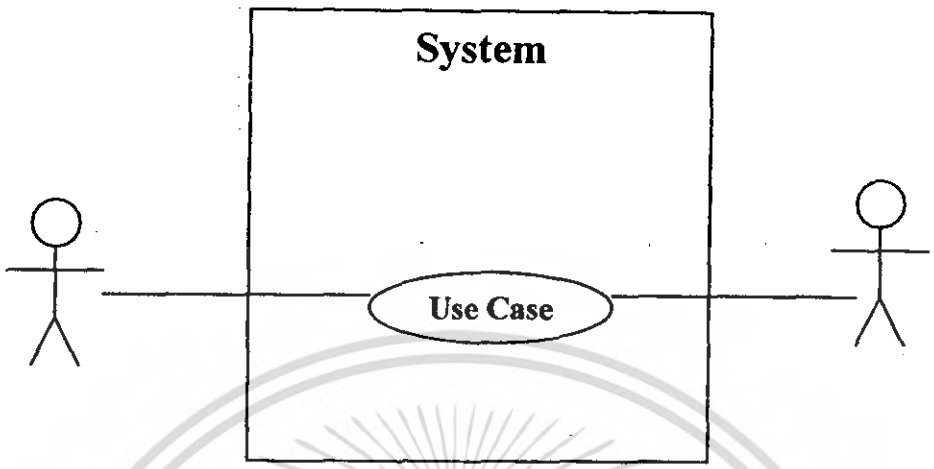
ยูสเคสไดอะแกรมใช้ในการแสดงลำดับของแต่ละเหตุการณ์ ว่ามีส่วนประกอบอะไรบ้าง และมีความเกี่ยวพันธ์กันอย่างไร ซึ่งจะช่วยให้ผู้พัฒนาระบบสามารถแยกแยะได้ว่ามีกิจกรรมใดที่น่าจะเกิดขึ้นในระบบได้บ้าง โดยยูสเคสเป็นไดอะแกรมที่ใช้รูปภาพในการอธิบายทำให้ผู้เขียนยูสเคสไดอะแกรมไม่จำเป็นต้องมีความรู้ทางคอมพิวเตอร์อย่างลึกซึ้งแต่ทางกลับกันยูสเคสจะออกมาอย่างมีประสิทธิภาพถ้าผู้เขียนมีความรู้ความเข้าใจในเรื่องโดเมน ของปัญหา แต่ละลำดับของเหตุการณ์ในยูสเคส จะเริ่มต้นโดยบุคคลหรือระบบ เรียกว่าผู้กระทำ (Actor) ในยูสเคสผลที่ได้จากลำดับจะเป็นสิ่งที่ใช้โดย ผู้กระทำที่เริ่มเหตุการณ์หรือผู้กระทำอื่นๆ

ยูสเคสจะสร้างจากมุมมององผู้ใช้ โดยจะมีลักษณะเหมือนกับสิ่งที่เกิดขึ้นจริง ซึ่งต่างจากการออกแบบระบบอื่น เช่น ดาต้าโฟลวไดอะแกรม (Dataflow Diagram) ที่ผู้ออกแบบระบบจะต้องหาอินพุท(Input) ของระบบในทางคอมพิวเตอร์ ซึ่งบางครั้งผู้ใช้ไม่สามารถระบุถึงได้อย่างชัดเจนครบถ้วน

สัญลักษณ์ที่ใช้ในยูสเคสไดอะแกรม

ในการเขียนยูสเคสไดอะแกรมเราใช้รูปวงรีในการแสดงยูสเคส สัญลักษณ์รูปคนแทนผู้กระทำ และใช้เส้นตรงเชื่อมระหว่างยูสเคสและผู้กระทำ เพื่อแสดงการใช้งานยูสเคสของผู้กระทำ โดยยูสเคสที่เริ่มต้นเหตุการณ์จะอยู่ด้านซ้ายของยูสเคส และผู้กระทำที่ได้รับผลจากยูสเคสจะอยู่ด้านขวา ชื่อของผู้กระทำจะอยู่ด้านล่างของสัญลักษณ์รูปคน ส่วนชื่อของยูสเคสจะอยู่ภายในวงรีหรืออยู่ใต้วงรี นอกจากนั้นยูสเคสทุกๆตัวจะต้องอยู่ภายในกรอบสี่เหลี่ยมเดียวกันซึ่งมีชื่อของระบบระบุอยู่เพื่อแสดงความเป็นยูสเคสของปัญหาโดเมนเดียวกัน ดังตัวอย่างในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 สัญลักษณ์การเขียนยูสเคสไดอะแกรม

การหาผู้กระทำของระบบ

ผู้กระทำ คือ สิ่งที่อยู่ภายนอกระบบและมีการติดต่อโดยตรงกับระบบ อาจอยู่ในรูปของการเข้ามาใช้ระบบ, ได้รับข้อมูลจากระบบ, หรือถูกระบบกระทำบางสิ่งบางอย่าง เช่น ในระบบยืมคืนหนังสือในห้องสมุด ผู้กะทำของระบบคือ บรรณารักษ์ ผู้ที่ทำหน้าที่ปรับปรุงข้อมูลของหนังสือและผู้ใช้บริการห้องสมุดซึ่งต้องกเรียมหรือคืนหนังสือ

ในการหาผู้กระทำนั้นจะต้องใช้ตัวบทบาทเป็นตัวกำหนดผู้กระทำ ยกตัวอย่างเช่น ในระบบจัดการคลังสินค้ามีพนักงานจัดสินค้า ชื่อนาย A และนาย A ก็เป็นพนักงานเช็คสต็อกด้วย ผู้กระทำของระบบนี้ไม่ใช่นาย A หากแต่เป็นพนักงานจัดสินค้า และพนักงานเช็คสต็อกด้วยการหายูสเคสของผู้กระทำแต่ละตัว ก่อนอื่นเราต้องทราบว่าผู้กระทำ แต่ละตัวจะใช้ระบบเมื่อต้องการทำงานอะไรและระบบจะติดต่อกับผู้กระทำภายนอก เมื่อต้องทำงานอะไรสิ่งที่พึงระวังคือ งานที่ตอบสนองความต้องการของผู้กระทำ ไม่ใช่งานที่ผู้กระทำต้องการมีกระบวนการ (Process)อะไรบ้าง

กล่าวโดยสรุป ข้อดีของการประยุกต์ใช้แนวคิดเชิงวัตถุในการพัฒนาระบบซอฟต์แวร์มีดังต่อไปนี้

- สนับสนุนการพัฒนาระบบที่ซับซ้อน (Problem Decomposition)

ในการพัฒนาโปรแกรมรูปแบบเดิม นักพัฒนาโปรแกรมจะทำงานกันที่ระดับฟังก์ชัน กล่าวคือตั้งแต่เริ่มการพัฒนาจนจบ จะเป็นการสร้างปรับปรุงแก้ไขรายละเอียดของฟังก์ชันต่างภายในตัวโปรแกรม แต่สำหรับแนวคิดเชิงวัตถุ นักพัฒนาจะทำการวิเคราะห์ออกแบบระบบกันใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับของออปเจ็กต์ซึ่งประกอบด้วยทั้งข้อมูลและฟังก์ชันภายในแต่ละออปเจ็กต์ ดังนั้นจึงกล่าวได้ว่าการอาศัยแนวคิดเชิงวัตถุจะช่วยให้นักพัฒนาสามารถสร้างโปรแกรมได้ง่ายขึ้นมาก

- สนับสนุนการนำกลับมาใช้งานซ้ำอีกครั้ง(Promotion of Reusability)

เนื่องจากแต่ละคลาสหรือออบเจ็กต์ที่กำหนดขึ้นนั้นจะมีความสมบูรณ์อยู่ในตัวของมันเองบนพื้นฐานแนวคิดของแต่ละออบเจ็กต์เอง รวมทั้งยังเป็นอิสระจากสภาพแวดล้อมอื่น ดังนั้นแต่ละคลาสจึงง่ายต่อการนำกลับมาใช้งานปรับปรุงเพิ่มเติม การนำกลับมาใช้งานอาจอยู่ในรูปแบบของการสืบทอดคุณสมบัติระหว่างออบเจ็กต์จากคลาสปุ่มเพื่อไปใส่ไว้ในโปรแกรมต่างๆ ได้ตามต้องการ นั่นคือคลาสปุ่มสามารถนำกลับมาใช้งานได้

- ปรับปรุงแก้ไขและบำรุงรักษาง่าย (Easy for Change and Extensibility)

เนื่องจากข้อมูลและฟังก์ชั่นการทำงานที่เกี่ยวกับออบเจ็กต์หนึ่งๆ จะถูกรวบรวมอยู่ที่เดียวกัน การทำงานภายในของแต่ละออบเจ็กต์จะไม่เกี่ยวข้องพึ่งพิงกับโค้ดที่อยู่ภายนอกออบเจ็กต์ ดังนั้นนักพัฒนาสามารถทำการแก้ไขปรับปรุงรายละเอียดของแต่ละคลาสได้โดยไม่กระทบต่อส่วนที่เรียกใช้งานภายนอกแต่อย่างใด นอกจากนี้ในการขยายระบบก็สามารถทำได้โดยง่าย โดยการสร้างออบเจ็กต์หรือคลาสเพิ่มเติมลงไปในตัวโปรแกรมนั่นเอง

2.3 ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ

วัตถุประสงค์หลักของทุกกระบวนการพัฒนาซอฟต์แวร์คือ การแปลงความต้องการของผู้ใช้ให้เป็นระบบที่มีคุณภาพและสามารถใช้งานได้จริง อีกทั้งยังช่วยลดระยะเวลาที่ต้องทำการเขียนโปรแกรมและแก้ไขข้อผิดพลาดให้น้อยลงอีกด้วย ซึ่งในระหว่างการพัฒนาจะมีการใช้เครื่องมือช่วยต่างๆ หลายอย่าง ในที่นี้ได้แก่ภาษามาตรฐานยูเอ็มแอล

กระบวนการที่จะอธิบายต่อไปนี้เป็นวิธีการทั่วไปในการพัฒนาซอฟต์แวร์เชิงวัตถุที่ประกอบไปด้วย 5 ขั้นตอน โดยในแต่ละขั้นตอนจะถูกเรียกว่า เฟส(Phase) ซึ่งผลลัพธ์ที่ได้จากการพัฒนาในแต่ละเฟสจะต้องถูกนำไปใช้ในการพัฒนาต่อในเฟสถัดไป และมีเพียงเฟสแรกที่จะถูกดำเนินการในช่วงเริ่มต้นของการพัฒนาครั้งเดียวและในสี่เฟสที่เหลือจะถูกทำซ้ำ (Iteration) เพื่อทำการขยายต่อเติมฟังก์ชั่นของระบบจนกว่าจะได้ระบบที่สมบูรณ์ นั่นคือในแต่ละรอบของการวนซ้ำสิ่งที่ได้จะเป็นเพียงต้นแบบ (Prototype) อันประกอบไปด้วยฟังก์ชั่นบางอย่างสอดคล้องกับความต้องการของผู้ใช้และสิ่งนี้เองที่ฝ่ายพัฒนาควรทำการส่งมอบให้ผู้ใช้พิจารณาเป็นระยะๆ ในช่วงของการพัฒนา และในความเป็นจริงสิ่งที่เกิดขึ้นตามมาซึ่งมักหลีกเลี่ยงไม่ได้คือความจำเป็นในการดำเนินการเฟสแรกมากกว่าหนึ่งครั้งอันเนื่องมาจากการเปลี่ยนแปลงความต้องการของผู้ใช้นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 การวิเคราะห์ความต้องการของระบบหรือผู้ใช้(Requirement Analysis)

โครงการที่ถือว่าประสบความสำเร็จจะต้องตอบสนองต่อความต้องการและความคาดหวังของผู้ใช้ได้อย่างครบถ้วนหรือมากกว่าอันเป็นจุดสำคัญของทุกกระบวนการพัฒนาซอฟต์แวร์ การวิเคราะห์ความต้องการของผู้ใช้เป็นการค้นหาขอบข่ายของระบบและเป็นการเตรียมข้อมูลด้านความสามารถของระบบ (System Function) จากมุมมองภายนอกที่จะต้องถูกทำการพัฒนาโดยไม่คำนึงถึงรายละเอียดหรือกรรมวิธีทางเทคนิคต่างๆ

ในความเป็นจริงการเริ่มต้นเฟสแรกนี้จะต้องทำในรูปของการกำหนดเป็นข้อตกลง (Agreement) ด้วยการเจรจากันระหว่างผู้ใช้งานกับผู้พัฒนา ซึ่งฝ่ายผู้พัฒนาจะต้องบันทึกความต้องการของผู้ใช้อย่างละเอียดที่สุดเท่าที่จะเป็นไปได้ เรียกขั้นตอนย่อยนี้ว่า User Requirement Elicitation อันหมายถึงการสัมภาษณ์ผู้ใช้งานระบบถึงความต้องการต่างๆ เฟสนี้จำเป็นอย่างมากโดยเฉพาะในการพัฒนาระบบธุรกิจ แต่ไม่จำเป็นสำหรับการสร้างระบบเชิงวิจัยพัฒนาซึ่งมีเป้าหมายที่ชัดเจนอยู่แล้ว สำหรับในส่วนที่ไม่สามารถลงรายละเอียดของความต้องการได้ก็ควรกำหนดเป้าหมายให้ชัดเจนในส่วนนั้นๆ เพื่อป้องกันการเข้าใจที่ไม่ตรงกันระหว่างผู้ใช้และผู้พัฒนา

นอกจากการค้นหาความสามารถที่ต้องมีอยู่ในตัวระบบแล้ว ผู้พัฒนายังจำเป็นต้องทำการสำรวจถึงความต้องการในแง่อื่นๆ ด้วย(Nonfunctional Requirement) ตัวอย่างเช่น ประสิทธิภาพช่วงเวลาในการตอบสนอง ส่วนอินเทอร์เฟส การใช้งานร่วมกับระบบเดิม เป็นต้น อย่างไรก็ตามหากมองในภาพที่กว้างขึ้น จะพบว่าการดำเนินการพัฒนาระบบจะต้องเกี่ยวข้องกับงบประมาณ ระยะเวลาในการส่งมอบผลงานในแต่ละช่วง การทำข้อสัญญา และการบริหารโครงการ เป็นต้น

2.3.2 การวิเคราะห์ระบบ (Domain Analysis)

การวิเคราะห์ระบบ (Domain Analysis) เป็นการวิเคราะห์โครงสร้าง (Structure) และพฤติกรรม (Behavior) ของระบบที่จะทำการพัฒนาซึ่งจะถูกนำไปกำหนดรายละเอียดเชิงเทคนิคในเฟสการออกแบบและจะถูกสร้างจริงเป็นลำดับต่อไปในอิมพลีเมนต์ระบบ กิจกรรมในเฟสนี้จะเป็นการทำความเข้าใจกับระบบกำลังจะพัฒนาในสองด้าน กล่าวคือ

- การวิเคราะห์โครงสร้าง เป็นการทำความเข้าใจกับระบบที่กำลังพัฒนาเพื่อค้นหาคลาสและความสัมพันธ์ระหว่างคลาสในระบบ ซึ่งผลที่ได้เหล่านี้จะถูกนำไปอิมพลีเมนต์สร้างเป็นระบบขึ้นจริงในเฟสถัดๆไป

- การวิเคราะห์พฤติกรรม เนื่องจากออบเจ็กต์ที่เกิดจากคลาสต่างๆในโครงสร้างของระบบจะทำงานร่วมกันเพื่อให้ระบบโดยรวมสามารถตอบสนองต่อการใช้งานของผู้ใช้งานได้ในทุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชั่นโดยกิจกรรมในระหว่างการปฏิบัติงานร่วมกันของแต่ละออบเจ็กต์จำเป็นต้องถูกทำความเข้าใจเพื่อนำไปอิมพลีเมนต์เป็นขั้นตอนการทำงานของแต่ละฟังก์ชั่นของระบบได้

2.3.3 การออกแบบ(Design)

ซึ่งเป็นกระบวนการคิดค้นวิธีแก้ไขปัญหาหรือพิจารณารายละเอียดเชิงเทคนิค เพื่อเตรียมที่จะอิมพลีเมนต์ระบบได้จริง ซึ่งจะเป็นการนำผลการวิเคราะห์จากเฟสที่สองมาทำการแก้ไขเพิ่มเติมรายละเอียดเชิงเทคนิค เพื่อสามารถที่จะถูกนำไปสร้างขึ้นเป็นระบบซอฟแวร์จริงได้อย่างสมบูรณ์ ตัวอย่างกิจกรรมที่เกิดขึ้นในเฟสนี้คือ

- การเลือกใช้ภาษาทางโปรแกรมคอมพิวเตอร์ที่จะใช้ในการพัฒนาระบบ
- การออกแบบฐานข้อมูล โดยเฉพาะในส่วนที่เรียกว่า ระดับแนวคิดซึ่งนิยมใช้ อีอาร์โมเดลเข้ามาช่วย
- การออกแบบสถาปัตยกรรมระบบ จำนวนเครื่องคอมพิวเตอร์
- การออกแบบลักษณะการเชื่อมต่อของเครือข่าย
- การนำกลับมาใช้งานของคลาสไลบลารีหรือซอฟแวร์คอมโพแนนต์ที่มีอยู่
- การออกแบบหน้าตาของตัวโปรแกรม (User Interface)
- การจัดการกับข้อผิดพลาดที่อาจเกิดขึ้นในการใช้งานระบบ เช่นระบบควรมีการจัดการอย่างไรหากผู้ใช้ป้อนข้อมูลผิดประเภท เป็นต้น

ดังนั้นเฟสนี้จำเป็นต้องอาศัยความรู้ทางเทคนิคและเทคโนโลยีต่างๆ ที่มีอยู่เพื่อสามารถเลือกได้อย่างเหมาะสม

2.3.4 การสร้างโปรแกรมระบบ (Construction, Coding, Implementation)

หลังจากที่ระบบได้ถูกออกแบบไว้อย่างสมบูรณ์พร้อมที่จะถูกนำไปอิมพลีเมนต์จริงโดยโปรแกรมเมอร์ ในเฟสนี้สิ่งที่เกิดขึ้นจากการออกแบบทั้งหมดจะถูกแปลงไปสู่ระบบจริง โดยส่วนใหญ่กิจกรรมในเฟสนี้จะเป็นการสร้างโปรแกรมหรือการอิมพลีเมนต์ระบบอันเป็นขั้นตอนของ OOP (Object Oriented Programming) ที่คุ้นเคยกันดี ซึ่งจะต้องอาศัยความรู้ในตัวภาษาโปรแกรมที่ใช้ในการสร้างโค้ด ในขั้นตอนนี้จะถูกดำเนินการโดยทีมโปรแกรมเมอร์ภายหลังจากได้รับข้อมูลการออกแบบที่เพียงพอจากฝ่ายวิเคราะห์ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 การทดสอบระบบ (Testing)

เป็นการทดสอบความถูกต้องของระบบที่พัฒนาเพื่อค้นหาข้อผิดพลาดเชิงเทคนิค และการตรวจสอบความสอดคล้องกับความต้องการที่ถูกระบุอยู่ในความต้องการของผู้ใช้งาน ทั้งนี้การค้นพบข้อผิดพลาดจะถือว่าเป็นความสำเร็จของการดำเนินงานในเฟสนี้มิใช่ความล้มเหลวแต่อย่างใด นอกจากนี้ยังเป็นการประเมินความสมบูรณ์ของระบบว่าจำเป็นต้องทำการวิเคราะห์ออกแบบเพิ่มเติมอีกครั้งหรือไม่ โดยปกติในเฟสนี้จะต้องมีการจัดเตรียมข้อมูลที่จะใช้สำหรับการทดสอบและการประเมินผลลัพธ์เรียกว่า Test Cases จะประกอบไปด้วยกิจกรรมในการทดสอบการป้อนข้อมูลและการทดสอบผลลัพธ์รวมถึงความคาดหวังจากการทำงานของระบบว่าน่าจะเป็นที่พึงพอใจแก่ผู้ใช้งานจริงหรือไม่ ผลของการทดสอบจะถูกบันทึกลงรายงานการทดสอบซึ่งรวมถึงการบรรยายรายละเอียดข้อผิดพลาดที่ปรากฏเพื่อทำการแก้ไขต่อไปโดยใช้โปรแกรมเมอร์

ในเฟสนี้สิ่งที่ทำให้นักพัฒนาปวดหัวอยู่เสมอคือข้อผิดพลาด (Error) ซึ่งแบ่งได้เป็นสองด้าน ได้แก่

-Functional เป็นความผิดพลาดในระดับฟังก์ชั่นการทำงานของระบบ ซึ่งมักเกิดขึ้นบ่อยครั้ง

-Nonfunctional เป็นความผิดพลาดที่ไม่เกี่ยวข้องกับระดับฟังก์ชันการทำงานของระบบที่พัฒนาโดยตรง เช่น การที่ประสิทธิภาพการทำงานของระบบต่ำเกินไป ช่วงเวลาในการตอบสนองนานเกินไป เป็นต้น

โดยปกติความต้องการของผู้ใช้งานมักไม่ถูกค้นพบทั้งหมดในช่วงเริ่มต้นของการสัมภาษณ์ผู้ใช้ จึงเป็นเหตุผลว่าทำไมผู้ใช้งานจึงมักร้องขอการเปลี่ยนแปลง ซึ่งเป็นปัญหาที่เกิดขึ้นกับนักพัฒนาทั้งหลายโดยตลอดมา ข้อเสนอแนะต่อไปนี้อาจช่วยป้องกันปัญหาดังกล่าวได้บ้างไม่มากก็น้อย

-กำหนดรายละเอียดความต้องการให้ชัดเจนถึงสิ่งที่ผู้ใช้ต้องการและไม่ต้องการทั้งนี้ห้ามสมมติเอาเอง เพื่อใช้ในการป้องกันหรือชะลอความเสียหายจากผู้ใช้ที่มีการเปลี่ยนแปลงความต้องการบ่อยๆ

-สิ่งที่ผู้ใช้เกริ่นว่าจะต้องการแต่ไม่ใช่ตอนนี้อาจถูกออกแบบพัฒนาเผื่อไว้โดยปกติ 10% เพื่อลดผลกระทบที่เกิดขึ้นเมื่อเวลาที่ต้องพัฒนาส่วนนั้นจริงๆ มาถึง

-กำหนดข้อตกลงให้ชัดเจนในการส่งมอบงานหรือระบบตัวอย่างแต่ละช่วงเวลาในการพัฒนา

-การร้องขอให้แก้ไขข้อผิดพลาดในระบบที่เพิ่งส่งมอบและตรวจรับเรียบร้อยอาจถูกคิดเป็นค่าบริการพิเศษ ทั้งนี้ต้องกำหนดไว้ในสัญญาอย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจำเป็นต้องมองการเปลี่ยนแปลงความต้องการของผู้ใช้งานเป็นเรื่องปกติธรรมดาที่ผู้พัฒนาจะต้องเผชิญ หากแต่สิ่งที่ควรจะทำคือการหาวิธีรับรองที่จะทำให้ได้รับผลกระทบน้อยที่สุดเมื่อความต้องการการเปลี่ยนแปลงดังกล่าวเกิดขึ้น

2.4 ภาษายูเอ็มแอล

UML (Unified Modeling Language) หมายถึง ภาษาสัญลักษณ์รูปภาพมาตรฐาน (Standard Modeling Language) ที่ใช้กำหนดลักษณะคลาส การสร้างคลาสและเป็นเอกสารที่บอกถึงรายละเอียดของระบบโครงสร้าง โดยที่ยูเอ็มแอลจะสามารถแสดงโครงสร้างของระบบเชิงวัตถุในรายละเอียดลึกๆได้ดี ในรูปแบบของแผนภาพไดอะแกรม แผนภาพเหล่านี้จะทำให้เกิดความเข้าใจที่ตรงกันระหว่างผู้ออกแบบระบบและโปรแกรมเมอร์ ทำให้การปรับปรุงแก้ไขโปรแกรมทำได้ง่ายขึ้น

2.4.1 ความเป็นมาของภาษายูเอ็มแอล

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

เนื่องจากวิธีการพัฒนาซอฟต์แวร์เชิงวัตถุของทั้งสามมีชื่อเสียงอยู่แล้วในช่วงเวลานั้น ดังนั้น ภาษายูเอ็มแอลที่ถูกพัฒนาขึ้นใหม่นี้กลายมาเป็นที่นิยมใช้อย่างแพร่หลายเช่นกัน ในปี 1996 มีหลายบริษัทที่ขอเข้าร่วมในการพัฒนาภาษายูเอ็มแอล ซึ่งได้แก่ บริษัทดิจิตอลอีควิปเมนต์ บริษัทเอชพี บริษัทไอบีเอ็ม รวมทั้งบริษัทออราเคิล บริษัทไมโครซอฟต์ และบริษัทอื่นๆอีกหลายบริษัท

2.4.2 ข้อดีของยูเอ็มแอล

ยูเอ็มแอลมีข้อดีหลายประการดังต่อไปนี้

- เป็นภาษารูปแบบมาตรฐานหรือภาษาสากลที่ใช้ในการพัฒนาซอฟต์แวร์วัตถุ และสามารถใช้ในการแลกเปลี่ยนโมเดลได้อย่างสื่อความหมาย รวมถึงการจัดสร้างเอกสารการวิเคราะห์ออกแบบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถนำเสนอและสนับสนุนหลักการเชิงวัตถุอย่างครบถ้วนชัดเจน ทำให้นักพัฒนาระบบสามารถทำความเข้าใจกับปัญหาและค้นพบวิธีแก้ไขได้อย่างรวดเร็วและง่ายยิ่งขึ้น

- ไม่ผูกติดกับภาษาโปรแกรมใดโปรแกรมหนึ่ง กล่าวคือโมเดลที่ถูกสร้างขึ้นจากภาษามาตรฐานซึ่งสามารถแปลงไปเป็นระบบจริงที่ถูกสร้างขึ้นด้วยภาษาโปแกรมเชิงวัตถุใดๆก็ได้

- เป็นภาที่ง่ายต่อการทำความเข้าใจ ผู้ที่ทำการศึกษาหรือนำไปใช้งานไม่จำเป็นต้องมีความรู้อื่นใดนอกจากแนวคิดเชิงวัตถุ

- สามารถถูกแปลงเป็นภาษาที่ใช้ในการสร้างระบบขึ้นได้จริงได้อย่างอัตโนมัติ จึงเป็นการช่วยลดภาระ เวลา และค่าใช้จ่ายการพัฒนาระบบได้อย่างมาก

- ยูเอ็มแอลใช้ในการบันทึกความคิดของนักพัฒนาในลักษณะของเอกสารที่พร้อมจะถูกนำมาทำความเข้าใจหรือสานต่ออีกครั้งได้อย่างรวดเร็ว

2.4.3 องค์ประกอบของยูเอ็มแอล

ยูเอ็มแอลเป็นภาษาการสร้างโมเดลในแบบกราฟฟิก ซึ่งใช้ในการประกอบกันเป็นไดอะแกรม โดยมีกฎในการประกอบกันของอีลีเมนต์ต่างๆ ไดอะแกรมจะแสดงถึงมุมมองต่างๆ (Multiple view) ของระบบซึ่งจะเรียกว่าโมเดล โมดูลยูเอ็มแอลจะบ่งบอกถึงรายละเอียดของระบบแต่จะไม่ระบุถึงวิธีการในการพัฒนาระบบ

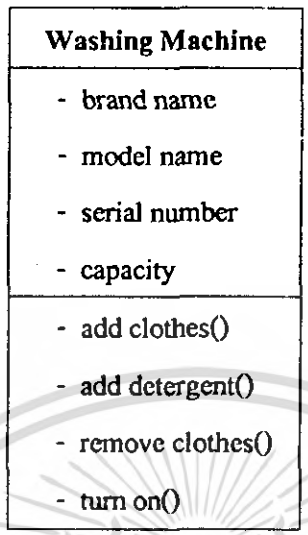
คลาสไดอะแกรม(Class Diagram)

คลาสไดอะแกรมเป็นแผนภาพที่ใช้แสดงคลาส และความสัมพันธ์ในแง่ต่างๆระหว่างคลาสเหล่านั้น ซึ่งความสัมพันธ์ที่กล่าวถึงในคลาสไดอะแกรมนี้ ถือเป็นความสัมพันธ์เชิงสถิต (Static Relationship) หมายถึงความสัมพันธ์ที่มีอยู่แล้วในคลาสต่างๆ ซึ่งเรียกว่าความสัมพันธ์เชิงกิจกรรม (Dynamic Relationship) สิ่งที่จะปรากฏในคลาสไดอะแกรมนั้นประกอบไปด้วยกลุ่มของคลาสและกลุ่มของความสัมพันธ์ โดยใช้สัญลักษณ์ที่ใช้ในการแสดงคลาสนั้นๆ แทนด้วยสี่เหลี่ยมที่แบ่งออกเป็น 3 ส่วนโดยส่วนบนสุดจะเป็นชื่อคลาส ส่วนกลางจะแสดงแอททริบิวท์ของคลาส และส่วนล่างจะแสดงโอเปอร์เรชั่นของคลาส

ตัวอย่างเช่น คลาส Washing Machine มีแอททริบิวท์คือ brand name, model, serial number และ capacity มีโอเปอร์เรชั่นคือ add clothes, add detergent, turn on และ remove clothes สามารถเขียนให้อยู่ในรูปแบบยูเอ็มแอล คลาสไดอะแกรมได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

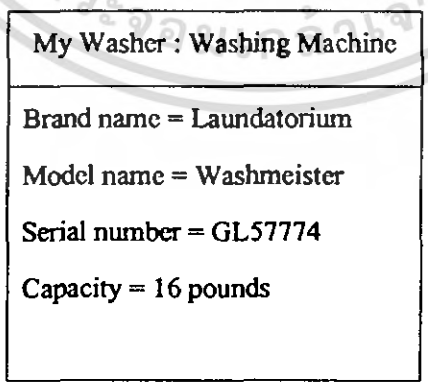
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.6 คลาสไดอะแกรมของระบบ Washing Machine

ออบเจ็กต์ไดอะแกรม (Object Diagram)

ออบเจ็กต์ คือ อินสแตนซ์ของคลาส การกำหนดรายละเอียดของออบเจ็กต์ คือการกำหนดค่าให้กับแอททริบิวต์ ออบเจ็กไดอะแกรมของยูเอ็มแอล จะแสดงเป็นรูปสี่เหลี่ยม โดยชื่อจะประกอบด้วยชื่อออบเจ็กต์และชื่อคลาส คั่นระหว่างชื่อด้วยเครื่องหมาย ":" และขีดเส้นใต้ที่ชื่อด้วย ตัวอย่างเช่น ออบเจ็กต์ Washing Machine ชื่อ My Washer มี brand name คือ laundatoriam, model คือ Washmeister, serial number คือ GL57774 และ capacity คือ 16 ปอนด์ สามารถแสดงได้ดังรูป



รูปที่ 2.7 ออบเจ็กต์ไดอะแกรมของระบบ washing machine

73016
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยูสเคสไดอะแกรม (Use Case Diagram)

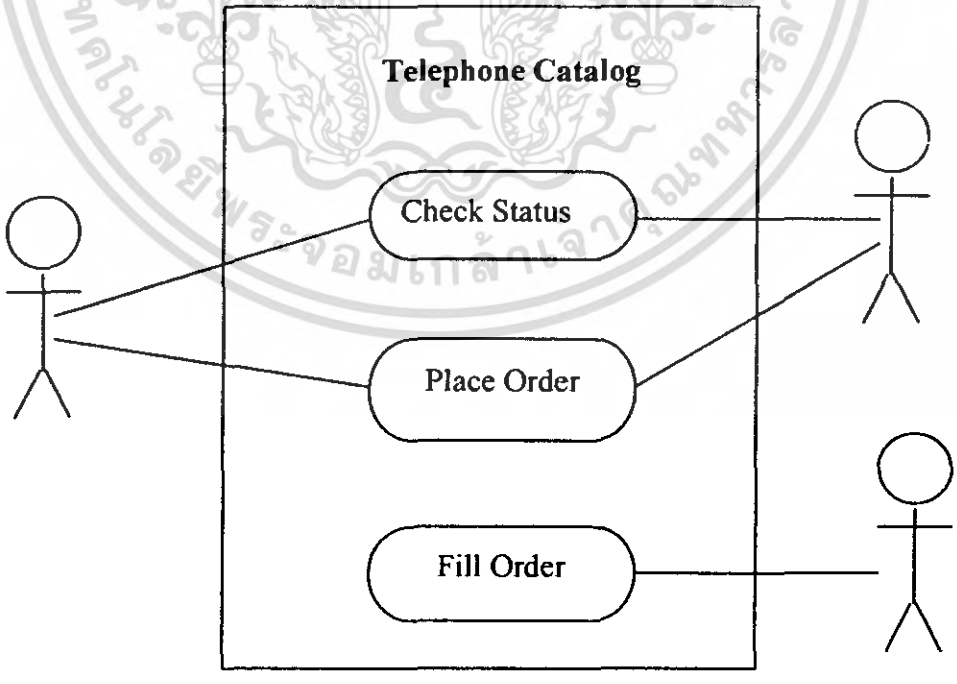
เป็นแผนภูมิที่ถูกออกแบบมาเพื่ออธิบายถึงกิจกรรมของระบบจากมุมมองของผู้ใช้ซึ่งจะทำให้ได้ระบบที่ตรงกับความต้องการและสามารถนำไปใช้งานได้จริง ยูสเคสไดอะแกรมจะประกอบไปด้วย แอ็กเตอร์ (Actor), ยูสเคส(Use Case), เส้นแสดงความสัมพันธ์(Relationsip)

- แอ็กเตอร์ คือผู้ที่กระทำกับระบบ (เป็นผู้ที่ใช้ยูสเคส) โดยแอ็กเตอร์ที่เริ่มต้นเหตุการณ์จะอยู่ด้านซ้ายของยูสเคส และแอ็กเตอร์ ที่ได้รับผลจากเหตุการณ์นั้นจะอยู่ด้านขวาของยูสเคสชื่อของแอ็กเตอร์จะอยู่ด้านล่างของรูปที่ใช้แทนแอ็กเตอร์นั้น

- ยูสเคส เป็นกิจกรรมที่ทำโดย แอ็กเตอร์ซึ่งเป็นฟังชั่นต่างๆ ที่ซอฟต์แวร์ต้องทำได้ทั้งหมดจึงจะถือว่าซอฟต์แวร์ไม่มีข้อผิดพลาดหรืออีกนัยหนึ่งยูสเคส คือสิ่งที่บอกว่าระบบหรือซอฟต์แวร์ทำอะไรได้บ้าง โดยยูสเคสจะใช้เครื่องหมายวงรีในการแสดง และชื่อของยูสเคสจะอยู่ภายในวงรีหรืออยู่ใต้วงรี

Use Case Dependency คือความสัมพันธ์ระหว่าง ยูสเคส ซึ่งมี 2 ลักษณะคือแบบ inclsion และแบบ extention แบบ include เป็นการมีการทำขั้นตอนนี้ทุกครั้ง ส่วนแบบ extend จะเป็นการทำงานนี้เป็นบางครั้ง

- เส้นแสดงความสัมพันธ์คือเส้นที่แสดงความสัมพันธ์ระหว่างกันทั้งของคลาส หรือ แอ็กเตอร์



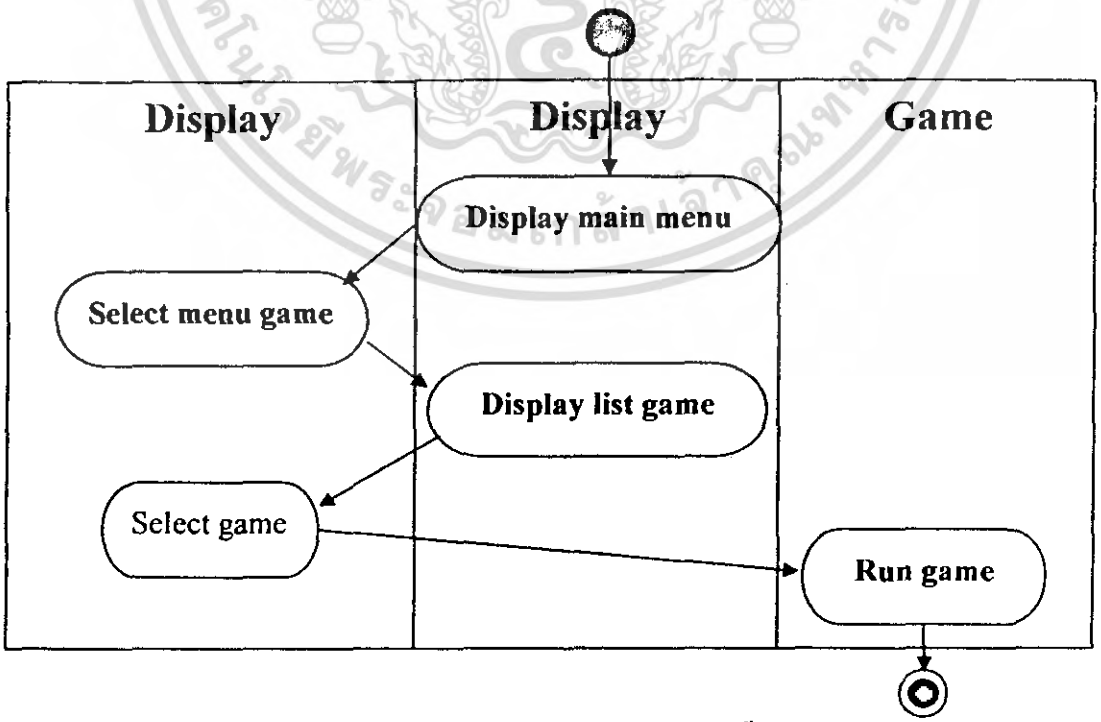
รูปที่ 2.8 แสดงตัวอย่างยูสเคสไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอ็กทิวิตี้ไดอะแกรม (Activity Diagram)

แอ็กทิวิตี้ไดอะแกรมจะแสดงขั้นตอนการทำงานของยูสเคส โดยจะเน้นที่งานย่อยของวัตถุ ซึ่งจะเจาะจงไปที่งานๆ หนึ่งของวัตถุนั้นจะรู้สึกว่าเหมือนกับ เสตตชาร์ดไดอะแกรมที่แสดงสถานะของวัตถุ แอ็กทิวิตี้ไดอะแกรมสามารถเปลี่ยนสถานะได้โดยไม่ต้องมีเหตุการณ์ที่กำหนดไว้ในไดอะแกรมมาทำก่อน แต่จะเปลี่ยนสถานะเองตามกระบวนการทำงานคล้ายกับโฟลชาร์ต สัญลักษณ์ที่ใช้ในแอ็กทิวิตี้มีดังนี้

สัญลักษณ์	ความหมาย
	เป็นจุดเริ่มต้นเข้าสู่แอ็กทิวิตี้ไดอะแกรม
	เป็นจุดสำหับออกจากแอ็กทิวิตี้ไดอะแกรม
	เป็นจุดตัดสินใจเงื่อนไขว่าเป็นจริงหรือเท็จ
	เป็นกิจกรรมที่ทำ
	แสดงขั้นตอนในการทำงานถัดไป



รูปที่ 2.9 แสดงตัวอย่างของแอ็กทิวิตี้ไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เทคโนโลยีดอทเน็ตเฟรมเวิร์ค (.NET Framework)

ดอทเน็ตเฟรมเวิร์ค คือ โครงสร้างหรือพิมพ์เขียวของไมโครซอฟท์ที่ประกอบด้วยคลาสหรือออบเจ็กต์ต่างๆมากมายสำหรับให้นักพัฒนาโปรแกรมเรียกไปใช้งานเมื่อพัฒนาแอพพลิเคชั่นต่างๆ ดอทเน็ตเฟรมเวิร์คนี้ไม่ใช่ระบบปฏิบัติการ (OS) แต่เปรียบเสมือนโปรแกรมหนึ่งที่จะสามารถสร้างสภาวะแวดล้อมหนึ่งซึ่งสามารถทำงานในระบบ

ในอดีตภาษาโปรแกรมต่างๆจะมีคลาสเฉพาะของตนเอง ไม่สามารถเรียกใช้งานข้ามภาษาโปรแกรมได้ทำให้นักพัฒนาโปรแกรมต้องพัฒนาโปรแกรมมากกว่าหนึ่งภาษาต้องเรียนรู้และจดจำคลาสของภาษานั้นๆแยกกันต่างหาก เช่น

- โปรแกรมที่ใช้ภาษา Visual C++ จะต้องเรียนรู้ MFC (Microsoft Foundation Classes)
- โปรแกรมที่ใช้ภาษา Visual J++ จะต้องเรียนรู้ WFC (Windows Foundation Classes)
- โปรแกรมที่ใช้ภาษา Visual Basic จะต้องเรียนรู้ Visual Basic Framework เป็นต้น

ซึ่งปัญหาดังกล่าวทำให้เกิดความยุ่งยากมากในการเปลี่ยนภาษาหรือเรียกใช้คอมโพแนนต์ของภาษาอื่นผู้พัฒนาจำเป็นต้องรู้คลาสของแต่ละภาษาใหม่จึงจะสามารถทำงานได้

แต่ในดอทเน็ตเฟรมเวิร์ค ได้ทำการสร้างคลาสรวมที่สามารถเรียกใช้งานได้จากภาษาของโปรแกรมใดก็ได้ที่ชื่อว่า Unified Classes เข้ามาครอบคลาสเดิมที่มีอยู่แล้วในแต่ละภาษา ซึ่ง Unified Classes จะมีความสามารถและคุณสมบัติของคลาสทุกภาษารวมกันอยู่ด้วย ดังนั้นเมื่อเราต้องการพัฒนาแอพพลิคชั่นใน ดอทเน็ตเฟรมเวิร์คเราสามารถที่จะเลือกใช้งานผ่านคลาสชื่อใหม่ Unified Classes ในการพัฒนาโปรแกรมหรือยังคงใช้งานคลาสชื่อเก่าๆที่มีประจำแต่ละภาษาตามความถนัดก็ได้ ซึ่งในอนาคตไมโครซอฟท์ก็หวังที่จะนำเอาระบบนี้ไปติดตั้งลงบนอุปกรณ์ทุกชนิดเพื่อทำให้อุปกรณ์ทุกอย่างมีระบบๆหนึ่งที่เหมือนกันหมด

ดอทเน็ตเฟรมเวิร์คนั้นมีส่วนประกอบภายในแบ่งออกเป็น3 ชั้นใหญ่ๆคือ

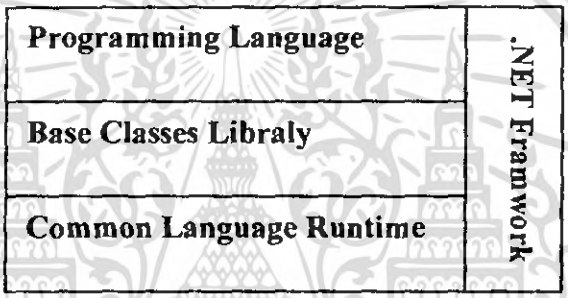
1. Programming Language: เป็นรูปแบบของภาษาที่ออกแบบมาเพื่อให้สามารถทำงานในสภาวะที่เป็น .NET ได้โดยที่ทางไมโครซอฟท์ได้เปิดตัวภาษาหลักๆที่จะใช้พัฒนาบน .NET นี้ 3 ภาษา ได้แก่ C# ที่พัฒนามาจาก C++ กับ JAVA เป็นหลัก ภาษา VB.NET เป็นภาษาที่พัฒนามาจาก Visual Basic และภาษา JScript.net เป็นภาษาที่พัฒนามาจาก JScript ซึ่งเป็น JavaScript

2. Base Classes Library: Library นั้นเปรียบเสมือนชุดคำสั่งสำเร็จรูปย่อยๆที่เพิ่มเข้ามาซึ่งส่วนใหญ่จะเป็นชุดคำสั่งที่ต้องใช้งานอยู่เป็นประจำ ดังนั้นจึงมีผู้คิดค้นเพื่ออำนวยความสะดวกในการเขียนโปรแกรม ซึ่ง Library ในภาษาต่างๆส่วนใหญ่จะอยู่ในรูปแบบไฟล์ incould แต่ถ้าเป็น ASP สิ่งที่เป็น library ก็คือ คอมโพเนนท์ต่างๆนั่นเอง ซึ่งภายในระบบ .NET จะสร้างสิ่งที่เรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น Library พื้นฐานขึ้น ทำให้ไม่ว่าจะใช้ภาษาใดในการพัฒนาโปรแกรมก็สามารถที่จะเรียกใช้ Libraryที่เป็นตัวเดียวกันได้หมด

3. Common Language Runtime (CLR) : นับเป็นสิ่งสำคัญแทบจะที่สุดของระบบ .NET นี้ก็ว่าได้ เพราะ CLR ที่ว่านี้มีหน้าที่ทำให้โปรแกรมที่เขียนขึ้นมาด้วยภาษาต่างๆกัน กลายเป็นภาษารูปแบบมาตรฐานเดียวกัน ทั้งหมด เราเรียกภาษาที่ว่านี้ว่า Intermediate language (IL) ซึ่งเมื่อต้องการที่จะรันโปรแกรมใด CLR ที่ว่านี้จะตรวจสอบเครื่องที่รันว่ามีสภาวะแวดล้อมการทำงานเช่นใดหลังจากนั้นก็จะคอมไพล์เป็นโปรแกรมที่เหมาะสมต่อการทำงานของเครื่องนั้น ทำให้เราสามารถใช้งานโปรแกรม ต่างๆได้อย่างมีประสิทธิภาพสูงสุดในแต่ละเครื่อง



รูปที่ 2.10 โครงสร้างของ .NET Framework

2.5.1 ประโยชน์และข้อดีของ .NET Framework

1. เป็นระบบที่มีไลบรารีที่เป็นมาตรฐานเดียวกัน : เนื่องจากมีไลบรารีที่เป็นมาตรฐานเดียวกันทั้งหมดทำให้เราไม่ต้องกังวล ว่าภาษาที่ใช้เขียนนั้นมีไลบรารีตัวนั้นตัวนี้หรือไม่ รวมทั้งไม่ต้องคอยกังวลว่าถ้าใช้ไลบรารีของภาษาหนึ่งแล้วอีกภาษาหนึ่งจะไม่มีไลบรารีตัวนั้น

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

3. ใช้ในการพัฒนาได้ทุกภาษา :ทำให้เราไม่ต้องคอยมาศึกษาภาษาใหม่ๆเมื่อต้องการสร้างโปรแกรมในแต่ละครั้ง นอกจากนั้นเรายังสามารถเลือก ใช้ภาษาที่เราถนัดที่สุดในการพัฒนาโปรแกรมต่างๆได้ด้วย

4. มีการควบคุมสิ่งแวดล้อมในการทำงานเป็นอย่างดี :เนื่องจากเป็นระบบที่เป็นมาตรฐานทำให้การควบคุมจัดสรรระบบต่างๆ ทำได้ง่ายขึ้น ไม่ว่าจะเป็นการจัดสรรหน่วยความจำ ด้านการใช้งานเครื่องก็มีความรวดเร็วมากขึ้น ลดโอกาสที่เครื่องจะแฮงก์ได้เป็นอย่างดี

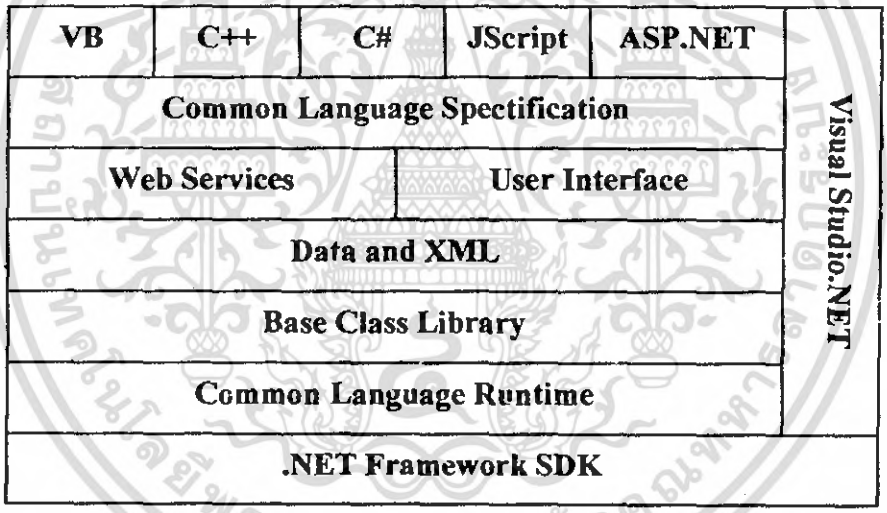
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ความปลอดภัยที่มีมากขึ้น :.NET Framework สามารถกำหนดสิทธิ์การใช้งานหรือ permission ของผู้ใช้งานได้มากขึ้นทำให้สมารถกำหนดว่า จะให้โปรแกรมในส่วนใดใช้งานได้หรือไม่ได้แล้วแต่เฉพาะบุคคล

ทั้งหมดนี้เป็นเพียงประโยชน์ส่วนหนึ่งในแนวคิดของไมโครซอฟท์ที่กำลังจะพัฒนาให้สำเร็จเท่านั้น บางข้อทำได้สำเร็จแล้ว แต่บางข้อก็ยังไม่สำเร็จดังนั้นจึงต้องคอยรอดูว่า ฝันของไมโครซอฟท์จะเป็นจริงและสำเร็จได้เมื่อไร

2.5.2 สถาปัตยกรรม .NET

โครงสร้างโดยรวมทั้งหมดของการสร้างแอพพลิเคชั่นด็อทเน็ตซึ่งแสดงได้ดังโครงสร้างด้านล่าง



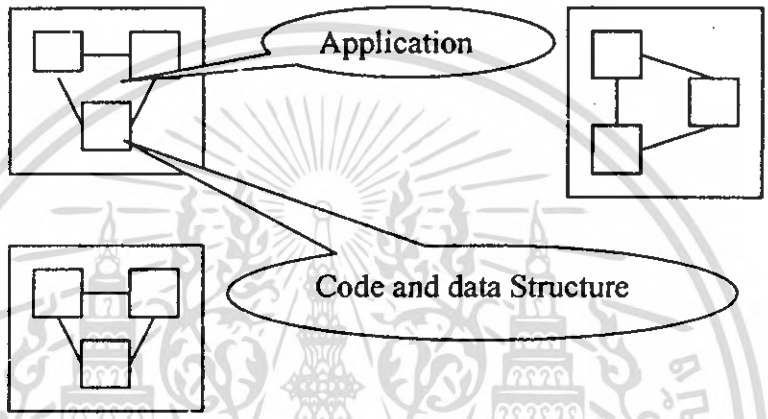
รูปที่ 2.11 รูปแสดงถึงสถาปัตยกรรมการพัฒนาแอพพลิเคชั่นด็อทเน็ต

จากรูป เป็นการแสดงถึงสถาปัตยกรรมของแอพพลิเคชั่นด็อทเน็ต ที่พัฒนาด้วยวิชวลสตูดิโอด็อทเน็ตทั้งเพลตฟอร์มที่ใช้ในการพัฒนาแอพพลิเคชั่น และสถาปัตยกรรมที่ใช้โดยมีเลเยอร์ล่างสุด คือ .NET Framework SDK เปรียบเสมือน Runtime Library ที่จะรันอยู่คอยสนับสนุนการทำงานของแอพพลิเคชั่นจากนั้นจะเป็นเลเยอร์ของ Common Language Runtime ซึ่งเป็นผลลัพท์ของการคอมไพล์แอพพลิเคชั่นด็อทเน็ต เลเยอร์ถัดขึ้นมาเป็นเครื่องมือ(Tools) และเทคนิคต่างๆที่เราสามารถใช้พัฒนาแอพพลิเคชั่นได้ทั้งในเรื่องของ Webservice, ADO.NET, ASP.NET จนกระทั่งถึงเลเยอร์บนสุดคือ ภาษาที่ใช้ในการพัฒนาแอพพลิเคชั่นด้วยวิชวลสตูดิโอด็อทเน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 เลเยอร์คอมมมอนแลงเกวจรันไทม์(Common Language Runtime)

ก่อนที่จะมีการพัฒนาโปรแกรมเป็นเชิงวัตถุ (Object Oriented) นั้นแอพพลิเคชั่นแต่ละตัวเปรียบเสมือนกล่อง ภายในแพลลิเคชั่นก็จะมโค้ด (Code) และมีโครงสร้างข้อมูล (Data Structure) ต่างๆของตัวเองมีฟังก์ชั่นต่างๆของตัวแอพพลิเคชั่นนั้นๆ

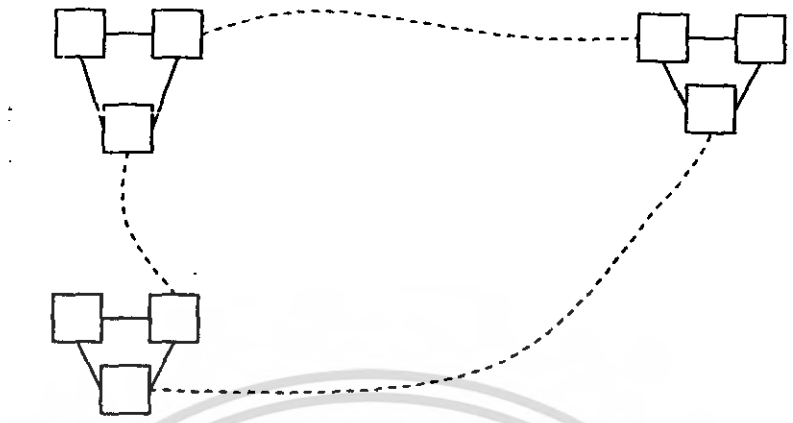


รูปที่ 2.12 โครงสร้างแอพพลิเคชั่นแรกๆ และการติดต่อระหว่างแอพพลิเคชั่น

การที่แอพพลิเคชั่นต่างๆจะมีการเรียกใช้การทำงานของกันและกัน หรือมีการส่งผ่านข้อมูลถึงกันและกัน เป็นสิ่งที่ทำได้ยาก ซึ่งอาจจะต้องมีการกำหนดว่าอะไรขึ้นมาเองระหว่าง 2 แอพพลิเคชั่นนั้นจนกระทั่งในยุคถัดมา ไมโครซอฟท์ได้คิดค้นเทคโนโลยี COM (Component Object Model) เป็นวิธีที่ทำให้เราเขียนโปรแกรมเป็นแบบเชิงวัตถุ และเรียกใช้การทำงานที่มาจากต่างแอพพลิเคชั่นได้

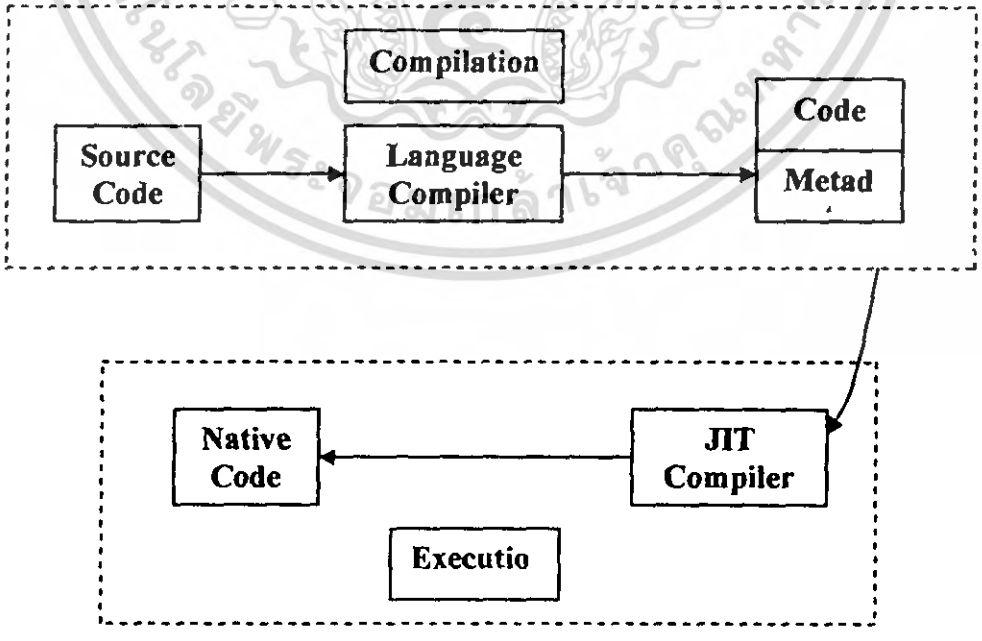
หากจะอธิบายให้ง่ายขึ้น ก็คือเปรียบเหมือนเราเอาแพ็กเกจอันหนึ่งห่อแอพพลิเคชั่นของเราไว้ โดยการพูดคุยหรือสื่อสารกันของแอพพลิเคชั่นก็พูดคุยผ่านแพ็กเกจที่เราห่อไว้ จนกระทั่งมาถึงตัววิชวลสตูดิโอด็อทเน็ตที่ได้รับการออกแบบใหม่ให้คลาสต่างๆสามารถติดต่อกันได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การพัฒนาแอพพลิเคชั่นบนเทคโนโลยีดอทเน็ตคลาสภายในแต่ละแอพพลิเคชั่นสามารถติดต่อกันได้

การพัฒนาแอพพลิเคชั่นด้วยวิชวลสตูดิโอดอทเน็ตนั้นเมื่อเราคอมไพล์ผ่านสิ่งที่เราจะได้จากการคอมไพล์ไม่ใช่โค้ดไบนารี่ (Binary Code) เลยทีเดียว แต่จะได้เป็นภาษากลางอันหนึ่งเรียกว่า MSE(Microsoft Intermediate Language) ซึ่งเป็นภาษาในระดับเลเยอร์ต่างๆ โครงสร้างภาษาจะเหมือนภาษา Assembly ภายในสิ่งที่เกิดจากการคอมไพล์ก็จะเป็น Microsoft Intermediate Language ตัวนี้ ภายในตัวมันจะประกอบด้วย 2 ส่วนคือ โค้ดกับตัวแอททริบิวต์หรือพร็อพเพอร์ตี้ต่างๆที่ใช้อธิบายตัวโค้ดนั้นซึ่งเรียกว่าเมต้าดาต้า (Meta Data)



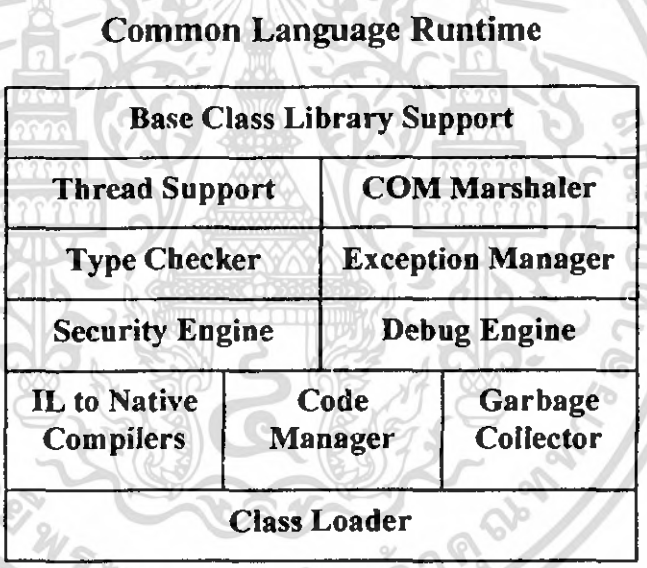
รูปที่ 2.14 โครงสร้างการคอมไพล์จากวิชวลสตูดิโอดอทเน็ตและเมื่อนำไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเมื่อโค้ดซึ่งเป็นภาษากลาง(Microsoft Intermediate Language) ถูกเรียกใช้งานจริงๆ จะมีตัวแปรภาษาตัวหนึ่งมาทำการคอมไพล์โค้ดตัวนั้นให้เป็นไบนารี่ ซึ่งตัวแปรภาษาตัวนั้นจะเรียกว่า JIT(Just In Time Compiler) เพราะว่าจะมีการคอมไพล์เมื่อมีการใช้งาน ฉะนั้นคลาสหรือโค้ดต่างๆที่เราพัฒนาขึ้นแล้วจะถูกคอมไพล์มาเป็นภาษากลาง ที่มีโครงสร้างภาษาเดียวกัน เพราะฉะนั้นคลาสต่างๆ ในแอพพลิเคชั่นจึงจะสามารถทำงานได้อย่างกลมกลืนกันและไม่มีข้อมูลติดขัดอะไร

การทำงานของ คอมมอนแลงเกวจรันไทม์(Common Langauge Runtime)

ภายในตัว คอมมอนแลงเกวจรันไทม์ จะมีโมดูล(Module) ย่อยๆซึ่งเป็นสถาปัตยกรรมภายในรูป



รูปที่ 2.15 แสดงสถาปัตยกรรมของคอมมอนแลงเกวจรันไทม์

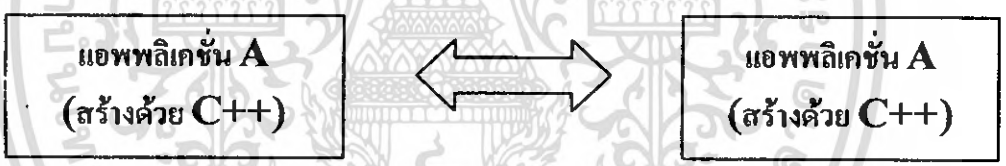
ด้านล่างสุดจะมี Class Loader ซึ่งเอาไว้โหลดโปรแกรมของเราขึ้นมาทำงาน นอกจากนี้ก็จะมีการคอมไพล์เลอร์ ซึ่งจะทำการคอมไพล์ภาษากลาง ให้เป็นภาษาไบนารีโดยจะมีตัวโค้ดเมเนเจอร์(Code Manager) และการ์เบจคอลเลคเตอร์(Garbage Collector) คอยจัดการกับหน่วยความจำที่เราจองเวลาเรียกใช้งาน นอกจากนี้ก็จะมีเรื่องความปลอดภัย (Security) ในการทำงานรวมทั้ง Debug Engine ในการดักจับรันไทม์เออเรอร์ (Runtime error) และเอ็กซ์เซ็ปชั่น การตรวจเช็คชนิดของตัวแปรต่างๆ และด้านบนสุดจะเป็นการใช้งานร่วมกับไลบรารีคลาสต่างๆจึงจัดเตรียมมาให้ เพราะฉะนั้นส่วนของเลเยอร์ของคอมมอนแลงเกวจรันไทม์จะสามารถคอมไพล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอพพลิเคชั่นใดก็ตาม ไม่ว่าจะเป็น ASP.NET หรือเขียนแอพพลิเคชั่นบนวินโดวส์ธรรมดา หรือจะเป็นการเขียนเว็บเซอร์วิสก็ตาม สิ่งที่ได้จากการคอมไพล์จะเป็น คอมมอนแลงเกวจรันไทม์ตามแผนภาพนี้

นั่นคือ จากการออกแบบเพื่อสนับสนุนการทำงานร่วมกันใน คอมมอนแลงเกวจรันไทม์จึงขจัดข้อเสียของ COM ไปได้เนื่องจากข้อจำกัดของ COM คือเป็นเพียงการเอาอะไรอย่างมาห่อคลาสไว้เท่านั้น ดังนั้นเวลาที่เอา COM ไปใช้งานจึงคอนข้างยุ่งยากรวมทั้งถ้ามีการเปลี่ยนแปลง COM นั้นๆก็จะทำให้เกิดปัญหาเรื่องการเข้ากันได้ (Compatibility) ระหว่างเวอร์ชั่นเดิมกับเวอร์ชั่นปัจจุบัน แต่ถ้าเราพัฒนาด้วยวิชวลสตูดิโอด็อทเน็ต ข้อเสียของ COM ทั้งหมดก็จะถูกขจัดเสีย

ความจริง คอมมอนแลงเกวจรันไทม์เป็นหลักการทำงานที่มีวิวัฒนาการมาจาก COM อีกทีหนึ่ง เป็นเชิงวัตถุที่แกนของภาษาเลย โดยวิชวลสตูดิโอด็อทเน็ตนั้นถูกออกแบบเพื่อสนับสนุนเชิงวัตถุโดยเฉพาะคลาสต่างๆที่อยู่ในแต่ละแอพพลิเคชั่นสามารถติดต่อถึงกันได้โดยตรงในแอพพลิเคชั่น A เราอาจจะเขียนคลาสด้วยภาษา C++ และแอพพลิเคชั่น A อาจจะติดต่อกับแอพพลิเคชั่น B ซึ่งเขียนด้วยภาษา Visual Basic ได้ คือการสืบทอดข้ามภาษานั้นสามารถทำได้



รูปที่ 2.16 แสดงความสามารถในการติดต่อข้ามแอพพลิเคชั่น

ใน วิชวลสตูดิโอด็อทเน็ทจะคอมไพล์เป็นภาษาเดียวกันคือ ภาษากลางตามรูปนอกจากนี้ยังสามารถทำงานด้วยกันกับ COM แบบเดิมที่เราเขียนมาแล้วได้ด้วยซึ่งใช้ วิชวลสตูดิโอเดิมโค้ดเหล่านี้ก็ไม่จำเป็นต้องโยนทิ้ง ในวิชวลสตูดิโอด็อทเน็ตเราสามารถเรียกใช้งานคอมโพเน็นต์ที่เขียนด้วยวิชวลสตูดิโอด็อทเน็ตได้เช่นกัน คือเป็นการเข้ากันได้ทั้งสองทาง (Backward Forward Compatibility) นี่คือข้อดีมากๆของด็อทเน็ต ทำให้เราไม่ต้องมาพิจารณาโค้ดใหม่

การจัดการเกี่ยวกับหน่วยความจำเมื่อทำการประมวณผล

เนื่องจากการทำงานทั้งหมดของตัวด็อทเน็ตจะมีการดูแลในเรื่องของคอมมอนแลงเกวจรันไทม์ ปัญหาหนึ่งที่มักจะพบ เมื่อเราพัฒนาแอพพลิเคชั่นคือเรื่องของการ์เบจคอลเลคชั่นในการจัดการกับหน่วยความจำ โดยเฉพาะบางภาษาอย่างเช่น ภาษาซีพลัสพลัสที่ต้องมีการเรียกใช้งานพ้อยน์เตอร์ (Pointer) ค่อนข้างมาก ในการใช้งานพ้อยน์เตอร์นั้นเราทำผิดไปนิดเดียวก็อาจจะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอพพลิเคชั่นของเราแฮงค์ในระหว่างรันหรือประมวลผลก็ได้ ทำให้เมื่อแอพพลิเคชั่นของเรารันไปเรื่อยๆหน่วยความจำส่วนนี้ก็จะขยายไปเรื่อยๆและสักพักหนึ่งก็จะหยุดไป

การแก้ไขข้อผิดพลาด (Debug) ตรงนี้ทำได้ยากมาก เพราะกว่าจะรู้ว่าเกิดความผิดพลาดขึ้นเราก็เสียเวลาในการรันไปแล้ว ซึ่งการจัดการกับหน่วยความจำนี้ ถ้าเราพัฒนาด้วยแพลตฟอร์มของด็อทเน็ตจะมี เครื่องมือหนึ่งเรียกว่า การ์เบจคอลเลคเตอร์ เป็นตัวที่คอยจัดการเกี่ยวกับหน่วยความจำให้เรา ความจริงตัวการ์เบจคอลเลคเตอร์จะเป็นตัวที่ทำการกำหนดค่าหน่วยความจำให้ว่างให้เราเองในส่วนที่เราไม่ได้ใช้งานโดยอัตโนมัติ ซึ่งนี่คือข้อดีอันหนึ่งในการพัฒนาด้วยแพลตฟอร์มของด็อทเน็ต

ระบบการตรวจจับความผิดพลาด

นอกจากนี้การคอมไพล์แอพพลิเคชั่นด็อทเน็ตมีให้เลือกหลายแบบ บางคนอาจจะคอมไพล์ให้เป็นโค้ดไบนารีเลย ไม่ต้องคอมไพล์ให้เป็นอินเทอร์มีเดียทแลงเกวจก็สามารถจะพัฒนาได้โดยใช้วิชวลซีพลัสพลัสด็อทเน็ต เรียกว่าเมเนจซีพลัสพลัสหรือถ้าเราเลือกใช้เครื่องมือภาษาอื่นๆที่มีความง่ายในการพัฒนายิ่งกว่า คือวิชวลเบสิค หรือ ซีชาร์ป ก็สามารถคอมไพล์เป็นอินเทอร์มีเดียทแลงเกวจได้

การคอมไพล์จะไม่มีตัวอินเทอร์รันไทม์ไลบรารี (Interpreter Runtime Library) ต่างๆ เช่น วิชวลเบสิค ที่ต้องต้องมีรันไทม์ไลบรารีของตัวเองในเวลาพัฒนาด้วย วิชวลสตูดิโออันนี้ก็จะไม่จำเป็นต้องมีอีกต่อไป เราใช้ตัวรันไทม์อันเดียวกันก็คือตัว ด็อทเน็ตเฟรมเวิร์คเป็นตัวรันไทม์ไลบรารีที่กล่าวข้างต้น

รูปแบบการทำงานร่วมกับภาษาอื่นๆ

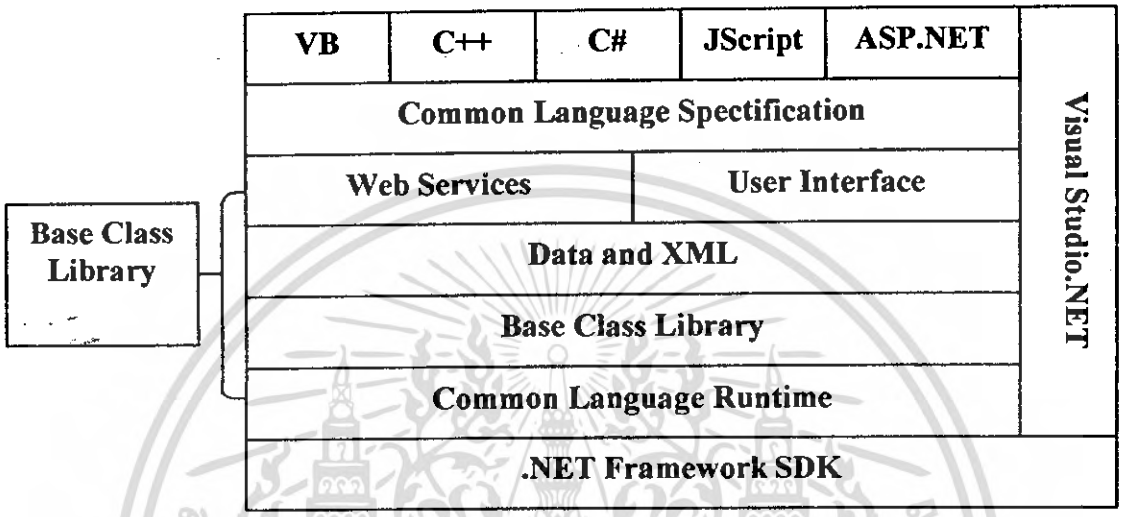
การพัฒนาแอพพลิเคชั่นด็อทเน็ตรวมไปถึงเทคนิคต่างๆที่กล่าวมาแล้วนั้นไม่จำเป็นจะต้องใช้ภาษาตระกูลวิชวลสตูดิโอไม่ว่าจะเป็นวิชวลเบสิคด็อทเน็ต (VB.NET), ซีชาร์ป (C#) หรือซีพลัสพลัส (C++) ถ้าภาษาใดก็ตามสมารถคอมไพล์มาเป็นคอมมอนแลงเกวจรันไทม์ได้ภาษานั้นก็จะสามารถใช้คุณลักษณะทั้งหมดของด็อทเน็ตได้ซึ่งตอนนี้มีภาษาอื่นๆที่สามารถทำงานร่วมกับแอพพลิเคชั่นด็อทเน็ตมากกว่า 20 ภาษาที่พัฒนาตัวคอมไพล์ขึ้นมาเพื่อคอมไพล์ภาษาต่างๆให้อยู่ในรูปของแพลตฟอร์มด็อทเน็ต ไม่ว่าจะเป็นภาษาโกบอล (Cobal), ปาสคาล (Pascal) และอื่นๆ

ในตัววิชวลสตูดิโอด็อทเน็ต ก็มีเครื่องมือระดับสูง (High Level Tools) ที่จะทำให้เราพัฒนาแอพพลิเคชั่นได้สะดวกและง่ายมากขึ้นนี่คือเรื่องของคอมมอนแลงเกวจรันไทม์ ซึ่งถือได้ว่าเป็นเลเยอร์ล่างสุดของการคอมไพล์แอพพลิเคชั่นที่เขียนด้วยวิชวลสตูดิโอด็อทเน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 เลเยอร์เบสคลาสไลบรารีแลงเกวจ (Base Class Library Language)

เลเยอร์ถัดมาเป็นของโครงสร้างของการพัฒนาแอพพลิเคชั่นดือทเน็ตต่อจากคอมมอนรันไทม์แลงเกวจ ก็คือเบสคลาสไลบรารี



รูปที่ 2.17 เบสคลาสไลบรารีของสถาปัตยกรรมดือทเน็ต

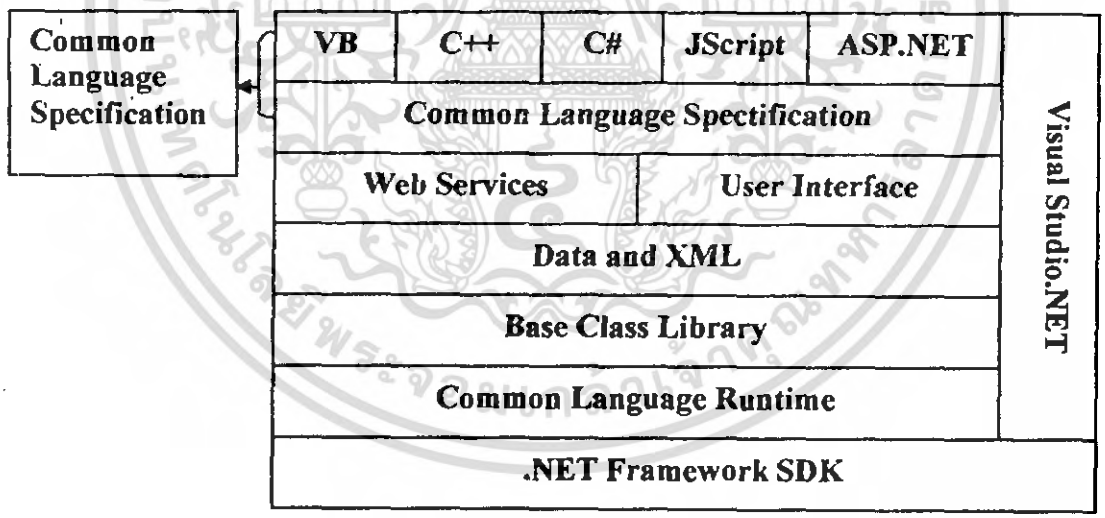
เทคนิคต่างๆที่วิชวลสตูดิโอดือทเน็ตจัดเตรียมให้เราในการใช้งานนั้นเบสคลาสไลบรารีเปรียบเสมือนเป็นการรวบรวมเอาฟังก์ชั่นของเอพีไอ (API: Application Programming Interface) ทั้งหมดสมมุติว่าตอนที่เราพัฒนาแอพพลิเคชั่นด้วยวิชวลสตูดิโอ- เวลาเราเรียกใช้งานฟังก์ชั่นบางอย่างในระดับสูงหรือต้องการทำงานแบบลึกๆกับระบบเรามักจะเรียกใช้เอพีไอ ภายในวิชวลสตูดิโอมีความสามารถในการเรียกฟังก์ชั่นเอพีไอได้ต่างกัน

ตัวเบสคลาสไลบรารี ก็คือ การที่เรารวบรวมเอาฟังก์ชั่นเอพีไอซึ่งกระจัดกระจายอยู่เวลาจะเรียกใช้เราต้องไปค้นหาในเมนูช่วยเหลือ (Help) นั่นคือเบสคลาสไลบรารีพยายามที่จะรวบรวมเอพีไอและฟังก์ชั่นทั้งหมดเกี่ยวกับระบบเข้ามาไว้ในลักษณะของเชิงวัตถุทำเป็นคาสอันหนึ่งซึ่งเป็นมาตรฐาน เป็นคลาสที่สร้างมาในตัวระบบเรียบร้อย ซึ่งคลาสทั้งหมดจะอยู่ภายใต้คลาสหลักอันหนึ่งที่เรียกว่า ซิสเต็ม (System) ทุกอย่างที่พัฒนาด้วยภาษาในวิชวลสตูดิโอดือทเน็ต จะเป็นเชิงวัตถุทั้งหมด โดยมีคลาสที่ใหญ่ที่สุดเรียกว่าคลาสซิสเต็ม ซึ่งภายในคลาสซิสเต็มจะมีคลาสย่อยๆมากมาย ซึ่งแต่ละอันจะสนับสนุนฟังก์ชั่นเอพีไอหรือสนับสนุนการทำงานที่เราต้องการได้ไม่ว่าจะเป็นการทำงานในเรื่องของกราฟริก การทำเกี่ยวกับโครงสร้างของข้อมูลการทำเกี่ยวกับเรื่องเครือข่าย ฟังก์ชั่นเอพีไอเหล่านี้จะถูกจัดกลุ่มให้เป็นเชิงวัตถุ อยู่ภายใต้คลาสซิสเต็มหรือเรียกใช้งานคลาสซิสเต็มถ้าเป็นวิชวลเบสิคกับซีชาร์ปก็ใช้งานได้ทั้ง 2 อย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.5 เลเยอร์คอมมอนแลงเกวจสเป็คซิฟิเคชั่น (Common Language Specification Layer)

เลเยอร์สุดท้ายในสถาปัตยกรรมด็อทเน็ตที่เราจะพูดถึงก็คือเลเยอร์คอมมอนแลงเกวจสเป็คซิฟิเคชั่น เรื่องของมาตรฐานบนพื้นฐานด็อทเน็ต ซึ่งคอมไพล์เลอร์จะต้องทำงานตามมาตรฐานดังกล่าว เพื่อให้สามารถทำงานร่วมกับภาษาบนพื้นฐานด็อทเน็ต และภาษาอื่นๆได้ ไมโครซอฟ์ได้ทำการปรับภาษาต่างๆ เช่น ซีชาร์ป, วิชวลเบสิค, ให้เข้ามาตรฐานด็อทเน็ต นอกจากนั้นผู้ผลิตรายอื่นสามารถพัฒนาตามข้อกำหนดนี้เพื่อให้สามารถทำงานบนพื้นฐานด็อทเน็ตได้ เลเยอร์ที่ 3 ของสถาปัตยกรรมด็อทเน็ต ก็คือเครื่องมือที่ใช้ในการสร้างแอพพลิเคชั่น แต่สิ่งที่อยู่เหนือกว่าทุกๆอย่างก็คือภาษาที่เราใช้งานด็อทเน็ตนั้นมีข้อดีคือ ต้องสนับสนุนมาตรฐานเดียวกัน เรียกว่าคอมมอนแลงเกวจสเป็คซิฟิเคชั่น ซึ่งไมโครซอฟท์ได้จดทะเบียนมาตรฐานนี้เข้ากับองค์กร ECMA แล้ว ซึ่งเป็นองค์กรที่ดูแลโครงสร้างของภาษาต่างๆ และเป็นแบบเปิดด้วยเพราะฉะนั้นเจ้าของภาษาอื่นก็สามารถสร้างตัวแปรภาษา หรือคอมไพล์เลอร์ เพื่อคอมไพล์ภาษาของเขาให้เข้ามาเป็นคอมมอนแลงเกวจสเป็คซิฟิเคชั่นอันนี้



รูปที่ 2.18 คอมมอนแลงเกวจสเป็คซิฟิเคชั่นของสถาปัตยกรรมด็อเน็ต

2.6 ระบบฐานข้อมูล(Database System)

ฐานข้อมูล คือ แหล่งข้อมูลที่อาจจะเป็นเท็กซ์ไฟล์ หรือระบบเก็บข้อที่เป็นแอพพลิเคชั่นอื่นๆซึ่งมีการเก็บข้อมูลอย่างเป็นระบบระเบียบ เพื่อให้สามารถเข้าถึงข้อมูลภายในได้อย่างสะดวก เราสามารถที่จะเรียกดู แก้ไข และทำงานกับชนิดของข้อมูลในฐานข้อมูลได้แต่ปกติเมื่อเรากล่าวถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลเรามักจะหมายถึงระบบเก็บข้อมูลที่เป็นแอพพลิเคชั่น (MS Access, MS SQL, My SQL ฯลฯ) เสียมากกว่า

2.6.1 ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

ระบบฐานข้อมูลมีอยู่หลายรูปแบบแต่ที่นิยมใช้กันมากที่สุดคือระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งจะประกอบไปด้วยข้อมูลที่อยู่ในตารางต่างๆและตารางเหล่านั้นจะมีการเชื่อมต่อความสัมพันธ์กันเพื่อประโยชน์ของการจัดเก็บ การอัพเกทและการค้นหาข้อมูล ซึ่งชนิดของความสัมพันธ์สามารถแบ่งออกได้เป็น 3 ความสัมพันธ์หลักๆคือ

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One – to – one - relationship) เป็นความสัมพันธ์ที่เข้าใจง่ายที่สุดเนื่องจากเร็คคอร์ด 1 เร็คคอร์ด ในตารางหนึ่งจะมีความสัมพันธ์กับอีก 1 เร็คคอร์ด ในอีกตารางหนึ่งเท่านั้น ไม่สามารถมีเกิน 1 ได้

- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One – to – many relationship) เป็นความสัมพันธ์ที่เร็คคอร์ด 1 เร็คคอร์ด ในตารางหนึ่งจะมีความสัมพันธ์กับอีก 2 เร็คคอร์ดหรือมากกว่า ในอีกตารางหนึ่ง ซึ่งความสัมพันธ์แบบนี้จะพบบ่อยมากในฐานข้อมูล

- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many – to – many relationship) เป็นความสัมพันธ์ที่ไม่ค่อยพบเห็นเป็นการสร้างตารางความสัมพันธ์แบบกลุ่มต่อหนึ่ง 2 ตารางมารวมกัน

2.6.2 คำศัพท์ต่างๆในระบบฐานข้อมูล

ตาราง (Table)

ตาราง คือ โครงสร้างที่เราใช้เก็บข้อมูลจริงในฐานข้อมูลโดยเก็บในรูปแบบของตารางย่อยที่มีความสัมพันธ์กัน แต่ละแถวในตารางเรียกว่า เร็คคอร์ด ส่วนแต่ละคอลัมน์ในแถวจะเรียกว่า ฟิวส์ (Field)โดยแต่ละเร็คคอร์ด คือข้อมูลหนึ่งชุดที่มีโครงสร้างซ้ำกัน และแต่ละฟิลด์ ก็คือข้อมูลตัวเดียวกันของแต่ละเร็คคอร์ด

คีย์ (Key)

การเชื่อมโยงข้อมูลระหว่างตารางต่างๆเข้าด้วยกัน โดยอาศัยฟิลด์ที่มีค่าตรงกันในแต่ละตารางเป็นตัวเชื่อม เรียกว่า คีย์ (Key) โดยปกติแล้วแต่ละเร็คคอร์ดในตารางจะต้องมีคีย์หรือข้อมูลที่จะเป็นกุญแจสำหรับเข้าถึงหรือบ่งชี้ว่าเร็คคอร์ดใดเป็นเร็คคอร์ดใด หรือต่างจากเร็คคอร์ดอื่นอย่างไร เหมือนกับที่ทุกคนต้องมีชื่อและนามสกุลที่แตกต่างกันเวลาอ้างอิงจะได้รู้ว่าเป็นใคร

คีย์หลัก (Primary Key) เป็นคีย์ที่ใช้ในการจัดเรียงแยกแยะข้อมูลแต่ละเร็คคอร์ด ในตารางนั้นออกจากกัน คีย์หลักนี้จะต้องมีค่าอยู่ในทุกเร็คคอร์ดจะปล่อยว่างไม่ได้อีกทั้งยังจะต้องไม่ซ้ำกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลย (Unique) ซึ่งอาจจะเลือกจากฟิลด์ใดฟิลด์หนึ่งหรือกำหนดขึ้นมาใหม่ก็ได้ เช่น เลขประจำตัวประชาชน 13 หลัก หรือ รหัสสินค้า (Bar Code) เป็นต้น

คีย์นอก (Foreign Key) เป็นฟิลด์ที่มีเก็บอยู่หลายตารางจึงสามารถใช้เป็นคีย์ในการเชื่อมโยงข้อมูลระหว่างตารางเข้าด้วยกัน ตัวอย่างเช่น ระบบงานสั่งอาหาร อาจมีตารางข้อมูลอาหารสำหรับเก็บรายละเอียดของอาหารแต่ละอย่าง ซึ่งจะมีชื่อหรือรหัสของอาหารเป็นคีย์หลัก ส่วนตารางการใช้บริการแต่ละโต๊ะก็จะต้องมีชื่อหรือรหัสของอาหารที่สั่งเช่นกันซึ่งทั้งสองตารางก็ต้องมีการเชื่อมโยงเข้าด้วยกัน ทำได้โดยเชื่อมโยงข้อมูลในเร็คคอร์ด จากทั้งสองตารางเข้าด้วยกันโยใช้ชื่อหรือรหัสอาหารเป็นคีย์นอก

คีย์คู่แข่ง (Candidate Key) เป็นคีย์ที่มีความสามารถเป็นคีย์หลักแต่ไม่ได้ใช้เป็นคีย์หลักสำหรับตาราง

คีย์ผสม (Composite Key) เป็นคีย์ที่ใช้ฟิลด์ร่วมกับฟิลด์อื่นๆที่เป็นคีย์ผสมเหมือนกันเพื่อใช้เป็นคีย์หลัก

2.7 เทคโนโลยีเอดีโอด็อทเน็ต(ADO.NET)

เทคโนโลยีเอดีโอด็อทเน็ต(ActiveX Data Object Dot Net) คือสื่อกลางที่แอพพลิเคชั่นใช้ในการทำงานน่วมกับแหล่งข้อมูล(Data Source) ซึ่งอาจอยู่ในรูปแบบไฟล์ข้อความ(Text File), ข้อมูล XML หรือระบบฐานข้อมูล เช่น Microsoft SQL Server, Microsoft Access, Oracle และ MySQL รวมทั้งแหล่งข้อมูลอื่นๆเราสามารถใช้เอดีโอด็อทเน็ตในการเชื่อมต่อกับแหล่งข้อมูล, รับข้อมูล และปรับปรุงข้อมูล ซึ่งภายในเอดีโอด็อทเน็ตจะประกอบไปด้วย .Net Data Providers ที่ใช้สำหรับการติดต่อกับฐานข้อมูล, การเอ็กซ์คิวต์คำสั่ง และรับผลลัพท์ ซึ่งในส่วนของเอดีโอด็อทเน็ตนั้นจะประกอบไปด้วยคลาสต่างๆ มากมายเนื่องจากว่าเอดีโอด็อทเน็ตได้ถูกพัฒนาขึ้นด้วยหลักของ OOP (Object – Oriented - Programming) แบบเต็มๆ

ในเอดีโอด็อทเน็ตนั้นสามารถติดต่อกับฐานข้อมูลได้ 2 รูปแบบหลักๆ ก็คือแบบที่ต้องจองแช่การเชื่อมต่อกับฐานข้อมูล(Connected) และแบบที่ไม่ต้องแช่การเชื่อมต่อ(Disconnected)

การติดต่อแบบ Connected เราต้องเปิดการติดต่อกับฐานข้อมูลอยู่ตลอดเวลาที่ใช้ ถ้าโปรแกรมส่วนอื่นหรือ Session อื่นต้องการใช้ฐานข้อมูลด้วย Connection อันใหม่ก็จะต้องถูกสร้างขึ้นมา ส่วนนี้จะเป็นปัญหาในกรณีที่มีผู้เข้าใช้งานมากๆ โดยเฉพาะกับฐานข้อมูลที่มีประสิทธิภาพการทำงานที่ไม่สูงนัก เช่นฐานข้อมูลอย่าง Access นั้นรองรับการใช้งานพร้อมๆ ได้ประมาณ 10 คนเท่านั้นเอง

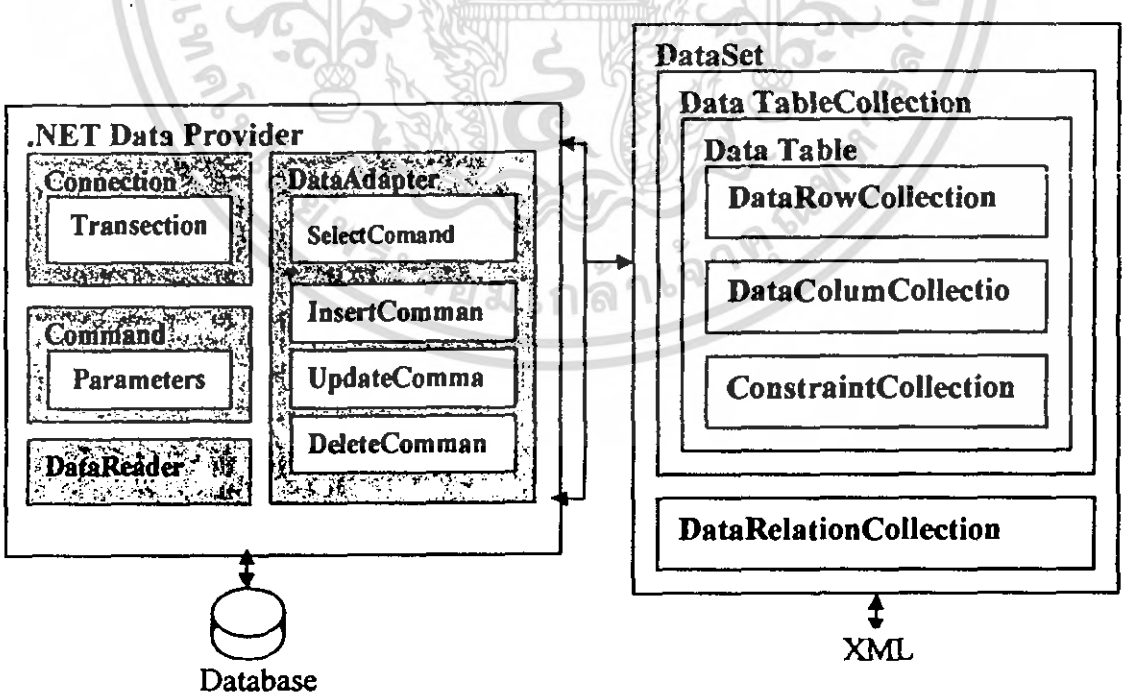
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อแบบ Disconnect นั้นเป็นรูปแบบใหม่ที่แทนที่จะต้องเปิด Connection แช่เอาไว้ เอดีโอดอทเน็ต จะทำ Snapshot ของฐานข้อมูลที่ผู้ใช้แต่ละคนกำลังติดต่ออยู่ เก็บไว้ใน หน่วยความจำของเครื่อง โดยจะทำการเก็บข้อมูลเฉพาะที่ผู้ใช้คนนั้นๆ ดึงมาใช้เท่านั้น พอได้ข้อมูลมาหมดก็ทำการตัดการเชื่อมต่อกับฐานข้อมูลได้ทันที ในขณะที่แบบ Connected นั้นจะเป็นการลักษณะการเรียกข้อมูลออกมาทีละแถว

ซึ่งการติดต่อแบบDisconnected นั้นภาระที่เกี่ยวกับการเชื่อมต่อจะอยู่ที่เซิร์ฟเวอร์ เนื่องจากต้องมีการเก็บข้อมูลที่เกิดจากการเชื่อมต่อของผู้ใช้แต่ละคนไว้ในหน่วยความจำตลอดเวลา ซึ่งข้อมูลก็มีปริมาณมากด้วย ส่วนการทำงานแบบConnection นั้นจะไม่ค่อยเป็นภาระให้กับเซิร์ฟเวอร์แต่จะเป็นภาระให้กับฐานข้อมูลแทนซึ่งก็จะเหมาะสำหรับการดงข้อมูลไม่มากนัก เนื่องจากเป็นการดึงข้อมูลออกมาทีละแถว

2.7.1 องค์ประกอบภายในของเอดีโอดอทเน็ต

ภายในเอดีโอดอทเน็ต ประกอบไปด้วยส่วนประกอบ 2 ส่วนใหญ่ๆ คือ .NET DataProvider และ Data Set ดังรูป



รูปที่ 2.19 แสดงโครงสร้างภายในของเอดีโอดอทเน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 ออปเจ็กต์ที่สำคัญในเอดีโอด็อทเน็ต

ออบเจ็กต์ที่ใช้อยู่บ่อยๆ ในเอดีโอด็อทเน็ตซึ่งจะกล่าวถึงในส่วนนี้ก็คือConnection, Dataset, DataReater และData Adapter

ออบเจ็กต์คอนเน็คชั่น(Connection Object)

ออปเจ็กต์Connection เป็นออบเจ็กต์ที่อยู่ในเอดีโอด็อทเน็ตหน้าที่คือ การเปิดการเชื่อมต่อ (Connection) ระหว่างไคล์เอนต์กับแหล่งข้อมูล โดยที่เราสามารถกำหนดรูปแบบการเชื่อมต่อไปยังฐานข้อมูลที่ต้องการ พร้อมค่ารายละเอียดต่างๆ เช่นรหัสผ่านเมื่อจะเชื่อมต่อกับฐานข้อมูลซึ่งยังคงคล้ายกับเอดีโอแบบเดิมนั่นเอง

ออบเจ็กต์คอมมานด์(Command Object)

ออบเจ็กต์ตัวนี้มีหน้าที่เอาไว้เขียนคำสั่งเอสคิวแอล (SQL) จากไคล์เอนต์ไปทำงานบนแหล่งข้อมูล เช่นเดียวกับออบเจ็กต์คอนเน็คชั่น (Connection Object) คือในเอดีโอด็อทเน็ตก็ยังมีออบเจ็กต์นี้อยู่โดยคำสั่งเอสคิวแอล (SQL) ที่ใช้ร่วมกับคลาสนี้ก็เช่น Insert, Update, Delete, Create Table เป็นต้น

ออบเจ็กต์ดาต้าเซ็ต (Dataset Object)

ออบเจ็กต์ดาต้าเซ็ต นี้มีเฉพาะในเอดีโอด็อทเน็ตเท่านั้น ดาต้าเซ็ตเปรียบเสมือนฐานข้อมูลทั้งก้อนเลย คือเมื่อเราเลือก (Select) ออกมาเราไม่ได้เลือกเฉพาะแถวของข้อมูลออกมา แต่ได้ข้อมูลอันใหม่อันหนึ่งโดยเป็นฐานข้อมูลที่อยู่ในหน่วยความจำของเรา

ภายในดาต้าเซ็ตนั้น จะมองว่าเป็นเหมือนฐานข้อมูลเหมือนกัน ในนั้นมีตาราง (Tables), คอลัมน์ (columns) และแถว (row) ต่างๆอยู่ภายใน มีความสัมพันธ์ระหว่างได้มีการกำหนดข้อบังคับความถูกต้อง (constrain), ดรรชนี(index key), คีย์ร่วม (public key) ได้ โดยการเก็บข้อมูลของดาต้าเซ็ตจะเก็บโครงสร้าง (schema) และข้อมูลของตัวมันด้วย เราสามารถเลือกข้อมูลออกมาเก็บไว้ในดาต้าเซ็ต จากนั้นก็สามารถออกจากการเชื่อมต่อได้โดยเราก็จะมีดาต้าเซ็ตเอาไว้ใช้งาน จากนั้นภายในโปรแกรมเราก็สามารถใช้คำสั่ง Insert, Update, Delete ลงไปใน Dataset ได้จนเสร็จเรียบร้อยแล้วเราสามารถกลับมาเชื่อมต่อกับฐานข้อมูลอีกทีหนึ่ง แล้วทำการซิ้งค์ (Synch) ข้อมูลกลับขึ้นไปได้ โดยตัวเอดีโอด็อทเน็ตจะคอยจัดการความแตกต่างระหว่างข้อมูลด้วยกันเอง

การซิ้งค์ข้อมูลกลับไปลักษณะนี้ คือการเขียนโปรแกรมแบบ Disconnected Programming โดยไม่จำเป็นต้องเชื่อมต่อฐานข้อมูลตลอดเวลา เพราะแอพพลิเคชั่นบนอินเตอร์เน็ตนั้น ถ้าเราเชื่อมต่อฐานข้อมูลไว้ตลอดเวลาจะเป็นการสิ้นเปลืองทรัพยากรของเซิร์ฟเวอร์มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจ็กต์ดาต้าอะแด็ปเตอร์(Data Adapter Object)

ออบเจ็กต์ดาต้าอะแด็ปเตอร์มีเฉพาะในเอดีโอดอทเน็ตเช่นกัน ออบเจ็กต์ดาต้าอะแด็ปเตอร์ก็เหมาะสำหรับเลือก(Select) ทั่วๆไปซึ่งเปรียบเสมือนเป็นสะพานเชื่อมระหว่างแหล่งข้อมูลที่มีข้อมูลกับตัวดาต้าเซ็ตของเราโดยลำเลียงข้อมูลจากแหล่งข้อมูลเอามาเก็บในดาต้าเซ็ต จากนั้นจึงสามารถใช้คำสั่งเอสคิวแอลในดาต้าเซ็ต ทำการเปลี่ยนแปลงข้อมูลในแถวนั้นได้ และหากต้องการซิ้งค์กับข้อมูลจากดาต้าเซ็ตกลับไปในแหล่งข้อมูลใหม่ เราก็ใช้ดาต้าอะแด็ปเตอร์ ทำการอัพเดตข้อมูลกลับขึ้นไปในแหล่งข้อมูล

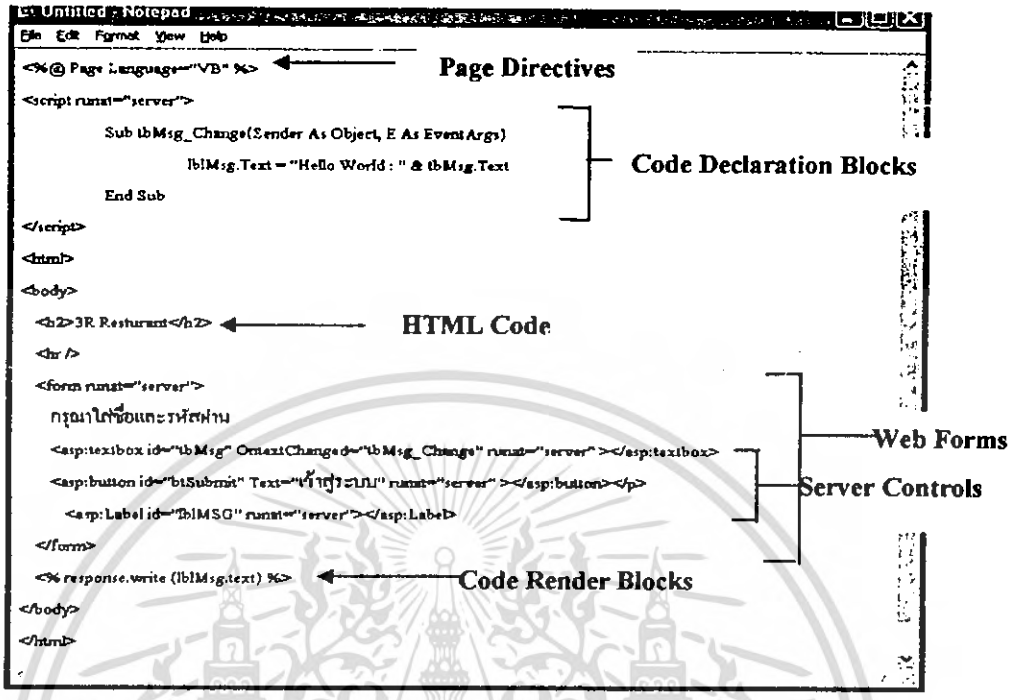
หากเปรียบเทียบภาพของดาต้าอะแด็ปเตอร์ ก็เหมือนสะพานเชื่อมโยงระหว่างฐานข้อมูลของเรากับดาต้าเซ็ตนั่นเองการที่จะเอาข้อมูลมาเก็บไว้ในดาต้าเซ็ตเราต้องสร้างออบเจ็กต์ขึ้นมาสองตัว ตัวแรกเป็นดาต้าเซ็ต ตัวที่ 2 เป็นดาต้าอะแด็ปเตอร์ หลังสร้างเสร็จแล้วให้ใช้คำสั่งเอสคิวแอลเพื่อดึงข้อมูลมาจากนั้นใช้คำสั่ง Fill เพื่อนำข้อมูลที่ได้จากการอ่านมาใส่ในดาต้าเซ็ต

2.8 โครงสร้างของเอกสาร ASP.NET

โครงสร้างของเอกสาร ASP.NET นั้นเป็นลักษณะของโครงสร้างในส่วนของหน้าที่ใช้ในการออกแบบซึ่งสามารถแบ่งออกเป็นส่วนใหญ่ๆได้หลายส่วนด้วยกัน ซึ่งส่วนประกอบที่สำคัญของโครงสร้างหลักๆของเอกสาร ASP.NET มีดังนี้

- Directives
- Web Forms
- Server-side Comments
- Code Declaration Blocks
- Server Controls
- Server-Side Include Directive
- Code Render Blocks
- HTML Code
- Data Binding Expression

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 แสดงโครงสร้างหลักของเอกสาร ASP.NET

ส่วน Directives : ระบุค่าต่างๆ ให้คอมไพล์เลอร์ทราบ

เป็นการระบุค่าข้อมูลที่จำเป็นบางอย่างภายในเอกสารของ Web Form Page(.aspx)และ User Control(.ascx) โค้ดส่วน Directives นี้เราสามารถพิมพ์เข้าไปในส่วนใดของโค้ดก็ได้ (จะพิมพ์ไว้ส่วนล่างสุด ส่วนกลางหรือบนสุดของโค้ดก็ได้) แต่ที่นิยมก็คือส่วนบนบรรทัดแรกสุดของโค้ด เพราะจะทำให้สังเกตได้ง่าย เวลาย้อนกลับมาอ่านโค้ดโปรแกรมในภายหลังใน Directives แต่ละอันสามารถประกอบไปด้วยแอตทริบิวต์หนึ่งหรือมากกว่าก็ได้ ซึ่งชนิดของ Directives สำหรับ ASP.NET นั้นมีดังนี้ @Page, @Control, @Import, @Implements, @Register, @Assembly, @OutputCache, @Reference เป็นต้น

บางครั้งเราไม่ต้องประกาศส่วน Directive ตัว @Page สำหรับ Page หรือ @Control สำหรับ User Control ก็ได้เพราะ ASP.NET ถือว่า Directive ดังกล่าวถือเป็นตัวสำคัญจึงมีค่า Default ให้อยู่แล้วซึ่ง Directive แต่ละตัวมีลักษณะที่สำคัญดังนี้

@Page และ @Control

@Page และ @Control เป็นการระบุคุณสมบัติประจำตัวของเอกสาร Page หรือ User Control ตามลำดับ เช่น บอกว่าจะใช้ภาษาใดในการเขียนเอกสารนี้ ซึ่ง @Page และ @Control มีรูปแบบการใช้งานที่เหมือนกัน เพียงแต่แค่เปลี่ยนคำว่า Page หรือ Control เท่านั้นเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<%@PageLanguage="language"Inheriites="inherite"Src="src"%>

โดยที่ Inherite หมายถึงชื่อคลาสที่เรียกมาใช้งานหรือเรียกว่า Code Behind

Src หมายถึงชื่อไฟล์ของคลาส (CodeBehind) ที่เรียกมาใช้งาน

Language หมายถึงภาษาที่ใช้ในการเขียนโค้ดโปรแกรม เช่น ภาษาวิชวลเบสิก (VB) ภาษาซีชาร์ป (C#) หรือจาวาสคริบ (Jscript)

ตัวอย่างเช่น ใช้ภาษาวิชวลเบสิกในการเขียน

<%@PageLanguage="VB"%>

@Import

เป็นการประกาศการนำเข้า namespace สำหรับในการใช้งานใน Page หรือ User Control

การใช้งาน namespace

สำหรับ APS.NET นั้น ก่อนเรียกใช้คลาสหรือออบเจ็กต์ใน .NET Framework เราต้องเรียก namespace ขึ้นมาก่อนเพราะ namespace จะเป็นที่จัดกลุ่มคลาสที่มีหน้าที่การทำงานคล้ายๆ กัน ดังนั้นก่อนที่จะมีการเลือกใช้คลาสใดๆเราต้องมีการประกาศใช้ namespace ที่บรรจุคลาสที่ต้องการใช้นั้นให้ถูกต้องด้วยทุกครั้ง เช่น หากต้องการใช้งานคลาสที่ทำงานกับฐานข้อมูล ก็ต้องมีการใช้งาน namespace ชื่อ System.Data เสียก่อน หรือการทำงานกับข้อมูลรูปแบบ XML ต้องมีการประกาศ namespace ชื่อ System.Xml เสียก่อน เป็นต้น การประกาศใช้หรือนำเข้า namespace มีรูปแบบคำสั่งดังนี้

<%@import Namespace="namespace"%>

โดยที่ namespace คือชื่อ namespace ที่ต้องการนำเข้ามาร่วมในเอกสาร ASP.NET

หากต้องการนำเข้า namespace มากกว่าหนึ่งตัวก็ให้ประกาศต่อกันได้เช่น

<%@import Namespace="System.IO"%>

<%@import Namespace="System.Xml"%>

จะเป็นการนำเข้า namespace สองตัวชื่อ System.IO และ System.Xml เข้ามาในเอกสาร ASP.NET

ในกรณีที่มี namespace ที่มี namespace อื่นอยู่ภายในอีก เมื่อมีการประกาศนำเข้า namespace นั้นเข้ามาจะไม่ได้ทำการนำเข้า namespace ที่อยู่ในระดับลึกลงไปเข้ามาด้วย ดังนั้นหากเราต้องการใช้ระดับ namespace ที่อยู่ในระดับลึกลงไป เราจำเป็นที่จะประกาศต่างหากด้วย เช่น

<%@import Namespace="System.Data"%>

<%@import Namespace="System.Data.OleDb"%>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายชื่อ namespace ที่เรามักมีการใช้งานบ่อยๆ มีดังนี้

- System.Data เกี่ยวกับการแสดงผลข้อมูลและออบเจ็กต์
- System.Data.OleDb เกี่ยวกับการเชื่อมต่อกับฐานข้อมูลผ่าน OLEDB Provider
- System.Data.SQLClient เกี่ยวกับการเชื่อมต่อกับฐานข้อมูล SQL Server
- System.Drawing เกี่ยวกับการแสดงผลสีสันตัวอักษร
- System.IO เกี่ยวกับการทำงานกับระบบไฟล์
- System.Xml เกี่ยวกับการทำงานกับข้อมูลรูปแบบ XML

การประกาศนำเข้า namespace มาใช้งานในเอกสารมีผลกระทบต่อปริมาณข้อมูลสำหรับการประมวลผลโดยรวมน้อยมาก ตัวคอมไพเลอร์จะทำการแปรโค้ดที่นิยามใน namespace เพียงบางส่วนที่มีการอ้างอิงใช้งานเท่านั้น ดังนั้นการนำเข้า namespace จึงไม่ทำให้เกิดความล่าช้าในการแปรโค้ดและการประมวลผล

ส่วน Declaration Blocks : ระบุขอบเขตโค้ดโปรแกรม

เป็นการระบุขอบเขตของโค้ดโปรแกรมส่วนของ ASP.NET ซึ่งโค้ดในส่วนนี้จะไม่ถูกส่งไปให้กับบราวเซอร์ของผู้ใช้โดยตรง โดยเมื่อมีการใช้งานโค้ดในส่วนนี้จะถูกทำการคอมไพล์ให้เป็น MSIL และเป็นภาษาเครื่อง ซึ่งจะทำให้การทำงานมีความรวดเร็วเมื่อเรียกใช้งานครั้งที่สองขึ้นไป ส่วนนี้ถือได้ว่าเป็นส่วนหลักของโค้ด ASP.NET สำหรับการประกาศโพรซิเยอร์, ฟังก์ชั่น หรือตัวแปรที่ต้องการใช้ร่วมกันทั้งหน้า ในเอกสาร ASP.NET หน้าหนึ่งๆสามารถที่จะกำหนด Code Declaration Block กี่บล็อกก็ได้

ส่วน Code Render Blocks : โค้ดที่เขียนในส่วน<%...%>

โค้ดที่อยู่ในส่วน Code Render Block นี้จะมีการใช้งานคล้ายๆกับ Code Declaration Block คือใช้ในการบรรจุโค้ดที่ต้องมีการประมวลผลที่ฝั่งเซิร์ฟเวอร์ก่อน แล้วจึงจะส่งผลลัพท์ที่ได้ไปยังบราวเซอร์ของผู้ใช้อีกที

ข้อแตกต่างระหว่าง Code Render Block และ Code Declaration Block

- **ประการแรก ลักษณะการคอมไพล์**

Code Render Block จะไม่ได้ถูกการคอมไพล์ตามหลักการทั่วไปของ .NET Framework แต่จะถูกแปลให้เป็นภาษาเครื่องโดยวิธีการอินเทอร์เพลต (Interpret) หรือการแปลโค้ดที่ละบรรทัดเมื่อตอนที่เว็บเพจถูกเรียกใช้งาน ซึ่งจะทำให้การทำงานโดยรวมช้ากว่าการแปลแบบการคอมไพล์ เนื่องจากเมื่อมีการเรียกใช้งานก็จะต้องมีการแปลอีกทุกครั้งอย่างนี้เรื่อยไป ต่างกับการคอมไพล์ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Declaration Block ที่จะมีการแปลผลเพียงครั้งเดียวตอนคอมไพล์ เมื่อมีการเรียกใช้ภายหลังก็จะเรียกจากที่คอมไพล์แล้วไปใช้เลย

- **ประการที่สอง การควบคุมการแสดงผล**

จะเห็นได้ว่าเมื่อทำงานที่มีความซับซ้อน เช่น การแสดงผลที่ซับซ้อน การใช้ Code Declaration Block จะทำงานได้สะดวกกว่า Code Render Block อย่างมากเพราะสามารถทำงานร่วมกับคอนโทรลต่างๆที่มีใน ASP.NET ได้ ดังนั้นจึงขอแนะนำว่าถ้าไม่จำเป็นก็ขอให้หลีกเลี่ยงการใช้งาน Code Render Block แต่พยายามใช้ Code Declaration ที่เป็นสิ่งเกิดขึ้นใหม่ใน ASP.NET จะดีกว่า

ส่วน Web Forms : สร้างส่วนติดต่อกับผู้ใช้(ทำงานฝั่งเซิร์ฟเวอร์)

Wed Forms ถือเป็นเรื่องใหม่ที่เกิดมาพร้อมกับ ASP.NET และเป็นส่วนสำคัญที่เราต้องทำความเข้าใจและสร้างความคุ้นเคยให้มากๆเพราะ ASP.NET จะเกี่ยวข้องกับการสร้าง Web Forms ชนิดที่แยกกันไม่ออกเลยทีเดียว Web Forms นั้นเป็นส่วนที่ใช้สำหรับเป็นส่วนที่ติดต่อกับผู้ใช้เช่นเดียวกับแบบฟอร์ม HTML ทั่วไป(หรือบางครั้งเรียกว่า Client Form) ต่างกันที่ Web Forms จะมีการทำงานที่ฝั่งเซิร์ฟเวอร์ก่อน แล้วจึงทำการสร้างเป็นแบบฟอร์ม HTML ส่งไปยังโปรแกรมบราวเซอร์ของผู้ใช้ซึ่งจากโปรแกรมตัวอย่างข้างต้น Web Forms นี้จะอยู่ภายในส่วนของ <Form>...</Form>

ส่วน Server Controls : คอนโทรลที่ทำงานฝั่งเซิร์ฟเวอร์

Server Control คือคอมโพแนนท์ (Component) ที่ทำงานฝั่งเซิร์ฟเวอร์ซึ่งคำว่า"ทำงานฝั่งเซิร์ฟเวอร์" โดยแต่ก่อนการใช้ฟอร์มรับข้อมูลจะใช้องค์ประกอบที่ทำงานฝั่งไคล์เอนต์เป็นหลัก เช่น ถ้าเราใช้ Textbox รับข้อมูลเราก็ต้องสร้างฟังก์ชั่นเพื่อรองรับอีเวนต์ที่เกิดจาก Textbox นั้นวิธีนี้โค้ดฟังก์ชั่นที่เราสร้างจะอยู่ที่ฝั่งไคลเอนต์

แต่ถ้าเป็น Server Control ฟังชั่นที่สร้างขึ้นเพื่อรองรับอีเวนต์จะรออยู่ที่เซิร์ฟเวอร์ซึ่งจะทำให้โค้ดที่เราสร้างถูกเก็บอย่างดีภายใต้ระบบรักษาความปลอดภัยของเซิร์ฟเวอร์ส่วนการทำงานของฝั่ง Server Control นั้นอีเวนต์ต่างๆและข้อมูลที่ผู้ใช้กรอกลงฟอร์มจะถูกเก็บไว้ทั้งหมด รอจนกว่าผู้ใช้จะทำการส่งอีเวนต์และข้อมูลนั้นไปให้เซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์รับอีเวนต์ที่ส่งมาก็จะเรียกฟังก์ชั่นที่รองรับอีเวนต์นั้นขึ้นมาประมวลผลข้อมูลที่ผู้ใช้ส่งกรอก ซึ่งหมายความว่าไม่ว่าจะเกิดอีเวนต์ใดที่ฝั่งไคลเอนต์ก็จะไม่มีการประมวลผลใดๆเกิดขึ้นทั้งสิ้น แต่จะเก็บอีเวนต์นั้นแล้วรอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จนกว่าผู้ใช้กดปุ่มส่งข้อมูล อีเวนต์เหล่านั้นทั้งหมดจึงจะถูกส่งให้เซิร์ฟเวอร์ตีความและประมวลผล ซึ่งการประมวลผลที่เกิดขึ้นที่เซิร์ฟเวอร์ เราจึงเรียกองค์ประกอบเหล่านี้ว่า Server Controls

Server Controls ที่มีอยู่ใน ASP.NET สามารถแบ่งออกเป็น 4 ประเภทใหญ่ๆ ได้แก่

1. HTML Server Controls เป็นคอนโทรลที่สร้างองค์ประกอบเบื้องต้นสำหรับเว็บเพจ สามารถใช้ภาษา HTML แทนก็ได้ แต่จะไม่มีความสามารถในการถูกควบคุม หรือทำงานร่วมกับโค้ดที่ทำงานฝั่งเซิร์ฟเวอร์ได้
2. Web Server Controls เป็นองค์ประกอบสำหรับเว็บเพจ แต่มีลักษณะที่ซับซ้อน หากสร้างด้วย HTML ก็จะต้องยุ่งยากมาก การสร้างด้วย Web Server Controls จึงช่วยอำนวยความสะดวกในการสร้างได้อย่างมากนอกจากนอกจากนี้ Web Server Controls ยังสามารถทำงานที่ซับซ้อน และสอดคล้องกับโค้ดใน ASP.NET ได้เป็นอย่างดี
3. Validation Control เป็นคอนโทรลที่ใช้สำหรับตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้กรอก ก่อนที่จะนำข้อมูลนั้นไปประมวลผลต่อไป
4. User Controls เป็นชุดของคอนโทรลที่มีอยู่แล้วใน ASP.NET ซึ่งเราสามารถนำคอนโทรลนั้นๆมาสร้างเป็นไฟล์แยกเก็บชุดคอนโทรลที่ใช้ซ้ำๆกันทุกหน้าได้ เช่น สร้างเมนู หรือ ทูลบาร์ เป็นต้น

ส่วน HTML Code : โค้ดที่ทำงานฝั่งบราวเซอร์

ในเอกสาร ASP.NET ก็มีความจำเป็นที่จะต้องใช้โค้ด HTML ประกอบด้วยเช่นกัน เพื่อใช้ในการจัดรูปแบบการแสดงผลบางอย่าง เพราะเอกสาร ASP.NET นั้นผลลัพท์ก็เพื่อจะสร้างเป็น HTML ดังนั้นเรื่อง HTML ก็จึงแยกไม่ขาดจากการพัฒนาเว็บเพจ โค้ดที่เป็น HTML จะไม่ต้องถูกประมวลโดยเซิร์ฟเวอร์แต่จะถูกส่งไปให้บราวเซอร์ของผู้ใช้โดยตรง

ส่วน Server Side Comments : หมายเหตุในเอกสาร ASP.NET

Server Side Comments เป็นส่วนที่ช่วยให้เราแทรกหมายเหตุ หรือคำอธิบายโปรแกรมภายในเอกสาร ASP.NET ได้ โดยสิ่งที่อยู่ระหว่างแทรกเปิดและแทรกปิดของ Server Side Comments ไม่ว่าจะเป็นโค้ดคำสั่งของ Asp.NET หรือข้อความใดๆ จะไม่ถูกทำการประมวลผลที่เซิร์ฟเวอร์ และจะไม่ปรากฏแสดงผลที่หน้าเว็บเพจด้วย ซึ่งจะว่าไปแล้ว ก็จะทำหน้าที่เหมือนกับการเขียนหมายเหตุในภาษาโปรแกรมทั่วไปนั่นเอง ซึ่ง Server Side Comments มีรูปแบบการใช้งานดังนี้

<%-- หมายเหตุ หรือโค้ดที่ไม่ต้องการให้แสดง --%>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Serer Side Include Directive : นำเข้าไฟล์ในเอกสาร

ส่วนนี้เป็นเทคนิคที่ใช้สำหรับนำเข้าหรือแทรกโค้ดที่บรรจุในไฟล์ภายนอกเข้ามาในเอกสาร ASP.NET ในตำแหน่งที่ประกาศ Server Side Include ซึ่งมีรูปแบบการใช้งานดังนี้

<!--#include file l virtual = ชื่อไฟล์ -->

การใช้งาน include ให้ประกาศไว้ภายนอกส่วน Block ของ ASP.NET

ส่วน Data Binding Expression : เชื่อมโยงพรอพเพอร์ตี้กับเว็บเพจ

Data Binding Expression เป็นตัวเชื่อมโยงระหว่างพรอพเพอร์ตี้ใดๆ กับเว็บเพจASP.NET รวมทั้งพรอพเพอร์ตี้ของ Server Controls และแหล่งข้อมูลซึ่งการทำงานจะกิดขึ้นเมื่อมีการเรียกใช้งานเมธอด DataBinding หน้าตาของส่วนที่เป็น Data Binding ในเอกสาร ASP.NET จะมีลักษณะการใช้งานดังนี้

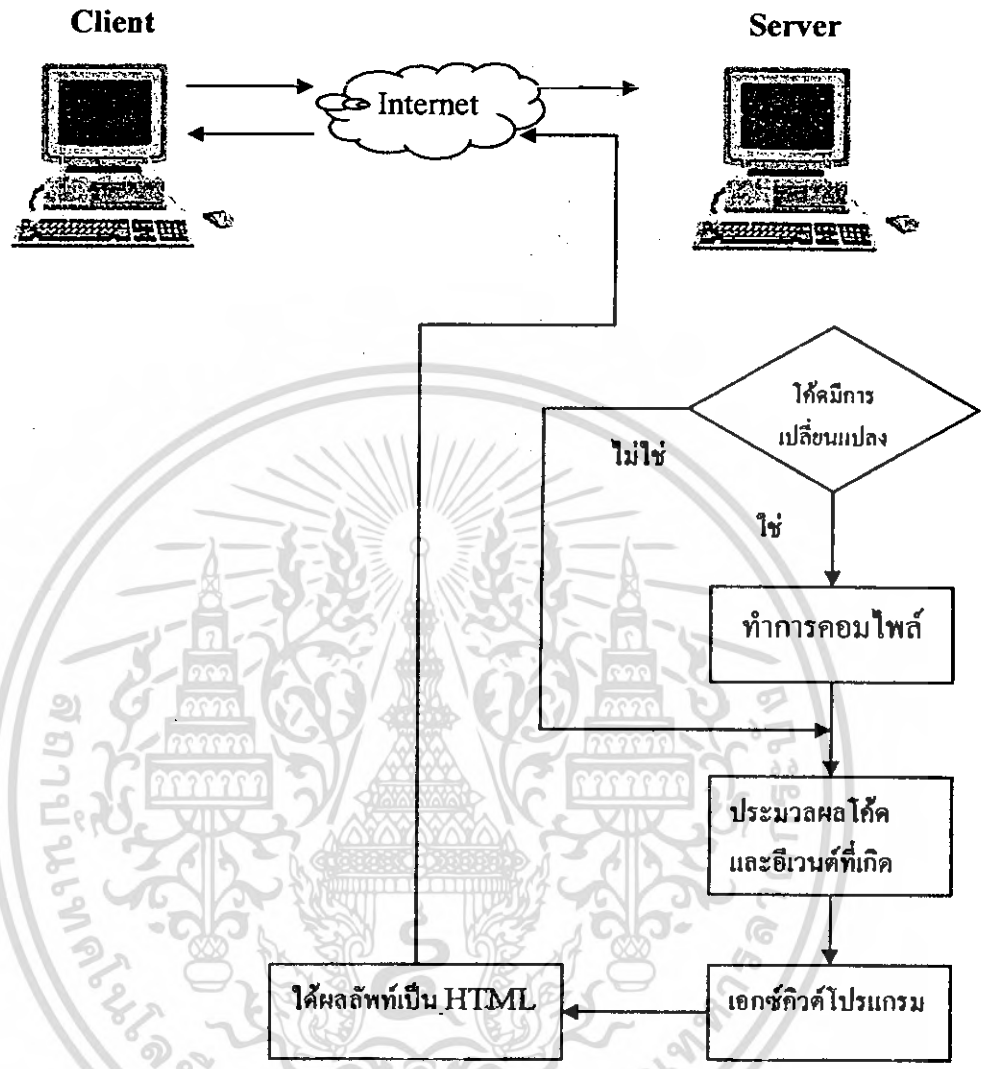
<%#data-binding expression%>

2.9 ลักษณะการทำงานของเว็บ ASP.NET

2.9.1 การทำงานที่เกิดขึ้นเมื่อเรียกใช้งานเว็บเพจ ASP.NET

เมื่อผู้ใช้มีการเรียกใช้เว็บเพจ ASP.NET ครั้งแรกโค้ดทั้งหมดในส่วน Code Declaration Block ของเอกสาร ASP.NET จะถูกแปลหรือคอมไฟล์ให้เป็น MSIL และสุดท้ายจะได้เป็นภาษาเครื่อง จากนั้นก็มีการตรวจสอบอีเวนต์หรือเหตุการณ์ที่เกิดขึ้นหรือถูกส่งมาจากฝั่งไคลเอนต์แล้วจึงทำการเอกต์คิวซ์โปรแกรมหรือสั่งให้โปรแกรมทำงานให้สอดคล้องกับอีเวนต์ที่ได้รับมา ซึ่งนี่ก็เป็นอีกสาเหตุหนึ่งที่ทำให้การเรียกใช้งานเว็บพจ ASP.NET ในครั้งแรกจะค่อนข้างช้ากว่าการเรียกใช้งานในครั้งต่อๆ ไป แต่เมื่อมีการเรียกเว็บเพจในครั้งที่สองและครั้งที่สามต่อๆไป จะข้ามขั้นตอนในการคอมไฟล์โค้ดจึทำให้การทำงานเร็วกว่าครั้งแรก และเร็วกว่าการทำงานแบบแปลทีละบรรทัดใน ASP แบบเดิมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงขั้นตอนการทำงานที่เกิดขึ้นเมื่อมีการเรียกวีบเพจ ASP.NET

2.9.2 การทำงานในรูปแบบ View State

View State เป็นส่วนที่ใช้อธิบายว่ามีอะไรเกิดขึ้น และอย่างไรกับแบบฟอร์มก่อนที่จะมีการส่งข้อมูลไปยังเซิร์ฟเวอร์และแสดงผลหรือรีเฟรชหน้าเว็บเพจใหม่ ซึ่งในเว็บเพจสมัยก่อนเมื่อกรอกข้อมูลลงในแบบฟอร์มแล้วกดปุ่มส่งข้อมูลและกลับมายังหน้าเว็บเพจเดิมอีกข้อมูลทุกอย่างที่กรอกไว้จะหายไป เพราะหลักการทำงานร่วมกันระหว่างเว็บเซิร์ฟเวอร์และเว็บบราวเซอร์นั้น เป็นแบบไม่มีการจดจำข้อมูลที่ได้รับส่งกัน ซึ่งเราเรียกการทำงานในลักษณะนี้ว่า Stateless

หากต้องการให้การทำงานมีลักษณะการจดจำที่เราเรียกว่า View State เราต้องเขียนโค้ดทำการดึงข้อมูลที่ส่งให้เซิร์ฟเวอร์กลับมาแสดงในฝั่งไคลเอนต์เอาเอง หรืออีกวิธีหนึ่งก็อาจจะใช้งานออบเจ็กต์ Session เข้าช่วยในเรื่องนี้ก็ได้ ซึ่งใน ASP.NET นั้นเราไม่จำเป็นต้องจัดการเกี่ยวกับการ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ในทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับส่งข้อมูลข้ามเว็บเพจ ASP.NET จะเก็บและส่งผ่านข้อมูลให้เราโดยอัตโนมัติ เราจึงไม่ต้องทำเอง แต่ได้ผลลัพท์ดังที่เราเคยทำด้วยความยุ่งยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

3.1 ข้อมูลเบื้องต้น

เนื่องจากโครงงานที่ทำขึ้นเกี่ยวข้องกับระบบร้านอาหาร ดังนั้นก่อนที่จะทำโครงงานนี้ทางคณะผู้จัดทำจึงต้องทำการศึกษาค้นคว้าและเก็บรวบรวมความรู้เรื่องอาหาร รวมถึงวิเคราะห์ถึงปัญหาต่างๆที่เกิดขึ้น และนอกจากนี้โครงงานได้มีการนำเครื่องพีดีเอ (Personal Digital Assistants) มาประยุกต์ใช้กับโครงงานด้วยซึ่งมีรายละเอียดดังนี้

3.1.1 วิเคราะห์ข้อมูลของร้านอาหาร

ในปัจจุบันธุรกิจร้านอาหารนับได้ว่าเป็นธุรกิจที่น่าสนใจอีกธุรกิจหนึ่ง เนื่องจากมีความเกี่ยวข้องกับชีวิตประจำวันของมนุษย์ ซึ่งเมื่อวิเคราะห์ปัญหาที่พบในธุรกิจนี้ แล้วเราจะพบว่าปัญหาหลักจะเกี่ยวข้องกับการบริการ ซึ่งเป็นจุดสำคัญและเป็นจุดขายของร้านอาหาร ดังนั้น การให้บริการที่ทำให้เกิดความประทับใจแก่ลูกค้าจึงเป็นสิ่งที่สำคัญ ซึ่งการบริการของร้านอาหารมีส่วนที่สำคัญอยู่ 2 ส่วน

3.1.1.1 การรับรายการอาหารและเครื่องดื่ม

สำหรับรูปแบบการรับรายการอาหารที่เกิดขึ้น จะมีบริกรทำการรับรายการอาหาร หลังจากนั้นบริกรก็จะทำการจดรายการอาหารตามที่ลูกค้าสั่งด้วยมือลงไปในใบรายการสั่งอาหาร แล้วบริกรก็จะนำส่งต่อไปยังห้องครัว และบาร์เครื่องดื่ม ถ้าร้านอาหารนั้นมีการแบ่งแยกห้องครัว หัวหน้าพ่อครัวก็จะเป็นคนจำแนกงานต่างๆให้ในแต่ละครัว ซึ่งการรับคำสั่งในวิธีนี้อาจเกิดปัญหาดังกล่าวดังนี้

1. หัวหน้าพ่อครัวอ่านลายมือของบริกรไม่ออก จุดนี้อาจทำให้เกิดการทำรายการอาหารที่ผิดพลาดจากรายการที่สั่งไว้ ซึ่งทำให้เกิดความล่าช้าในการบริการอันเนื่องมาจากการทำอาหารไม่ตรงกับความต้องการของลูกค้า

2. เมื่อลูกค้าต้องการสั่งอาหารเพิ่มเติม อาจจะต้องเสียเวลารอเป็นเวลานานกว่าที่จะได้สั่งอาหารเพิ่มเติมทำให้การบริการไม่เป็นที่น่าพอใจแก่ลูกค้า

3. เมื่อลูกค้าต้องการยกเลิกรายการอาหารที่สั่งไป บริกรจะต้องเดินไปที่ห้องครัว เพื่อดูว่ารายการอาหารที่สั่งไปนั้นได้ทำไปแล้วหรือยัง และเดินกลับมาบอกให้ลูกค้าทราบว่าสามารถยกเลิกรายการอาหารนั้นได้หรือไม่ แทนที่ลูกค้าจะทราบผลได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อร้านมีลูกค้าจำนวนมากทำให้การบริการนั้นไม่ทั่วถึงและไม่รวดเร็ว เกิดความล่าช้า แต่ถ้าหากจ้างพนักงานเพิ่มก็จะทำให้เกิดค่าใช้จ่ายเพิ่มขึ้น

5. เมื่อลูกค้าต้องการทราบส่วนประกอบของอาหารของรายการอาหารนั้นๆ บริกรอาจจะไม่ทราบถึงส่วนประกอบของรายการอาหารนั้นๆได้

6. การให้บริการต่างๆของบริกร จะเกี่ยวข้องกับครัว และพนักงานคิดเงิน (Cashier) ถ้าหากหาวิธีที่บริกรไม่ต้องเดินไปที่ครัว ก็จะทำให้การบริการรวดเร็วขึ้น

7. ในกรณีที่รายการอาหารที่สั่งไม่ครบถ้วนตามที่ลูกค้าสั่ง บริกรจะไม่สามารถทราบได้เลยถ้าไม่มีการทักท้วงจากลูกค้า

3.1.1.2 การชำระเงิน

ในส่วนของการชำระเงินนั้น มีลำดับขั้นตอนดังนี้

1. ลูกค้าจะทำการเรียกบริกรเพื่อทำการชำระเงิน

2. บริกรทำการแจ้งแก่พนักงานคิดเงินให้ทราบว่าลูกค้าโต๊ะใดต้องการชำระเงิน

3. ลูกค้าตรวจดูรายการอาหารว่าครบถ้วนหรือไม่ ถ้าไม่ครบถ้วนก็ทำการทักท้วง หลังจากนั้นก็ทำการชำระเงิน

4. บริกรนำเงินมาส่งให้พนักงานคิดเงิน

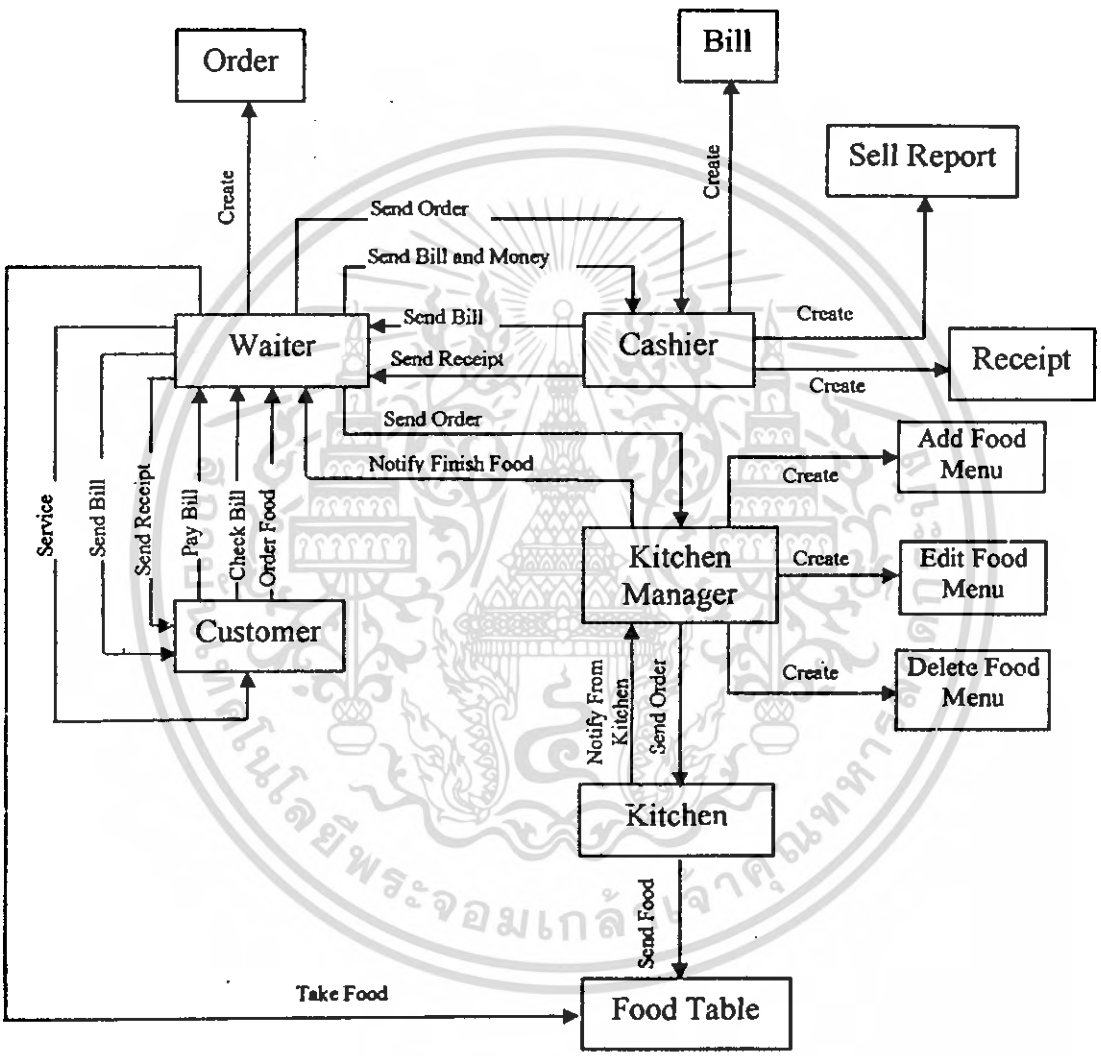
5. บริกรนำเงินทอนมาให้แก่ลูกค้า

จากขั้นตอนที่เกิดขึ้นต่างๆ จะทำให้เกิดความล่าช้า และมีเรื่องการเดินไปกลับของบริกร ดังนั้นเมื่อลูกค้าต้องการชำระเงิน พนักงานคิดเงินก็ควรที่จะทราบทันที พร้อมกับแจ้งใบรายการทั้งหมดให้แก่ลูกค้าทราบได้ทันที จะทำให้การบริการเป็นไปอย่างรวดเร็วและสะดวกมากขึ้น นอกจากนี้สำหรับในส่วนเรื่องการบันทึกรายการชำระเงินนั้น โดยมากแล้วจะเป็นเพียงการจดบันทึก ซึ่งการทำในลักษณะนี้อาจจะทำให้เกิดความผิดพลาดทางด้านข้อมูลขึ้น

ดังนั้นหากพัฒนาโปรแกรมและอุปกรณ์ซึ่งช่วยลดปัญหาดังกล่าวได้ จึงคาดว่าน่าจะได้รับความสนใจจากผู้ประกอบการร้านอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 โครงสร้างโดยทั่วไปของระบบ

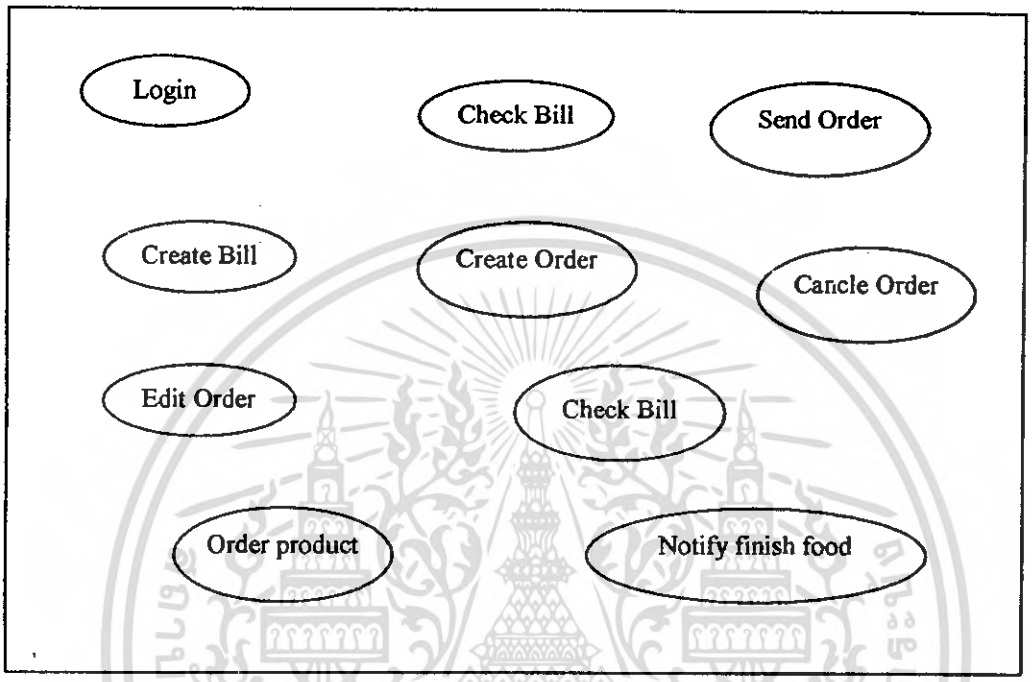


รูปที่ 3.1 โดเมนโมเดล (Domain Model)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

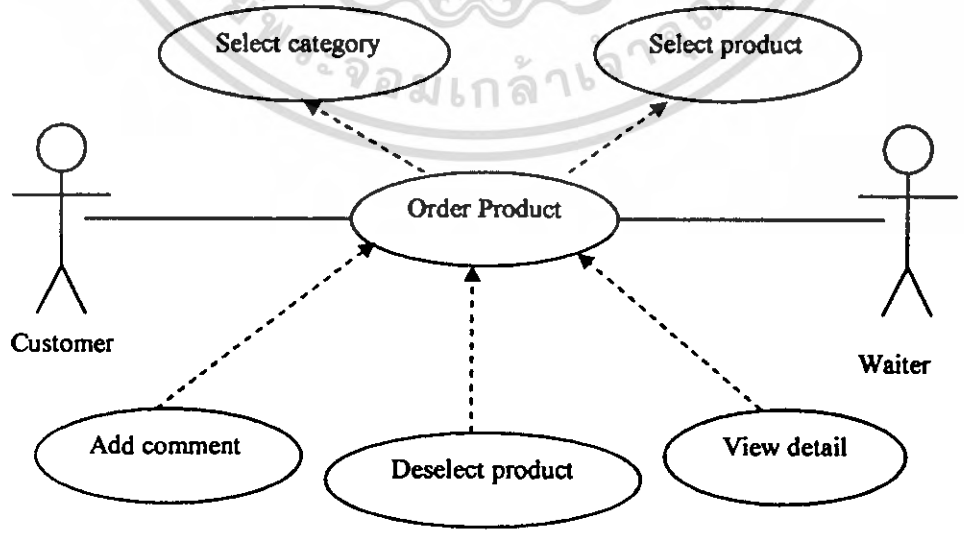
3.2 การออกแบบระบบ

3.2.1 ไฮเลเวลยูสเคสไดอะแกรมของระบบ(Hight level Use Case Diagram)



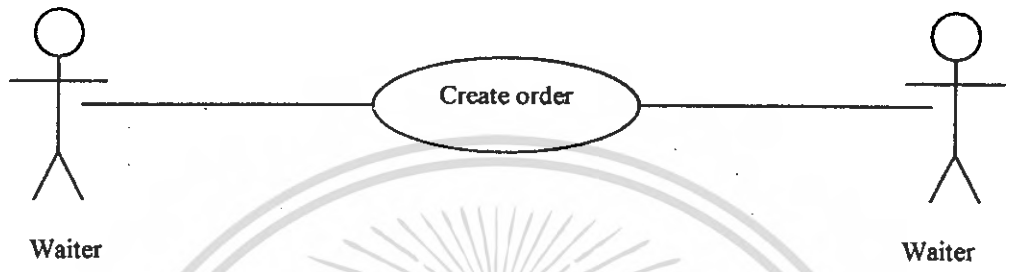
รูปที่ 3.2 แสดงไฮเลเวลยูสเคส (Hight level Use Case)

3.2.2 ยูสเคสไดอะแกรม (Use Case Diagram)

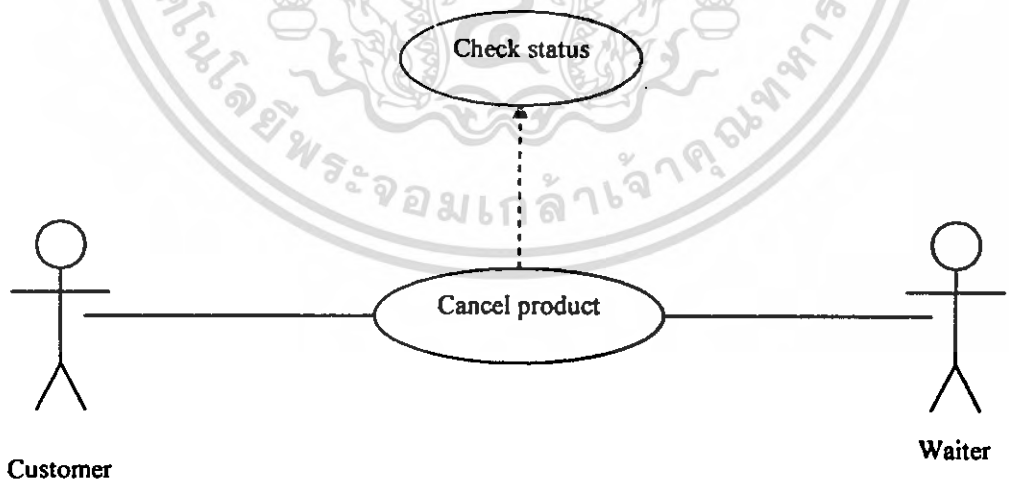


รูปที่ 3.3 ยูสเคสไดอะแกรมสำหรับการจัดการเกี่ยวกับรายการสั่งสินค้า(Order product)

เอกสารนี้เป็น[illegible]ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

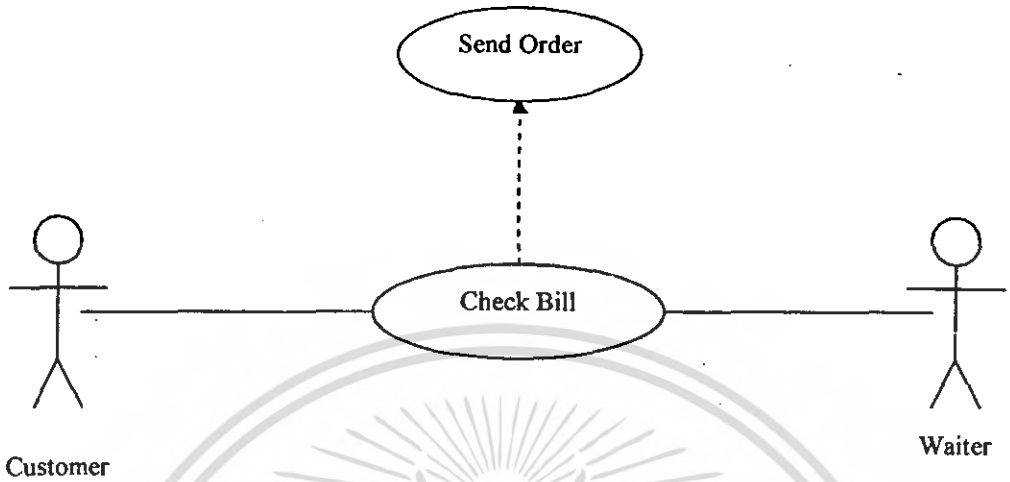


รูปที่ 3.4 ยูสเคสไดอะแกรมการจัดการเกี่ยวกับการสร้างรายการสั่งสินค้า(Create Order)

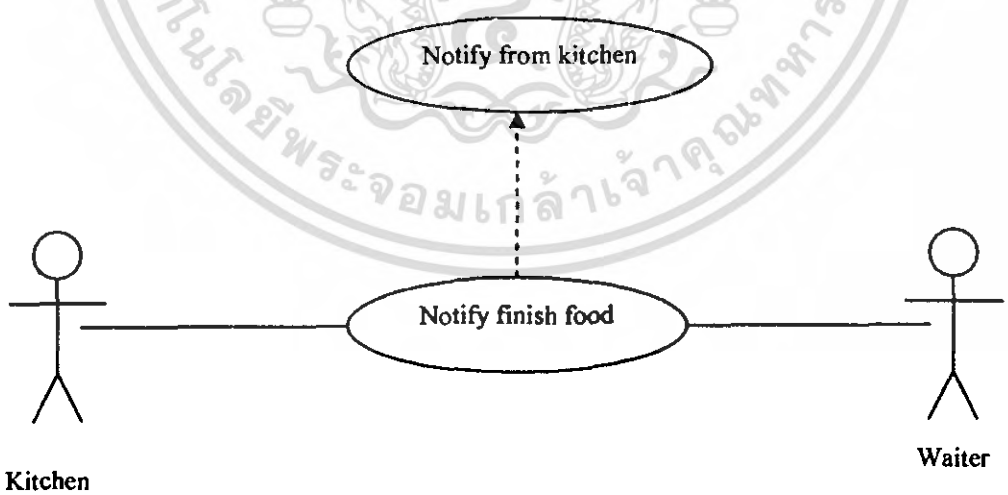


รูปที่ 3.5 ยูสเคสไดอะแกรมการจัดการเกี่ยวกับการยกเลิกรายการสั่งสินค้า(Cancel Product)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

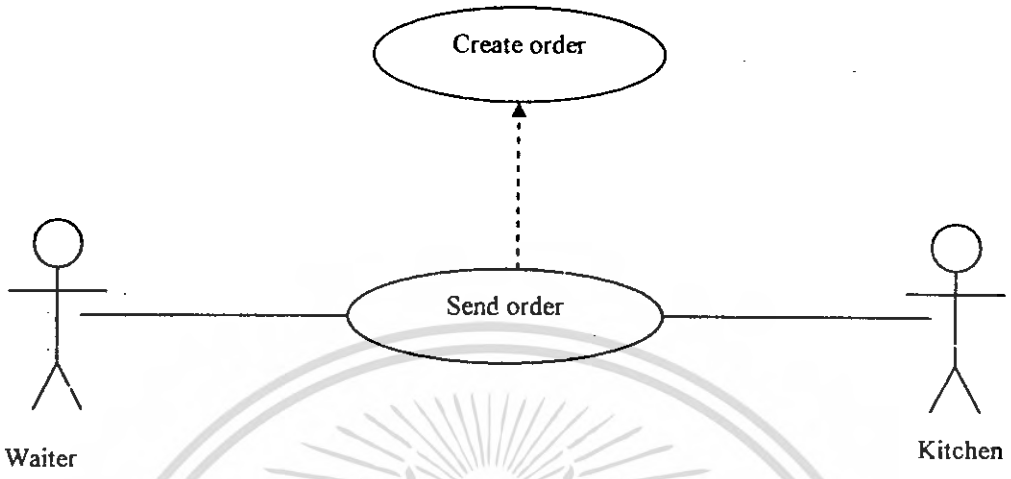


รูปที่ 3.6 ยูสเคสไดอะแกรมสำหรับการเรียกเก็บเงิน(Check Bill)

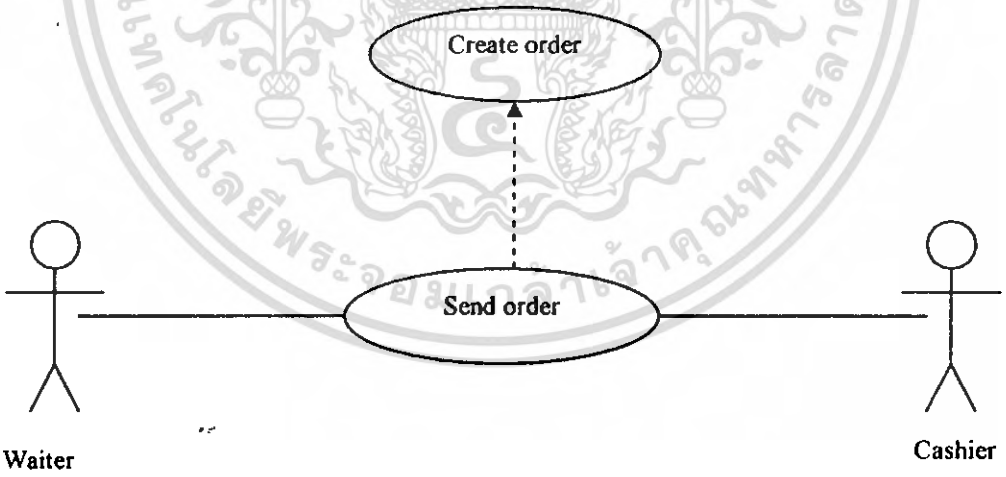


รูปที่ 3.7 ยูสเคสไดอะแกรมสำหรับแจ้งการทำอาหารเสร็จ(Notify Finish Food)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



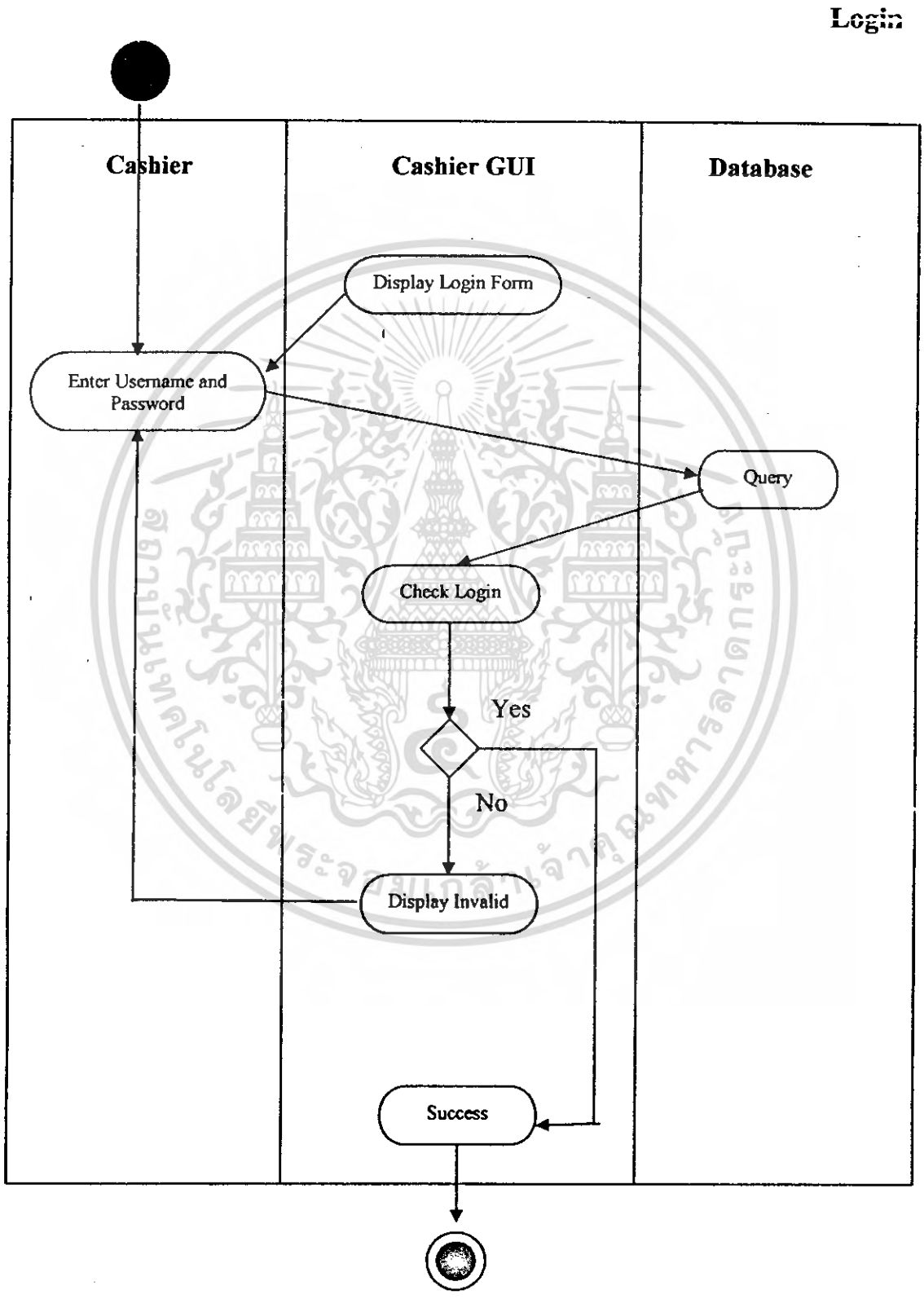
รูปที่ 3.8 ยูสเคสไดอะแกรมสำหรับการส่งรายการอาหารจากบริกรไปห้องครัว(Send Order)



รูปที่ 3.9 ยูสเคสไดอะแกรมการส่งรายการอาหารจากบริกรไปยังพนักงานคิดเงิน(Send Order)

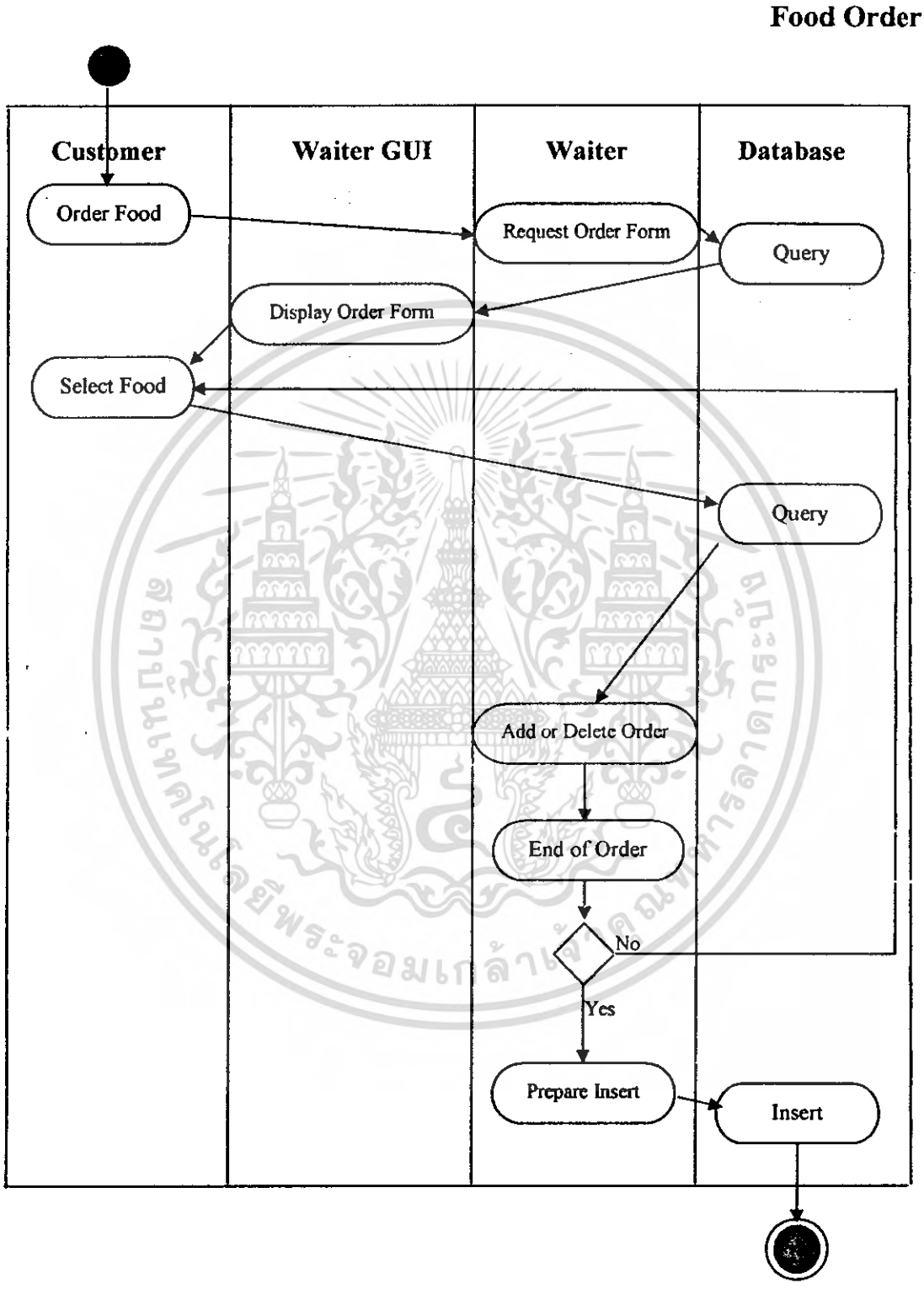
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 แอคทิวิตี้ไดอะแกรม(Activity Diagram)



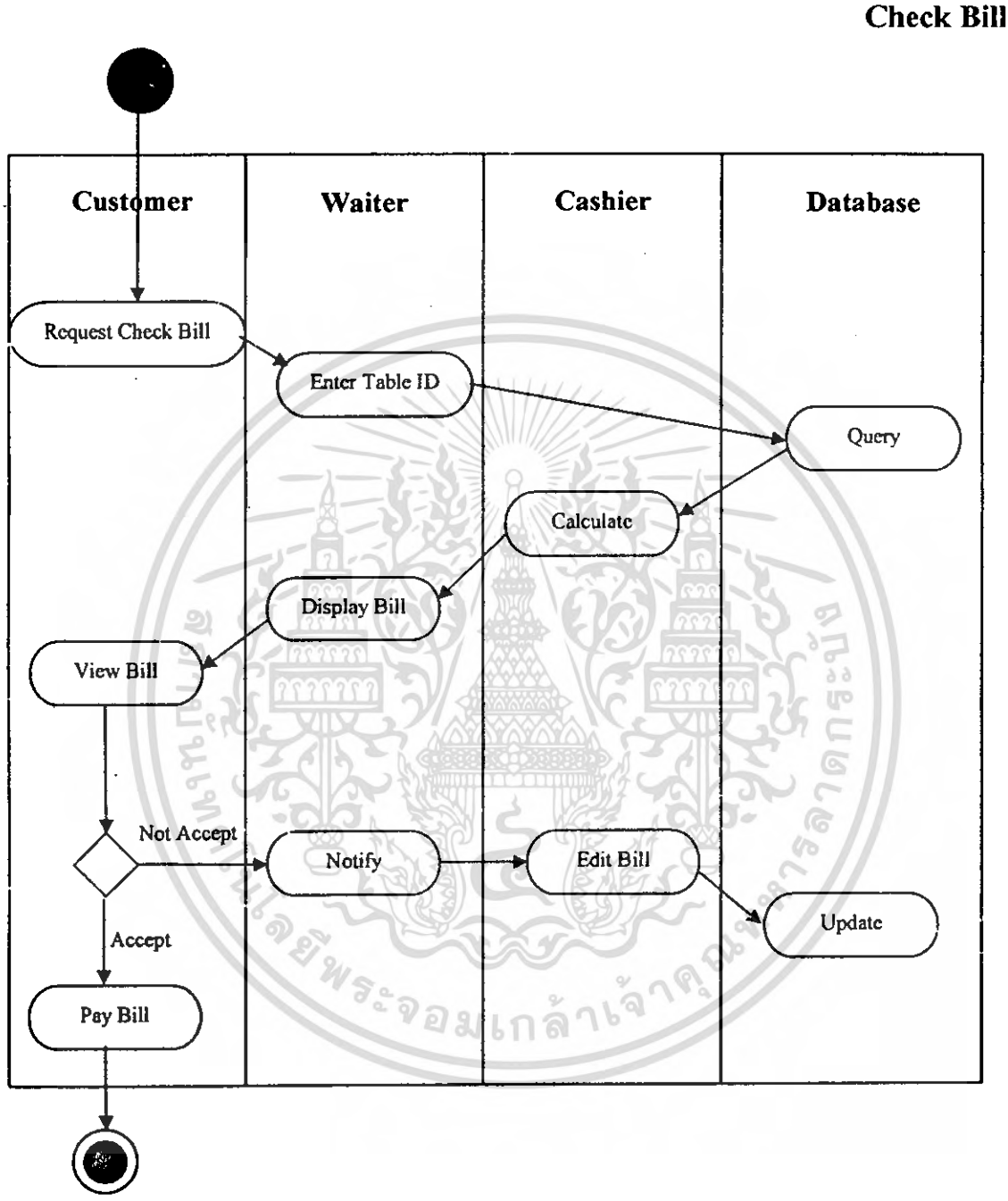
รูปที่ 3.10 แอคทิวิตี้ไดอะแกรมสำหรับการล็อกอินเข้าใช้งานระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



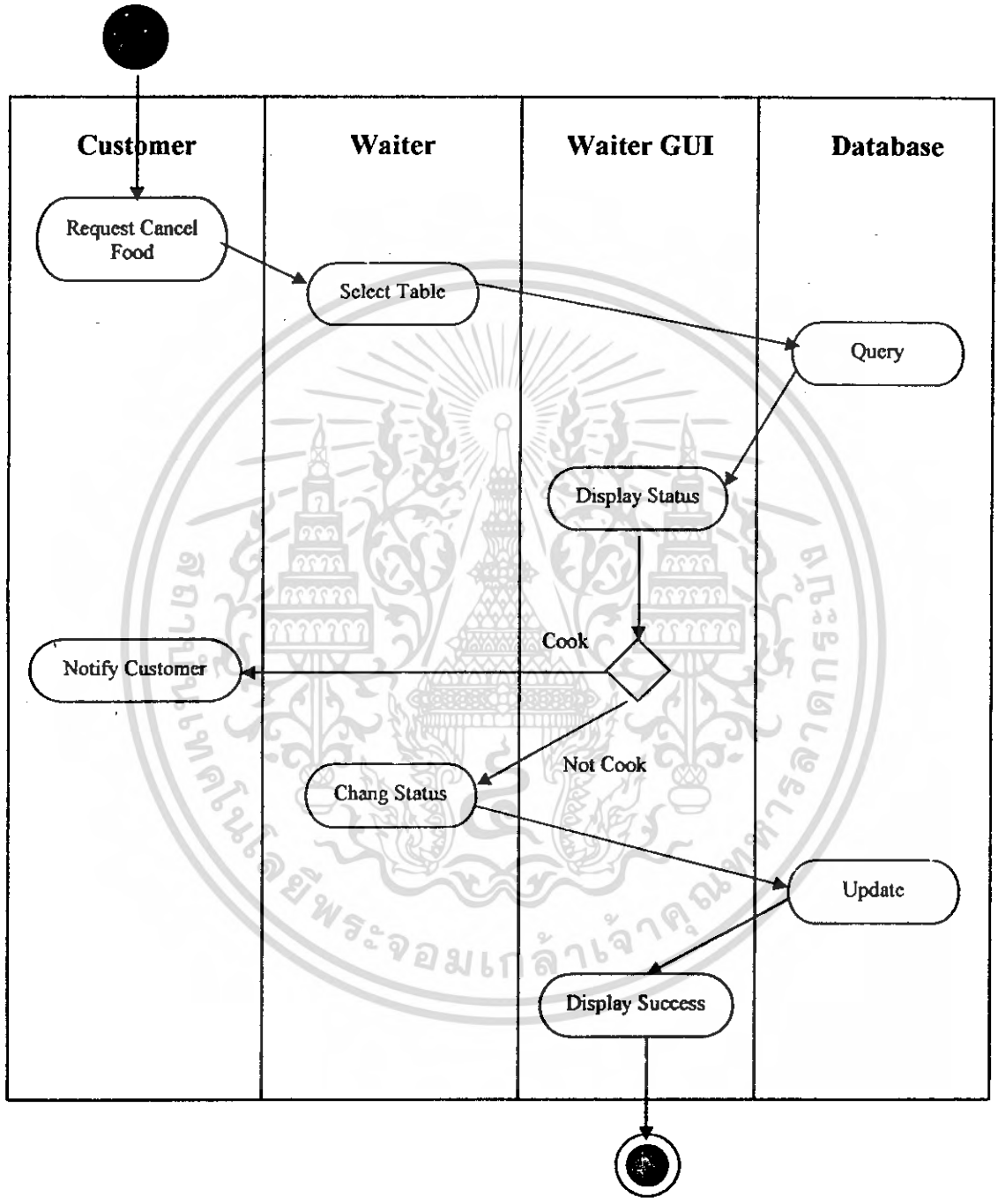
รูปที่ 3.11 แอคทิวิตี้ไดอะแกรมสำหรับการสั่งอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แอคทิวิตี้ไดอะแกรมสำหรับการเรียกเก็บเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แอคทิวิตี้ไดอะแกรมสำหรับการยกเลิกรายการอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 ดาต้าดิกชันนารี (Data Dictionary)

ตารางที่ 3.1 ตารางข้อมูลชนิดอาหาร

Name	Type	Key	Null	Size
รหัสชนิดอาหาร	Long Integer	PK	False	0
รหัสอาหาร	Long Integer	PK	False	0
ชื่ออาหาร	Text		True	50
ราคาต่อหน่วย	Long Integer		True	0
สถานะอาหาร	Text		True	50
รายละเอียดอาหาร	Text		True	50

ตารางที่ 3.2 ตารางข้อมูลโต๊ะ (Table)

Name	Type	Key	Null	Size
ลำดับที่	Long Integer		True	0
รหัสโต๊ะ	Long Integer	PK	False	0
รหัสชนิดอาหาร	Long Integer	PK	False	0
รหัสอาหาร	Long Integer	PK	False	0
ชื่ออาหาร	Text		True	50
สถานะอาหาร	Text		True	50
จำนวนที่สั่ง	Long Integer		True	0
ราคาต่อหน่วย	Long Integer		True	0
ราคารวม	Currency		True	0
เวลาที่สั่งอาหาร	Text	PK	False	50
อาหารเสร็จเวลา	Text		True	50
ชื่อพนักงานที่สั่ง	Text		True	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 ตารางข้อมูลพนักงาน (Employee)

Name	Type	Key	Null	Size
รหัสพนักงาน	Long Integer	PK	False	0
ชื่อพนักงาน	Text		True	50
เวลาเข้าทำงาน	Text		True	50
เวลาเลิกงาน	Text		True	50
วันที่มาทำงาน	Text		True	50

ตารางที่ 3.4 ตารางข้อมูลห้องครัว

Name	Type	Key	Null	Size
ลำดับที่	Auto Number	PK	False	0
รหัสโต๊ะ	Long Integer	PK	False	0
รหัสชนิดอาหาร	Long Integer	PK	False	0
รหัสอาหาร	Long Integer	PK	False	0
ชื่ออาหาร	Text		True	50
สถานะอาหาร	Text		True	50
จำนวนที่สั่ง	Long Integer		True	0
เวลาที่สั่งอาหาร	Text	PK	False	50
อาหารเสร็จเวลา	Text		True	50

ตารางที่ 3.5 ตารางข้อมูลวัตถุดิบ (InStock)

Name	Type	Key	Null	Size
รหัสสินค้า	Long Integer	PK	False	0
จำนวนคงเหลือ	Long Integer		True	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

จากการดำเนินงานที่ผ่านมาได้ระบบจัดการร้านอาหาร ซึ่งแบ่งการทำงานเป็นส่วนของพนักงาน ส่วนของบาร์เครื่องดื่ม ส่วนของห้องครัวและส่วนของการจัดการระบบ โดยทุกส่วนนี้จะมีความสัมพันธ์กันมีการทำงานร่วมกันซึ่งสามารถอธิบายการทำงานได้ดังนี้

4.1 ส่วนของพนักงาน

4.1.1 ส่วนของหน้าจอสำหรับการเข้าสู่ระบบ (Login)

รูปที่ 4.1 หน้าจอสำหรับการล็อกอินเข้าใช้ระบบ

พนักงานสามารถเข้าใช้ระบบได้โดยทำการล็อกอินเข้าใช้งานระบบ ซึ่งพนักงานต้องใส่ยูสเซอร์เนม (Username) และรหัสผ่าน(Password) ที่ถูกต้อง ซึ่งถ้าหากใส่ยูสเซอร์เนม หรือ รหัสผ่านผิดก็จะมีข้อความเตือนให้ใส่ใหม่ ซึ่งยูสเซอร์เนมและรหัสผ่านนี้จะต้องถูกต้องตรงกันกับที่มีอยู่ในส่วนของเซิร์ฟเวอร์ เมื่อทำการใส่ยูสเซอร์เนมและรหัสผ่านถูกต้องเรียบร้อยแล้วก็จะสามารถเข้าสู่ระบบการใช้งานของส่วนต่างๆของเมนูได้ ซึ่งพนักงานสามารถทำการใช้งานระบบในการที่จะสั่งอาหาร เครื่องดื่ม ชำระเงิน ทำการแก้ไขหรือยกเลิกรายการอาหารที่สั่งไปแล้วได้ ซึ่งส่วนของการล็อกอินเข้าใช้งานระบบนี้จะเป็นหน้าแรกของการทำงานในส่วนของพนักงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ส่วนของหน้าจอหลักสำหรับพนักงาน

รูปที่ 4.2 หน้าจอหลักสำหรับพนักงาน

ส่วนนี้จะเป็นส่วนของหน้าจอหลักในการให้บริการซึ่งพนักงานจะสามารถใช้งานระบบต่างๆ ได้เช่น ทำการสั่งอาหาร เครื่องดื่ม เช็คบิล และเข้าไปตรวจสอบที่ส่วนของห้องครัวและบาร์เครื่องดื่มได้

โดยในส่วนของหน้าจอหลักนี้จะประกอบไปด้วยเมนูที่สำคัญอยู่ 2 ส่วนคือส่วนแรกเป็นส่วนของการสั่งอาหารซึ่งจะประกอบด้วยเมนูสั่งอาหารและเมนูสั่งเครื่องดื่ม ส่วนที่สองจะเป็นส่วนของเมนูหลักที่สำคัญได้แก่ ส่วนของเมนูหลักที่จะเข้าไปยังหน้าของเมนูหลักซึ่งก็คือหน้าที่แสดงนี้เอง เมนูเช็คบิล ส่วนนี้จะเข้าไปยังส่วนของการเลือกโต๊ะที่ต้องการตรวจสอบรายการอาหารหรือต้องการที่จะคิดเงิน ส่วนของเมนูห้องครัวจะเข้าไปยังส่วนของห้องครัวเพื่อตรวจสอบรายการอาหารว่ามีการปรุงแล้วหรือยัง และส่วนของแผนกน้ำ เป็นเมนูที่เข้าไปยังส่วนของบาร์เครื่องดื่มเพื่อตรวจสอบรายการเครื่องดื่มที่ทำการสั่งไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ส่วนของรายการอาหาร

รูปที่ 4.3 หน้าจอแสดงเมนูเลือกชนิดอาหาร

ส่วนนี้จะแสดงเมนูของชนิดของอาหารต่างๆ ที่มีอยู่ภายในร้าน โดยจะแยกเป็น 2 ส่วนเพื่อสะดวกในการที่จะเลือกทำรายการ โดยส่วนด้านล่างจะประกอบไปด้วยส่วนของการเลือกชนิดของรายการอาหารซึ่งจะแบ่งออกเป็นเมนูอาหารชนิดต่างๆ ประกอบไปด้วย อาหารชนิดผัด ชนิดต้ม ชนิดยำ ชนิดทอด ชนิดนึ่ง ชนิดตุ๋น และยังประกอบด้วยเมนูที่เป็นอาหารตามสั่งและอาหารว่างไว้อีกด้วย ส่วนด้านบนก็จะเป็นส่วนของเมนูหลักๆ ประกอบด้วยเมนูหลักเพื่อที่จะกลับไปยังเมนูหน้าเมนูให้บริการหลัก เมนูเช็คบิลเพื่อที่จะทำการตรวจสอบรายการอาหารที่มีการสั่งจากโต๊ะต่างๆ และทำการคิดเงิน และเมนูห้องครัวเพื่อที่จะเข้าไปตรวจสอบว่ารายการอาหารที่สั่งนั้นมีการปรุงหรือยังในกรณีที่ลูกค้าต้องการที่จะยกเลิกรายการที่สั่งไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 ส่วนของรายการอาหารชนิดต่างๆ

รูปที่ 4.4 หน้าจอแสดงรายการอาหารชนิดต่างๆ

เมื่อทำการเลือกชนิดของอาหารที่จะสั่งเช่น อาหารประเภทผัด อาหารประเภทต้ม อาหารประเภททอด อาหารประเภทนึ่ง เป็นต้นแล้วก็จะปรากฏหน้าจอของรายการอาหารประเภทนั้นขึ้นมาเพื่อที่จะให้ทำการเลือกรายการอาหารจากการสั่งของลูกค้า ซึ่งจะมีรายการอาหารที่มีอยู่ภายในร้านซึ่งเป็นชนิดที่เราเลือก ในส่วนของหน้าจอนี้สามารถที่จะเลือกโต๊ะได้ว่าลูกค้าโต๊ะใดสั่งและสั่งเป็นจำนวนเท่าใด เมื่อทำการเลือกรายการอาหารแล้ว รายการอาหารที่ถูกเลือกก็จะไปปรากฏที่ส่วนของห้องครัวเพื่อที่จะสั่งให้ทางห้องครัวทำการปรุงอาหารนั้นๆ ตามรายการที่สั่ง โดยส่วนบนจะเป็นเมนูของการเลือกโต๊ะที่สั่งอาหาร และมีเมนูสั่งอาหารเพื่อที่จะกลับไปยังหน้าของการเลือกชนิดอาหารที่จะสั่ง และเมนูเครื่องดื่มเพื่อไปยังส่วนของการเลือกเครื่องดื่ม ซึ่งในหน้าของรายการอาหารนี้หากรายการอาหารภายในร้านมีจำนวนมากจำเป็นที่จะต้องมีการเลื่อนแถบด้านข้างเพื่อทำการเลือกอาหารที่อยู่ด้านล่างของหน้าจอ เนื่องจากความสามารถในการแสดงผลของ Pocket PC มีจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 ส่วนของการเลือกเครื่องดื่ม

รูปที่ 4.5 หน้าจอแสดงรายการเครื่องดื่ม

หน้าจอนี้จะเป็นหน้าจอที่แสดงรายการของเครื่องดื่มประเภทที่เราเลือกว่าต้องการเครื่องดื่มที่ทางร้านมีไว้ให้บริการ ส่วนบนก็จะเป็นส่วนของเมนูสั่งอาหาร เมนูหลัก และการเลือกโต๊ะที่สั่ง ส่วนล่างก็จะเป็นส่วนของรายการเครื่องดื่ม ราคาและจำนวนที่ต้องการสั่ง ซึ่งรายการเครื่องดื่มที่สั่งทั้งหมดจะไปปรากฏที่ส่วนของบาร์เครื่องดื่ม เพื่อให้บริกรหรือพนักงานทำการจัดเครื่องดื่มเพื่อนำมาให้บริการแก่ลูกค้าตามที่ลูกค้าโต๊ะนั้นๆ ทำการสั่งไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 ส่วนของการคิดเงิน

รูปที่ 4.6 หน้าจอสำหรับทำการคิดเงินโต๊ะต่างๆ

หน้าจอสำหรับทำการคิดเงินนี้เป็นส่วนของพนักงานที่ต้องการที่จะทำการเช็คยอดการใช้บริการของลูกค้า เมื่อลูกค้าต้องการที่จะทำการจ่ายเงินก็จะแจ้งแก่พนักงาน พนักงานก็จะเข้ามายังส่วนของการคิดเงิน ซึ่งจะแยกออกเป็นโต๊ะต่างๆ ที่มีทั้งหมดภายในร้านเพื่อให้สามารถเลือกว่าโต๊ะใดที่ต้องการให้คิดเงิน โดยส่วนของหน้าจอก็จะแบ่งเป็นสองส่วน ส่วนบนจะประกอบด้วยส่วนของเมนูหลัก เมนูห้องครัว และเมนูบาร์เครื่องดื่ม ส่วนล่างได้แก่ส่วนของเมนูโต๊ะต่างๆที่ต้องการคิดค่าบริการซึ่งในส่วนของหน้าจอนี้นอกจากจะใช้ในการคิดค่าบริการแล้วยังทำหน้าที่ตรวจสอบแก้ไขหรือเปลี่ยนแปลงข้อมูลกรณีที่ทางลูกค้าหรือผู้มาใช้บริการต้องการทราบรายการอาหารที่ตัวเองได้สั่งไปทั้งหมดว่ามีอะไรบ้างหากทางลูกค้าเกิดทักท้วงหรือมีการผิดพลาดเกิดขึ้นทางบริกรก็สามารถที่จะทำการแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.7 ส่วนของโต๊ะต่างๆ

รูปที่ 4.7 หน้าจอแสดงส่วนของโต๊ะต่างๆ

เมื่อพนักงานทำการเลือกโต๊ะที่จะทำการคิดเงินแล้วเข้ามายังส่วนของโต๊ะที่จะทำการคิดเงิน ก็จะปรากฏรายการอาหารที่ได้มีการสั่งไว้ทั้งหมดของโต๊ะนั้นๆ ซึ่งประกอบไปด้วย รายการอาหารที่สั่งทั้งหมด จำนวนของอาหารที่สั่งแต่ละชนิด ราคาต่อหน่วยของอาหารแต่ละชนิด ซึ่งรายการทุกอย่างจะสามารถที่จะทำการแก้ไขได้กรณีที่มีการผิดพลาดอันเนื่องมาจากการสั่งโดยสามารถที่จะทำการยกเลิกรายการอาหารนั้นๆได้ นอกจากนี้ยังมีส่วนของการเช็คเงินทั้งหมดที่ลูกค้าจะต้องจ่ายทั้งหมด และยังมีส่วนของการลบรายการอาหารทั้งหมดของโต๊ะนั้นในกรณีที่ลูกค้าได้ทำการชำระเงินเรียบร้อยแล้ว โดยส่วนบนก็จะเป็นส่วนของเมนูหลักๆ ซึ่งได้แก่เมนูหลัก ส่วนของการเช็คบิล และส่วนของห้องครัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.8 ส่วนของการตรวจสอบที่ห้องครัว

รูปที่ 4.8 หน้าจอแสดงส่วนของการตรวจสอบที่ห้องครัว

ในส่วนของการตรวจสอบที่ห้องครัวนั้นจะเป็นส่วนที่ตรวจสอบรายการอาหารทั้งหมดที่สั่งเข้ามาจากบริกรที่ให้บริการอยู่ภายในร้านโดยจะมาจากโต๊ะต่างๆ เรียงตามการสั่งมาเพื่อที่ทางห้องครัวจะได้ทำการปรุงอาหารได้ตามลำดับการสั่งก่อนหลัง โดยถ้าหากว่ารายการอาหารใดที่ทำการปรุงแล้วก็จะปรากฏสถานะว่าปรุงแล้ว ซึ่งก็จะไม่สามารถที่จะทำการยกเลิกรายการอาหารนั้นได้ แต่ถ้าหากว่ารายการอาหารนั้นยังไม่ได้มีการปรุงก็จะไม่ปรากฏสถานะเพื่อที่ทางพนักงานหรือลูกค้าที่ต้องการที่จะยกเลิกหรือเปลี่ยนแปลงรายการอาหาร ซึ่งการที่จะยกเลิกหรือเปลี่ยนแปลงรายการได้นั้นรายการอาหารนั้นจะต้องยังไม่มีการปรุงเกิดขึ้นเท่านั้นถ้าหากรายการอาหารนั้นมีการปรุงแล้วจะไม่สามารถที่จะยกเลิกรายการอาหารได้ ซึ่งทางพนักงานสามารถที่จะเข้ามาตรวจสอบดูได้ และทำการแก้ไขได้เลย เมื่อทางพนักงานทำการยกเลิกข้อมูลหรือรายการอาหารนั้นก็จะถูกลบออกจากส่วนของห้องครัวจะไม่ปรากฏที่ห้องครัวเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

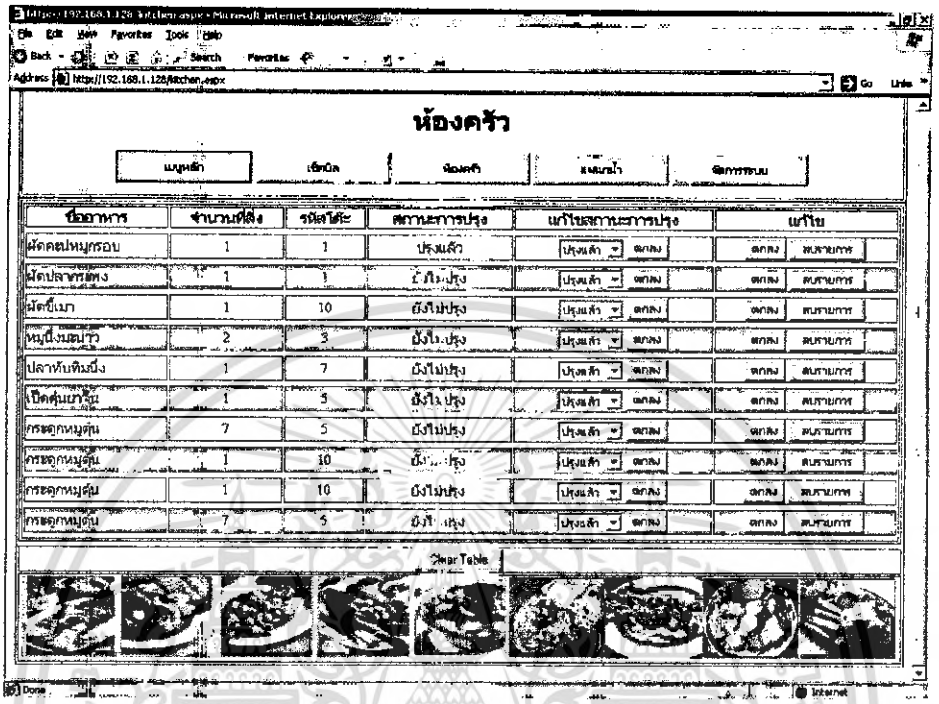
4.1.9 ส่วนของการตรวจสอบที่บาร์เครื่องดื่ม

รูปที่ 4.9 หน้าจอแสดงส่วนของการตรวจสอบที่บาร์เครื่องดื่ม

ส่วนของการตรวจสอบที่บาร์เครื่องดื่มจะเป็นการเข้าไปทำการตรวจสอบว่ารายการเครื่องดื่มที่ได้ถูกสั่งไปจากโต๊ะต่างๆหรือโต๊ะที่เราต้องการจะตรวจสอบว่าได้มีการนำมาส่งหรือยังหากยังไม่ได้มีการส่งก็จะปรากฏรายการของเครื่องดื่มนั้นๆ อยู่ แต่ถ้าหากว่ารายการเครื่องดื่มนั้นได้มีการนำมาส่งแล้วก็จะไม่ปรากฏรายการของเครื่องดื่มนั้นๆ หรือหากทางลูกค้าต้องการที่จะยกเลิกรายการเครื่องดื่มใดๆ โดยการที่จะยกเลิกได้ก็ต่อเมื่อรายการเครื่องดื่มนั้นยังไม่ได้มีการจัดส่งหรือยังมีรายการของเครื่องดื่มนั้นๆ อยู่ในรายการของเครื่องดื่มที่ส่วนของบาร์เครื่องดื่มเท่านั้น รายการเครื่องดื่มที่ไม่มีในรายการแสดงว่าได้มีการจัดหรือเตรียมที่จะส่งแล้วไม่สามารถที่จะทำการยกเลิกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของห้องครัว

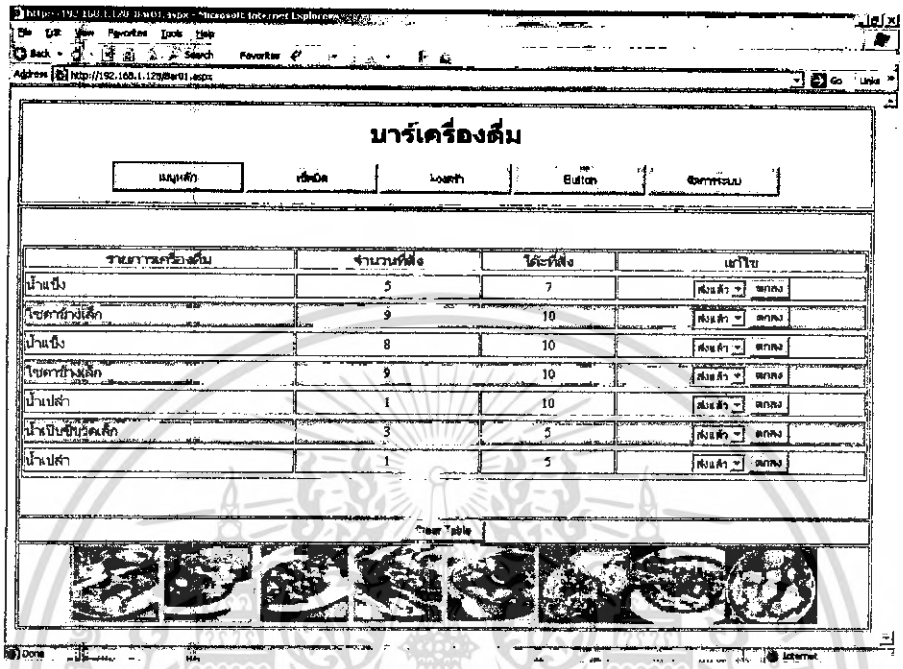


รูปที่ 4.10 แสดงส่วนของห้องครัว

ส่วนนี้จะเป็นส่วนที่อยู่ภายในห้องครัวโดยจะทำการแสดงรายการอาหารทั้งหมดที่มีการสั่งเข้ามาจากทางบริกรแสดงตามลำดับเพื่อที่จะให้ทางห้องครัวได้ทำการปรุง เมื่อทางห้องครัวเห็นรายการอาหารที่ปรากฏขึ้นและจะเริ่มทำการปรุงก็จะทำการตอบตกลง สถานะของรายการอาหารนั้นก็จะเปลี่ยนไปจากที่ไม่ปรากฏสถานะคือยังไม่มีการปรุงเป็นสถานะที่ปรุงแล้วปรากฏในส่วนของสถานะของการปรุง ซึ่งจะเป็นการแสดงว่ารายการอาหารนั้นจะไม่สามารถที่จะทำการยกเลิกหรือแก้ไขได้จากทางลูกค้า เมื่อรายการอาหารนั้นทำการปรุงเสร็จเรียบร้อยแล้วทางห้องครัวจะทำการยกเลิกรายการอาหารนั้น รายการอาหารนั้นก็จะถูกลบออกไปจากส่วนของห้องครัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ส่วนของบาร์เครื่องดื่ม



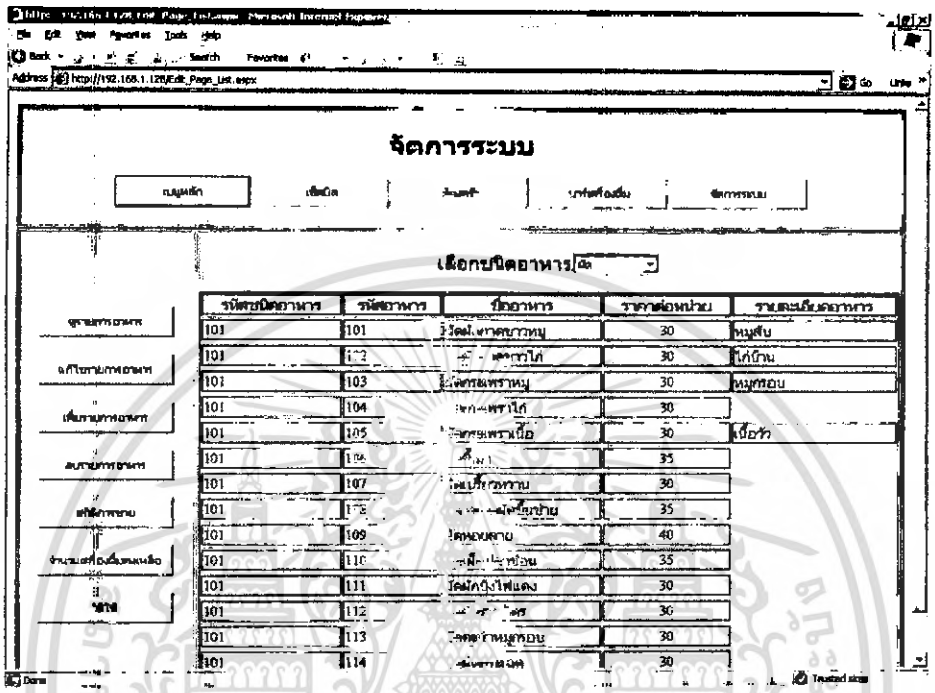
รูปที่ 4.11 แสดงส่วนของบาร์เครื่องดื่ม

ส่วนนี้จะเป็นส่วนที่แสดงรายการของเครื่องดื่มทั้งหมดที่มีการสั่งเข้ามาจากทางบริกรซึ่งจะปรากฏรายการเครื่องดื่ม จำนวนที่สั่ง โต๊ะที่สั่ง และส่วนของการจัดส่ง ซึ่งเมื่อทางพนักงานที่บาร์เครื่องดื่มเห็นรายการของเครื่องดื่มที่ปรากฏขึ้นก็จะทำการเตรียมการจัดส่งแล้วทำการตอบตกลงเพื่อที่จะทำการลบรายการของเครื่องดื่มออกจากส่วนของบาร์เครื่องดื่มซึ่งเป็นการแสดงว่ารายการเครื่องดื่มนั้นได้ทำการจัดส่งแล้วไม่สามารถที่จะทำการเปลี่ยนแปลงหรือแก้ไขใดๆ ในกรณีที่ลูกค้าต้องการที่จะยกเลิกหรือเปลี่ยนรายการของเครื่องดื่มนั้น ซึ่งในส่วนของบริกรสามารถที่จะเข้ามาตรวจสอบรายการได้ในส่วนบาร์เครื่องดื่มนี้ได้ หากรายการใดยังไม่ถูกจัดส่งก็จะสามารถที่จะเปลี่ยนแปลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ส่วนการจัดการระบบ

4.4.1 ส่วนของการดูรายการอาหาร

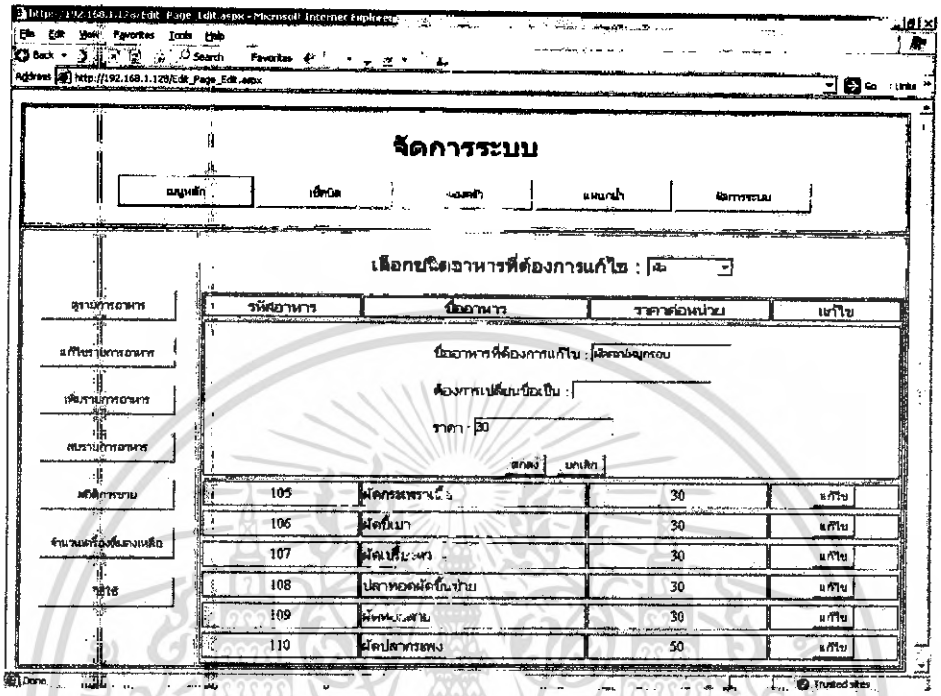


รูปที่ 4.12 แสดงส่วนของการดูรายการอาหารทั้งหมด

ส่วนนี้จะเป็นการแสดงรายการของอาหารทั้งหมดที่มีอยู่ภายในร้านซึ่งจะถูกจัดแยกไว้เป็นชนิดต่างๆ ซึ่งแต่ละชนิดจะแสดงรหัสชนิดอาหาร รหัสอาหาร ราคาต่อหน่วยและอาจจะมีรายละเอียดของอาหารนั้นๆ ด้วย โดยจะสามารถที่จะเลือกชนิดของอาหารที่เราต้องการจะดูรายการได้ทั้งหมดรวมทั้งรายการของเครื่องดื่มต่างๆ ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ส่วนของการแก้ไขรายการอาหาร

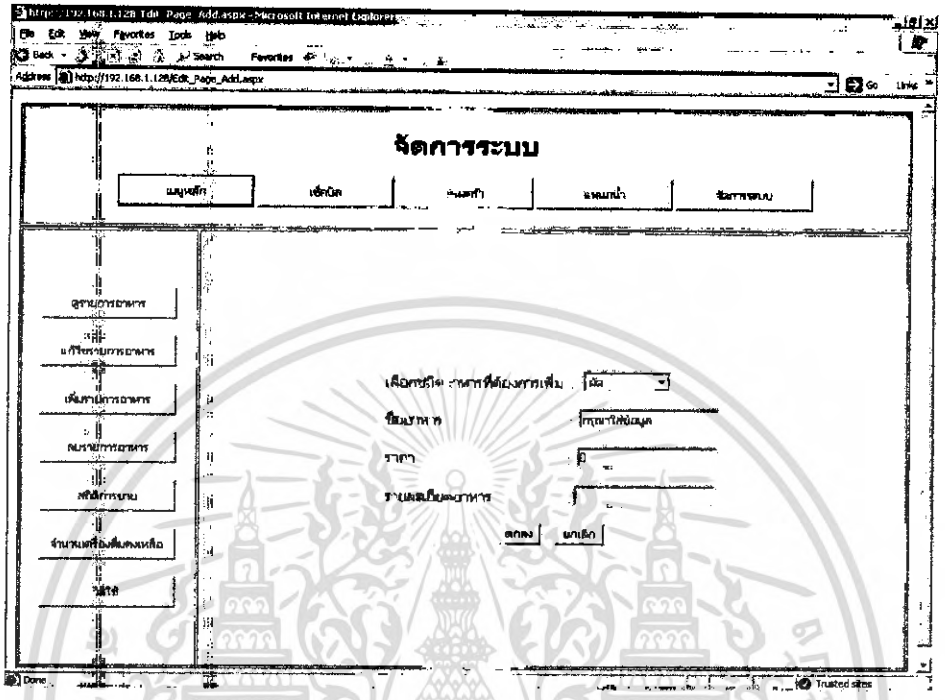


รูปที่ 4.13 แสดงส่วนของการแก้ไขรายการอาหาร

ส่วนนี้จะเป็นส่วนที่ทำการแก้ไขรายการอาหารในกรณีที่มีการเปลี่ยนแปลงชื่อหรือราคาของรายการอาหารนั้นๆ โดยสามารถเลือกชนิดและรายการที่ต้องการแก้ไข ซึ่งการแก้ไขนี้จะทำการแก้ไขได้เฉพาะชื่อ และราคาของรายการนั้นได้ เมื่อทำการแก้ไขแล้วรายการอาหารเดิมจะหายไปโดยจะถูกแทนที่ด้วยรายการอาหารมีที่แก้ไขนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ส่วนของการเพิ่มรายการอาหาร

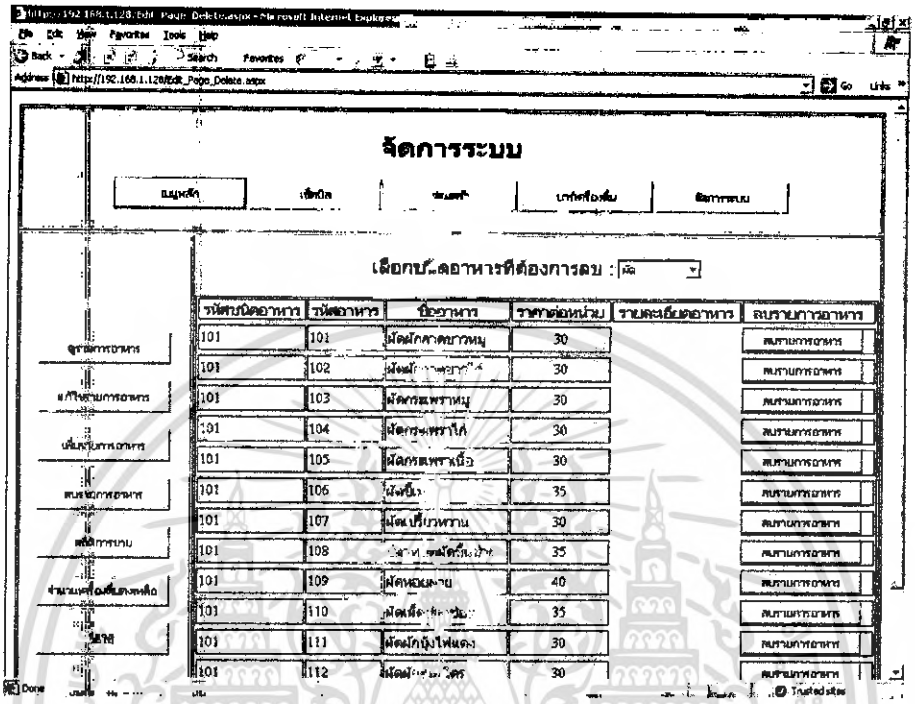


รูปที่ 4.14 แสดงส่วนของการเพิ่มรายการอาหาร

ส่วนของการเพิ่มรายการอาหารจะเป็นส่วนที่ใช้ในการที่ทางห้องครัวมีรายการอาหารใหม่เพิ่มเข้ามาในเมนูเดิมที่มีอยู่ โดยการเพิ่มรายการอาหารนั้นจะต้องเลือกชนิดของอาหารที่ต้องการเพิ่มเข้ามาเพื่ออาหารที่ทำการเพิ่มเข้าไปนั้นอยู่ที่ชนิดที่ถูกต้อง ซึ่งการเพิ่มรายการอาหารนั้นจะเป็นการเพิ่มในส่วนของชื่ออาหาร ราคาอาหาร และรายละเอียดของอาหารนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 ส่วนของการลบรายการอาหาร

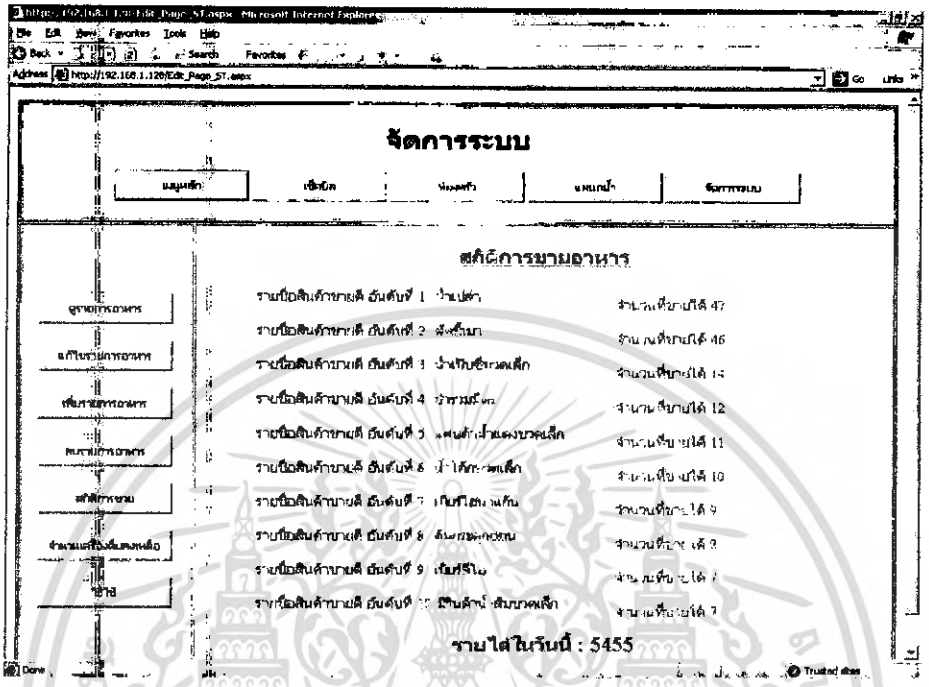


รูปที่ 4.15 แสดงส่วนของการลบรายการอาหาร

ส่วนนี้จะเป็นการแสดงส่วนของการลบรายการอาหาร ซึ่งทางห้องครัวต้องการที่จะทำการลบรายการอาหารนั้นๆ ออก โดยวัตถุประสงค์ในการลบก็แล้วแต่ทางห้องครัว ซึ่งอาจจะมาจากวัตถุดิบที่ใช้ในการปรุงอาหารนั้นๆ หมด รายการอาหารนั้นไม่ได้รับความนิยมของผู้บริโภค หรือรายการอาหารนั้นทางห้องครัวไม่สามารถที่จะทำการปรุงได้เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 ส่วนของสถิติการขายอาหาร

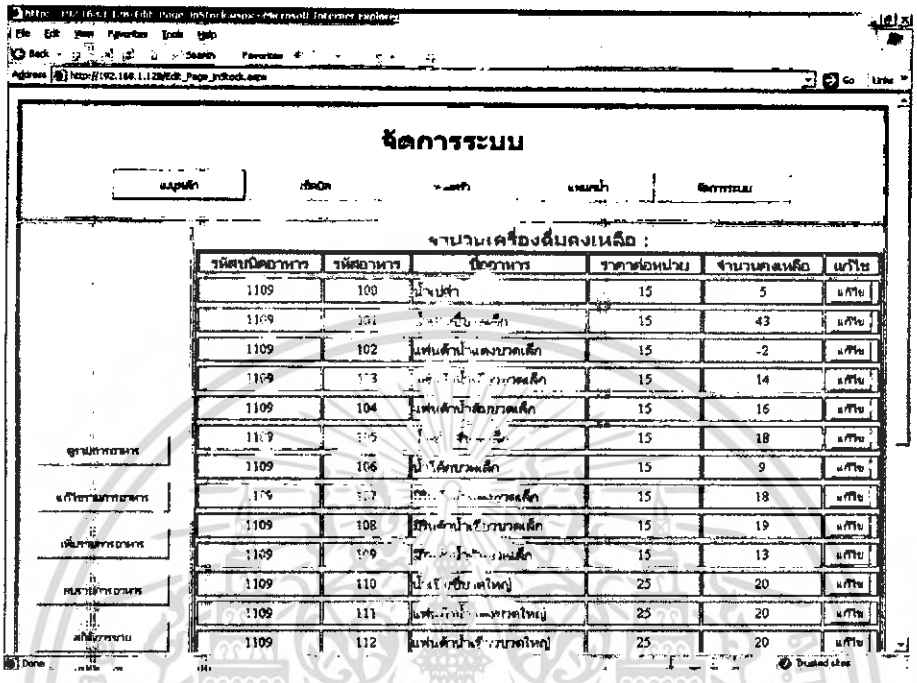


รูปที่ 4.16 แสดงส่วนของสถิติการขายอาหาร

ส่วนนี้จะแสดงรายการของอาหารที่ขายดีหรือขายได้มากที่สุดประจำวันโดยจะเรียงลำดับจากที่ขายได้มากที่สุดจำนวน 10 ลำดับ และแสดงจำนวนที่ขายได้ทั้งหมดของรายการอาหารนั้นๆ นอกจากนี้ยังมีส่วนของการแสดงจำนวนรายได้ที่ขายได้ในวันนั้นๆ ว่าในวันนั้นยอดขายทั้งหมดในร้านขายได้เท่าไรซึ่งจะแสดงออกมาเป็นจำนวนเงิน เพื่อที่จะนำยอดขายนี้ไปทำการคิดหาผลกำไรของการประกอบการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 ส่วนของรายการเครื่องดื่มคงเหลือ



รูปที่ 4.17 แสดงจำนวนเครื่องดื่มคงเหลือ

ส่วนนี้จะแสดงส่วนของจำนวนเครื่องดื่มคงเหลือภายในร้านว่าเครื่องดื่มแต่ละชนิดเหลือจำนวนเท่าไรเพื่อที่จะได้ทำการจัดซื้อหากว่าเครื่องดื่มนั้นๆ ใกล้ที่จะหมด โดยถ้าหากว่ามีการจัดซื้อเครื่องดื่มชนิดนั้นมาเพิ่มก็ทำการเพิ่มจำนวนของเครื่องดื่มนั้นเข้าไปที่ส่วนของการแก้ไขแล้วใส่จำนวนเครื่องดื่มที่ซื้อเพิ่มเข้ามา จำนวนคงเหลือของเครื่องดื่มก็จะเพิ่มขึ้นตามที่เราจัดซื้อ โดยจำนวนของเครื่องดื่มนี้จะลดลงตามจำนวนที่มีการสั่งของลูกค้าที่มาใช้บริการซึ่งเมื่อลูกค้าสั่งเครื่องดื่มเป็นจำนวนเท่าไรจำนวนของเครื่องดื่มนั้นๆ ก็จะลดลงตามจำนวนที่สั่งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิเคราะห์ผลการทดลอง

จากผลการทดลองที่ผ่านมาจะเห็นได้ว่าการใช้งานระบบ ไคลเอนต์ – เซิร์ฟเวอร์ ผ่านระบบเครือข่าย WLAN นั้นจะสามารถแบ่งการทำงานเป็นส่วนต่างตามลักษณะการใช้งานซึ่งส่วนหลักๆ มีดังนี้

1. ส่วนของพนักงาน
2. ส่วนของห้องครัว
3. ส่วนของบาร์เครื่องดื่ม
4. ส่วนของการจัดการระบบ

ซึ่งอาจกล่าวได้ว่าการติดต่อกันระหว่างเครื่องเซิร์ฟเวอร์กับเครื่องไคลเอนต์นั้นใช้การติดต่อกันแบบเว็บเซิร์ฟเวอร์(Web Server) ซึ่งเครื่องไคลเอนต์จะทำการติดต่อไปยังเซิร์ฟเวอร์โดยใช้ระบบการสื่อสารแบบไร้สาย(Wireless Lan) โดยที่ทางไคลเอนต์จะทำการดึงข้อมูลจากทางเซิร์ฟเวอร์มาแสดงผลที่หน้าจอ ให้พนักงานสามารถที่จะใช้งานในส่วนของหน้าต่างๆ ของการทำงานได้ โดยข้อมูลจะถูกเก็บไว้ในส่วนของฐานข้อมูล(Database) ที่อยู่ภายในเซิร์ฟเวอร์ เมื่อพนักงานต้องการที่จะทำการสั่งอาหารเพิ่มหรือแก้ไขรายการอาหารที่สั่งไว้แล้วก็สามารถที่จะทำการแก้ไขได้ในส่วนที่ทางเซิร์ฟเวอร์ยอมให้มีการแก้ไขได้เท่านั้น ซึ่งจะเป็นในส่วนของการเพิ่มหรือลบรายการอาหารที่สั่งไป

จากการทำโครงงานที่ผ่านมาสามารถที่วิเคราะห์ปัญหาที่เกิดขึ้นของโครงงานได้ดังนี้

1. ปัญหาจากข้อจำกัดทางอุปกรณ์ฮาร์ดแวร์และซอร์ฟแวร์ที่ใช้ในการทำงานร่วมกันในระบบเครือข่ายซึ่งต้องอาศัยความเข้ากันได้ของเครือข่ายกับอุปกรณ์ที่ใช้ เช่น ข้อจำกัดของเครือข่ายแบบไร้สายที่จะมีมากกว่าระบบเครือข่ายแบบมีสาย

2. ข้อจำกัดของการแสดงผลและการจัดเก็บข้อมูล ที่มักเกิดความผิดพลาด ในบางประเภทตัวแปรของข้อมูลที่ใช้ในการแสดงผล

ในการวิเคราะห์การติดต่อกันระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์นั้นทั้งในส่วนของการแสดงผลและส่วนของฐานข้อมูลนั้นมีความสำคัญทั้งสิ้น ซึ่งในปริญญานิพนธ์เล่มนี้จะเป็นการติดต่อกับฐานข้อมูลที่มีขนาดเล็กซึ่งมีข้อมูลไม่มากนัก ดังนั้นหากต้องการที่จะใช้กับฐานข้อมูลที่มีขนาดใหญ่กว่านี้ก็ควรมีการศึกษาเกี่ยวกับระบบฐานข้อมูลที่สามารถใช้กับข้อมูลที่มีขนาดใหญ่กว่านี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. จำลอง ครูอุดสาหะ, "ASP.NET ฉบับโปรแกรมเมอร์", เคทีพี คอมพ์แอนด์ คอนซัลท์, 2545, 498หน้า.
2. ณัฐภัชร ณ เขาวงกต," IIS 6 แค่คลิก-คลิก-คลิก ก็บริหารเซิร์ฟเวอร์ได้แล้ว", วิตตี้ กรุ๊ป, 2548, 336 หน้า.
2. ทวีชัยหงส์สุมาลย์, "อินไซท์ ASP.NET ฉบับสมบรูณ์", โปรวิชัน, 2546, 608หน้า.
3. ธวัชชัย สุริยะทองธรรม, "พัฒนาเว็บแอพพลิเคชัน ASP.NET", ซัคเซส มีเดีย, 2548, 587หน้า.
4. มณีโชติ สมานไทย, "คู่มือการออกแบบฐานข้อมูลและภาษา SQL ฉบับผู้เริ่มต้น", อินโฟเพรส, 2546, 304หน้า.
5. สุทธา ศรีวิริยาจารย์, "Microsoft Windows Server 2003", ซีเอ็ดยูเคชั่น, 2548, 256หน้า.
6. สุรสิทธิ์ ดิวประสพศักดิ์, "อินไซท์ Visual Basic.NET ฉบับสมบูรณ์", โปรวิชัน, 2546, 536หน้า.

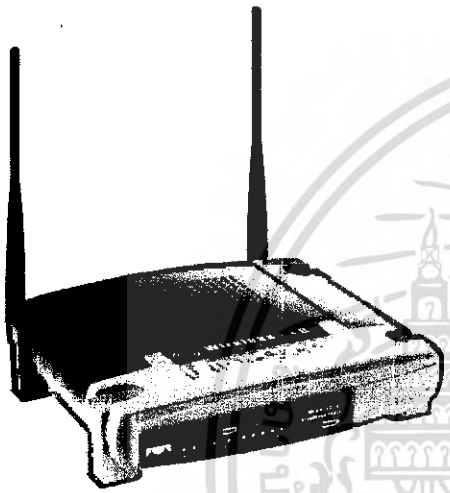
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LINKSYS®
A Division of Cisco Systems, Inc.

An Internet-Sharing Router and Switch, with Universal Wireless Access (A, B, and G)



The Dual-Band Wireless A+G Broadband Router is like four devices in one box! The Router function lets you securely share one high-speed Internet connection among your entire network, while the 4-port full duplex 10/100 Switch jump-starts your wired-Ethernet network. Connect four PCs directly, or daisy-chain out to more hubs and switches to create as big a network as you need.

The Dual-Band Wireless A+G Broadband Router also contains two Wireless Access Points, supporting all three wireless networking specifications. The first Access Point uses the 2.4GHz radio band, supporting both the popular and inexpensive Wireless-B (802.11b) standard at 11Mbps, and the new, almost five times faster, Wireless-G (802.11g) at 54Mbps. The second Access Point radio operates in the 5GHz band, and supports Wireless-A (802.11a) networking, also at 54Mbps. Since the two radios operate in different bands, they can work simultaneously, blanketing your wireless zone with bandwidth.

To protect your data and privacy, the Dual-Band Wireless A+G Broadband Router can encode all wireless transmissions with industrial strength WPA encryption. The MAC Address filter lets you decide exactly who has access to your wireless network. The Router also serves as a DHCP Server, supports VPN pass-through, and can be configured to filter internal users' access to the Internet. Configuration is a snap with the web browser-based configuration utility.

With the Linksys Dual-Band Wireless A+G Broadband Router at the center of your home or office network, you can share a high-speed Internet connection, files, printers, and multi-player games with the flexibility, speed, and security you need!

Wi-Fi CERTIFIED a b g

Designed for Microsoft® Windows® XP Media Center Edition

Dual-band, tri-standard Access Point communicates with Wireless-A (802.11a), Wireless-B (802.11b), and Wireless-G (802.11g) wireless networks

Protect your wireless investment while preparing your infrastructure for the future

Built-in 4-port Switch jump-starts your wired network, while the Router functionality securely shares your high-speed Internet connection with wired and wireless computers

Advanced Security: Stateful Packet Inspection Firewall, 152-bit Wireless data encryption (WEP), VPN pass-through, Internet access control and Wireless filters

Dual-Band Wireless A+G Broadband Router

Product Data

Model No. **WRT55AG**

CISCO SYSTEMS

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dual-Band Wireless
Broadband Router

Features

- Performance Investment Protection: Compatibility with Wireless-A (802.11a), Wireless-B (802.11b) and Wireless-G Standards
- Built-In 4-Port 10/100 Switch Supports Wired Ethernet Clients
- All Ethernet Ports Support Auto-MDI/MDI-X -- No Need for Crossover Cables or Uplink Port
- Wireless Security with WEP or WPA Encryption
- Enhanced Security Management Functions: Anonymous Internet Requests, Web, MAC Address and IP Address Filtering, and NAT Technology
- Access Your Corporate Network Remotely through Virtual Private Networking (VPN) -- Supports IPSec and PPTP Pass-Through
- Easily Configurable through a Setup Wizard or Web Browser.
- Supports QoS (Quality of Service)
- Supports Static and Dynamic Routing (RIP)

Specifications

Model Number	WRT55AG
Standards	IEEE 802.3, IEEE 802.3u, IEEE 802.11a, IEEE 802.11g, IEEE 802.11b
Ports	Internet: One 10/100 RJ-45 Port for Cable/DSL Modem LAN: Four 10/100 RJ-45 Switched Ports One Power Port
Buttons	One Reset Button
Cabling Type	UTP Cat 5 Ethernet cable or better
LEDs	Power, DMZ, Interent, Ethernet (1, 2, 3, 4)
Security Features	Internet Policy, Wireless Filters, Port Filters
Encryption Key Bits	WEP and WPA, 64 and 128

Environmental

Dimensions	7.32" x 1.89" x 6.89" (186 mm x 48 mm x 175 mm)
Unit Weight	0.88 lbs (0.40 kg)
Power	External, 5V DC, 2.5A
Certifications	FCC,
Operating Temp.	32°F to 104F (0°C to 40°C)
Storage Temp.	-4°F to 158°F (-20°C to 70°C)
Operating Humidity	10% to 85% Non-Condensing
Storage Humidity	5% to 90% Non-Condensing
Warranty	3-Year Limited

Linksys
A Division of Cisco Systems, Inc.
16582 Teller Avenue
Irvine, CA 92612 USA

E-mail: sales@linksys.com
support@linksys.com

Web: http://www.linksys.com

Linksys products are available in more than 50 countries, supported by 12 Linksys Regional Offices throughout the world. For a complete list of local Linksys Sales and Technical Support contacts, visit our Worldwide Web Site at www.linksys.com.

Minimum Requirements

- One 200 MHz or Faster Processor
- 64 MB RAM
- Internet Explorer 5.5 or Netscape Navigator 6.1 or higher for web-based configuration
- CD-ROM Drive
- 802.11a, 802.11b, or 802.11g Wireless Adapter with TCP/IP Protocol Installed or
- Network Adapter with TCP/IP Protocol Installed and Category 5 Ethernet Network Cable

Package Contents

- One Dual-Band Wireless A+G Broadband Router
- One Setup CD-ROM Setup CD-ROM with Symantec Internet Security
- User Guide on CD-ROM
- One Power Adapter
- One Ethernet Network Cable
- One Quick Installation
- One Registration Card

ifications are subject to change without notice. Linksys is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries. Copyright © 2005 o Systems, Inc. All rights reserved. Other brands and product names are trademarks or registered trademarks of their respective holders.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้