

ระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม

WEB BASED MEMBERSHIP MANAGEMENT SYSTEM FOR SMALL
AND MEDIUM ENTERPRISES

นายกรกฎ วีรกุลนิยม
นางสาววิรัชญา มากจัน
นางสาวสินนภา ประดิษฐ์รัฐสินธุ์

โครงการพิเศษเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม

**WEB BASED MEMBERSHIP MANAGEMENT SYSTEM FOR SMALL
AND MEDIUM ENTERPRISES**

นายกรรณ วีรกุลนิยม

นางสาววิรัชญา มากจีน

นางสาวสินนภา ประสิทธิ์รัฐสินธุ์

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

**WEB BASED MEMBERSHIP MANAGEMENT SYSTEM FOR SMALL
AND MEDIUM ENTERPRISES**

KORAKOT WIRAKUNNIYOM

WIRANYA MAKJEEN

SINNAPA PRASITH-RATHSINT

**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE**

IN COMPUTER SCIENCE


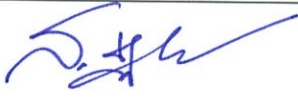

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2013

หัวข้อโครงการพิเศษ	ระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและ ขนาดย่อม Web Based Membership Management System for Small and Medium Enterprises		
ชื่อนักศึกษา	นายกรกฎ	วีรกุลนิยม	53050925
	นางสาววิริญญา	มากจิ้น	53051082
	นางสาวสินนภา	ประสิทธิ์รัฐสินธุ์	53051103
ปริญญา	วิทยาศาสตรบัณฑิต		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
อาจารย์ที่ปรึกษา	ดร.สายชล ใจเย็น		

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้
โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการ
คอมพิวเตอร์ ประจำปีการศึกษา 2556

คณะกรรมการสอบ	ลายมือชื่อ
ดร.อนันตพร หารรรษคุณาดัย ประธานกรรมการ	
ดร.สันติภูษณ์ นรบิน กรรมการ	
ดร.สายชล ใจเย็น กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงการพิเศษ	ระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม		
ชื่อนักศึกษา	นายกรกฎ	วีรกุลนิยม	53050925
	นางสาววิรัชญา	มากจิน	53051082
	นางสาวสินนภา	ประสิทธิ์รัฐสินธุ์	53051103
ปริญญา	วิทยาศาสตร์บัณฑิต		
ภาควิชา	คณิตศาสตร์และคอมพิวเตอร์		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2556		
อาจารย์ที่ปรึกษา	ดร.สายชล ใจเย็น		

บทคัดย่อ

ในปัจจุบันการซื้อสินค้าตามห้างสรรพสินค้าและร้านค้าต่างๆ จะมีการแจกบัตรสมาชิกเพื่อดึงดูดลูกค้าใหม่และลูกค้าทั่วไปให้กลับมาเป็นลูกค้าประจำของร้านและเพื่อรักษฐานลูกค้าประจำ ลูกค้าที่เป็นสมาชิกและมีบัตรสมาชิกจะได้รับสิทธิพิเศษตามเงื่อนไขของแต่ละร้านค้า ซึ่งทำให้เกิดปัญหาความยุ่งยากและไม่สะดวกในการพกพาบัตรสมาชิกจำนวนหลายๆใบ เพราะเป็นสมาชิกกับหลายๆ ร้านค้า ลูกค้าจึงมักที่จะลืมไม่พกบัตรเป็นประจำ มีความสับสนในการใช้บัตรสมาชิก เนื่องจากบัตรสมาชิกบางร้านค้ามีขนาดใกล้เคียงกัน รูปร่างเหมือนกัน สีและรูปภาพบนบัตรคล้ายๆกัน โครงการปัญหาพิเศษฉบับนี้ได้พัฒนาระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม เพื่อใช้แก้ปัญหาที่จะเกิดขึ้นและพัฒนาระบบให้มีประสิทธิภาพเพื่อความสะดวกสบายมากยิ่งขึ้น เนื่องจากลูกค้าไม่จำเป็นต้องพกพาบัตรสมาชิกติดตัวอีกต่อไป เพียงแค่ใช้เบอร์โทรศัพท์แทนบัตรสมาชิกก็สามารถได้รับสิทธิพิเศษตามเงื่อนไขของร้านค้าต่างๆ โดยระบบจะให้บริการระบบจัดการข้อมูลสมาชิกที่เป็นศูนย์กลางและระบบการบริหารจัดการร้านค้า เมื่อแต่ละร้านค้ามีการเพิ่มสมาชิกจะได้ข้อมูลสมาชิกภายในระบบเหมือนกัน

Special Project Title	Web Based Membership Management System for Small and Medium Enterprises
Students	Mr.Korakot Wirakunniyom 53050925 Miss.Wiranya Markjean 53051082 Miss.Sinnapa Prasith-rathsint 53051103
Degree	Bachelor of Science
Department	Mathematics and Computer Science, Faculty of Science
Programme	Computer Science
Academic Year	2013
Special Project Advisor	Saichon Jaiyen, Ph.D.

ABSTRACT

Currently, department stores and stores provide membership cards to attract customers to become their regular customers. Members will be given special privileges base on each store options. It is inconvenient for customers to carry many membership cards. Many customers do not want to, or forget to, carry their cards. It is confusing to use many cards because of some of them are similar in shape, color and pictures. To overcome these problems, this special project develop a membership management system for small and medium enterprises. Therefore, the customers no longer need to carry membership cards. In addition, the telephone number is used instead of a membership card to receive the membership privilege. This project will provide the membership information management system and store managing system when the new members are added to the system.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมสามารถสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอขอบพระคุณ ดร.สายชล ใจเย็น ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำปรึกษาและคำแนะนำด้านการศึกษาปัญหาการออกแบบระบบและแนวทางการแก้ไขปัญหา ตลอดจนการตรวจสอบและแก้ไขการเขียนรายงานปัญหาพิเศษเล่มนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา และผู้ปกครองที่ได้ให้คำปรึกษาในเรื่องต่างๆ รวมทั้งเป็นกำลังใจที่ดีเสมอมา

ขอบคุณเพื่อนๆ ที่ช่วยให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษ จนทำให้ปัญหาพิเศษสำเร็จลุล่วงไปได้

คณะผู้จัดทำ

มีนาคม 2557

สารบัญ

	หน้า
บทคำย่อภาษาไทย	I
บทคำย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูปภาพ	X
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของปัญหา	2
1.3 ขอบเขตของปัญหา	2
1.4 ส่วนประกอบของโครงการพิเศษ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	4
2.1 ASP.NET MVC	4
2.1.1 MVC Architecture Pattern	4
2.1.2 หลักการทำงานของ MVC	6
2.1.3 การทำงานของ ASP.NET MVC Request	6
2.1.4 หน้าที่ของ Folder	7
2.1.5 Routing System	8
2.1.6 Pass Data ระหว่าง Controller กับ View	8
2.1.7 การส่ง Data จาก View มา Controller	10
2.1.8 Return View	11
2.1.9 การสร้าง Model จากฐานข้อมูลที่มีอยู่แล้ว	12

สารบัญ (ต่อ)

	หน้า
2.2 Entity Framework (EF).....	16
2.2.1 โครงสร้างของ Entity Framework.....	16
2.2.2 Database First.....	17
2.2.3 Model First.....	17
2.2.4 Code First.....	17
2.2.5 Entity Framework 4.....	18
2.3 View and Razor Engine.....	19
2.3.1 รูปแบบของ Razor.....	19
2.3.2 Razor Syntax.....	19
2.3.3 การเข้าถึง root เสมือน.....	22
2.3.4 การแปลง Root เสมือนเป็น path บนไดรว์.....	23
2.4 Data Annotations.....	23
2.4.1 Required.....	23
2.4.2 MinLength.....	23
2.4.3 MaxLength.....	23
2.4.4 Display.....	23
2.4.5 DataType.....	24
2.4.6 Compare.....	24
2.5 jQuery.....	24
2.5.1 อธิบายการทำงานของ jQuery.....	26
2.5.2 การเลือกตำแหน่งที่ต้องกระทำด้วย Selectors.....	27
2.5.3 Method และคำสั่งของ jQuery.....	28
2.6 JavaScript.....	35
2.6.1 หลักการเขียน JavaScript.....	35
2.6.2 การเขียน JavaScript กับ Web Page.....	35

สารบัญ (ต่อ)

	หน้า
2.7 CSS.....	36
2.7.1 การเขียน css กับเว็บเพจ.....	37
2.7.2 Selector.....	41
2.7.3 การเลือกใช้ Selectors.....	44
2.8 กรณีศึกษา แอปพลิเคชัน “Stamp” บนบน โทรศัพท์สมาร์ทโฟน iOS.....	44
บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	45
3.1 แนวคิดการออกแบบระบบ.....	45
3.2 การใช้งานของระบบ.....	45
3.3.1 ระบบลูกค้า.....	45
3.3.2 ระบบร้านค้า.....	46
3.3 Use Case Diagram.....	47
3.3.1 คำอธิบาย Use Case.....	48
3.4 Entity Relationship Diagram.....	52
3.4.1 ตารางของระบบฐานข้อมูล.....	53
3.5 Software ที่เกี่ยวข้อง.....	60
บทที่ 4 การใช้งานโปรแกรม.....	61
4.1 ส่วนประกอบต่างๆจากการพัฒนาระบบ.....	61
4.1.1 โปรแกรมส่วนของลูกค้า.....	61
4.1.1.1 การสมัครสมาชิก.....	63
4.1.1.2 การใช้บริการ (Login).....	64
4.1.1.3 บัตรสะสมแต้ม.....	65
4.1.1.4 การเลือกร้านค้า.....	68
4.1.1.5 การเลือกร้านค้าแบ่งตามประเภท.....	68
4.1.1.6 การตรวจโปรโมชัน.....	69
4.1.1.7 การแก้ไขข้อมูล.....	70

สารบัญ (ต่อ)

	หน้า
4.1.2 โปรแกรมส่วนของร้านค้า.....	73
4.1.2.1 การสมัครสมาชิก.....	74
4.1.2.2 การเข้าใช้บริการ (Login).....	75
4.1.2.3 การจัดการกับสินค้า.....	77
4.1.2.4 ระบบชำระเงิน.....	79
4.1.2.5 สถิติการขาย.....	81
4.1.2.6 การจัดการกับโปรโมชั่น.....	82
4.1.2.7 การแลกเปลี่ยนสินค้า.....	85
4.1.2.8 การแก้ไขข้อมูล.....	85
บทที่ 5 สรุปและข้อเสนอแนะ.....	90
5.1 สรุป.....	90
5.2 ข้อเสนอแนะ.....	92
บรรณานุกรม.....	93

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางแสดงตัวอย่าง ViewBag.....	9
2.2 ตารางแสดงตัวอย่าง @Model.....	9
2.3 ตารางแสดงตัวอย่าง @Model List.....	10
2.4 ตารางแสดงตัวอย่าง Selectors Expressions(1).....	29
2.5 ตารางแสดงตัวอย่าง Selectors Expressions(2).....	30
2.6 ตารางแสดงตัวอย่าง CSS Method(1).....	30
2.7 ตารางแสดงตัวอย่าง CSS Method(2).....	31
2.8 ตารางแสดงตัวอย่าง Effect Method.....	32
2.9 ตารางแสดงตัวอย่าง DOM Traversal Method.....	32
2.10 ตารางแสดงตัวอย่าง Dom Manipulation Method.....	33
2.11 ตารางแสดงตัวอย่าง AJAX Method.....	34
2.12 ตารางแสดงตัวอย่าง Utilities Method.....	34
3.1 ตารางที่ใช้ในการพัฒนาระบบจัดการระบบการจัดการสมาชิกบนเว็บ.....	53
3.2 แสดงตาราง UseStampPromotionCategory มี useStampPromotionId เป็นคีย์หลัก.....	53
3.3 แสดงตาราง Store มี storeIdEmail เป็นคีย์หลัก.....	54
3.4 แสดงตาราง ReceiptDescription มี receiptDescriptionId เป็นคีย์หลัก.....	54
3.5 แสดงตาราง ReceiptDescription มี ProductId เป็นคีย์หลัก.....	55
3.6 แสดงตาราง GetStampPromotionCategory มี getStampPromotionCategoryId เป็นคีย์หลัก.....	55
3.7 แสดงตาราง Stamp มี StampId เป็นคีย์หลัก.....	55
3.8 แสดงตาราง UseStampPromotion มี useStampPromotionId เป็นคีย์หลัก.....	56
3.9 แสดงตาราง GetStampPromotion มี getStampPromotionId เป็นคีย์หลัก.....	57
3.10 แสดงตาราง Receipt มี receiptId เป็นคีย์หลัก.....	58
3.11 แสดงตาราง LogGetStamp มี logGetStampId เป็นคีย์หลัก.....	58
3.12 แสดงตาราง DeleteStampSchedule มี deleteStampScheduleId เป็นคีย์หลัก.....	59
3.13 แสดงตาราง StoreCategory มี storeCategoryId เป็นคีย์หลัก.....	59
3.14 แสดงตาราง Member มี memberTel เป็นคีย์หลัก.....	59

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
3.15 แสดงตาราง LogUseStamp มี logUseStampId เป็นคีย์หลัก.....	60
5.1 ตารางเปรียบเทียบระหว่างระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและ ขนาดย่อม(PocketCards) และ Application “Stamp” บน Smartphone iOS.....	91

สารบัญรูปภาพ

รูปที่	หน้า
2.1 MVC Architecture Pattern.....	4
2.2 แสดง Sequence Diagram การทำงานของ ASP.NET MVC Request.....	6
2.3 แสดงผลการ Post จะรับจาก Request (1).....	10
2.4 แสดงผลการ Post จะรับจาก Request (2).....	11
2.5 แสดงผลการ Post จะรับจากแบบ Parameter ของ Action (1).....	11
2.6 แสดงผลการ Post จะรับจากแบบ Parameter ของ Action (2).....	11
2.7 แสดงโค้ด Return View.....	12
2.8 การสร้าง Model จากฐานข้อมูลที่มีอยู่แล้ว.....	12
2.9 การกำหนดค่าให้มี connection string ที่เชื่อมต่อระบบของเราเข้ากับ database.....	13
2.10 การเลือก Generate จาก Database.....	13
2.11 การ connection เลือกไปยัง database ที่สร้างไว้.....	13
2.12 การบันทึก entity connection ถึง web.config.....	14
2.13 การเลือกตารางที่จะใช้.....	14
2.14 การตั้งชื่อ Model.....	14
2.15 การสร้าง class จาก edm.....	15
2.16 การเลือกโค้ดแบบ Self tracking.....	15
2.17 แสดง class ตามที่ edm สร้างไว้ ภายในโครงสร้างโปรเจก.....	15
2.18 แสดงโครงสร้างของ Entity Framework.....	16
2.19 การพัฒนาของ Entity Framework.....	17
2.20 แสดง My MVC Application.....	18
2.21 การสร้างเป็น Single Statement.....	19
2.22 การปิด Block ของ Statement ด้วย {...}.....	20
2.23 การใส่เครื่องหมายซิโคลอนหลังจากจบคำสั่งใน Block.....	20
2.24 การสร้างตัวแปรใหม่เก็บค่าคงที่.....	20
2.25 การระบุข้อความให้ปิดด้วยเครื่องหมายอัฒประกาศ (“ ”).....	21
2.26 การใช้เครื่องหมายแบคสแลช และเครื่องหมายอัฒประกาศ.....	21

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
2.27 การใช้เครื่องหมายอัฒประกาศ 2 ครั้ง	21
2.28 ในคำหลัก C# แบบ case sensitive	21
2.29 การใส่หมายเหตุแบบ Razor	22
2.30 ตัวแปรและประเภทข้อมูล	22
2.31 การเข้าถึง root เสมือน	22
2.32 การแปลง Root เสมือนเป็น path บนไดรว์	23
2.33 ตัวอย่างการสร้างปฏิทินด้วย jQuery	25
2.34 การดาวน์โหลด jQuery จากหน้าเว็บ http://www.jquery.com	25
2.35 ตัวอย่างการใช้ jQuery เพื่อตรวจสอบข้อมูลก่อนการส่งข้อมูลไปประมวลผล	26
2.36 การเลือกตำแหน่งที่ต้องกระทำด้วย Selectors	27
2.37 การเขียน JavaScript กับ Web Page ในรูปแบบ Embedded Script	35
2.38 แสดงผลกับ Web Browser แบบ Class Selector	41
2.39 แสดงผลกับ Web Browser แบบ TAG Selector	42
2.40 แสดงผลกับ Web Browser แบบ ID Selector	43
2.41 โลโก้แอปพลิเคชัน “Stamp” และหน้าจอการใช้งานต่างๆ	44
3.1 Use Case Diagram	47
3.2 Entity Relationship Diagram	52
4.1 หน้าหลักระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม	61
4.2 หน้าหลักแสดงการเข้าใช้งานของส่วนลูกค้า	62
4.3 หน้าจอแสดงแสดงการแจ้งเตือนอัตโนมัติเมื่อใกล้จะถึงวันครบสแตมป์	62
4.4 หน้าจอแสดงการสมัครสมาชิกของลูกค้า	63
4.5 หน้าจอแสดงการ login เข้าสู่ระบบของลูกค้า	64
4.6 หน้าจอแสดงการขอรหัสผ่านใหม่ โดยกรอก E-mail ของลูกค้า	64
4.7 หน้าจอแสดงบัตรสะสมสแตมป์	65
4.8 หน้าจอแสดงข้อมูลรายละเอียดของร้านค้า (1)	66
4.9 หน้าจอแสดงข้อมูลรายละเอียดของร้านค้า (2)	66

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
4.10 หน้าจอแสดงข้อมูลสินค้าของแต่ละร้าน	67
4.11 หน้าจอแสดงร้านค้าทั้งหมด	68
4.12 หน้าจอแสดงรายการร้านค้าตามที่เราเลือกประเภทร้านค้า	69
4.13 หน้าจอแสดงรายการโปรโมชั่นที่สามารถแลกได้	69
4.14 หน้าจอแสดงการแก้ไขข้อมูลของลูกค้า	70
4.15 หน้าจอแสดงการแก้ไขเบอร์โทรศัพท์ของลูกค้า	70
4.16 หน้าจอแสดงการแก้ไข E-mail ของลูกค้า	71
4.17 หน้าจอแสดงการแก้ไขวันเกิดของลูกค้า	71
4.18 หน้าจอแสดงการแก้ไขรูปลูกค้า	72
4.19 หน้าจอแสดงการเปลี่ยนรหัสผ่านของลูกค้า	72
4.20 หน้าจอแสดงการบันทึกการแก้ไขข้อมูลของลูกค้า	73
4.21 หน้าหลักแสดงการเข้าใช้งานของส่วนร้านค้า	74
4.22 หน้าจอแสดงการสมัครสมาชิกของร้านค้า	75
4.23 หน้าจอแสดงการ Log in เข้าสู่ระบบของร้านค้า	76
4.24 หน้าจอแสดงการขอรหัสผ่านใหม่ โดยกรอก E-mail	76
4.25 หน้าจอแสดงการเพิ่มรายการสินค้า	77
4.26 หน้าจอแสดงรายการสินค้าที่ขาย	78
4.27 หน้าจอแสดงการแก้ไขข้อมูลสินค้า	78
4.28 หน้าจอแสดงรายการสินค้าที่ไม่ขาย	79
4.29 หน้าจอแสดงร้านค้าจัดการการชำระเงินของลูกค้า	80
4.30 หน้าจอแสดงระบบชำระเงินในการใช้ส่วนลด	81
4.31 หน้าจอแสดงสถิติรายได้แบบกราฟ	81
4.32 หน้าจอแสดงสถิติยอดขายแบบแผนภูมิ	82
4.33 หน้าจอแสดงการกำหนดคลับแอสตัมปี	82
4.34 หน้าจอแสดงโปรโมชั่นให้แอสตัมปี	83
4.35 หน้าจอแสดงโปรโมชั่นใช้แอสตัมปี	84

สารบัญรูปรภาพ (ต่อ)

รูปที่	หน้า
4.36 หน้าจอแสดงการแลกสินค้า.....	85
4.37 หน้าจอแสดงการแก้ไขข้อมูลของร้านค้า.....	86
4.38 หน้าจอแสดงการแก้ไข E-mail ของร้านค้า.....	86
4.39 หน้าจอแสดงการแก้ไขประเภทร้านค้า.....	87
4.40 หน้าจอแสดงการแก้ไขชื่อร้านค้า.....	87
4.41 หน้าจอแสดงการแก้ไขรูปร้านค้า.....	88
4.42 หน้าจอแสดงการเปลี่ยนรหัสผ่านของร้านค้า.....	88
4.43 หน้าจอแสดงการบันทึกการแก้ไขข้อมูลของร้านค้า.....	89

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ทุกวันนี้มีการซื้อสินค้าตามห้างสรรพสินค้า หรือร้านค้าต่างๆ จะมีการแจกบัตรสมาชิกเพื่อดึงลูกค้าใหม่และลูกค้าทั่วไปให้กลับมาเป็นลูกค้าประจำของร้าน และเพื่อรักษาฐานลูกค้าประจำ ลูกค้าที่เป็นสมาชิกและมีบัตรสมาชิกจะได้สิทธิพิเศษตามเงื่อนไขของแต่ละร้านค้า เมื่อลูกค้ามีการสมัครสมาชิกกับร้านค้าต่างๆ เพื่อรับสิทธิพิเศษ ลูกค้าต้องพกพาบัตรสมาชิกหลายใบ ปัญหาที่จะเกิดขึ้นคือทำให้เกิดความยุ่งยากและไม่สะดวกในการพกพาบัตรสมาชิกจำนวนหลายๆ ใบ เพราะเป็นสมาชิกกับหลายๆ ร้านค้า ลูกค้ามักจะลืมไม่พกบัตรเป็นประจำ มีความสับสนในการใช้บัตรสมาชิก เนื่องจากบัตรสมาชิกบางร้านค้ามีขนาดใกล้เคียงกัน รูปร่างเหมือนกัน สีและรูปภาพบนบัตรคล้ายๆ กัน และถ้ากระเป๋าใส่บัตรหาย หรือบัตรสมาชิกหลายๆ ใบเกิดการสูญหาย ลูกค้าจะต้องไปทำบัตรใหม่ จะทำให้เกิดความยุ่งยาก เสียเวลา และในบางร้านค้าลูกค้าอาจจะต้องเสียเงินค่าบัตรสมาชิกใหม่

เมื่อมีการสมัครสมาชิกเพิ่มมากขึ้น ร้านค้าจะมีต้นทุนในการทำบัตรสมาชิกมากขึ้น และถ้าลูกค้าทำบัตรสมาชิกหายก็ต้องทำบัตรสมาชิกใหม่ ร้านค้าต้องเสียเวลาและต้นทุนในการทำบัตรสมาชิกทั้งสมาชิกเก่าและสมาชิกใหม่รวมทั้งต้นทุนในการจ้างพนักงานมาดูแลในส่วนนี้ และทำให้เกิดความยุ่งยากในการบริหารของร้านค้าเอง ถ้าร้านค้าจะสร้างฐานข้อมูลทำระบบสมาชิก และระบบการบริหารจัดการร้านค้า ทางร้านค้าจะต้องเสียค่าใช้จ่ายในการพัฒนาระบบ เสียค่าบำรุงรักษาระบบ และค่าจ้างพนักงานมาดูแลในส่วนนี้

ดังนั้นระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม (Web Based Membership Management System for Small and Medium Enterprises) จึงมีความสำคัญขึ้นมา เนื่องจากทางร้านค้าไม่ต้องทำบัตรสมาชิกให้กับลูกค้า ลดขั้นตอนในการบริหารงานลงไปได้ และลูกค้าก็ไม่ต้องพกบัตรสมาชิกหลายๆ ใบ เพียงแค่ใช้เบอร์โทรศัพท์แทนบัตรสมาชิกก็สามารถได้สิทธิพิเศษตามเงื่อนไขของร้านค้าต่างๆ โดยระบบจะให้บริการระบบจัดการข้อมูลสมาชิกที่เป็นศูนย์กลาง และระบบการบริหารจัดการร้านค้า เมื่อแต่ละร้านค้ามีการเพิ่มสมาชิกจะได้ข้อมูลสมาชิกภายในระบบเหมือนกัน ภายในระบบจะประกอบไปด้วย

ระบบลูกค้า ลูกค้าสามารถสมัครสมาชิก แก้ไขข้อมูลภายในระบบ ดูรายละเอียดร้านค้า ดูรายละเอียดรายการ โปรโมชั่น คูณสินค้าที่ขายของแต่ละร้านค้า เมื่อลูกค้าต้องซื้อสินค้าจะแจ้งเบอร์โทรศัพท์มือถือของลูกค้าทุกครั้งกับร้านค้า ระบบจะเก็บบันทึกข้อมูลทั้งหมดไว้ให้ในรูปแบบบัตรสมาชิกและบัตรสะสมแต้มป้ออนไลน์ซึ่งสามารถดูแต้มปีสะสมคงเหลือ ตรวจสอบโปรโมชั่นที่สามารถใช้แต้มปีได้ โดยระบบจะมีการบันทึกการแลกสินค้า บันทึกการให้แต้มปี และบันทึกการใช้แต้มปีรูปแบบออนไลน์

ระบบร้านค้า ร้านค้าสามารถสมัครสมาชิก แก้ไขข้อมูลภายในระบบ เพิ่มและแก้ไข รายการสินค้า กำหนดและแก้ไขโปรโมชั่นในการให้แต้มปี กำหนดและแก้ไขโปรโมชั่นในการใช้แต้มปี กำหนดและลบกำหนดลบแต้มปี ดูสถิติรายได้และยอดขาย การชำระเงิน และการใช้แต้มปีเป็นส่วนลด

1.2 วัตถุประสงค์ของปัญหา

1. เพื่อพัฒนาระบบเว็บแอปพลิเคชันสำหรับผู้ใช้ โดยสร้างระบบสมาชิกไว้ที่เดียว ทางร้านค้าสามารถได้รับข้อมูลของสมาชิกเหมือนกัน เพื่อเป็นตัวกลางในการติดต่อและการให้บริการที่เหมาะสม ลูกค้าจะได้ไม่ต้องพกบัตรสมาชิกหลายใบ
2. เพื่อพัฒนาฝึกฝนกระบวนการพัฒนาโปรแกรมด้วยภาษา ASP.NET C#, CSS, jQuery และโปรแกรมการจัดการฐานข้อมูล SQL

1.3 ขอบเขตของปัญหา

สร้างเว็บแอปพลิเคชัน โดยใช้ภาษา ASP.NET C#, CSS, jQuery ด้วยเครื่องมือ Visual Studio 2012 และใช้ Microsoft SQL Server 2012 เป็นระบบจัดการฐานข้อมูล

1.4 ส่วนประกอบของโครงการพิเศษ

โครงการพิเศษชิ้นนี้ได้แบ่งเนื้อหาออกเป็น 5 บท โดยบทอื่นๆ มีเนื้อหาดังนี้

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง จะกล่าวถึงทฤษฎีต่างๆ ที่เกี่ยวกับการพัฒนาเว็บแอปพลิเคชัน และการเชื่อมต่อกับระบบจัดการฐานข้อมูล

บทที่ 3 การวิเคราะห์และออกแบบระบบ จะกล่าวถึงแนวทางการออกแบบโครงสร้าง และการพัฒนาโปรแกรมในส่วนต่างๆ

บทที่ 4 การใช้งานโปรแกรม จะกล่าวถึงแนวทางการใช้งานและส่วนประกอบต่างๆ ของระบบ

บทที่ 5 สรุปและข้อเสนอแนะ จะกล่าวถึงความสามารถในการตอบสนองความต้องการ ประโยชน์ที่ได้รับ และแนวทางการพัฒนา

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยให้ลูกค้าไม่ต้องพกพาบัตรสมาชิก
2. ช่วยให้ร้านค้ามีการบริหารลูกค้าสัมพันธ์ได้ดีขึ้น
3. ช่วยให้ร้านค้าไม่ต้องเสียเวลาและต้นทุนในการทำบัตรสมาชิก สร้างฐานข้อมูลระบบสมาชิกและระบบการบริหารจัดการร้านค้า
4. ได้รับแนวคิดจากการพัฒนาโปรแกรมสำหรับการบริหารจัดการสมาชิกและระบบร้านค้า
5. เสริมสร้างทักษะในการเขียน โปรแกรม ASP.NET C#, CSS, jQuery ของผู้เขียนโปรแกรม

บทที่ 2

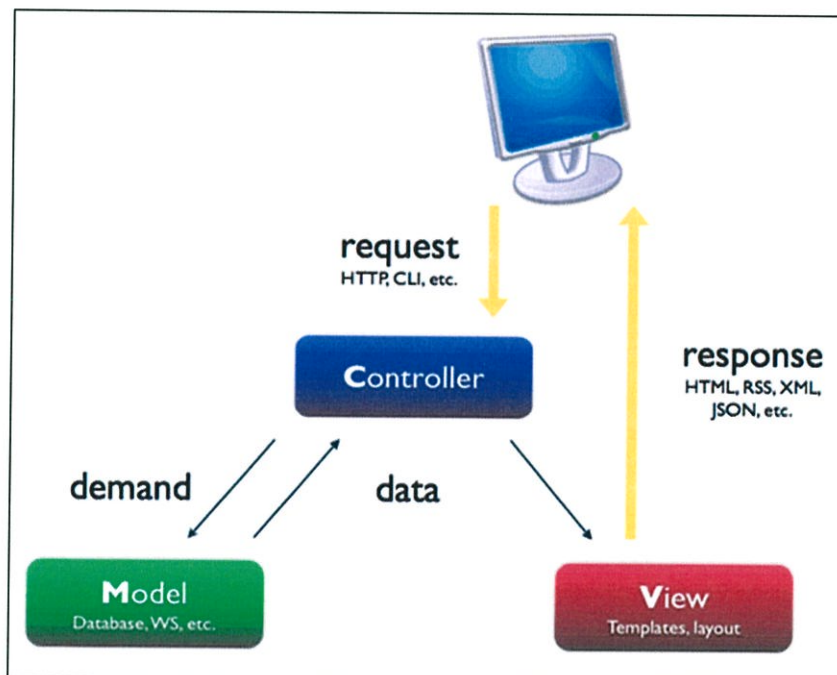
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ASP.NET MVC

ในปัจจุบันมีการพัฒนาระบบซึ่งมีขนาดของโครงการจะใหญ่ขึ้นเรื่อยๆ เนื่องจากความต้องการมีมากขึ้นเรื่อยๆ ดังนั้นการที่จะพัฒนา Software ด้วยตัวคนเดียวเป็นเรื่องยาก จึงมีการประยุกต์ใช้งาน Enterprise Architecture Pattern อย่าง MVC กันมากขึ้นในทุกๆ platform ด้วยเหตุนี้การที่จะใช้งาน MVC Framework ตัวใดก็ได้แต่ไม่ว่าจะเป็น PHP .NET หรือ Java ล้วนแล้วแต่ต้องอาศัยแนวความคิดที่ถูกต้องจึงจะเป็นประโยชน์อย่างแท้จริง ไม่เช่นนั้นนอกจากจะไม่มีประโยชน์ใดๆ ในการใช้ Framework แล้ว ยังจะทำให้แก้ไขได้ยากกว่าปกติ ดังนั้นจึงเริ่มจากการแนะนำเรื่องแนวคิดของการแบ่งแยกส่วนของ code ต่าง ๆ

2.1.1 MVC Architecture Pattern

การแบ่งแยกส่วนของ code ออกเป็น 3 ส่วนด้วยกัน คือ



รูปที่ 2.1 MVC Architecture Pattern

1. Model (M)

เป็นส่วนของ Business Logic และส่วนของ Entity ซึ่งจะต่างจากแนวคิดแบบ 3-tier ซึ่งจะแยกจากส่วนล่างสุดคือ Data Access Layer (DAL) ซึ่งจะทำหน้าที่ติดต่อกับ Database เท่านั้น นอกจากนั้นยังรวมถึงเรื่องของ Business Logic ต่างๆ เช่น การคำนวณ VAT 7% หรือการคำนวณค่าต่างๆให้อยู่ในชั้นนี้ ชั้นนี้ควรจะทำการสร้าง Service Layer ขึ้นมา

2. View (V)

เป็นส่วนของการแสดงผล คือ User Interface ของระบบ ซึ่งการทำงานในส่วนของวิวนั้นจะอ้างถึงกล่องโมเดล โดยเพื่อทำการค้นหาและดึงข้อมูลขึ้นมาและแสดงข้อมูลออกมาในรูปแบบที่เราได้กำหนดไว้ในวิวนั้น ซึ่งวิวแต่ละวิวขึ้นอยู่กับ Controller จะเรียกใช้เพื่อ render การทำงาน ไม่ว่าจะเป็วิวแบบ Add/Edit/Delete และ Detail เป็นส่วนของ HTML CSS และ JavaScript ในการทำ Web Application แต่ถ้าเป็น Windows Application ก็จะเป็นพวก forms ต่างๆ ในการใช้งาน PHP Framework บางตัวจะพ่วงเอา Template Engine มาให้ใช้งานด้วย เช่น Symfony จะมี Twig เป็น Template Engine ซึ่งข้อดีของการใช้งาน Template Engine ก็คือจะทำการแยกส่วนของการแสดงผลกับ logic ได้ชัดเจนมากขึ้น คนที่ทำ HTML CSS และ JavaScript ไม่จำเป็นต้องเขียน PHP ได้

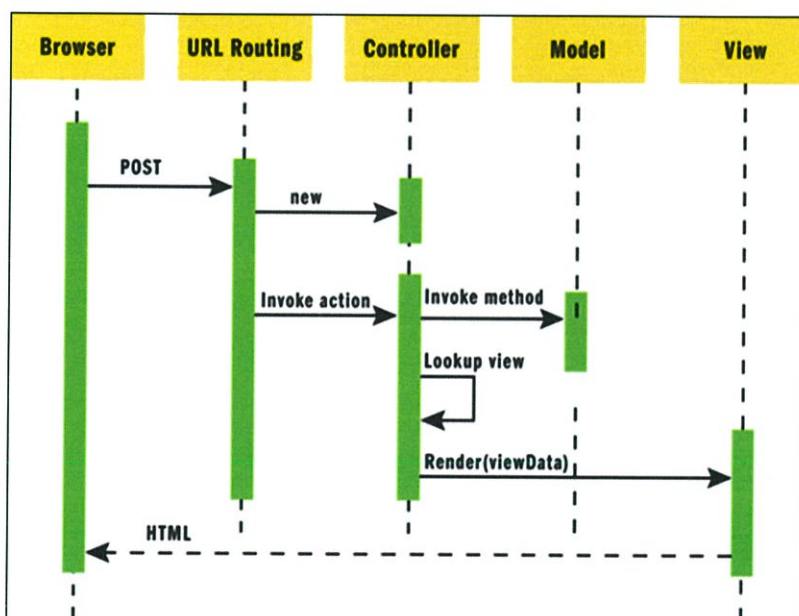
3. Controller (C)

เป็นเสมือนตัวกลางที่ทำงานระหว่างโมเดล (Model) และวิว (View) ซึ่งคอนโทรลเลอร์จะทำงานตามรีเควสจากผู้ใช้งาน ซึ่งจะรู้ว่าเมื่อมีการรีเควสโมเดลนี้ ต้องการคอนโทรลเลอร์ตัวไหน แอคชั่นอะไร ภายในแอคชันก็จะไปเรียก Model เพื่อทำงานตาม business อีกที จากนั้นเมื่อทำงานเสร็จเช่นการ Update หรือขอดู Detail ก็จะส่ง data ไปยัง view นั้นๆ กลับไปแสดงผลพร้อมทั้งนำอ็อปเจ็กต์ที่สำเร็จไปแสดงผล

2.1.2 หลักการทำงานของ MVC

1. เมื่อ Client มีการร้องขอผ่าน View
2. ข้อมูลจะส่งผ่านไปหา Controller โดย Controller จะเป็นตัวจำแนก Action ต่างๆ หรือเป็น Business Logic ของระบบ
3. จากนั้น Controller ก็จะร้องขอไปยัง Model จาก Action นั้นๆ เช่น ร้องขอการเลือกข้อมูลทั้งหมดจากรายง
4. เมื่อ Model รับการร้องขอก็จะทำการ Query ข้อมูลตามที่ Controller ส่งมา
5. Model จะส่งข้อมูลที่ได้กลับมามาหา Controller
6. Controller ก็จะทำการ Set ค่าลงในตัวแปรเพื่อส่งให้ View ต่อไป
7. View ก็จะนำตัวแปรเหล่านั้นไปทำการแสดงผลตามต้องการได้

2.1.3 การทำงานของ ASP.NET MVC Request



รูปที่ 2.2 แสดง Sequence Diagram การทำงานของ ASP.NET MVC Request

1. Browser เมื่อผู้ใช้ได้ส่งคำสั่ง request คำสั่งใด ๆ จาก Browser เข้ามา เช่น ขอหยิบ ยกตัวอย่างด้านบน `http://localhost:63282/Home/About`
2. URL Routing จะนำ URL Request ที่ได้ไปแม็พเข้ากับ pattern ที่ได้กำหนดไว้ใน `Global.asax.cs` ตั้งแต่ตอนที่ Application Start เรียกเมธอดที่ชื่อว่า `RegisterRoutes ()` ซึ่งกำหนดค่าเริ่มต้นไว้ดังนี้ `{controller}/{action}/{id}`

เมื่อเทียบกับบริเวณ Home/About ก็จะได้ค่า controller = Home action = About ดังนั้น RouteHandler จะไปขอ HomeController จาก ControllerFactory และทำการเรียก About Method ใน HomeController ตาม action ที่ร้องขอมา

3. Controller ในกรณีนี้ About Action นั้น เป็นแค่การกำหนดค่า ViewData และสั่ง render Views/Home/About.aspx แค่นั้น แต่ในการทำงานส่วนใหญ่มักจะมีการเรียกใช้ Model เข้ามาทำงานตามไดอะแกรมด้านบนซึ่งใน controller จะทำการเรียก Model เพื่อรัน business method ให้ได้ผลลัพธ์กลับออกมา จากนั้นก็ส่ง Data ที่ได้เข้าไป Render เพื่อให้ได้ ViewResult และหา ViewEngine ได้ที่ตั้งค่าเลือกไว้ ซึ่ง WebFormViewEngine เป็นของที่มีมาให้ใช้งาน
4. View เมื่อเลือก ViewEngine ได้ก็ทำการ Render HTML กลับไปยัง Browser

2.1.4 หน้าที่ของ Folder

Folder ทั้งหมดที่มีการสร้างขึ้นจะถูกจัดไว้สำหรับการสร้างทรัพยากรของเว็บ โดยเฉพาะแต่ละเรื่อง แต่ละหน้าที่ ซึ่งจะอธิบายดังต่อไปนี้

- Content folder คือโฟลเดอร์ที่เอาไว้เก็บพวก style sheet image และ icon
- Controllers folder คือโฟลเดอร์ที่เอาไว้เก็บโดยแยกเก็บเป็น Controller class ที่ใช้ในการสร้าง method ในการควบคุมสั่งงานต่าง ๆ
- Models folder คือโฟลเดอร์ที่เอาไว้เก็บพวกกลุ่ม Domain class ต่าง ๆ ที่จะใช้ในการทำงานกับ Controller และ View
- Scripts folder คือเอาไว้เก็บ JavaScript jQuery และ Microsoft Script ต่างๆ เป็นต้น
- Views folder คือโฟลเดอร์สำหรับเก็บ View ที่เราจะสร้างขึ้นเพื่อแสดงผล รับค่าต่าง ๆ
- Shared คือ view ที่จะมองเห็นเรียกใช้ได้จาก view อื่น ๆ เช่น master page เป็นต้น

2.1.5 Routing System

เป็นการ Mapping ไปยัง Controller และ Action มองได้ง่ายๆ ว่า Controller คือ Class และ Action คือ Method ที่อยู่ใน Class นั้นๆ โดยค่า Default ของ Routing จะอยู่ใน Global.asax คือ Host/{Controller}/{Action} เช่น http://localhost/Home/About จะมาจาก Controller Class ชื่อ HomeController และ Method ชื่อ About ซึ่ง Host/{Controller} จะเป็นการเข้า Action ชื่อ Index เพราะที่กำหนด Default

2.1.6 Pass Data ระหว่าง Controller กับ View

เราสามารถ Pass Data ซึ่งหมายถึง Objects Model Value etc. จาก Controller ไปยัง View ได้ด้วย

- ViewBag เป็น Dynamic Collection ที่เก็บ Object ใดๆ จะเขียนแบบ Dynamic Property หรือเป็น Array ก็ได้
 - ใน Controller
 - ViewBag.TestData1="My Test Data1";
 - //ViewBag["TestData1"]="My Test Data1";
 - ใน View
 - @ViewBag.TestData
- Model Binding เป็นการส่ง Data Object ผ่านไปกับ View โดยที่ View จะใส่ keyword @model {Full Name Class} หรือ @model IList<> เพื่อส่งเป็น List ไปก็ได้ และใช้ Keyword @Model ในการแสดงผล
 - ใน View
 - @model MvcApplication1.Models.RegisterModel
 - User Name: @Model.UserName หรือ
 - @model List<MvcApplication1.Models.RegisterModel>
 - @Model[0].UserName
- AJAX ใช้ technique AJAX รับส่ง Data จะใช้ ViewBag เพื่อแทน Model Binding ก็ได้ แต่ว่าจะไม่มี Code Intellisense ขึ้นให้

ตัวอย่าง ViewBag

Controller	<p>แก้ไข Method ListData</p> <pre>public ActionResult ListData() { ViewBag.Data = "My Data from Controller"; ViewBag.DataInt = 1001; ViewBag.CreateDate = DateTime.Now; return View(); }</pre>
View	<p>แก้ไข /Views/Task/ListData.cshtml</p> <pre>@{ ViewBag.Title = "ListData"; } <h2>ListData</h2> My Data = @ViewBag.Data
 Data Int = @ViewBag.DataInt
 Create Date = @ViewBag.CreateDate</pre>

ตารางที่ 2.1 ตารางแสดงตัวอย่าง ViewBag

ตัวอย่าง @Model

Controller	<p>เพิ่ม Method ShowData</p> <pre>public ActionResult ShowData() { var reg = new MvcApplication1.Models.RegisterModel(); reg.UserName = "User 1"; reg.Email = "user1@aspmvc.net"; return View(reg); }</pre>
View	<p>สร้าง /Views/Task/ShowData.cshtml</p> <pre>@model MvcApplication1.Models.RegisterModel @{ ViewBag.Title = "ShowData"; } <h2>ShowData</h2> User Name:@Model.UserName
 Email: @Model.Email
</pre>

ตารางที่ 2.2 ตารางแสดงตัวอย่าง @Model

ตัวอย่าง @Model List

Controller	<p style="text-align: center;">เพิ่ม Method ListTask</p> <pre>public ActionResult ListTask() { IList<MvcApplication1.Models.RegisterModel> list = new List<MvcApplication1.Models.RegisterModel>(); for (int i = 0; i < 10; i++) { var reg = new MvcApplication1.Models.RegisterModel(); reg.UserName = "User "+i; reg.Email = "user"+i+"@aspmvc.net"; list.Add(reg); } return View(list); }</pre>
View	<p style="text-align: center;">สร้าง /Views/Task/ListTask.cshtml</p> <pre>@model IList<MvcApplication1.Models.RegisterModel> @{ ViewBag.Title = "ListTask"; } <h2>ListTask</h2> @foreach (var m in Model){ User Name: @m.UserName
 Email: @m.Email
 <hr /> }</pre>

ตารางที่ 2.3 ตารางแสดงตัวอย่าง @ Model List

2.1.7 การส่ง Data จาก View มา Controller

เหมือนการ Post จะรับจาก Request หรือจากที่ Parameter ของ Action เลขก็ได้ จาก Request

```
public string TestRequest1()
{
    string firstName = Request["FirstName"];
    string lastName = Request["LastName"];
    if (firstName == null)
        firstName = "John";
    if (lastName == null)
        lastName = "Doe";
    return "Hello " + firstName + " " + lastName;
}
```

รูปที่ 2.3 แสดงผลการ Post จะรับจาก Request (1)

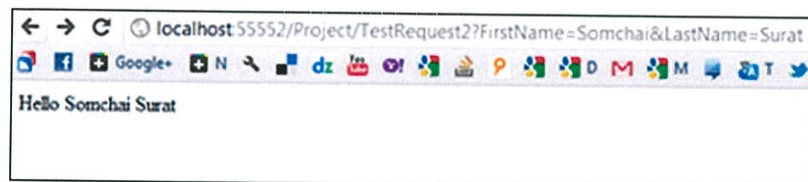


รูปที่ 2.4 แสดงผลการ Post จะรับจาก Request (2)

แบบ Parameter ของ Action

```
public string TestRequest2(string firstName="John", string lastName="Doe")
{
    return "Hello " + firstName + " " + lastName;
    ....
}
```

รูปที่ 2.5 แสดงผลการ Post จะรับจากแบบ Parameter ของ Action (1)



รูปที่ 2.6 แสดงผลการ Post จะรับจากแบบ Parameter ของ Action (2)

จะเห็นว่าแบบเป็น Parameter ของ Action จะสวยงามกว่า เขียน Code สั้น กระชับ เห็นชัดเจนว่า Action นี้ต้องการ Parameter อะไรบ้าง และยังใส่ค่า Default ลงไปได้ทันที

2.1.8 Return View

โดย Default ถ้า Action ใน Controller Return View โดยไม่มี Parameter ใดๆ ASP.NET Engine ก็จะใช้ชื่อของ Action ไปหา View ใน Folder ของ Controller แต่ถ้าต้องการให้ไปหา View ชื่ออื่นก็ทำได้โดยใส่ Path ของ View ที่จะให้ Render หรือใช้ RedirectToAction เช่นตัวอย่างโค้ดด้านล่างนี้

```

public ActionResult About()
{
    return RedirectToAction("About", "Home");
}

public ActionResult AboutPath()
{
    return View("~/Views/Home/About.cshtml");
}

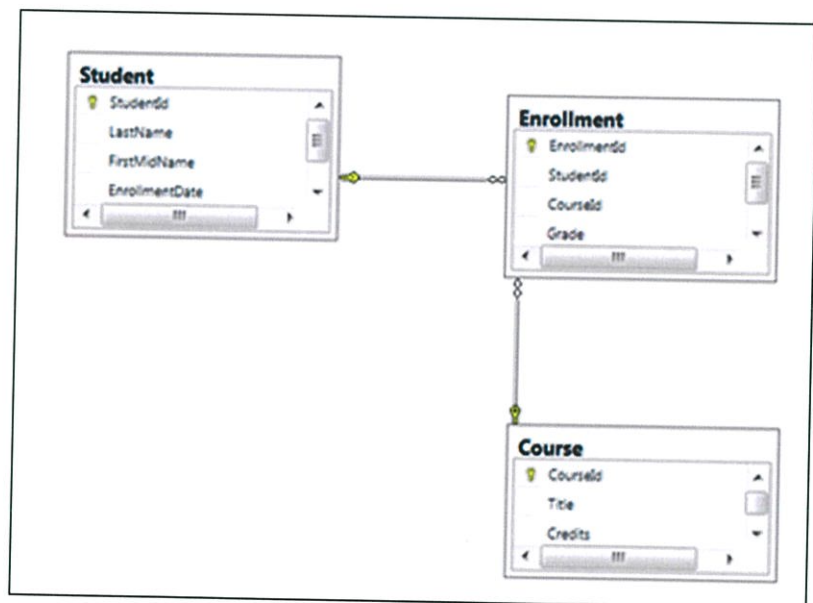
```

รูปที่ 2.7 แสดงโค้ด Return View

2.1.9 การสร้าง Model จากฐานข้อมูลที่มีอยู่แล้ว

การสร้าง Model จากฐานข้อมูลที่มีอยู่เรียกว่า Database First

1. เมื่อเราได้สร้างฐานข้อมูลแล้ว สร้างตารางและความสัมพันธ์เอาไว้ก่อนแล้ว ดังรูปที่ 2.8



รูปที่ 2.8 การสร้าง Model จากฐานข้อมูลที่มีอยู่แล้ว

2. เพื่อความแน่ใจให้ดูใน web.config ว่ามี connection string ที่เชื่อมต่อระบบของเราเข้ากับ database ที่เราได้สร้างเอาไว้หรือยัง หากยังให้กำหนดค่า ดังรูปที่ 2.9

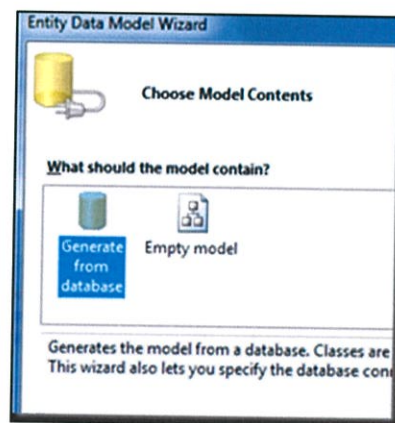
```

7 <configuration>
8   <connectionStrings>
9
10    <add name="SchoolDB" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=|
11    DataDirectory|School.mdf;Integrated Security=True;User Instance=True" />

```

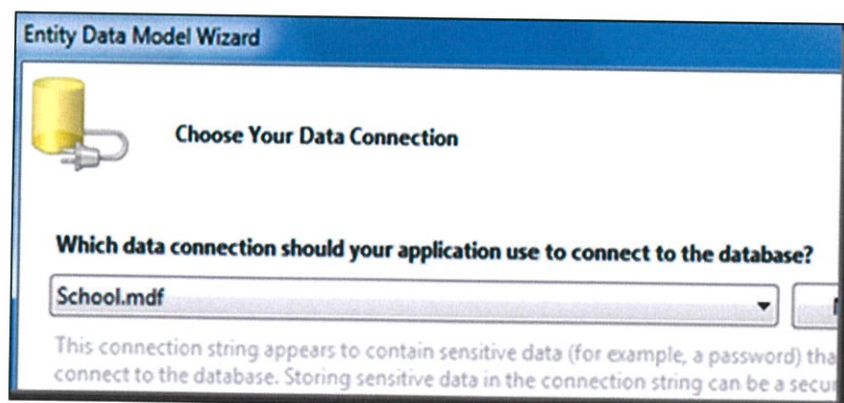
รูปที่ 2.9 การกำหนดค่าให้มี connection string ที่เชื่อมต่อระบบของเราเข้ากับ database

3. ให้สร้าง Entity Data Model ขึ้นมา ที่โฟลเดอร์ Model ให้เพิ่ม Entity data model ลงไป แล้วตั้งชื่อ SchoolModel
4. เลือก Generate จาก Database ดังรูปที่ 2.10



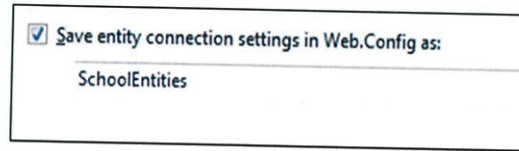
รูปที่ 2.10 การเลือก Generate จาก Database

5. ที่ connection เลือกไปยัง database ที่เราได้สร้างเอาไว้ ดังรูปที่ 2.11



รูปที่ 2.11 การ connection เลือกไปยัง database ที่สร้างไว้

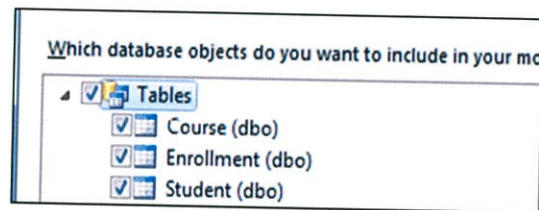
6. ด้านล่างให้บันทึก entity connection ลง web.config ดังรูปที่ 2.12



Save entity connection settings in Web.Config as:
SchoolEntities

รูปที่ 2.12 การบันทึก entity connection ลง web.config

- เลือกตารางที่ใช้ ดังรูปที่ 2.13



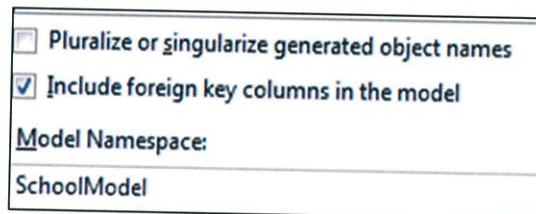
Which database objects do you want to include in your model?

Tables

- Course (dbo)
- Enrollment (dbo)
- Student (dbo)

รูปที่ 2.13 การเลือกตารางที่จะใช้

- และตั้งชื่อว่า SchoolModel ดังรูปที่ 2.14



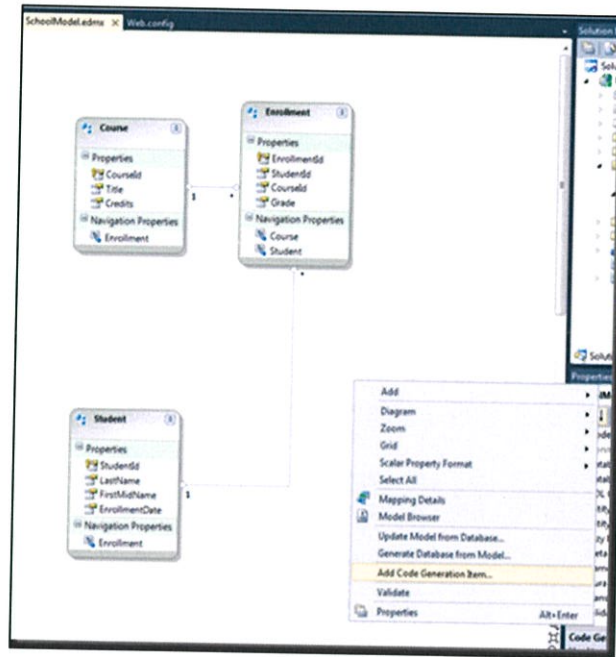
Pluralize or singularize generated object names

Include foreign key columns in the model

Model Namespace:
SchoolModel

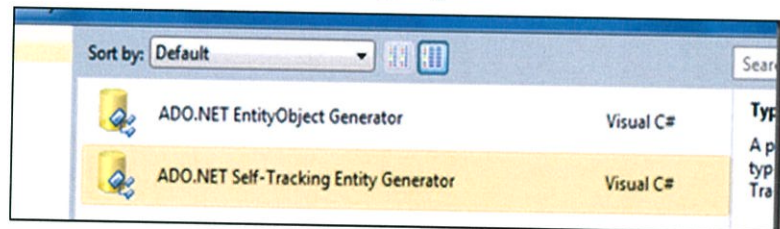
รูปที่ 2.14 การตั้งชื่อ Model

7. เลือก build project แล้วสร้าง class จาก edm ด้วยการคลิกขวาในหน้าการออกแบบ edm ดังรูปที่ 2.15



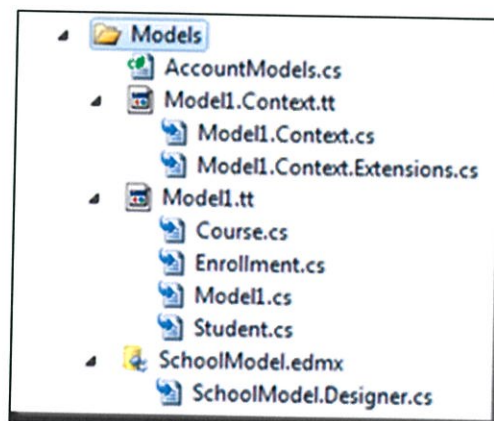
รูปที่ 2.15 การสร้าง class จาก edm

8. เลือก code แล้วเลือก Self tracking ดังรูปที่ 2.16



รูปที่ 2.16 การเลือกโค้ดแบบ Self tracking

9. เราจะได้ Class ตามที่ edm ได้สร้างเอาไว้ ดังรูปที่ 2.17

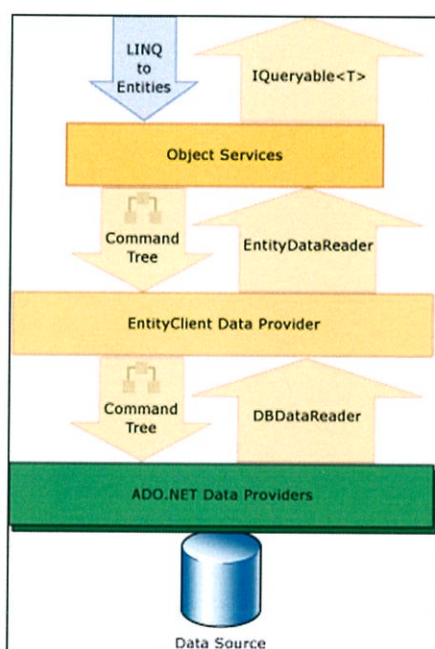


รูปที่ 2.17 แสดง class ตามที่ edm สร้างไว้ ภายในโครงสร้างโปรเจก

2.2 Entity Framework (EF)

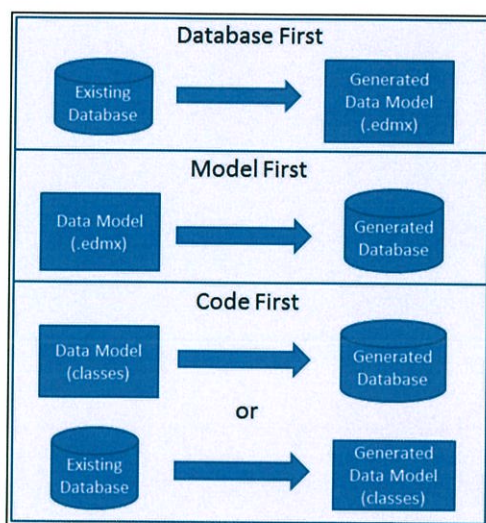
Entity Framework หรือ ADO.NET Entity Framework หรือ Object Relation Mapping คือ การทำให้ Object และ Database รู้จักกัน โดยผ่านล่ามตัวหนึ่ง ซึ่งจะเป็นตัวกลางทำให้เราสามารถที่จะทำการเขียนโปรแกรมจัดการกับข้อมูลใน Database ได้ง่ายขึ้น และเป็น Framework ชนิดหนึ่งที่จะช่วยให้นักพัฒนาสามารถพัฒนาระบบสามารถพัฒนาได้รวดเร็วมากขึ้น เพราะ Framework ตัวนี้จะช่วยให้การจัดการเกี่ยวกับฐานข้อมูลต่างๆ เป็นไปได้ง่าย โดยเราสามารถที่จะทำการเขียนได้แบบ OO ซึ่งทำให้ลดเวลาในการทำงานกับ Database ได้อย่างมาก อีกทั้งยังจะช่วยลดปริมาณ Code ที่ไม่จำเป็นลง และยังช่วยให้การดูแลจัดการระบบภายหลังเป็นไปได้ง่ายมากยิ่งขึ้น

2.2.1 โครงสร้างของ Entity Framework



รูปที่ 2.18 แสดงโครงสร้างของ Entity Framework

จากภาพจะเห็นว่า เมื่อใช้ Entity Framework มันจะทำหน้าที่เป็น Data Access layer ให้กับเราโดยการเชื่อมต่อกับฐานข้อมูล จะใช้บนพื้นฐานของ ADO.NET Data Provider จากนั้น Entity Framework จะทำการ map Database Table ต่างๆ เป็น Entity เข้าสู่ระบบ และสุดท้ายนักพัฒนาจะใช้ LINQ TO Entity ในการ Query Entity ที่ map มา Entity Framework (EF) รองรับรูปแบบการพัฒนา 3 รูปแบบด้วยกัน ดังรูปที่ 2.19



รูปที่ 2.19 การพัฒนาของ Entity Framework

2.2.2 Database First

หากมี database อยู่แล้ว จะสามารถสร้าง EF data model ซึ่งประกอบไปด้วย entity class และ property สำหรับทำหน้าที่แทน table และ column ใน database ได้ทันที โดย EF data model จะเป็นไฟล์นามสกุล .edmx เป็นไฟล์ xml format ประกอบไปด้วย store schema conceptual model และ mapping นอกจากนี้ Visual Studio ยังเตรียม Entity Framework designer ซึ่งเป็น graphical tool สำหรับใช้ดูและจัดการ EF data model

2.2.3 Model First

ถ้ายังไม่มี database สามารถเริ่มต้นด้วยการสร้าง EF data model ขึ้น โดยใช้ Entity Framework designer ใน Visual Studio ซึ่งจะได้ไฟล์ .edmx เช่นเดียวกับวิธี Database First จากนั้นจึง generate DDL (Data Definition Language) statements เพื่อใช้ในการสร้าง table

2.2.4 Code First

ไม่ว่าจะมี database อยู่แล้วหรือไม่ก็ตาม สามารถสร้าง class และ property ขึ้นเอง สำหรับทำหน้าที่แทน table และ column โดยไม่จำเป็นต้องมี EF data model (.edmx) แต่อย่างใด บางครั้งก็เรียกรูปแบบนี้ว่า Code Only (ซึ่งอย่างไม่เป็นทางการ) การ mapping ระหว่าง store schema และ conceptual schema จะใช้ convention และ api พิเศษ นอกจากนี้ หาก class มีการเปลี่ยนแปลงยังสามารถ sync กับ database เพื่อให้ model และ database มีโครงสร้างที่สอดคล้องกันได้อีกด้วย

2.2.5 Entity Framework 4

Entity Framework ได้ประกาศใช้งานครั้งแรกในสมัยของ .Net Framework 3.5 SQL เมื่อเวลาผ่านไป ทาง SQL Team ของ Microsoft ก็ได้ทำการปรับปรุง เพิ่มคุณสมบัติที่ได้ถูกปรับปรุงเพิ่มขึ้นมากมาย โดยตัวที่สำคัญมีดังนี้

- POCO SUPPORT : (Plain Old C# / CLR Object) ซึ่งก็คือแนวทางในการเขียน code โดยไม่มีการ inherit/implement framework หรือ Library ใดๆ ซึ่งการเขียนในรูปแบบนี้จะค่อนข้างอ่านง่าย ช่วยให้การพัฒนาเพิ่มเติมทำได้ง่ายมากยิ่งขึ้น
- Lazy Loading Support : ดึงข้อมูลขึ้นมาโดยดึงเท่าที่จำเป็นขึ้นมาก่อน ซึ่งช่วยลดการใช้งานของทรัพยากรลงได้
- N-Tier Support and Self-Tracking Entities : รองรับการดำเนินงานจัดการ Entities เมื่อมีการส่งข้าม Tier กัน
- Better SQL Generation and SPROC support : ปรับปรุงในเรื่องของการ Generate SQL และรองรับ stored procedure แล้ว Automatic Pluralization Support : รองรับการแก้คำ เช่น Categories -> Category
- Improved LINQ Operator Support : รองรับการใช้งาน LINQ Operator ทั้งหมด

ในการพัฒนาตัว Entity Framework4 ในครั้งนี้ จะใช้ ASP.NET MVC 3 ในการพัฒนา เพราะตัว MVC3 นั้นได้รองรับ ตัว EF4 มาในตัวแล้ว นอกจากนี้ยังมี Code First Support และ Data Scaffolding Support มาในตัวแล้วอีกด้วย ซึ่ง Feature ทั้งสองตัวนี้จะช่วยให้การพัฒนาระบบที่อยู่ในรูปแบบ Data-Driven เป็นไปได้ง่ายมากยิ่งขึ้น



รูปที่ 2.20 แสดง My MVC Application

2.3 View and Razor Engine

View engines ถูกใช้ใน ASP.NET MVC3 Framework คือ Razor View Engine และ Web Forms View engine. ทั้งสอง View engines เป็นส่วนหนึ่งของ MVC3 framework โดยปกติ View engine ใน MVC framework ใช้ Razor ซึ่งมีนามสกุล .cshtml และ .vbhtml หรือ Web Forms .aspx หน้าออกแบบเพจเลย์เอาต์ของ หน้า user interface ซึ่งประกอบด้วยข้อมูล แทนที่ ASP.NET Web Forms โดยใช้ postback model ทุก ๆ การกระทำถูกควบคุมเส้นทางจากกลไก ASP.NET Routing. Views สามารถ map กับ REST-friendly URLs.

2.3.1 รูปแบบของ Razor

- Compact Expressive and Fluid: มีขนาดเล็กและลื่นไหล
- Easy to Learn: เรียนรู้ง่ายหรือใช้ความรู้เดิมที่มีอยู่เช่น HTML
- Is not a new language: ใช้ความรู้ทางภาษาโปรแกรมมิ่งเดิม เช่น C#/VB(or other)
- Works with any Text Editor: ทำงานกับเครื่องมือหรือใดๆ ก็ได้ เช่น notepad
- Has great Intellisense: มีตัวช่วยเติมเต็มประโยคที่ดี

2.3.2 Razor Syntax

1. การสร้างเป็น Single Statement หรือ Multi-Statement ให้ขึ้นต้นบรรทัดด้วย @ ดังรูปที่ 2.21

```

<!-- Single statement blocks -->
@{ var total = 7; }
@{ var myMessage = "Hello World"; }

<!-- Inline expressions -->
<p>The value of your account is: @total </p>
<p>The value of myMessage is: @myMessage</p>

<!-- Multi-statement block -->
@{
    var greeting = "Welcome to our site!";
    var weekDay = DateTime.Now.DayOfWeek;
    var greetingMessage = greeting + " Today is: " + weekDay;
}
<p>The greeting is: @greetingMessage</p>

```

รูปที่ 2.21 การสร้างเป็น Single Statement

2. ปิด Block ของ Statement ด้วย {...} ใส่โค้ดในวงเล็บปีกกา การเขียนโค้ดคำสั่งในบรรทัดเดียวหรือหลายบรรทัด จะต้องใส่ไว้ในวงเล็บปีกกาทุกครั้ง ดังรูปที่ 2.22

```
<!-- Single statement block. -->
@{ var theMonth = DateTime.Now.Month; }
<p>The numeric value of the current month: @theMonth</p>

<!-- Multi-statement block. -->
@{
    var outsideTemp = 79;
    var weatherMessage = "Hello, it is " + outsideTemp + " degrees.";
}
<p>Today's weather: @weatherMessage</p>
```

รูปที่ 2.22 การปิด Block ของ Statement ด้วย {...}

3. เมื่อจบคำสั่งใน Block แล้วให้ใส่เครื่องหมายซีมิโคลอน(;) รูปที่ 2.23

```
<!-- Single-statement block -->
@{ var theMonth = DateTime.Now.Month; }

<!-- Multi-statement block -->
@{
    var outsideTemp = 79;
    var weatherMessage = "Hello, it is " + outsideTemp + " degrees.";
}

<!-- Inline expression, so no semicolon -->
<p>Today's weather: @weatherMessage</p>
```

รูปที่ 2.23 การใส่เครื่องหมายซีมิโคลอนหลังจากจบคำสั่งใน Block

4. สร้างตัวแปรใหม่โดยใช้ variables(var) เพื่อเก็บค่าคงที่ซึ่งได้แก่ strings numbers และ date อื่นๆ และเมื่อเรียกใช้ var จะสามารถใช้ @ นำหน้าตัวแปร ดังรูปที่ 2.24

```
<!-- Storing a string -->
@{ var welcomeMessage = "Welcome, new members!"; }
<p>@welcomeMessage</p>

<!-- Storing a date -->
@{ var year = DateTime.Now.Year; }

<!-- Displaying a variable -->
<p>Welcome to our new members who joined in @year!</p>
```

รูปที่ 2.24 การสร้างตัวแปรใหม่เก็บค่าคงที่

5. ในการระบุข้อความ string ให้ปิดด้วยเครื่องหมายอัฒประกาศ (" ") หรือเครื่องหมายฟันทนหรือเครื่องหมายคำพูดก็ได้ รูปที่ 2.25

```
@{ var myString = "This is a string literal"; }
```

รูปที่ 2.25 การระบุข้อความให้ปิดด้วยเครื่องหมายอัฒประกาศ (" ")

ถ้าตัวอักษรที่ต้องการแสดงผลมีเครื่องหมายแบคสแลช (\) หรือมีเครื่องหมายอัฒประกาศ (" ") ให้ใช้ @ นำหน้าเพื่อเป็นการดำเนินการ (ใน C# เครื่องหมาย \ มีความหมายพิเศษเว้นแต่จะใช้ @ นำหน้าโปรแกรมก็จะรู้) ดังรูปที่ 2.26

```
<!-- Embedding a backslash in a string -->
@{ var myFilePath = @"C:\MyFolder\"; }
<p>The path is: @myFilePath</p>
```

รูปที่ 2.26 การใช้เครื่องหมายแบคสแลช และเครื่องหมายอัฒประกาศ

ใช้เครื่องหมายอัฒประกาศ 2 ครั้งเพื่อเป็นการแสดงข้อความในเครื่องหมายอัฒประกาศ (" ") และเครื่องหมาย @ จะถูกใช้เพิ่มเมื่อมีการใช้เครื่องหมายอัฒประกาศซ้ำ ดังรูปที่ 2.27

```
<!-- Embedding double quotation marks in a string -->
@{ var myQuote = @"The person said: ""Hello, today is Monday."""; }
<p>@myQuote</p>
```

รูปที่ 2.27 การใช้เครื่องหมายอัฒประกาศ 2 ครั้ง

6. ในคำหลัก C# (เช่น var true และ if) ชื่อตัวแปรเป็นแบบ case sensitive คือ อักษรตัวเล็กตัวใหญ่มีผลต่างกัน ด้านล่างการเขียนโค้ด 2 ตัวจะแตกต่างกัน ทั้งๆ ที่อ่านเหมือนกันก็ตาม lastName และ LastName. ดังรูปที่ 2.28

```
@{
    var lastName = "Smith";
    var LastName = "Jones";
}
```

รูปที่ 2.28 ในคำหลัก C# แบบ case sensitive

ถ้าประกาศตัวแปรเป็น var lastName = "Smith"; และไปเรียกใช้ @LastName มันจะเกิดข้อผิดพลาดขึ้นทันทีเพราะว่า LastName ไม่เหมือนกับ lastName ถ้าใช้ Visual Basic ตัวแปรจะไม่มีผลสำคัญ จะเล็กหรือจะใหญ่ก็ได้

7. การใส่หมายเหตุ ดังรูปที่ 2.29

```

@* A one-line code comment. *@
@*
   This is a multiline code comment.
   It can continue for any number of lines.
*@

```

รูปที่ 2.29 การใส่หมายเหตุแบบ Razor

8. ตัวแปรและประเภทข้อมูล สามารถประกาศตัวแปรแบบไม่ระบุประเภทข้อมูลได้โดยใช้ var นำหน้า หรือจะระบุประเภทตัวแปรเลยก็ได้ ดังรูปที่ 2.30

```

@{
    // Assigning a string to a variable.
    var greeting = "Welcome!";

    // Assigning a number to a variable.
    var theCount = 3;

    // Assigning an expression to a variable.
    var monthlyTotal = theCount + 5;

    // Assigning a date value to a variable.
    var today = DateTime.Today;

    // Assigning the current page's URL to a variable.
    var myPath = this.Request.Url;

    // Declaring variables using explicit data types.
    string name = "Joe";
    int count = 5;
    DateTime tomorrow = DateTime.Now.AddDays(1);
}

```

รูปที่ 2.30 ตัวแปรและประเภทข้อมูล

2.3.3 การเข้าถึง root เสมือน

ใช้ ~ เรียกว่า ตัวหนอน หรือเรียกว่า ทิลด้า ดังรูปที่ 2.31

```

@{
    var myImagesFolder = "~/images";
    var myStyleSheet = "~/styles/StyleSheet.css";
}

```

รูปที่ 2.31 การเข้าถึง root เสมือน

2.3.4 การแปลง Root เสมือนเป็น path บนไดรว์

ใช้คำสั่ง `server.MapPath` จะแสดงพาทออกมาเป็นตำแหน่งใน hdd ดังรูปที่ 2.32

```
@{
    var dataFilePath = "~/dataFile.txt";
}
<!-- Displays a physical path C:\Websites\MyWebSite\datafile.txt -->
<p>@Server.MapPath(dataFilePath)</p>
```

รูปที่ 2.32 การแปลง Root เสมือนเป็น path บนไดรว์

2.4 Data Annotations

Data Annotation มีไว้เพื่อแสดง format ของ state ตนเองไว้ด้วย ซึ่ง Data Annotation จะอยู่ใน library `System.ComponentModel.DataAnnotations` มันสำคัญมากต่อการสร้าง view กับการ validate model โดย Html Helper

2.4.1 Required

เป็นการ validate เพื่อตรวจสอบว่า input นั้นมีการใส่ข้อมูลหรือยังใช้ระบุเพื่อบอกว่าต้องการ field นี้ห้ามปล่อยทิ้งว่างไว้ เมื่อมีการ validate ที่ view

2.4.2 MinLength

จำกัดจำนวนตัวอักษรภายใน input ให้มีไม่น้อยกว่า {number} ตัวอักษร สามารถใช้ได้โดย `ng-minlength="{number}"`

2.4.3 MaxLength

จำกัดจำนวนตัวอักษรภายใน input ให้มีตัวอักษรไม่เกิน {number} ตัวอักษร สามารถเขียนได้แบบนี้ `ng-maxlength="{number}"`

2.4.4 Display

เมื่อระบุไว้ที่ field ใดๆแล้วมันจะแสดงตามข้อความที่เราระบุ เพื่อแสดงให้เห็น user เห็นในแบบเข้าใจได้ง่ายขึ้น

2.4.5 DataType

ใช้ระบุว่า field นี้เป็นข้อมูลชนิดอะไร เช่น เป็น EmailAddress ซึ่ง validator จะ validate field นี้ด้วย format ของ email

2.4.6 Compare

ใช้เปรียบเทียบ field สอง field ว่ามีค่าเหมือนกันหรือไม่ เช่น validate ข้อมูล password กับ confirm password ที่กรอกใน view ต้องเหมือนกัน

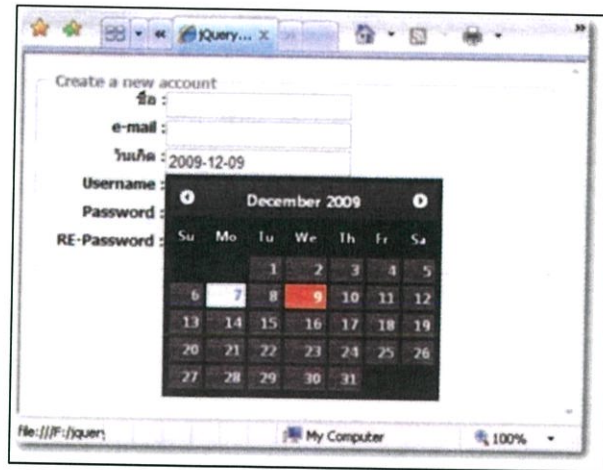
2.5 jQuery

jQuery เป็น Library ที่พัฒนาขึ้นมาจากภาษา JavaScript (JavaScript Library) เพื่อช่วยอำนวยความสะดวกให้แก่ักพัฒนาเว็บ สำหรับการพัฒนาลูกเล่น Action ต่างๆ ที่จำเป็นต้องใช้ในการพัฒนาเว็บ โดยมีการทำงานควบคู่ไปกับ DOM และ Event ต่างๆ ที่ใช้งานเว็บได้กระทำต่อหน้า Web page รวมถึงในการพัฒนา Web Application ในรูปแบบของ AJAX Technology เพื่อการติดต่อ Server Side Script อย่างเช่น PHP หรือ ASP.NET เป็นต้น

JavaScript Library ที่นักพัฒนาเว็บให้ความสนใจมีอยู่มากมายหลายตัว ส่วนใหญ่ก็จะเปิดโอกาสให้ผู้ที่สนใจดาวน์โหลดไปใช้งานกันได้ฟรี เช่น Dojo Prototype และ jQuery ซึ่งก็เป็นหนึ่งใน JavaScript Library ที่ได้รับความสนใจจากนักพัฒนาเว็บทั่วโลก และมี Community ที่ใหญ่พอสมควร ทำให้ผู้ที่ต้องการใช้งาน jQuery มีโอกาสได้รับคำปรึกษาจากผู้ศึกษาและใช้งาน jQuery ได้อย่างง่ายดาย หรือมี Plugin ออกมาให้เราดาวน์โหลดไปใช้งานคู่กับ jQuery อย่างมากมาย

แม้แต่ Microsoft ก็ได้ประกาศสนับสนุน jQuery โดยนำเอา jQuery ผสมลงไปกับชุด Visual Studio ตั้งแต่เวอร์ชัน 2008 ขึ้นมา เพื่อใช้เป็น JavaScript Framework ในการพัฒนาเว็บควบคู่ไปกับ Tools ของ Microsoft เอง

เราสามารถสร้างเอฟเฟ็คต์แปลกๆ สร้างเว็บให้มีลูกเล่นที่น่าสนใจได้อย่างง่ายดาย และไม่ยึดติดกับแพลตฟอร์ม (Platform) ใดๆ ไม่ว่าเว็บไซต์จะพัฒนาด้วย PHP หรือ ASP.NET หรืออื่นๆ ก็สามารถใช้งาน jQuery ร่วมด้วยได้ทั้งหมด ตัวอย่างเช่น หากเราต้องการสร้างปฏิทินสำหรับอำนวยความสะดวกให้แก่ผู้ที่กำลังจะใช้งานระบบ Web Application ของเรา เราสามารถสร้างได้โดยใช้โดยไม่ถึง 10 บรรทัด ดังรูปที่ 2.33



รูปที่ 2.33 ตัวอย่างการสร้างปฏิทินด้วย jQuery

jQuery เป็นไลบรารีที่พัฒนามาจากภาษา JavaScript ซึ่งเป็น Open Source เราสามารถดาวน์โหลดจากเว็บไซต์ของผู้พัฒนา jQuery ได้โดยตรง โดยเข้าไปที่ <http://www.jquery.com> ดังรูปที่ 2.34



รูปที่ 2.34 การดาวน์โหลด jQuery จากหน้าเว็บ <http://www.jquery.com>

ซึ่งไฟล์ที่ดาวน์โหลดมานั้น จะอยู่ในรูปแบบของ JavaScript นั่นก็คือ เป็นไฟล์ที่มีนามสกุลไฟล์ *.js นั่นเอง ตัวอย่างชื่อไฟล์ เช่น jquery-1.4.1min.js หมายเลขด้านหลังชื่อ jQuery คือชื่อ Version ของ jQuery Library

ความสามารถของตัว jQuery ในการพัฒนาเว็บไซต์ได้โดยสรุปมีดังนี้

- พัฒนาลูกเล่นบนหน้าเว็บ เช่น การซ่อน-การแสดงผล Contents ต่างๆ บนหน้าเว็บ
- พัฒนาในส่วนของการ Valid หรือการตรวจสอบข้อมูลที่ผู้ใช้ต้องกรอกที่หน้า Web Form
- พัฒนาในส่วนของ UI หรือ User Interface หน้าเว็บต่างๆ
- พัฒนาในส่วนของ AJAX สำหรับการเขียนโปรแกรมที่ต้องมีการติดต่อกับ Server Side Script อย่างเช่น PHP ASP.NET เป็นต้น

- พัฒนา Plugin เพิ่มเติมอื่นๆ ก็สามารถทำได้โดยง่าย

รูปที่ 2.35 ตัวอย่างการใช้ jQuery เพื่อตรวจสอบข้อมูลก่อนการส่งข้อมูลไปประมวลผล

2.5.1 อธิบายการทำงานของ jQuery

jQuery เริ่มต้นที่ `$(document).ready(fn)` จากที่เราได้เห็นโค้ดตัวอย่างกันไปจากขั้นตอนที่แล้ว ที่ `$(document).ready(fn)` จะหมายถึงการโหลดหน้าเอกสารของ DOM เพื่อให้พร้อมใช้งาน jQuery ถ้าหากเราเปรียบเทียบกับ JavaScript แบบปกติ ก็จะเทียบได้กับคำสั่ง `onload` โดยที่เครื่องหมาย `$` แทน jQuery ซึ่งการเขียนนี้เราสามารถเขียนได้ 4 รูปแบบ ดังนี้

รูปที่ 1

```

jQuery(document).ready(function() {
  // jQuery Code
});

```

รูปที่ 2

```

$(document).ready(function() { //เปลี่ยนคำว่า jQuery เป็นเครื่องหมาย $
  // jQuery Code
});

```

รูป
 รูปแบบ
 ที่ 3

```

$(function() { //ไม่ต้องพิมพ์ document ลงไป
  // jQuery Code
});

```

รูป
 รูปแบบ
 ที่ 4

```

$(function() { //เขียนแบบ Short Hand
  // jQuery Code
});

```

ทั้ง 4 รูปแบบที่กล่าวมาีการทำงานเหมือนกัน แตกต่างกันที่รูปแบบการเขียนเท่านั้น แต่วิธีที่นักพัฒนาเว็บด้วย รูปแบบที่ 1

2.5.2 การเลือกตำแหน่งที่ต้องกระทำด้วย Selectors

การใช้งาน jQuery จะมีการใช้งาน Selectors เข้ามาเกี่ยวข้อง โดย jQuery จะนำเอา Selectors มาเป็นสิ่งที่กำหนดการจับการกระทำ (Event) ที่ต้องการให้เกิดขึ้นกับสิ่งนั้นๆ โดยสามารถพิจารณาจากรูปที่ 2.36

The diagram shows a snippet of HTML and JavaScript code with five numbered annotations:

- 1 Selectors**: Points to the `$("#btnAlert")` selector in the `click` event handler.
- 2 Event**: Points to the `click` event name in the `click` method.
- 3 Function ที่ต้องการให้โปรแกรมทำ**: Points to the `function()` block containing `alert("Hello jQuery");`.
- 4 โค้ดสร้างปุ่มบนเว็บเพจ**: Points to the `<input type="button" name="btnAlert" id="btnAlert" value="Click Me !" />` HTML tag.
- 5 ID ที่กำหนดให้ปุ่ม**: Points to the `id="btnAlert"` attribute in the HTML tag.

รูปที่ 2.36 การเลือกตำแหน่งที่ต้องกระทำด้วย Selectors

โค้ดข้างต้นที่ยกตัวอย่าง เป็นการกำหนดว่าถ้ามีการคลิกปุ่มจะมีหน้าต่าง Alert เกิดขึ้นมา โดยบรรทัดที่ 6 เป็นการกำหนดว่า Selector ที่ต้องการตรวจจับ Event ที่จะเกิดขึ้นกับ TAG ที่มี Selector ตรงกับที่ระบุไว้ โดย Event ที่ใช้ในการตรวจจับในที่นี้คือ click นั้นหมายความว่า ถ้ามีการ Click ไปที่ TAG ที่มี ID ตรงกับที่ระบุใน Selector จะให้ Function

ภายในทำงาน ซึ่งการทำงานของ Function คือการสร้าง Alert ข้อความนั่นเอง ถ้าเปรียบเทียบ Event click ของ jQuery กับ JavaScript โดยปกติจะเทียบได้กับ Event onclick นั่นเอง

2.5.3 Method และคำสั่งของ jQuery

รวมชื่อเมธอดและคำสั่งของ jQuery เพื่อไว้ใช้อ้างอิงกับการเขียนโปรแกรมด้วย jQuery โดยที่สัญลักษณ์ที่จะเห็นต่อไปในการใช้อธิบายเมธอด มีดังนี้

- Dot (.) หน้าชื่อเมธอด หมายถึงต้องมีการอ้างถึง jQuery หรือ Selector ก่อนใช้งาน เมธอด
- Comma (,) หมายถึง การคั่นระหว่าง Parameter ที่ส่งไป หากมีการส่งมากกว่า 1 ค่า
- [...] หมายถึง Options ของ Method ที่จะมีการส่งค่าไปด้วยหรือไม่ก็ได้
- Fn หมายถึง Function ที่ต้องการนำมาใช้งานกับเมธอด
- Bool หมายถึง ค่าความเป็นจริง Boolean ที่เป็นได้เพียง 2 ค่า คือ True (1) หรือ False (0)
- Exp หมายถึง expression ข้อความตามรูปแบบที่กำหนด

Selectors Expressions

ตารางที่ 2.4 ตารางแสดงตัวอย่าง Selectors Expressions (1)

Selectors	ความหมาย
:not(x)	Element ที่ไม่มี Element ที่ระบุ เช่น p:not(:last) Tag p ทุก Tag ที่ไม่ใช่ Tag สุดท้าย เป็นต้น
:even	เลือกกระทำเฉพาะ Tag ลำดับที่เป็นเลขคู่ โดยเริ่มนับที่ 0 เช่น p:even เป็นต้น
:odd	เลือกกระทำเฉพาะ Tag ลำดับที่เป็นเลขคี่ โดยเริ่มนับที่ 0 เช่น p:odd เป็นต้น
:eq(index)	เลือกกระทำเฉพาะลำดับ Element ที่ตรงกับ index ที่ระบุ โดยเริ่มนับลำดับที่ 0 เช่น p:eq(2) เป็นต้น
:gt(index)	เลือกกระทำเฉพาะลำดับ Element ที่มากกว่า index ที่ระบุ โดยเริ่มนับลำดับที่ 0 เช่น p:gt(1) เป็นต้น
:lt(index)	เลือกกระทำเฉพาะลำดับ Element ที่น้อยกว่า index ที่ระบุ โดยเริ่มนับลำดับที่ 0 เช่น p:lt(5) เป็นต้น
:header	เลือกกระทำเฉพาะ Header Tag เท่านั้น เช่น <h1>, <h2> เป็นต้น
:animate	เลือกกระทำกับ Element ที่ระบุ และที่กำลังมีการเคลื่อนไหวทำงาน เช่น \$("p:animate") เป็นต้น
:contains(text)	เลือกกระทำเฉพาะ Element ที่มีข้อความตามที่ระบุ เช่น p:contains('Website')
:empty	เลือกกระทำเฉพาะ Element ไม่มีข้อมูล เช่น p:empty เป็นต้น
:has(x)	เลือกกระทำเฉพาะ Element ที่มี Element ที่ระบุ เช่น p:has(u) เป็นต้น
:parent	เลือกกระทำเฉพาะ Element ที่มี Element ลูก หรือมีข้อมูลภายใน Element เช่น p:parent เป็นต้น
:hidden	เลือกกระทำเฉพาะ Element ที่มีสถานะเป็นการซ่อนการแสดงผล (มีการกำหนด display:none) เช่น p:hidden
:visible	เลือกกระทำเฉพาะ Element ที่มีสถานะเป็นการแสดงผล เช่น p:visible
[attr]	Element ที่มีชื่อ Attribute ที่กำหนดในเครื่องหมาย [..] เช่น [src]
[attr=value]	Element ที่มีชื่อ Attribute และ Value ที่กำหนดในเครื่องหมาย [...] เช่น [src=cartoon.gif]
[attr!=value]	Element ที่มีชื่อ Attribute และมี Value ที่ไม่ใช่ที่กำหนด เช่น [src != cartoon.gif]
[attr^=value]	Element ที่มีชื่อ Attribute และมี Value ขึ้นต้นด้วยค่าที่กำหนด เช่น [src ^= car]
[attr\$=value]	Element ที่มีชื่อ Attribute และมี Value ลงท้ายด้วยค่าที่กำหนด เช่น [src \$= car]
[attr*=value]	Element ที่มีชื่อ Attribute และมี Value ที่มีข้อความภายในตรงกับ Value ด้วยค่าที่กำหนด เช่น [src *= car]
ตัวอย่างการเขียน \$("div[align=center]")	

ตารางที่ 2.5 ตารางแสดงตัวอย่าง Selectors Expressions (2)

Selectors	ความหมาย
*	ทุก Element บนหน้าเว็บเพจ
#id	Element ที่มี ID ที่ระบุ
TagName	Element ที่มีชื่อตรงกับชื่อ TAG ที่ระบุ
.ClassName	ทุก Element ที่มี Class ตรงกับที่ระบุ
x, y	Element ที่มีค่าตรงกับ x หรือ y
x y	Element y ที่อยู่ภายใน Element x
x > y	Element y ที่เป็น Element ลูก Element x
x + y	Element y ที่เป็น Element ถัดต่อไปจาก Element x
x ~ y	Element y ที่เป็น Element หลังจาก Element x
:first	Element แรกสุด เช่น p:first เป็นต้น
:last	Element สุดท้าย เช่น p:last เป็นต้น

CSS Method

ตารางที่ 2.6 ตารางแสดงตัวอย่าง CSS Method (1)

Method	ความหมาย
.css(name)	การเข้าถึง CSS จาก Properties Name ที่ระบุ เช่น <code>\$("#ShowText").css("border-color")</code>
.css(properties)	กำหนดคุณสมบัติ CSS ให้กับ Selector ที่เรียกใช้ เช่น <code>\$("#ShowText").css({color: '#005566', 'font-size': '2em'})</code> ;
.css(name, value)	กำหนดคุณสมบัติ CSS ให้กับ Selector ที่เรียกใช้ โดยกำหนดได้ครั้งละ 1 คุณสมบัติ เช่น <code>\$("#ShowText").css('color', '#FF0000')</code> ;
.addClass(className)	เพิ่ม CSS Class ให้กับ Selector ที่เรียกใช้ เช่น <code>\$("#ShowText").addClass("font24")</code>
.removeClass(className)	ลบ CSS Class ออกจาก Selector ที่เรียกใช้
.toggleClass(class)	สลับการทำงานระหว่าง Add และ Remove CSS Class ที่ระบุ
.hasClass(class)	ตรวจสอบ CSS Class กับ Selector ที่เรียกใช้ว่ามีอยู่หรือไม่ โดยส่งค่ากลับเป็น Boolean เช่น <code>\$("#ShowText").hasClass("font24")</code>
.offset()	คืนเป็นค่าระยะความห่าง Left, Top จาก Element ที่เรียกใช้ โดยเทียบกับหน้าเอกสาร
.position()	คืนค่าตำแหน่งของ Element โดยเทียบกับ Parent Element
.scrollTop([val])	ค่า Scroll Bar จากทางด้านบน สามารถกำหนดค่าเป็นตัวเลขได้
.scrollLeft([val])	ค่า Scroll Bar จากทางด้านซ้าย สามารถกำหนดค่าเป็นตัวเลขได้
.height([val])	ค่าความสูงของ Element สามารถกำหนดค่าเป็นตัวเลขได้
.width([val])	ค่าความกว้างของ Element สามารถกำหนดค่าเป็นตัวเลขได้
.innerHeight()	ค่าความสูงของ Element โดยรวมค่า padding
.innerWidth()	ค่าความกว้างของ Element โดยรวมค่า padding
.outerHeight(margin)	ค่าความสูงของ Element โดยรวมค่า padding และค่า Border
.outerWidth(margin)	ค่าความกว้างของ Element โดยรวมค่า padding และค่า Border

ตัวอย่างการเขียน `$("#ShowText").css("color", "#FF0000");`

ตารางที่ 2.7 ตารางแสดงตัวอย่าง CSS Method (2)

Method หรือ คำสั่ง	ความหมาย
.ready(fn)	โหลด DOM เพื่อพร้อมที่จะจัดการการทำงาน (ใช้เป็นเมธอดเริ่มต้นในการเขียน jQuery)
.bind(eventType, [data], fn)	ผูก Event เข้ากับ Selector เช่น <code>\$("#button1").bind("click", {str: 'Hello'}, function(e){alert(e.data.str);})</code>
.one(eventType, [data], fn)	ผูก Event เข้ากับ Selector โดยจะมีการกระทำเพียง 1 ครั้ง ต่อ 1 Object เท่านั้น
.unbind([eventType],[fn])	ยกเลิกการผูก Event
.live(type, fn)	ผูก Event เข้ากับ Selector แต่เป็นในรูปแบบของ Auto Bind สามารถใช้กับ Element ที่เกิดขึ้นในแบบ Auto ในหน้าเว็บเพจได้
.die(type, [fn])	ยกเลิกการผูก Event ที่ได้จาก .live()
.blur(fn)	ยกเลิกหรือย้ายจาก Object ที่เคย Focus
.change(fn)	มีการเปลี่ยนแปลงข้อมูล
.click(fn)	มีการคลิก
.dblclick(fn)	มีการดับเบิลคลิก
.error(fn)	เมื่อมีการทำงานผิดพลาดเกิดขึ้น เช่น รูปภาพบนหน้าเว็บไม่ปรากฏ (<code>\$('img').error(fn)</code>) เป็นต้น
.focus(fn)	เมื่อมีการ Focus ที่ Object
.keydown(fn)	เมื่อมีการกดปุ่มบนแป้นคีย์บอร์ดทุกปุ่ม
.keypress(fn)	เมื่อมีการกดปุ่มที่แป้นคีย์บอร์ด ยกเว้นปุ่มที่เป็น Function ของคีย์บอร์ด เช่น ปุ่ม Num Lock เป็นต้น
.keyup(fn)	เมื่อมีการยกนิ้วออกจากปุ่มคีย์บอร์ด
.load(fn)	เมื่อมีการโหลดเกิดขึ้น เช่น การโหลดรูปภาพของ <code></code>
.mousedown(fn)	เมื่อมีการคลิกเมาส์ (คลิกขวาหรือคลิกซ้าย)
.mouseenter(fn)	เมื่อมีเมาส์มาชี้
.mouseleave(fn)	เมื่อนำเมาส์ที่ชี้ออกไป
.mousemove(fn)	เมื่อมีการเคลื่อนย้ายเมาส์
.mouseout(fn)	เมื่อนำเมาส์ออกจาก Object
.mouseover(fn)	เมื่อนำเมาส์มาวางเหนือ Object
.mouseup(fn)	เมื่อมีการคลิกแล้วปล่อย
.resize(fn)	เมื่อมีการเปลี่ยนแปลงขนาด เช่น การเปลี่ยนขนาดของ Browser เป็นต้น
.scroll(fn)	เมื่อมีการเลื่อน Scroll
.select(fn)	เมื่อมีการเลือก Object
.submit(fn)	เมื่อมีการคลิกเพื่อส่งข้อมูลของ Form
.unload(fn)	เมื่อมีการยกเลิกการโหลด เช่น การปิดหน้าต่างของ Browser เป็นต้น
.hover(fn)	เมื่อมีเมาส์มาวางเหนือ Object
.toggle(fn1, fn2, fn3,)	สลับการทำงานระหว่าง Function 1, 2, 3,
.trigger(eventType, [data])	เมื่อมีการกระทำที่ Object อื่น Object ที่ถูกกำหนดเป็น trigger() จะเกิดการกระทำขึ้นด้วย

ตัวอย่างการเขียน `$("#button1").click(fn)`

Effect Method

ตารางที่ 2.8 ตารางแสดงตัวอย่าง Effect Method

Method	ความหมาย
.show([speed], [fn])	แสดงผล ("slow", "normal", "fast") เช่น show("slow")
.hide([speed], [fn])	ซ่อนการแสดงผล ("slow", "normal", "fast")
.toggle([speed], [fn])	สลับการแสดงผลระหว่างการ show และ hide
.slideDown([speed], [fn])	แสดงผลจากบนสู่ล่าง ("slow", "normal", "fast")
.slideUp([speed], [fn])	แสดงผลจากล่างสู่บน ("slow", "normal", "fast")
.slideToggle([speed], [fn])	สลับการแสดงผลระหว่าง up และ down ("slow", "normal", "fast")
.fadeIn([speed], [fn])	วัตถุค่อยๆ ปรากฏขึ้น ("slow", "def", or "fast") เช่น fadeIn("def")
.fadeOut([speed], [fn])	วัตถุค่อยๆ จางหายไป
.fadeTo(speed, opacity, [fn])	วัตถุค่อยๆ จางหายไปจนถึงค่า Opacity ที่กำหนด ซึ่งมีค่าตั้งแต่ 0-1 เช่น .fadeTo("slow", 0.50)
.animate()	สร้าง Custom Animations
.stop()	หยุดการเคลื่อนไหวของ Animations
ตัวอย่างการเขียน \$('#ShowText').hide("slow")	

DOM Traversal Method

ตารางที่ 2.9 ตารางแสดงตัวอย่าง DOM Traversal Method

Method	ความหมาย
.filter(selector)	เลือกเฉพาะ Element ที่ตรงกับ Selector ที่ระบุ เช่น \$('div').filter("last")
.filter(fn)	เลือกเฉพาะ Element ที่ตรงกับ Function ทำงาน
.eq(index)	เลือกเฉพาะ Element ที่ตรงกับ Index ระบุ
.slice(start [, end])	แบ่ง Element ที่มี โดยการระบุจุดเริ่มต้นและสิ้นสุด โดยเริ่มที่ 0
.not(expr)	เลือกเฉพาะ Element ที่ไม่ตรงกับเงื่อนไขที่ระบุ
.find(expr)	เลือกเฉพาะ Element ที่ค้นหาพบ เช่น \$('p').find("u")
.contents()	เข้าถึงข้อมูลภายใน เช่น \$('p:last').contents()
.children()	เข้าถึง Element ลูก เช่น \$('p').children("a")
.next()	เข้าถึง Element ในลำดับถัดไป
.nextAll()	เข้าถึง Element ในลำดับถัดไปทั้งหมด
.prev()	เข้าถึง Element ในลำดับก่อนหน้า
.prevAll()	เข้าถึง Element ในลำดับก่อนหน้าทั้งหมด
.siblings()	เข้าถึง Element ที่อยู่ถัดจาก Selector ที่กำหนด เช่น \$('#ShowText').siblings()
.parent()	เข้าถึง Element ที่เป็น Element ต้นของ Element ที่ Selector ระบุ
.parents()	เข้าถึง Element ที่เป็น Element ต้นของ Element ที่ Selector ระบุทั้งหมด (รวมถึง Root Element ด้วย)
.closest(expr)	หา Element ที่ระบุ โดยที่ Element ที่ Element ที่ใกล้ที่สุดจะนำคุณสมบัติ นั้นมาใช้ เช่น \$('input').closest("div,p").addClass("bgBlock");
.andSelf()	เข้าถึง Element ทั้งก่อนหน้าและ Element ที่ระบุ แล้วทำการเพิ่มเติมสิ่งที่ต้องการ เช่น \$("label").find("input").andSelf().css("background-color", "#550022");
.end()	เข้าถึง Element ที่ระบุ โดยครอบคลุมข้อมูลทั้งหมดของ Element เช่น \$("strong").find("u").end().css("color", "#FF0000"); Query Traversing/<u>end()</u>
ตัวอย่างการเขียน \$('#ShowText').prev()	

Dom Manipulation Method

ตารางที่ 2.10 ตารางแสดงตัวอย่าง Dom Manipulation Method

Method	ความหมาย
<code>attr(name)</code>	การเข้าถึง Attribute ของ Element
<code>attr(key, value)</code>	การเข้าถึง Attribute และค่าของ Attribute
<code>attr(key, fn)</code>	การเข้าถึง Attribute และนำมาใช้งานกับ Function
<code>removeAttr(name)</code>	ลบ Attribute ที่ระบุออกจาก Element
<code>html()</code>	นำ TAG HTML ออกจาก Element ระบุ เช่น <code>\$("p").html()</code>
<code>html(value)</code>	เขียนข้อความที่ Element ที่ระบุ โดยจะแปลความหมายของ TAG HTML
<code>text()</code>	เข้าถึงเฉพาะข้อความที่เป็น Text จาก Element ที่ระบุ
<code>text(value)</code>	เขียนข้อความที่ Element ตามที่ระบุ โดยไม่แปลความหมาย TAG HTML
<code>val()</code>	เข้าถึง Attribute ที่ชื่อ value ของ Element
<code>val(value)</code>	กำหนดค่าให้กับ Attribute value
<code>append(content)</code>	เพิ่มข้อมูล (content) ต่อท้ายของเดิมที่ Element ที่ระบุ เช่น <code>\$("#ShowText").append("ข้อความต่อท้าย")</code>
<code>appendTo(selector)</code>	นำข้อมูลใน Selector ที่ระบุ ไปแสดงด้านหน้า Selector ที่เรียกใช้ เช่น <code>\$("#ShowText1").appendTo("#ShowText2")</code>
<code>prepend(content)</code>	นำข้อมูล (content) ไปวางด้านหน้าข้อมูล Element ที่ระบุ
<code>prependTo(selector)</code>	นำข้อมูลใน Selector ที่ระบุ ไปแสดงด้านหลังต่อจาก Selector ที่เรียกใช้งาน โดยแทรกไปยัง Element ที่เรียกใช้
<code>.after(content)</code>	นำข้อมูล (content) ไปแสดงหลัง Element ที่ระบุ
<code>.insertAfter(selector)</code>	นำข้อมูลของ Selector ที่เรียกใช้ มาแสดงหลัง Selector ที่ระบุ เช่น <code>\$("#ShowText1").insertAfter("#ShowText2")</code>
<code>.before(content)</code>	ความหมายตรงข้ามกับ <code>.after()</code>
<code>.insertBefore(selector)</code>	ความหมายตรงข้ามกับ <code>.insertAfter()</code>
<code>.wrap(html or elem)</code>	นำ TAG HTML ที่ระบุไปครอบ Selector ที่เรียกใช้ เช่น <code>\$("#ShowText").wrap("")</code> เป็นการครอบทีละ Element
<code>.wrapAll(html or elem)</code>	นำ TAG HTML ที่ระบุไปครอบ Selector ที่เรียกใช้ โดยเป็นการครอบในแบบทั้งหมด
<code>.wrapInner(html or elem)</code>	นำ TAG HTML ที่ระบุไปใช้กับ Selector ที่เรียกใช้ โดยการครอบด้านในของ Selector ที่เรียกใช้
<code>.replaceWith(content)</code>	แทนที่ข้อความด้วยข้อมูลที่ระบุ
<code>.replaceAll(selector)</code>	แทนที่ทั้งหมดไปที่ Selector ที่ระบุ เช่น <code>\$("ข้อความ").replaceAll("div")</code>
<code>.empty()</code>	ลบ Child Element ต่างๆ ที่อยู่ใน Selector ให้เป็นค่าว่าง
<code>remove([expr])</code>	ลบ Element ที่ระบุใน Selector เช่น <code>\$("#ShowText").remove()</code>
<code>clone([bool])</code>	ทำการ Copy จาก Selector ที่เรียกใช้ไปยัง Element ที่ต้องการ หากกำหนดค่า Boolean เป็น True จะหมายถึง Copy ให้นำ Event ไปใช้ด้วย เช่น <code>\$("#ShowText").clone().appendTo("div last");</code>

ตัวอย่างการเขียน `$("#ShowText").wrap("");`

AJAX Method

ตารางที่ 2.11 ตารางแสดงตัวอย่าง AJAX Method

Method	ความหมาย
<code>\$ajax(options)</code>	สร้าง AJAX Request เป็นเมธอดระดับแรกที่ใช้กับการทำงานด้าน AJAX (รายละเอียดการใช้งานบทที่ ๑)
<code>load(url, [data, callback])</code>	สร้าง AJAX Request ด้วยการโหลดข้อมูลจาก URL ที่ระบุ แล้วนำมาแสดงกับ Element ที่เรียกใช้งาน
<code>\$get(url, [data, callback, type])</code>	สร้าง HTTP GET Request (รายละเอียดการใช้งานบทที่ ๑)
<code>\$post(url, [data, callback, type])</code>	สร้าง HTTP POST Request (รายละเอียดการใช้งานบทที่ ๑)
<code>\$getJSON(url, [data, callback])</code>	สร้าง HTTP GET Request โดยรับข้อมูลกลับในรูปแบบของ JSON
<code>\$getScript(url, [callback])</code>	สร้าง HTTP GET Request โดยรับข้อมูลกลับในรูปแบบของ JavaScript
<code>\$ajaxSetup(option)</code>	กำหนดค่าเริ่มต้นให้ AJAX Request
<code>.ajaxComplete(callback)</code>	เมื่อ AJAX ได้ทำงานสมบูรณ์แล้ว
<code>.ajaxError(callback)</code>	เมื่อมีการ Request ผิดพลาดเกิดขึ้น
<code>.ajaxSend(callback)</code>	ทำงานก่อนมีการส่งข้อมูลกับ AJAX
<code>.ajaxStart(callback)</code>	เริ่มต้นการ AJAX Request
<code>.ajaxStop(callback)</code>	เมื่อการ Request สิ้นสุดแล้ว
<code>.ajaxSuccess(callback)</code>	เมื่อมีการ Request สำเร็จ
ตัวอย่างการเขียน <code>\$("ShowText").load("dataFile.php", {name: 'Jack'})</code>	

Utilities Method

ตารางที่ 2.12 ตารางแสดงตัวอย่าง Utilities Method

Method	ความหมาย
<code>\$support()</code>	ตรวจสอบการรองรับการใช้งาน
<code>\$browser</code>	ตรวจสอบ Web Browser เช่น <code>\$browser.msie</code> ดังนี้ msie mozilla opera safari
<code>\$browser.version</code>	ตรวจสอบรุ่นของ Web Browser
<code>\$each(obj, fn)</code>	เข้าถึงสมาชิกแต่ละตัวของ Object
<code>\$extend([deep], target, obj, [objN])</code>	เปลี่ยนแปลงค่าใน target เป็นค่าใน object โดยที่ถ้ามีค่าซ้ำกัน จะนำค่าใน object ไปกำหนดค่า แต่ถ้าค่าไม่ซ้ำกันจะใช้ค่าของ target เอง (อ่านตัวอย่างการใช้งานบทที่ 10)
<code>\$grep(array, callback, [invert])</code>	เข้าถึงสมาชิกแต่ละตัวของ Array
<code>\$makeArray(obj)</code>	เปลี่ยน Object เป็น Array
<code>\$unique(array)</code>	ลบค่า Array ที่ซ้ำกันออก
<code>\$inArray(value, array)</code>	ตรวจสอบว่าค่ามีอยู่ใน Array หรือไม่ ถ้าไม่มีอยู่ใน Array จะคืนค่า -1 หากมีอยู่จะคืนค่าเป็น index ของ Array
<code>\$merge(first, second)</code>	รวมค่า Array เข้าด้วยกัน
<code>\$isArray(obj)</code>	ตรวจสอบว่าเป็น Array หรือไม่
<code>\$isFunction(obj)</code>	ตรวจสอบว่าเป็น Function หรือไม่
<code>\$trim(str)</code>	ตัดช่องว่างหน้าและหลัง String ออก (remove whitespace)
<code>\$param(obj)</code>	สร้างเป็น Parameter เช่น <code>var p = {user: 'admin', pwd: '1234'}</code> <code>var p2 = \$param(p)</code> <code>// user=admin&pwd=1234</code>
<code>jQuery.noConflict()</code>	กลับไปใช้ jQuery แทน \$
ตัวอย่างการเขียน <code>\$("Hello jQuery")</code>	

2.6 JavaScript

JavaScript เป็นภาษา Script ที่ทำงานบนฝั่ง Client ซึ่งก็คือทำงานที่ Web Browser นั้นเอง โดยนักพัฒนาเว็บนิยมนำมาใช้ในการสร้างลูกเล่นต่างๆ บนหน้าเว็บเพจ ปัจจุบัน JavaScript ยังเป็นหัวใจหลักในการพัฒนาของเทคโนโลยีในแบบ AJAX ซึ่งกำลังเป็นที่สนใจของนักพัฒนาเว็บเป็นอย่างมากในทุกวันนี้

2.6.1 หลักการเขียน JavaScript

การเขียน JavaScript ก็เหมือนกับการเขียนภาษาสคริปต์ทั่วไปคือ มีการเขียนในลักษณะ Text File โดยต้องมีการเขียนให้ถูกหลักไวยากรณ์ของภาษานั้นๆ ซึ่งไวยากรณ์ของภาษา JavaScript จะมีรูปแบบคล้ายกับภาษาในกลุ่มของ C Style นั่นก็คือ มีการใช้เครื่องหมายปีกกา { ... } มีการใช้เครื่องหมาย semi-colon (;) ในการจบประโยคคำสั่ง เป็นต้น (กรณีเป็นการจบบรรทัดคำสั่ง แล้วไม่มีคำสั่งใดๆ ต่อท้ายในบรรทัดนั้นๆ จะมีการจบประโยคคำสั่งด้วยการขึ้นบรรทัดใหม่ โดยเราไม่ต้องพิมพ์เครื่องหมาย semi-colon ก็ได้)

2.6.2 การเขียน JavaScript กับ Web Page

เราสามารถเขียนโค้ดของ JavaScript ร่วมกับโดเมนของเว็บเพจได้ ซึ่งนิยมเขียนกัน 2 รูปแบบ คือ แบบ Embedded Script กับ External Script ซึ่งมีรายละเอียดดังนี้

รูปแบบที่ 1 Embedded Script

การเขียน JavaScript แบบ Embedded Script เป็นการเขียน JavaScript ร่วมกับโค้ดของเว็บเพจในไฟล์เดียวกัน โดยนิยมเขียนไว้ระหว่าง TAG <head> ... </head> และมี TAG <script type = "text/javascript"> ... </script> กำกับโค้ดในส่วนที่เป็นของ JavaScript ไว้ ดังตัวอย่าง รูปที่ 2.37

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>JavaScript </title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <script type="text/javascript">
      <!--
        function MyAlert() {
          alert("Hello JavaScript");
        }
      </script>
    </head>
  </body>
</body>
</html>

```

JavaScript เขียนไว้ระหว่าง <head> </head>

รูปที่ 2.37 การเขียน JavaScript กับ Web Page ในรูปแบบ Embedded Script

รูปแบบที่ 2 External Script

การเขียน JavaScript แบบ External Script เป็นการเขียนโค้ด JavaScript แยกเก็บเป็นไฟล์ไว้โดยเฉพาะ ซึ่งไฟล์ที่แยกเก็บไว้จะมีนามสกุลไฟล์เป็น *.js เช่น my-funtion.js เป็นต้น โดยเมื่อเว็บเพจใดต้องการเรียกใช้งาน จะเรียกใช้งานผ่าน Attribute src ของ Tag <script> เช่น <script type = "text/javascript" src=" my-funtion.js"></script> ลองพิจารณาจากโค้ดดังนี้

```
function MyAlert(){
    alert("Hello JavaScript");
}
```

โค้ดในส่วนของไฟล์ my-funtion.js มีดังนี้

โค้ดในส่วนของไฟล์เว็บเพจ mypage.html มีดังนี้

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title> new document </title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <script type="text/javascript src="my-funtion.js"></script>
</head>
<body>
</body>
</html>
```

เรียกใช้งาน JavaScript ที่แยกเก็บเป็นไฟล์ไว้

2.7 CSS

CSS หรือ Cascading Style Sheets เป็นภาษาหรือเทคโนโลยีที่พัฒนาขึ้นมาเพื่อใช้ในการตกแต่งการแสดงผลข้อมูลโดยเฉพาะ css ไม่ใช่เพียงแต่นำมาใช้กับเว็บเพจเท่านั้น แต่ยังมีการนำไปใช้ตกแต่งข้อมูลที่ได้มาจาก XML สำหรับความสามารถของ css นั้นมีมากมาย ขึ้นอยู่กับว่าเราจะไปนำไปประยุกต์ใช้กันในทิศทางไหน แต่ที่นิยมนำมาใช้กับเว็บเพจปัจจุบัน เช่น

- นำ css มาประยุกต์กับการเลย์เอาต์เว็บเพจ ซึ่งการเลย์เอาต์ด้วย css มีผลทำให้เว็บเพจเราโหลดได้รวดเร็วมากยิ่งขึ้น และ โค้ดของเว็บเพจมีความเป็นระเบียบมากยิ่งขึ้น
- css ทำให้เราสามารถเปลี่ยนการแสดงผลได้อย่างง่ายดาย โดยไม่ต้องไปยุ่งเกี่ยวกับข้อมูล หรือ โครงสร้างของหน้าเว็บเพจ
- เมื่อนำ css ไปใช้ร่วมกับ JavaScript ทำให้เราสามารถสร้างความแปลกใหม่ให้กับข้อมูลที่อยู่ในหน้าเว็บเพจได้โดยง่าย

2.7.1 การเขียน css กับเว็บเพจ

อย่างที่ทราบกันคืออยู่แล้วว่า เราสามารถนำ css เข้ามาใช้งานตกแต่งข้อมูลกับเว็บเพจของเราได้โดยการเขียน css ร่วมกับ Web Page มีอยู่ด้วยกัน 3 รูปแบบ คือ

1. Embedded Style เป็นการเขียน css เข้าไปโดยตรงที่ไฟล์เว็บเพจ โดยมีการ

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS : Embedded Style </title>
<style type="text/css">
<!--
.fontRed { font-family: Tahoma, Geneva, sans-serif; font-size: 16px; color: #F00; }
-->
</style>
</head>
<body>
<div class="fontRed">Embedding Style Sheet </div>
</body>
</html>

```

TAG <style type="text/css">...</style> ตัวอย่าง

2. External Style เป็นการเขียน css แยกเก็บไฟล์ไว้โดยเฉพาะ เมื่อต้องการจะนำมาใช้งานกับเว็บเพจ จะมีการประกาศการเรียกใช้งาน ซึ่งการประกาศการเรียกใช้งานไฟล์ css ที่แยกเก็บไว้นั้น ทำได้ 2 รูปแบบ คือ

- i. Linking เป็นการเรียกไฟล์ css เมื่อเว็บเพจใดต้องการเรียกใช้ ก็จะใช้เรียกใช้งานไฟล์ css โดยผ่าน TAG

```
<link href="CssFile.css" type="text/css" rel="stylesheet" />
```

๒

```
.fontRed {
  font-family: Tahoma, Gevena, sans-serif;
  font-size: 16px;
  color: #F00;
}
```

esign.css มีโค้ดดังนี้

ไฟล์ index.html หรือไฟล์เว็บเพจใดๆ มีโค้ดดังนี้

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS : Linking Style </title>
</head>
<body>
<div style="font-family: Tahoma, Gevena, sans-serif; font-size: 16px; color: #F00;">
<div style="font-family: Tahoma, Gevena, sans-serif; font-size: 16px; color: #F00;">
</div>
</div>
</body>
</html>
```

เป็นการเรียกใช้งานไฟล์ css ผ่าน TAG <style>...</style> โดยผ่าน @import และเมื่อนำไฟล์มาใช้งานกับเว็บเพจ เรายังสามารถสร้าง css ใหม่เข้าไปเพิ่มเติมได้เลยทันที ตัวอย่างเช่น ไฟล์ desing.css มีโค้ดดังนี้

๒

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>CSS : Import Style</title>
  <style type="text/css">
    <!--
      @import url("design.css");
      .fontBlue{ font-size: 16px; color: #39; }
    -->
  </style>
</head>
<body>
  <div class="fontRed">Import Style Sheet </div>
</body>
</html>

```

m

1 หรือ ไฟล์เว็บเพจใดๆ มีโค้ดดังนี้

3. Inline Style เป็นการเขียน css ลงไปโดยตรงกับ TAG ที่ต้องการกำหนดค่า โดยการเขียนผ่าน Attribute Style ตัวอย่างดังนี้

การเขียน css ร่วมกับเว็บเพจทั้ง 3 วิธีที่กล่าวมา เราต้องมีการเลือกใช้อย่างเหมาะสม เช่น ถ้าต้องการนำเอา css ซึ่งมีรูปแบบใช้งานเหมือนกันในทุกๆ เว็บเพจ เราก็ควรเลือกการเขียนในรูปแบบ Linking หรือ Import Style แทนที่จะใช้ Embedded Style เนื่องจากว่าถ้าเราเลือกใช้ Embedded Style ในกรณีแบบนี้ เราต้องทำงานซ้ำซ้อน เมื่อมีการแก้ไขโค้ด css เราก็ต้องตามไปแก้ไขกับทุกๆ ไฟล์เว็บเพจ แต่ถ้าเราเปลี่ยนมาใช้ Linkin หรือ Import Style ตามที่กล่าวมา เราเพียงแค่อัปเดตที่ไฟล์ css ไฟล์เดียว ทำให้เราประหยัดเวลาการพัฒนาเว็บลงไปได้มากเลยทีเดียว

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>CSS : Import Style</title>
</head>
<body>
  <strong style="font-size: 22px">Inline Style Sheet </strong>
</body>

```

2.9.2 ไวยากรณ์ของ css

ไวยากรณ์หรือรูปแบบการเขียน css นั้น มีโครงสร้างของไวยากรณ์ให้เราจำกันง่ายๆ อยู่ 4 ส่วนประกอบหลัก ดังนี้

1. Selector
2. เครื่องหมายปีกกา { ... } ครอบ Properties กับ Value ที่ต้องการกำหนดค่า
3. ชื่อ Properties
4. ค่า Value ที่กำหนดให้กับ Properties

CSS มี

โครงสร้าง CSS ดังนี้

```
SelectorName {
  Properties: Value; Properties;
  Value;
}
```

ตัวอย่างโค้ด

```
.fontRed {
  font-size: 20px;
  color: #FF0000
}
```

จากโค้ดข้างต้น หากพิจารณาตามโครงสร้างที่กล่าวมา จะได้

1. Selector คือ .fontRed
2. เครื่องหมายปีกกา {...} ที่มี Properties กับ Value เขียนอยู่ภายใน
3. ชื่อ Properties คือ font-size กับ color
4. ค่า Value คือ 20px ที่กำหนดให้แก่ font-size และ #FF0000 คือค่าที่กำหนดให้กับ color

จากโค้ดเราจะพบว่า มีเครื่องหมาย colon (:) ที่ต้องคั่นระหว่างชื่อ Properties กับ Value และในทุกๆ Properties ต้องจบด้วยเครื่องหมาย semi-colon (;) ซึ่งเป็นไวยากรณ์ของโค้ด CSS

2.7.2 Selector

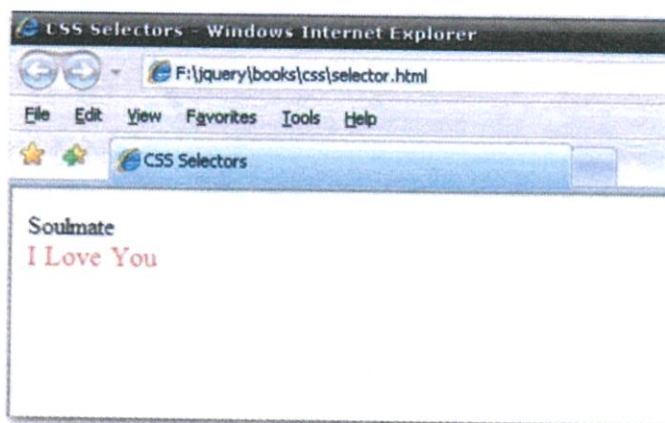
คือ ตัวเลือกที่ใช้เป็นตัวบ่งชี้ว่า จะนำค่าของ CSS ที่ได้กำหนดค่าต่างๆ ไว้ใช้กับ Elements หรือ TAG ใดได้บ้างภายในเว็บเพจ โดยมีการเขียน Selector ไว้หน้ากลุ่มของ Properties ที่ได้มีการกำหนดค่าไว้ ซึ่งพื้นฐาน Selector มีอยู่ 3 รูปแบบที่เราควรจะจดจำ และนำไปใช้ให้เป็น ดังนี้

1. Class Selector

เป็นรูปแบบการเขียนโดยการสร้างชื่อ Class ขึ้นมา ซึ่งชื่อคลาสที่สร้างขึ้นมานำหน้าด้วยเครื่องหมาย Dot (.) เช่น .fontRed{...} เป็นต้น คลาสเมื่อสร้างขึ้นมาแล้ว จะยังไม่แสดงผลอะไรกับข้อมูลในเว็บเพจ หากต้องการให้แสดงผลกับข้อมูลใด ต้องเรียกใช้งาน โดยเรียกใช้ผ่าน Attribute Class ที่เขียนกำกับ

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>CSS Selectors</title>
    <style type="text/css">
      .fontBlue{ color: #FF0000; font-size: 18px; }
    </style>
  </head>
  <body>
    <div>Soulmate </div>
    <div class="fontRer">I Love you </div>
  </body>
</html>
```

ที่ใช้ในการกำกับข้อมูลนั้น เช่น



รูปที่ 2.38 แสดงผลกับ Web Browser แบบ Class Selector

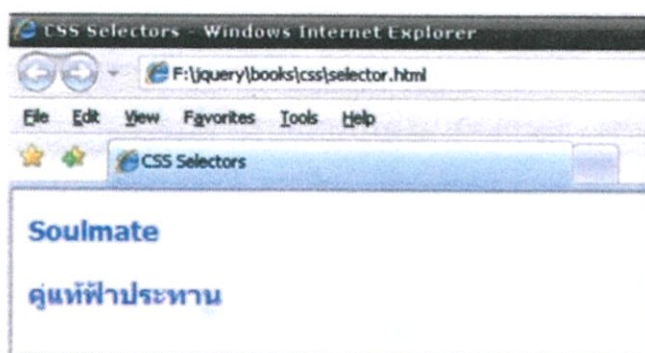
2. TAG Selector

เป็นการกำหนด CSS ให้กับ TAG ของ HTML/XHTML ที่ต้องการ โดยการเขียน Selector ในรูปแบบนี้ จะเป็นการเขียนโดยอ้างชื่อ TAG ต่างๆ เช่น `h1{...}` เป็นต้น ผลของการกำหนดในรูปแบบของ TAG Selector นี้ จะทำให้ TAG ทั้งหมดที่ตรงกับที่ประกาศไว้ใน CSS ของเว็บเพจ มีคุณสมบัติตาม CSS ที่เราได้กำหนดไว้โดยไม่ต้องมีการเรียกใช้ Selector ชนิดนี้กับ TAG ใดๆ ในเว็บเพจ

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>CSS Selectors</title>
  <style type="text/css">
    h1 {color: #0000CC; font: bold 18px Tahoma;}
  </style>
</head>
<body>
  <h1>Soulmate</h1>
  <h1>คู่แท้ฟ้าประทาน</h1>
</body>
</html>
```

Diagram annotations:

- An arrow points from the text box "h1 เป็น TAG ของ HTML/XHTML" to the `h1` selector in the CSS block.
- An arrow points from the text box "ไม่มีการเรียกใช้ CSS จาก TAG" to the `<h1>คู่แท้ฟ้าประทาน</h1>` tag in the body.



รูปที่ 2.39 แสดงผลกับ Web Browser แบบ TAG Selector

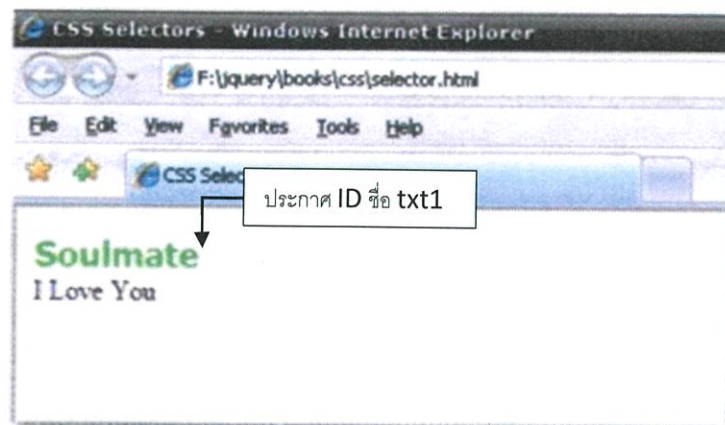
3. ID Selector

เป็นการกำหนดค่า CSS ให้กับ TAG ซึ่งมี Attribute ID ซึ่งมีชื่อตรงกับ ID Selector ของ CSS ที่ประกาศไว้ โดยการเขียน ID Selector จะมีรูปแบบการเขียนเครื่องหมายชาร์ป (#) ให้นำหน้าชื่อ ID เช่น #txt1 { ... } เป็นต้น ซึ่ง ID ในบนเว็บ

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>CSS Selectors</title>
<style type="text/css">
  #txt1 {color: #33CC00; font: bold 20px Tahoma;}
</style>
</head>
<body>
  <div id="txt1">Soulmate</div>
  <div class="txt2">I Love You</div>
</body>
</html>
```

๕๕

กาศไว้ ก็จะนำเอาค่าคุณสมบัติ CSS ไปใช้งาน ตัวอย่างเช่น



รูปที่ 2.40 แสดงผลกับ Web Browser แบบ ID Selector

ที่กล่าวมาทั้ง 3 รูปแบบของ Selector เป็นเพียงแค่พื้นฐานของการกำหนด CSS Selector เท่านั้น ซึ่งการเขียน Selector ยังมีรูปแบบการเขียนซับซ้อนมากกว่าข้างต้นที่กล่าวมา ทั้งนี้และทั้งนั้นขึ้นอยู่กับประเภทที่ต้องการใช้งาน

2.7.3 การเลือกใช้ Selectors

คงที่ได้อธิบายไปข้างต้นเรื่องของการเขียน Selectors หลายคนอาจจะสงสัยว่าเราควรมีหลักการเลือกใช้งาน Selectors ต่างๆ อย่างไร และใช้เมื่อใด ซึ่งหลักการใช้งาน Selectors ก็ขอให้ยึดตามหลักการใช้งาน ดังนี้

- หากมีการใช้งานคุณสมบัติของ CSS ที่ได้ประกาศขึ้นมามากกว่า 1 ครั้งภายในหน้าเว็บเพจ ให้เราเลือกใช้ Class Selector
- หากมีการใช้งานคุณสมบัติของ CSS ที่ได้ประกาศขึ้นมาเพียง 1 ครั้งภายในหน้าเว็บเพจ ให้เราเลือกใช้ ID Selector เนื่องจากวัตถุประสงค์ของ ID คือ Identity หรือการที่มีได้เพียงหนึ่งเดียวเป็นเอกลักษณ์เฉพาะในหน้าเว็บเพจนั้น
- หากต้องการเปลี่ยนคุณสมบัติของ TAG HTML/XHTML ทั้งหน้าเว็บเพจนั้น ให้เราเลือกใช้งานแบบ TAG Selector เช่น TAG `<h1> ... </h1>` ถ้าเราไม่เปลี่ยนคุณสมบัติของ TAG นี้ ในหน้าเว็บเพจของเราก็จะนำเอาค่าเริ่มต้นของ TAG มาใช้งานโดยทันที แต่คุณสมบัติเริ่มต้นของ TAG นี้ อาจจะไม่เหมาะสมกับเว็บเพจของเรา เราจึงต้องประกาศเปลี่ยนคุณสมบัติของ TAG `<h1> ... </h1>` กับ CSS เพื่อให้มีรูปแบบเข้ากับเว็บเพจของเรา เป็นต้น

2.8 กรณีศึกษา แอปพลิเคชัน “Stamp” บนบนโทรศัพท์ที่สมาร์ตโฟน iOS



รูปที่ 2.41 โลโก้แอปพลิเคชัน “Stamp” และหน้าจอการใช้งานต่างๆ

Stamp เป็นแอปพลิเคชันหนึ่งบนโทรศัพท์สมาร์ทโฟน ที่ให้บริการบนระบบปฏิบัติการ iOS โดยเมื่อเริ่มต้นเข้าใช้งาน ระบบจะทำการเก็บข้อมูลของลูกค้า เมื่อลูกค้าเป็นสมาชิกแล้วก็จะเห็นแต่ละร้านค้าที่แบ่งเป็นหมวดหมู่แต่ละประเภท ซึ่งแต่ละร้านก็จะมีข้อมูลของร้านค้า โปรโมชั่น รวมถึงบัตรสะสมแต้มที่สามารถบันทึกแต้มโดยใช้เครื่องมือแตะลงบนหน้าจอโทรศัพท์สมาร์ทโฟนได้เลย โดยที่ร้านค้าจะต้องสมัครเป็นสมาชิกกับ Stamp ก่อนแล้วจึงได้เครื่องมือสำหรับบันทึกแต้ม

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 แนวคิดการออกแบบระบบ

ระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมจะทำการเก็บรวบรวมข้อมูลของลูกค้าโดยใช้เบอร์โทรศัพท์มือถือ 10 หลัก เป็นส่วนที่ใช้ในการระบุข้อมูลลูกค้าซึ่งมีระบบจัดการข้อมูลสมาชิกที่เป็นศูนย์กลางและระบบการบริหารจัดการร้านค้า โดยเมื่อมีลูกค้าเข้ามาใหม่จะทำการตรวจสอบว่าลูกค้าเคยเป็นสมาชิกของแต่ละร้านค้าหรือไม่ ซึ่งถ้าลูกค้าเคยสมัครเป็นสมาชิกแล้ว จะได้รับสิทธิพิเศษตามเงื่อนไขของแต่ละร้านค้าผ่านศูนย์กลางบนระบบแทนการใช้กระดาษ เมื่อแต่ละร้านค้ามีการเพิ่มสมาชิกจะได้ข้อมูลสมาชิกภายในระบบเหมือนกัน ภายในระบบจะประกอบไปด้วยระบบลูกค้าและระบบร้านค้า ระบบลูกค้าเมื่อลูกค้าต้องการซื้อสินค้าจะแจ้งเบอร์โทรศัพท์มือถือของลูกค้าทุกครั้งกับร้านค้า ระบบจะทำการเก็บบันทึกข้อมูลทั้งหมดไว้ในรูปแบบบัตรสมาชิกและบัตรสะสมแต้มป้อนออนไลน์ โดยมีการบันทึกการแลกสินค้า บันทึกการให้แต้ม และบันทึกการใช้แต้มรูปแบบออนไลน์ และในส่วนของระบบร้านค้า สามารถสมัครสมาชิก กำหนดและแก้ไขโปรโมชันในการให้แต้มและการใช้แต้ม มีการแสดงรายการขายแบบแผนภูมิ การชำระเงิน และการใช้แต้มเป็นส่วนลด

ระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมจะนำเอาข้อมูลลูกค้าและร้านค้ามาจัดเก็บไว้ในฐานข้อมูลที่เป็นศูนย์กลาง โดยมีเงื่อนไขในการติดต่อกับฐานข้อมูลคือลูกค้าที่ทำการติดต่อกับระบบเพื่อดูข้อมูลจะต้องมีเบอร์โทรศัพท์มือถือ 10 หลัก และรหัสผ่านเพื่อทำการ Login เข้าสู่ระบบ และในส่วนของร้านค้าก็จะต้องมี E-mail และรหัสผ่านเช่นเดียวกันในการดำเนินการภายในระบบจัดการร้านค้า

3.2 การใช้งานของระบบ

3.3.1 ระบบลูกค้า

ผู้ใช้งานระบบจะต้องเป็นสมาชิกกับระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมก่อน ซึ่งการสมัครสมาชิกผู้ที่ทำการสมัครจะเป็นผู้กำหนดรหัสผ่านด้วยตัวเอง ก่อนใช้งานต้องทำการ Login ก่อนเสมอและทำการ Logout

เมื่อใช้งานเสร็จเรียบร้อยแล้วเพื่อความปลอดภัย หากสมาชิกสมัครรหัสผ่านจะมีระบบขอรหัสผ่านใหม่โดยต้องใส่ E-mail ที่ได้ใส่ไว้ในตอนสมัครสมาชิก จากนั้นระบบจะทำการส่งรหัสผ่านใหม่ไปยัง E-mail ของตนเอง

ผู้ใช้งานในส่วนของระบบลูกค้า เมื่อลูกค้าต้องการซื้อสินค้าจะต้องแจ้งเบอร์โทรศัพท์มือถือกับร้านค้าเพื่อรับสิทธิพิเศษตามเงื่อนไขของแต่ละร้านค้า ระบบจะเก็บบันทึกข้อมูลแสดมป์สะสมทั้งหมดไว้ให้ในรูปแบบบัตรสมาชิกและบัตรสะสมแสดมป์ออนไลน์ และลูกค้ายังสามารถดูบัตรสะสมแสดมป์ ดูรายละเอียดร้านค้า และรายการโปรโมชั่นของแต่ละร้านค้าได้ ตรวจสอบโปรโมชั่นที่สามารถใช้แสดมป์ได้

3.3.2 ระบบร้านค้า

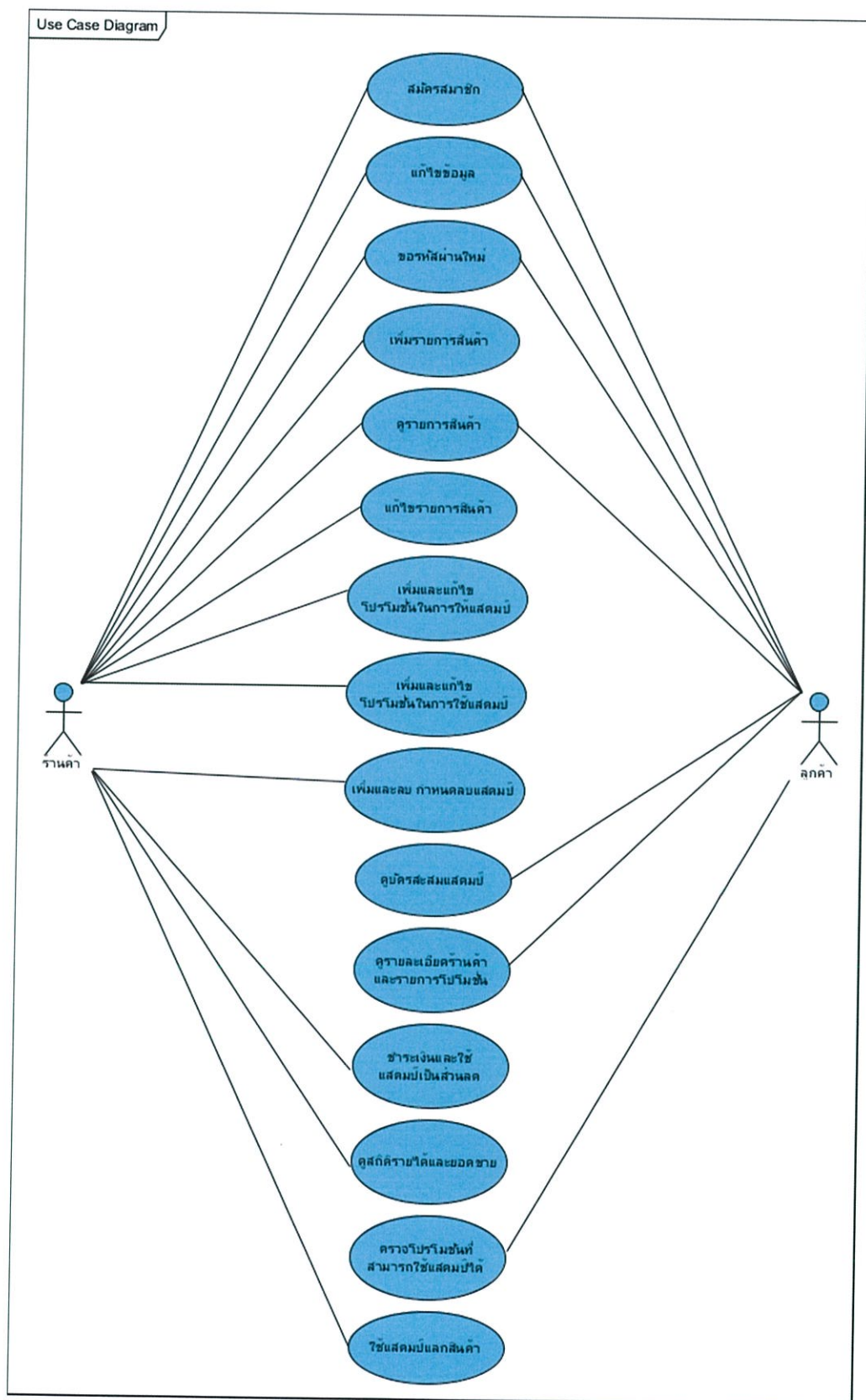
ผู้ใช้งานระบบจะต้องเป็นสมาชิกกับระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมก่อน ซึ่งการสมัครสมาชิกผู้ที่ทำการสมัครจะเป็นผู้กำหนดรหัสผ่านด้วยตัวเอง ก่อนเข้าใช้งานต้องทำการ Login ก่อนเสมอและทำการ Logout เมื่อใช้งานเสร็จเรียบร้อยแล้วเพื่อความปลอดภัย หากสมาชิกสมัครรหัสผ่านใหม่โดยต้องใส่ E-mail ที่ได้ใส่ไว้ในตอนสมัครสมาชิก จากนั้นระบบจะทำการส่งรหัสผ่านใหม่ไปยัง E-mail ของตนเอง

เมื่อลูกค้าต้องการซื้อสินค้า ทางร้านค้าจะพิมพ์เบอร์โทรศัพท์ของลูกค้า (ในกรณีที่ เป็นลูกค้าทั่วไปไม่ได้เป็นสมาชิกกับระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม ร้านค้าจะต้องใช้เบอร์โทรศัพท์ คือ 0000000000) รหัสสินค้า และจำนวนสินค้าที่ซื้อ หากลูกค้าต้องการใช้แสดมป์เป็นส่วนลด ภายในระบบชำระเงินของร้านค้าก็มีให้พร้อมใช้งาน และทางร้านค้ายังมีระบบการใช้แสดมป์แลกสินค้า

เมื่อมีการชำระเงินของลูกค้าแล้ว ภายในระบบจะมีรายงานสถิติแบบแผนภูมิ เพื่อให้ร้านค้าสามารถวางแผนธุรกิจได้ ซึ่งสถิติที่จัดเก็บได้แก่ สถิติรายได้ สถิติยอดขาย

ร้านค้าสามารถแก้ไขข้อมูลร้านค้าภายในระบบ ดูและแก้ไขรายการสินค้า กำหนด และแก้ไขโปรโมชั่นในการใช้แสดมป์ กำหนดและแก้ไขโปรโมชั่นในการใช้แสดมป์ได้

3.3 Use Case Diagram



รูปที่ 3.1 Use Case Diagram

Use Case Diagram แสดงถึงการใช้งานของผู้ที่เข้ามาใช้งานในระบบ โดยจะประกอบไปด้วย

- ร้านค้าจะสามารถทำการสมัครสมาชิก แก้ไขข้อมูล และมีหน้าที่ในการกำหนดและแก้ไขโปรโมชั่นในการให้แถมปี กำหนดและแก้ไขโปรโมชั่นในการให้แถมปี จัดการเรื่องการชำระเงินของลูกค้า การให้แถมปีเป็นส่วนลดหรือให้แถมปีแลกเปลี่ยนค่า เข้าคูสติรายได้และยอดขาย
- ลูกค้าจะสามารถทำการสมัครสมาชิก แก้ไขข้อมูล เข้าดูรายการสินค้าในแต่ละร้าน เข้าดูรายละเอียดร้านค้าและโปรโมชั่น และเข้าดูจำนวนบัตรสะสมแถมปี ตรวจสอบโปรโมชั่นที่สามารถให้แถมปีได้

3.3.1 คำอธิบาย Use Case

1. Use Case : สมัครสมาชิก

Actor : ร้านค้า ลูกค้า

Overview : ผู้ใช้บริการจะทำการสมัครสมาชิกโดยการกรอกข้อมูลในหน้า “สมัครสมาชิก”

2. Use Case : แก้ไขข้อมูล

Actor : ร้านค้า ลูกค้า

Overview : ผู้ใช้บริการจะทำการแก้ไขข้อมูลใหม่ๆ โดยแก้ไขข้อมูลต่างๆ ในหน้า “แก้ไขข้อมูล” หลังจากสมัครสมาชิก และทำการ Login แล้ว

3. Use Case : ขอรหัสผ่านใหม่

Actor : ร้านค้า ลูกค้า

Overview : หากผู้บริการลืมรหัสผ่าน จะทำการคลิก “ลืมรหัสผ่าน” ในหน้า “Login” เพื่อขอรหัสผ่านใหม่

4. Use Case : เพิ่มรายการสินค้า

Actor : ร้านค้า

- Overview : ร้านค้าสามารถเพิ่มรายการสินค้าในร้านของตัวเอง โดยเพิ่มข้อมูล รายละเอียดต่างๆเกี่ยวกับสินค้า ได้แก่ รหัสสินค้า ราคาสินค้า ชื่อสินค้า รูปสินค้าในหน้า “เพิ่มสินค้า”
5. Use Case : ดูรายการสินค้า
 Actor : ร้านค้า ลูกค้า
 Overview : ร้านค้าสามารถดูรายการสินค้าต่างๆในร้านของตัวเอง ส่วนลูกค้าสามารถดูรายการสินค้าต่างๆในแต่ละร้านในหน้า “สินค้า”
6. Use Case : แก้ไขรายการสินค้า
 Actor : ร้านค้า
 Overview : ร้านค้าสามารถทำการแก้ไขข้อมูลรายละเอียดของแต่ละสินค้าในหน้า “รายการสินค้าที่ใช้งาน” หรือในหน้า “รายการสินค้าที่ไม่ใช้งาน”
7. Use Case : เพิ่มและแก้ไขโปรโมชันในการให้แสดมปี
 Actor : ร้านค้า
 Overview : ร้านค้าจะทำการกำหนดและสามารถแก้ไขโปรโมชันในการให้แสดมปี ในหน้า “โปรโมชันให้แสดมปี”
8. Use Case : เพิ่มและแก้ไขโปรโมชันในการใช้แสดมปี
 Actor : ร้านค้า
 Overview : ร้านค้าจะทำการกำหนดและสามารถแก้ไขโปรโมชันในการใช้แสดมปีใน หน้า “โปรโมชันใช้แสดมปี”
9. Use Case : เพิ่มและลบกำหนดลบแสดมปี
 Actor : ร้านค้า
 Overview : ร้านค้าจะทำการกำหนดและสามารถลบกำหนดลบแสดมปีในหน้า “กำหนดลบแสดมปี”

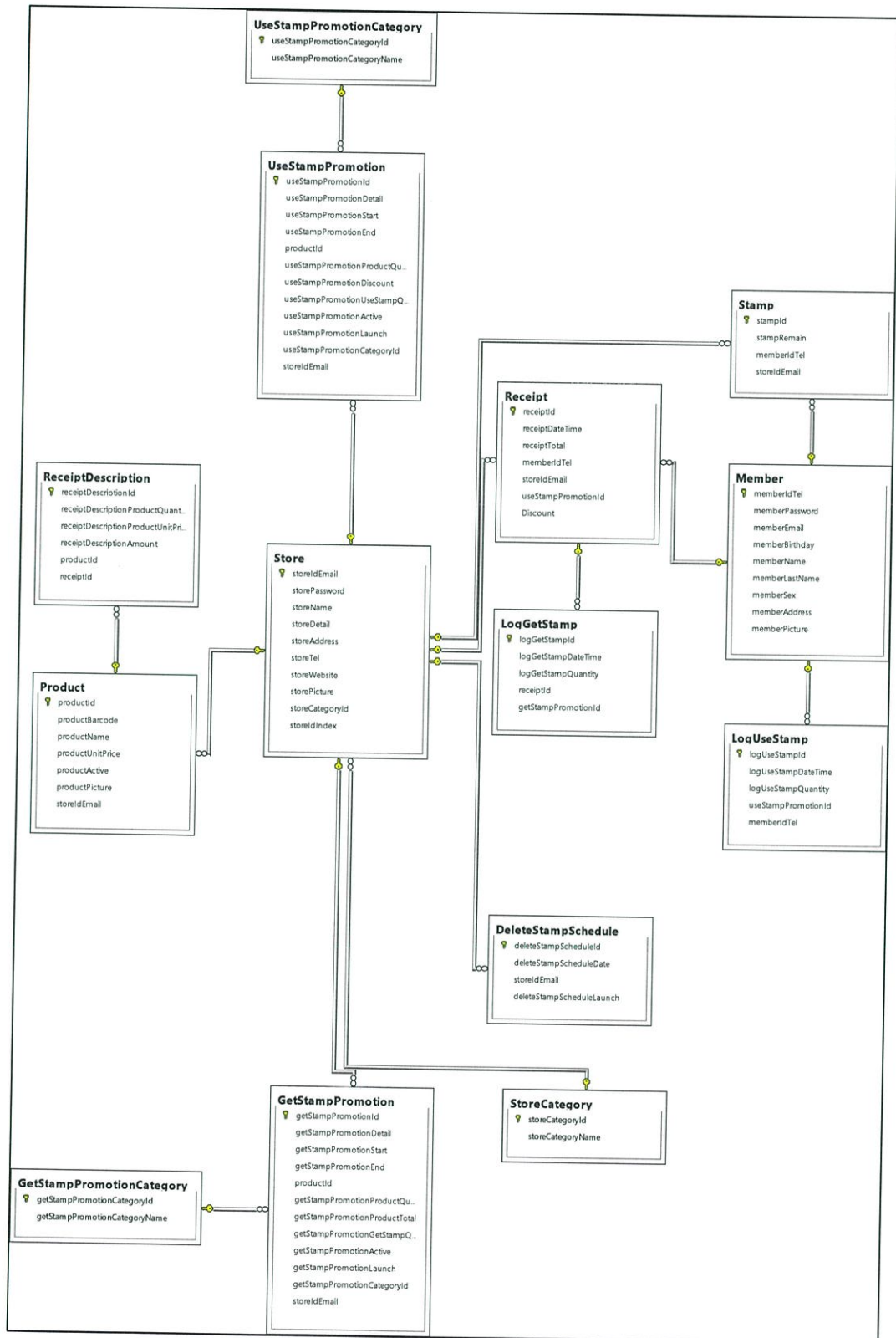
10. Use Case : ดูบัตรสะสมแต้มปี
 Actor : ลูกค้า
 Overview : ลูกค้าสามารถดูบัตรสะสมแต้มปี หลังจากการซื้อสินค้าในแต่ละร้านว่าลูกค้ามีแต้มปีคงเหลือของแต่ละร้านมีอยู่เป็นจำนวนเท่าไรในหน้า “บัตรสะสมแต้มปี”
11. Use Case : ดูรายละเอียดร้านค้าและรายการโปรโมชั่น
 Actor : ลูกค้า
 Overview : ลูกค้าสามารถเข้าดูรายละเอียดร้านค้าและรายการโปรโมชั่นแต่ละประเภทในหน้า “ร้านค้าทั้งหมด”
12. Use Case : ชำระเงินและการใช้แต้มปีส่วนลด
 Actor : ร้านค้า
 Overview : ร้านค้าจะทำการชำระเงินโดยใส่เบอร์โทรศัพท์ของลูกค้า รหัสสินค้า และจำนวนสินค้าที่ซื้อ หากลูกค้าต้องการใช้ส่วนลดจะสามารถใช้แต้มปีเป็นส่วนลดได้ในหน้า “ระบบชำระเงิน”
13. Use Case : ดูสถิติรายได้และยอดขาย
 Actor : ร้านค้า
 Overview : ร้านค้าสามารถดูสถิติรายได้และยอดขายแบบแผนภูมิ ได้แก่ สถิติรายได้แบบแผนภูมิตามรายเดือน และรายปี และสามารถดูสถิติยอดขายแบบแผนภูมิ โดยเลือกแสดงรายวัน รายเดือนหรือรายปี ในหน้า “สถิติการขาย”
14. Use Case : ตรวจสอบโปรโมชั่นที่สามารถใช้แต้มปีได้
 Actor : ลูกค้า
 Overview : ลูกค้าสามารถตรวจสอบหาโปรโมชั่นที่ลูกค้าสามารถใช้แต้มปีได้ กล่าวคือโปรโมชั่นที่ลูกค้ามีแต้มปีสะสมคงเหลือมากพอที่จะแลกสินค้าหรือส่วนลด สามารถตรวจได้ในหน้า “ตรวจสอบโปรโมชั่น”

15. Use Case : ใช้แต้มปีแลกสินค้า

Actor : ร้านค้า

Overview : เมื่อลูกค้ามีแต้มปีสะสมครบจำนวนตามโปรโมชั่นที่ร้านค้ากำหนด ร้านค้าจะทำการใส่เบอร์โทรศัพท์ของลูกค้าลงไปแล้วคลิกตรวจสอบว่ามีรายชื่อลูกค้าที่ได้เป็นสมาชิกของร้านค้าของตัวเองหรือไม่ หากมีลูกค้าที่เป็นสมาชิกของร้านตัวเอง ให้เลือกสินค้าที่สามารถแลกได้ในหน้า “แลกสินค้า”

3.4 Entity Relationship Diagram



รูปที่ 3.2 Entity Relationship Diagram

3.4.1 ตารางของระบบฐานข้อมูล

ตารางที่ 3.1 ตารางที่ใช้ในการพัฒนาระบบจัดการระบบการจัดการสมาชิกบนเว็บ

ลำดับ	ชื่อตาราง	ความหมายของตาราง
1.	Store	ข้อมูลร้านค้า
2.	ReceiptDescription	ข้อมูลรายละเอียดของใบเสร็จ
3.	Product	ข้อมูลสินค้า
4.	UseStampPromotionCategory	ประเภทโปรโมชั่นการใช้แสตมป์
5.	UseStampPromotion	ข้อมูลโปรโมชั่นการใช้แสตมป์
6.	GetStampPromotionCategory	ประเภทโปรโมชั่นการใช้แสตมป์
7.	GetStampPromotion	ข้อมูลโปรโมชั่นการใช้แสตมป์
8.	Receipt	ข้อมูลใบเสร็จ
9.	LogGetStamp	ข้อมูลบันทึกการใช้แสตมป์
10.	DeleteStampSchedule	ข้อมูลกำหนดการลบแสตมป์
11.	StoreCategory	ประเภทร้านค้า
12.	Stamp	ข้อมูลแสตมป์สะสม
13.	Member	ข้อมูลสมาชิก
14.	LogUseStamp	ข้อมูลบันทึกการใช้แสตมป์

ตารางที่ 3.2 แสดงตาราง UseStampPromotionCategory มี useStampPromotionId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	useStampPromotionId	int	รหัสประเภท โปรโมชั่นใช้แสตมป์	Primary Key
2.	useStampPromotion CategoryName	nvarchar(50)	ชื่อประเภทโปรโมชั่น ใช้แสตมป์	

ตารางที่ 3.3 แสดงตาราง Store มี storeIdEmail เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	storeIdEmail	nvarchar(50)	E-mail ร้านค้า	Primary Key
2.	storePassword	nvarchar(50)	รหัสผ่าน	
3.	storeName	nvarchar(50)	ชื่อร้านค้า	
4.	storeDetail	nvarchar(MAX)	รายละเอียดของร้านค้า	
5.	storeAddress	nvarchar(MAX)	ที่อยู่ของร้านค้า	
6.	storeTel	nvarchar(50)	เบอร์โทรศัพท์ร้านค้า	
7.	storeWebsite	nvarchar(50)	เว็บไซต์ของร้านค้า	
8.	storePicture	nvarchar(MAX)	ภาพร้านค้า	
9.	storeCategoryId	int	รหัสประเภทร้านค้า	Foreign Key
10.	storeIdIndex	int	รหัสดัชนีร้านค้า	

ตารางที่ 3.4 แสดงตาราง ReceiptDescription มี receiptDescriptionId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	receiptDescriptionId	int	รหัสข้อมูลรายละเอียดใบเสร็จ	Primary Key
2.	receiptDescription ProductQuantity	int	จำนวนสินค้าที่ซื้อ	
3.	receiptDescription ProductUnitPrice	money	ราคาต่อหน่วยของสินค้า	
4.	receiptDescription Amount	money	ราคารวมที่ซื้อสินค้า	
5.	ProductId	int	รหัสสินค้า	Foreign Key
6.	receiptId	int	รหัสใบเสร็จ	Foreign Key

ตารางที่ 3.5 แสดงตาราง ReceiptDescription มี ProductId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	ProductId	int	รหัสสินค้า	Primary Key
2.	productBarcode	nvarchar(50)	รหัสสินค้าหรือบาร์โค้ด	
3.	productName	nvarchar(50)	ชื่อสินค้า	
4.	productUnitPrice	money	ราคาสินค้าต่อหน่วย	
5.	productActive	bit	สถานการณ์ใช้งานสินค้า	
6.	productPicture	nvarchar(MAX)	รูปภาพสินค้า	
7.	storeIdEmail	nvarchar(50)	E-mail ร้านค้า	Foreign Key

ตารางที่ 3.6 แสดงตาราง GetStampPromotionCategory มี getStampPromotionCategoryId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	getStampPromotion CategoryId	int	รหัสประเภทโปรโมชั่น ให้แสดมปี	Primary Key
2.	getStampPromotion CategoryName	nvarchar(50)	ชื่อประเภทโปรโมชั่น ให้แสดมปี	

ตารางที่ 3.7 แสดงตาราง Stamp มี StampId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	StampId	int	รหัสแสดมปีสะสม	Primary Key
2.	stampRemain	int	แสดมปีสะสมคงเหลือ	
3.	memberIdTel	nvarchar(50)	เบอร์โทรศัพท์ลูกค้า	Foreign Key
4.	storedEmail	nvarchar(50)	E-mail ร้านค้า	Foreign Key

ตารางที่ 3.8 แสดงตาราง UseStampPromotion มี useStampPromotionId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	useStampPromotion Id	int	รหัสข้อมูลโปรโมชั่น ใช้แสดมป์	Primary Key
2.	useStampPromotion Detail	nvarchar(MAX)	รายละเอียดโปรโมชั่น ใช้แสดมป์	
3.	useStampPromotion Start	date	วันเริ่มต้นโปรโมชั่น	
4.	useStampPromotion End	date	วันสิ้นสุดโปรโมชั่น	
5.	productId	int	รหัสสินค้า	Foreign Key
6.	useStampPromotion ProductQuantity	int	จำนวนสินค้าที่ซื้อ	
7.	useStampPromotion Discount	money	ใช้แสดมป์แลก ส่วนลด	
8.	useStampPromotion UseStampQuantity	int	จำนวนแสดมป์ที่ใช้	
9.	useStampPromotion Active	bit	สถานะโปรโมชั่น	
10.	useStampPromotion Launch	bit	สถานะเปิดใช้ โปรโมชั่น	
11.	useStampPromotion CategoryId	int	รหัสประเภท โปรโมชั่นใช้แสดมป์	Foreign Key
12.	StoreIdEmail	nvarchar(50)	E-mail ร้านค้า	Foreign Key

ตารางที่ 3.9 แสดงตาราง GetStampPromotion มี getStampPromotionId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	getStampPromotionId	int	รหัสข้อมูลโปรโมชั่นให้แสดมปี	Primary Key
2.	getStampPromotionDetail	nvarchar(MAX)	รายละเอียดโปรโมชั่นให้แสดมปี	
3.	getStampPromotionStart	date	วันเริ่มต้นโปรโมชั่น	
4.	getStampPromotionEnd	date	วันสิ้นสุดโปรโมชั่น	
5.	ProductId	int	รหัสสินค้า	Foreign Key
6.	getStampPromotionProductQuantity	int	จำนวนสินค้าที่ซื้อ	
7.	getStampPromotionTotal	money	ราคารวมสุทธิที่ซื้อ	
8.	getStampPromotionGetStampQuantity	int	จำนวนแสดมปีที่ให้	
9.	getStampPromotionActive	bit	สถานะโปรโมชั่น	
10.	getStampPromotionLaunch	bit	สถานะเปิดใช้โปรโมชั่น	
11.	getStampPromotionCategoryld	int	รหัสประเภทโปรโมชั่นให้แสดมปี	Foreign Key
12.	storeldEmail	nvarchar(50)	E-mail ร้านค้า	Foreign Key

ตารางที่ 3.10 แสดงตาราง Receipt มี receiptId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	receiptId	int	รหัสใบเสร็จ	Primary Key
2.	receiptDateTime	datetime	วันที่และเวลาที่ออกใบเสร็จ	
3.	receiptTotal	money	ราคารวมสุทธิของใบเสร็จ	
4.	memberIdTel	nvarchar(50)	เบอร์โทรศัพท์ลูกค้า	Foreign Key
5.	storeIdEmail	nvarchar(50)	E-mail ร้านค้า	Foreign Key
6.	useStampPromotionId	int	รหัสข้อมูลโปรโมชั่นให้แต้ม	Foreign Key
7.	Discount	money	ส่วนลด	

ตารางที่ 3.11 แสดงตาราง LogGetStamp มี logGetStampId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	logGetStampId	int	รหัสข้อมูลบันทึกการให้แต้ม	Primary Key
2.	logGetStampDateTime	datetime	วันที่และเวลาการบันทึก	
3.	logGetStampQuantity	int	จำนวนแต้มที่ให้	
4.	receiptId	int	รหัสใบเสร็จ	Foreign Key
5.	getStampPromotionId	int	รหัสข้อมูลโปรโมชั่นให้แต้ม	Foreign Key

ตารางที่ 3.12 แสดงตาราง DeleteStampSchedule มี deleteStampScheduleId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	deleteStamp ScheduleId	int	รหัสกำหนดการลบ แสดมปี	Primary Key
2.	deleteStamp ScheduleDate	date	วันที่ลบแสดมปี	
3.	storeIdEmail	nvarchar(50)	E-mail ร้านค้า	Foreign Key
4.	deleteStamp ScheduleLaunch	bit	สถานะเปิดใช้ กำหนดการลบแสดมปี	

ตารางที่ 3.13 แสดงตาราง StoreCategory มี storeCategoryId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	storeCategoryId	int	รหัสประเภทร้านค้า	Primary Key
2.	storeCategoryName	nvarchar(50)	ชื่อประเภทร้านค้า	

ตารางที่ 3.14 แสดงตาราง Member มี memberTel เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	memberTel	nvarchar(50)	เบอร์โทรศัพท์ลูกค้า	Primary Key
2.	memberPassword	nvarchar(50)	รหัสผ่าน	
3.	memberEmail	nvarchar(50)	E-mail ลูกค้า	
4.	memberBirthday	date	วันเกิดลูกค้า	
5.	memberName	nvarchar(50)	ชื่อลูกค้า	
6.	memberLastName	nvarchar(50)	นามสกุลลูกค้า	
7.	memberSex	nvarchar(10)	เพศลูกค้า	
8.	memberAddress	nvarchar(MAX)	ที่อยู่ลูกค้า	
9.	memberPicture	nvarchar(MAX)	ภาพลูกค้า	

ตารางที่ 3.15 แสดงตาราง LogUseStamp มี logUseStampId เป็นคีย์หลัก

ลำดับ	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	ประเภทคีย์
1.	logUseStampId	int	รหัสข้อมูลบันทึก การใช้แสดมป์	Primary Key
2.	logUseStampDate Time	datetime	วันที่และเวลา การบันทึก	
3.	logUseStamp Quantity	int	จำนวนแสดมป์ที่ใช้	
4.	useStampPromotion Id	int	รหัสข้อมูลโปรโมชั่น ใช้แสดมป์	Foreign Key
5.	memberIdTel	nvarchar(50)	เบอร์โทรศัพท์ลูกค้า	Foreign Key

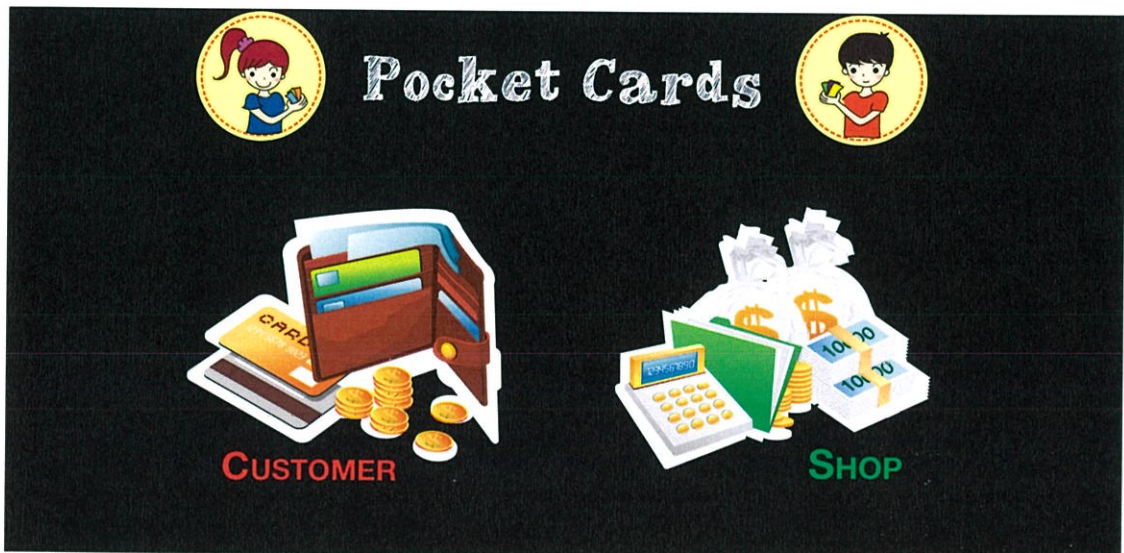
3.5 Software ที่เกี่ยวข้อง

- Visual Studio 2012
- Microsoft SQL Server 2012

บทที่ 4

การใช้งานโปรแกรม

4.1 ส่วนประกอบต่างๆจากการพัฒนาระบบ

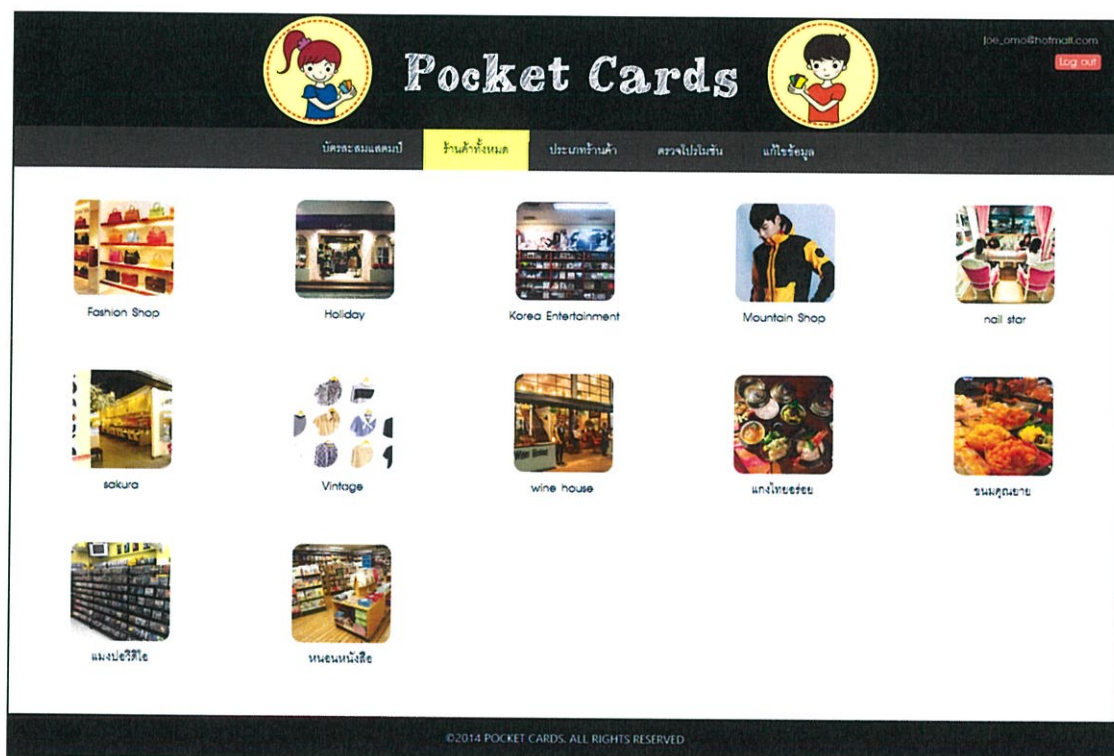


รูปที่ 4.1 หน้าหลักระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม

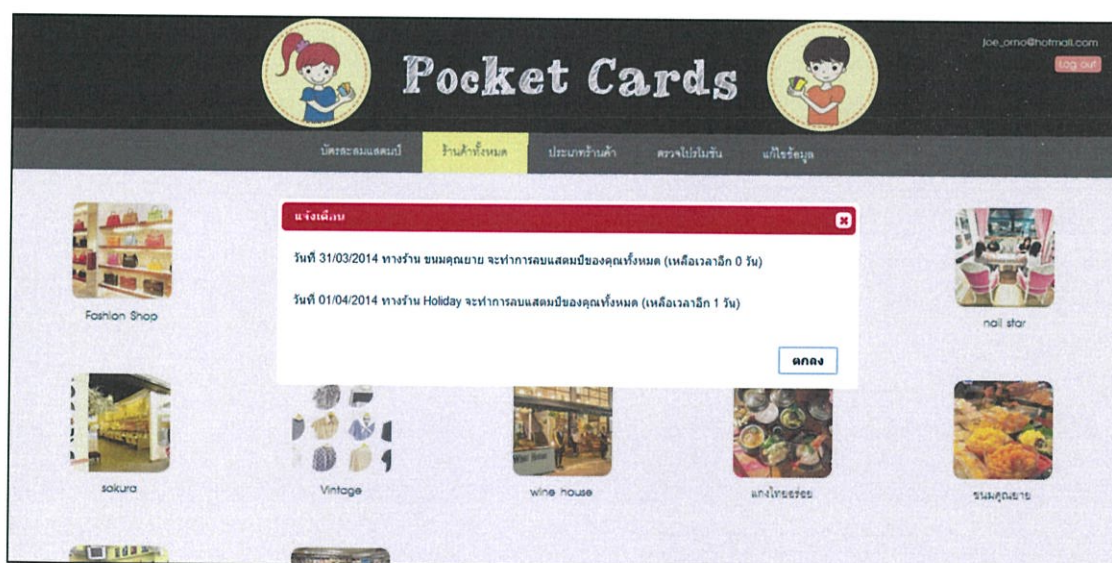
ส่วนประกอบของระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม เป็นผลที่ได้จากการพัฒนาระบบด้วย ASP.NET(MVC) jQuery JavaScript และ CSS ซึ่งประกอบไปด้วยหน้าหลักของระบบ และส่วนหลักคือ ส่วนของลูกค้าและส่วนของร้านค้า

4.1.1 โปรแกรมส่วนของลูกค้า

การเข้าใช้งานภายในระบบลูกค้าจะต้องเป็นสมาชิกของระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมก่อน ซึ่งจะทำการสมัครสมาชิกได้จากหน้าจอหลักของผู้ให้บริการ หน้าจอหลักในส่วนของลูกค้าจะประกอบไปด้วย 5 รายการคือ บัตรสะสมแต้มปี ร้านค้าทั้งหมด ประเภทร้านค้า ตรวจสอบโปรโมชั่น และแก้ไขข้อมูล



รูปที่ 4.2 หน้าหลักแสดงการเข้าใช้งานของส่วนลูกค้า



รูปที่ 4.3 หน้าจอแสดงแสดงการแจ้งเตือนอัตโนมัติเมื่อใกล้จะถึงวันลบสต็อก

4.1.1.1 การสมัครสมาชิก

จากหน้าจอหลักผู้ใช้บริการเข้าสู่รายการสมัครสมาชิก ทำการกรอกข้อมูลที่เป็นต่อระบบซึ่งประกอบไปด้วยเบอร์โทรศัพท์ รหัสผ่าน ยืนยันรหัสผ่าน E-mail วันเกิด ชื่อ นามสกุล เพศ ที่อยู่ และรูปประจำตัว โดยตัวอักษรใหญ่ และตัวอักษรเล็กจะให้ผลต่างกัน หากทำการสมัครเสร็จถูกต้อง ระบบจะทำการส่งข้อมูลการสมัครให้ทาง E-mail แต่หากเกิดข้อผิดพลาด ระบบจะแจ้งเตือนข้อผิดพลาดขึ้นมาตรงท้ายของช่อง ดังรูปที่ 4.4

ลูกด่าสมัครสมาชิก

เบอร์โทรศัพท์ * * ข้อมูลที่ต้องใส่

รหัสผ่าน * * ข้อมูลที่ต้องใส่

ยืนยันรหัสผ่าน * * ข้อมูลที่ต้องใส่

E-mail * * ข้อมูลที่ต้องใส่

วันเกิด

ชื่อ

นามสกุล

เพศ

ที่อยู่

รูปประจำตัว

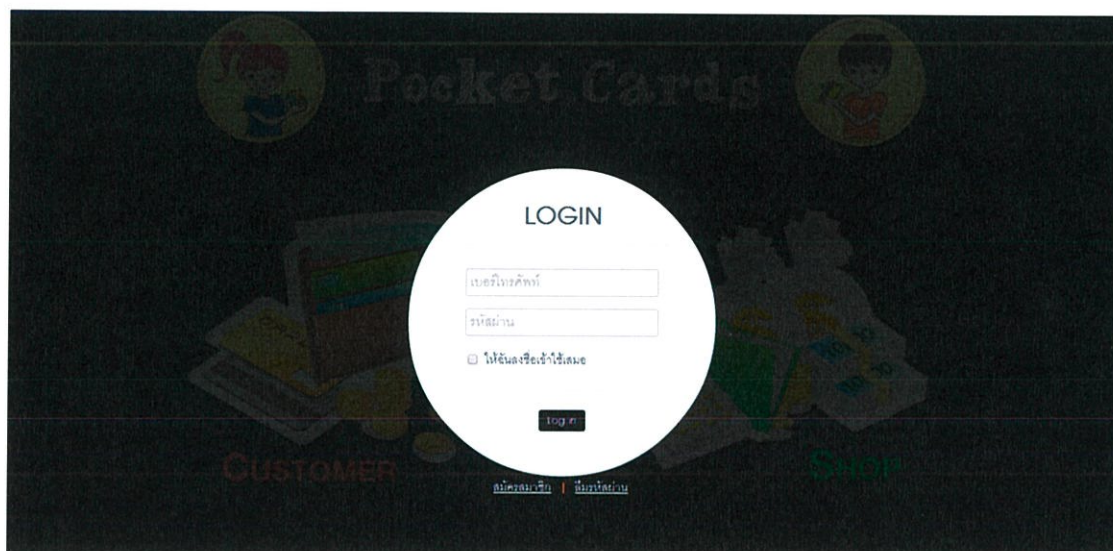
* ข้อมูลที่ต้องใส่

©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.4 หน้าจอแสดงการสมัครสมาชิกของลูกด่า

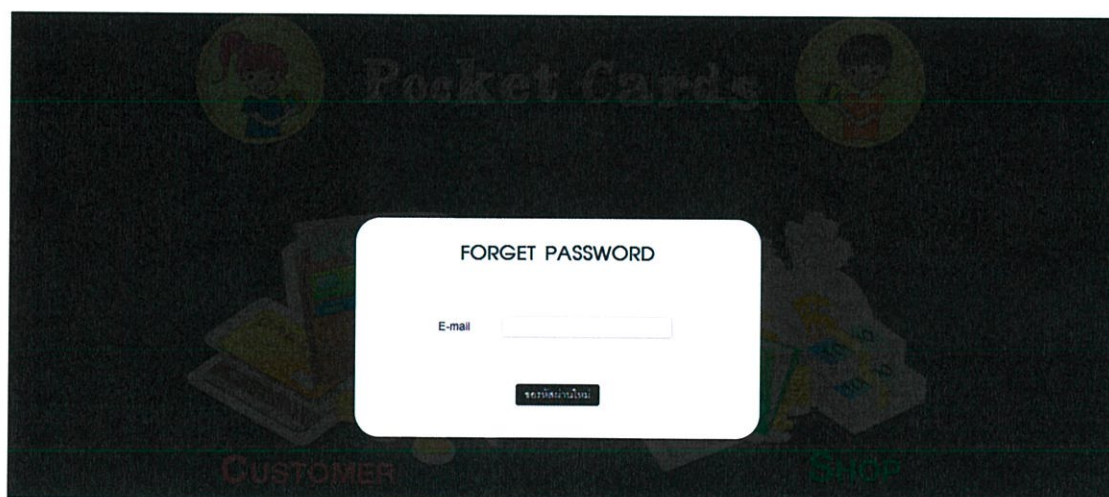
4.1.1.2 การใช้บริการ (Login)

กรอกเบอร์โทรศัพท์ และรหัสผ่านให้ถูกต้องในหน้า “Login” ของลูกค้า หลังจากนั้นกดปุ่ม Login ถ้าหากต้องให้ฉันลงชื่อเข้าใช้เสมอ ก็ทำการเลือกถูก เพื่อให้สามารถเข้าระบบอีกครั้งได้โดยไม่ต้อง login อีกครั้ง ดังรูปที่ 4.5



รูปที่ 4.5 หน้าจอแสดงการ login เข้าสู่ระบบของลูกค้า

หากลืมรหัสผ่าน ให้ทำการกดปุ่ม “ลืมรหัสผ่าน” เพื่อขอรหัสผ่านใหม่ โดยกรอก E-mail เพื่อระบบจะทำการส่งรหัสผ่านใหม่ให้ทาง E-mail ดังรูปที่ 4.6



รูปที่ 4.6 หน้าจอแสดงการขอรหัสผ่านใหม่ โดยกรอก E-mail ของลูกค้า

4.1.1.3 บัตรสะสมแต้มปี

หน้า “บัตรสะสมแต้มปี” มีการบอกจำนวนแต้มปีคงเหลือที่ลูกค้ามีจากการใช้บริการในแต่ละร้านค้า และวันหมดเขตให้แต้มปี ดังรูปที่ 4.7

The screenshot shows the 'Pocket Cards' website interface. At the top, there are two cartoon characters in circular frames and the text 'Pocket Cards'. Below this is a navigation bar with the following items: 'บัตรสะสมแต้มปี' (highlighted), 'ร้านค้าทั้งหมด', 'ประเภทร้านค้า', 'รางวัลโปรโมชั่น', and 'แก้ไขข้อมูล'. The main content area displays a list of stores under the heading 'ดูร้านค้า'. Each store entry includes a thumbnail image, the store name, a 'Read More' button, the number of points, the currency (บาท), and the expiration date.

ดูร้านค้า	ชื่อร้านค้า	จำนวนแต้มปีคงเหลือ	วันหมดเขตให้แต้มปี
	นมคุณยาย Read More	50 บาท	31/03/2014
	Holiday Read More	5 บาท	01/04/2014
	Vintage Read More	10 บาท	ไม่มีวันหมดเขต
	Fashion Shop Read More	5 บาท	ไม่มีวันหมดเขต
	แมงป่องวิดีโอ Read More	0 บาท	ไม่มีวันหมดเขต

At the bottom of the page, there is a copyright notice: ©2014 POCKET CARDS. ALL RIGHTS RESERVED.

รูปที่ 4.7 หน้าจอแสดงบัตรสะสมแต้มปี

เมื่อคลิก Read more ที่อยู่ใต้ชื่อร้านค้า จะมีข้อมูลรายละเอียดของแต่ละร้านค้า ได้แก่ ชื่อร้านค้า ประเภทร้านค้า รายละเอียดร้านค้า ที่อยู่ของร้านค้า เบอร์โทรศัพท์ของร้านค้า Website ร้านค้า โปรโมชั่นให้แต้มปี โปรโมชั่นใช้แต้มปี และกำหนดคลับสะสมแต้มปี ดังรูปที่ 4.8

บัตรสะสมแต้ม | **ร้านค้าทั้งหมด** | ประเภทร้านค้า | ครัวไปรษณีย์ | แก้ไขข้อมูล

ชื่อร้านค้า : ขนมคุณยาย
ประเภทร้านค้า : Entertainment
รายละเอียดร้านค้า : เป็นร้านขายขนมไทยสูตรดั้งเดิมใช้วัตถุดิบสดใหม่สะอาด
ที่อยู่ : ตลาดน้ำนันทาร ลาดกระบัง ซอย A ติดร้านขายเสื้อผ้าแฟชั่น
เบอร์โทรศัพท์ : 08978346939
Website :
รายการสินค้า : [Show More](#)

ไปรษณีย์ให้แต้ม
 ซื้อครบ 200.00 บาท ได้รับแต้มจำนวน 1 ดวง หมดเขตวันที่ 31/01/2014
 ซื้อสินค้า ขนมสด จำนวน 5 หน่วย ได้รับแต้มจำนวน 2 ดวง หมดเขตวันที่ 31/01/2014

ไปรษณีย์ใช้แต้ม
 แลตมป์จำนวน 10 ดวง แลกส่วนลด 50.00 บาท หมดเขตวันที่ 31/01/2014
 แลตมป์จำนวน 5 ดวง แลกสินค้า ขนมสด จำนวน 2 หน่วย หมดเขตวันที่ 31/03/2014

กำหนดครบแต้ม
 31/03/2014

©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.8 หน้าจอแสดงข้อมูลรายละเอียดของร้านค้า (1)

บัตรสะสมแต้ม | **ร้านค้าทั้งหมด** | ประเภทร้านค้า | ครัวไปรษณีย์ | แก้ไขข้อมูล

ชื่อร้านค้า : Vintage
ประเภทร้านค้า : Entertainment
รายละเอียดร้านค้า : เสื้อผ้าสไตล์วินเทจ
ที่อยู่ : Siam Square 2 Alley Pathum Wan, Bangkok
เบอร์โทรศัพท์ : 02 251 7448
Website :
รายการสินค้า : [Show More](#)

ไปรษณีย์ให้แต้ม
 ไม่มีไปรษณีย์ให้แต้ม

ไปรษณีย์ใช้แต้ม
 ไม่มีไปรษณีย์ใช้แต้ม

กำหนดครบแต้ม
 ไม่มีกำหนดครบแต้ม

©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.9 หน้าจอแสดงข้อมูลรายละเอียดของร้านค้า (2)

เมื่อกดคลิก Show more ที่อยู่ในช่องรายการสินค้า จะแสดงข้อมูลสินค้าที่ขายของร้านค้านั้น ประกอบด้วยชื่อสินค้าและราคาขาย ดังรูปที่ 4.10

Screenshot of the Pocket Cards website showing a grid of clothing items for sale. The header includes the brand name "Pocket Cards" and navigation links. The main content area is titled "สินค้าของร้าน VINTAGE" and displays 16 items in a 4x4 grid. Each item has a "SALE" tag, a photo, a name, and a price.

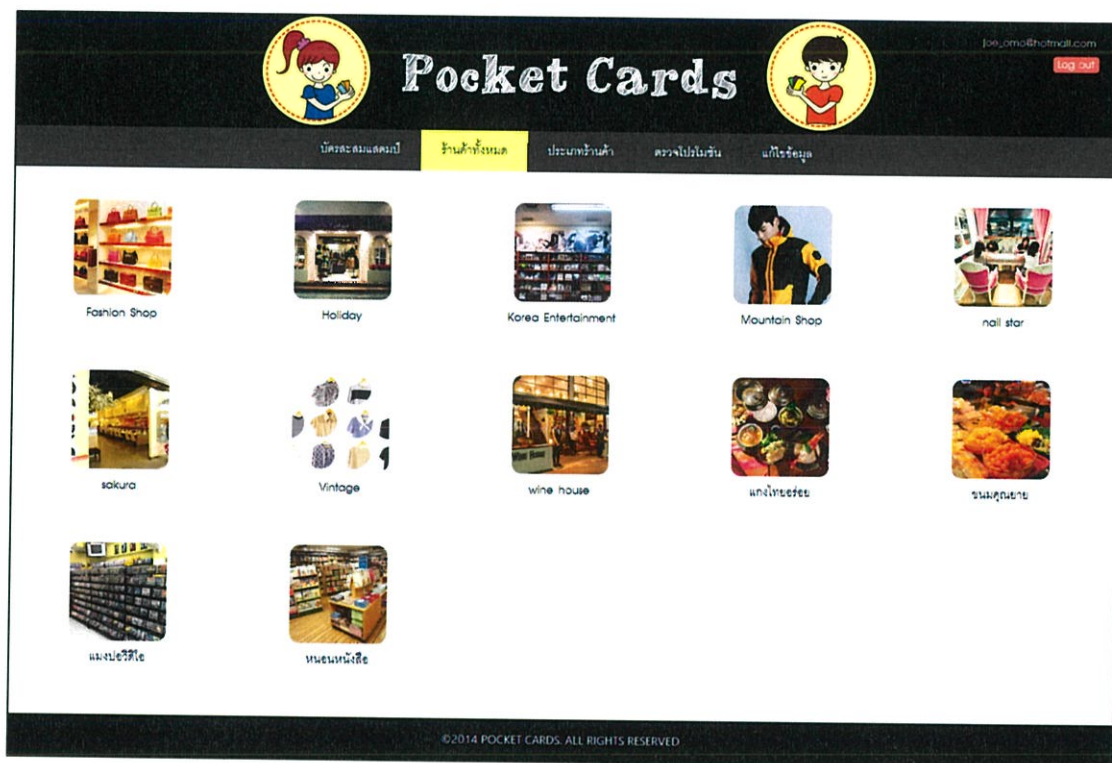
Item Name	Price
เดรสแขนงุด ลายดอกไม้	฿350.00
เดรสแขนงุด สีโรส	฿350.00
เดรสแขนงุดฟ้า-เหลือง	฿300.00
เดรสแขนงุดดอกไม้ ลายขวาง	฿350.00
เดรสผ่าขาด ลายจุด	฿300.00
เดรสยาว ลายดอกไม้	฿350.00
เดรสลาย	฿300.00
เดรสลายดอกไม้หวาน	฿350.00
เสื้อ+กระโปรง ขาวแดง	฿500.00
เสื้อ+กระโปรง เขียวแดง	฿500.00
เสื้อ+กระโปรง ฟ้าดำ	฿500.00
เสื้อ+กางเกง คอวีลายดอกไม้	฿550.00
เสื้อแขนงุด ลายจุด	฿250.00
เสื้อคอกลมขาว คายมแนว	฿250.00
เสื้อคอปก ชมพู-ฟ้า	฿250.00
เสื้อคอระบาย สีขาว	฿250.00

©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.10 หน้าจอแสดงข้อมูลสินค้าของแต่ละร้าน

4.1.1.4 การเลือกร้านค้า

ลูกค้าสามารถเข้าไปดูแต่ละร้านค้า เพื่อดูข้อมูลรายละเอียดต่างๆ ของแต่ละร้านค้า ดังรูปที่ 4.11 เมื่อเลือกร้านค้าแล้ว จะแสดงหน้าต่างข้อมูลรายละเอียดต่างๆ ดังรูปที่ 4.8



รูปที่ 4.11 หน้าจอแสดงร้านค้าทั้งหมด

4.1.1.5 การเลือกร้านค้าแบ่งตามประเภท

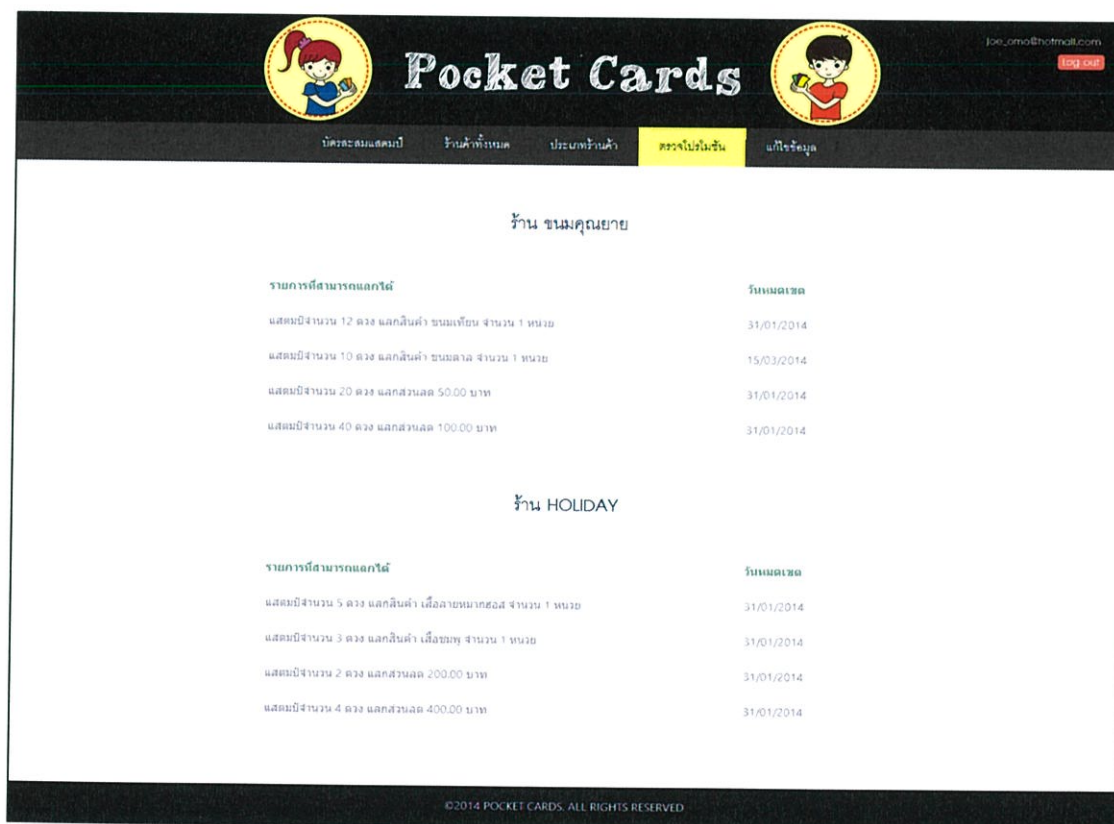
ลูกค้าสามารถเลือกร้านค้าแบ่งตามประเภท เมื่อคลิกเลือกแล้ว จะแสดงรายการร้านค้าตามที่เลือกประเภท ดังรูปที่ 4.12 เมื่อเลือกร้านค้าแล้ว จะแสดงหน้าต่างข้อมูลรายละเอียดต่างๆ ดังรูปที่ 4.8



รูปที่ 4.12 หน้าจอแสดงรายการร้านค้าตามที่เลือกประเภทร้านค้า

4.1.1.6 การตรวจโปรโมชั่น

ลูกค้าสามารถตรวจหาโปรโมชั่นที่ลูกค้ามีแต้มปีสะสมคงเหลือมากพอที่จะแลกสินค้าหรือส่วนลด ดังรูปที่ 4.13



รูปที่ 4.13 หน้าจอแสดงรายการโปรโมชั่นที่สามารถแลกได้

4.1.1.7 การแก้ไขข้อมูล

การแก้ไขข้อมูลของลูกค้า ทำได้โดยการกรอกหรือการคลิก “แก้ไข” ทั่วยช่องเพื่อแก้ไขข้อมูลส่วนตัวของลูกค้าที่เคยกรอกก่อนหน้านี้ ดังรูปที่ 4.14

The screenshot shows the 'Pocket Cards' website interface. At the top, there are navigation tabs: 'บัตรสะสมแต้ม', 'ร้านค้าทั้งหมด', 'ประเภทร้านค้า', 'ตรวจไปรษณีย์', and 'แก้ไขข้อมูล'. The 'แก้ไขข้อมูล' tab is active. The main content area contains a user profile form with the following fields and values:

- เบอร์โทรศัพท์: 1111111111 (with an 'แก้ไข' button)
- E-mail: joe_omo@hotmail.com (with an 'แก้ไข' button)
- วันเกิด: 24/02/1449 (with a 'เลือก' button)
- ชื่อ: Korakot
- นามสกุล: Wirakunniyom
- เพศ: ชาย (with a dropdown arrow)
- ที่อยู่: 13/2 หมู่10 ตำบลสองพี่น้อง อำเภอสว่างวีรวัฒน์ จังหวัดจันทบุรี 22120

On the right side, there is a profile picture placeholder and a 'เลือกไฟล์' button. Below it are 'อัปเดตข้อมูล' and 'ลบรูป' buttons, and a 'เปลี่ยนรหัสผ่าน' link. At the bottom of the form is a 'บันทึก' (Save) button. The footer text reads '©2014 POCKET CARDS. ALL RIGHTS RESERVED'.

รูปที่ 4.14 หน้าจอแสดงการแก้ไขข้อมูลของลูกค้า

หากต้องการแก้ไขเบอร์โทรศัพท์ ให้ทำการคลิก “แก้ไข” ทั่วยช่อง จะปรากฏดังรูปที่ 4.15

This screenshot shows the same user profile form as in Figure 4.14, but with a modal dialog box open for editing the phone number. The modal has a title 'แก้ไขเบอร์โทรศัพท์' and a text input field for the new number. Below the input field are two buttons: 'แก้ไข' and 'ยกเลิก'. The background form is dimmed.

รูปที่ 4.15 หน้าจอแสดงการแก้ไขเบอร์โทรศัพท์ของลูกค้า

หากต้องการแก้ไข E-mail ให้ทำการคลิก “แก้ไข” ที่ข้างช่องกรอก E-mail จะปรากฏดังรูปที่ 4.16

The screenshot shows the 'Pocket Cards' website registration page. A modal window titled 'แก้ไขอีเมล' (Edit Email) is open over the email field. The modal contains a text input field for the new email address and two buttons: 'แก้ไข' (Edit) and 'ยกเลิก' (Cancel). The background form shows the following details:

- เบอร์โทรศัพท์: 1111111111
- E-mail: joe_omo@hotmail.com
- วันเกิด: 24/02/1449
- ชื่อ: Korakot
- นามสกุล: Wirakunnyom
- เพศ: ชาย
- ที่อยู่: 13/2 หมู่10 ตำบลสองพี่น้อง อำเภอสว่างวีรวัฒน์ จังหวัดจันทบุรี 22120

รูปที่ 4.16 หน้าจอแสดงการแก้ไข E-mail ของลูกค้า

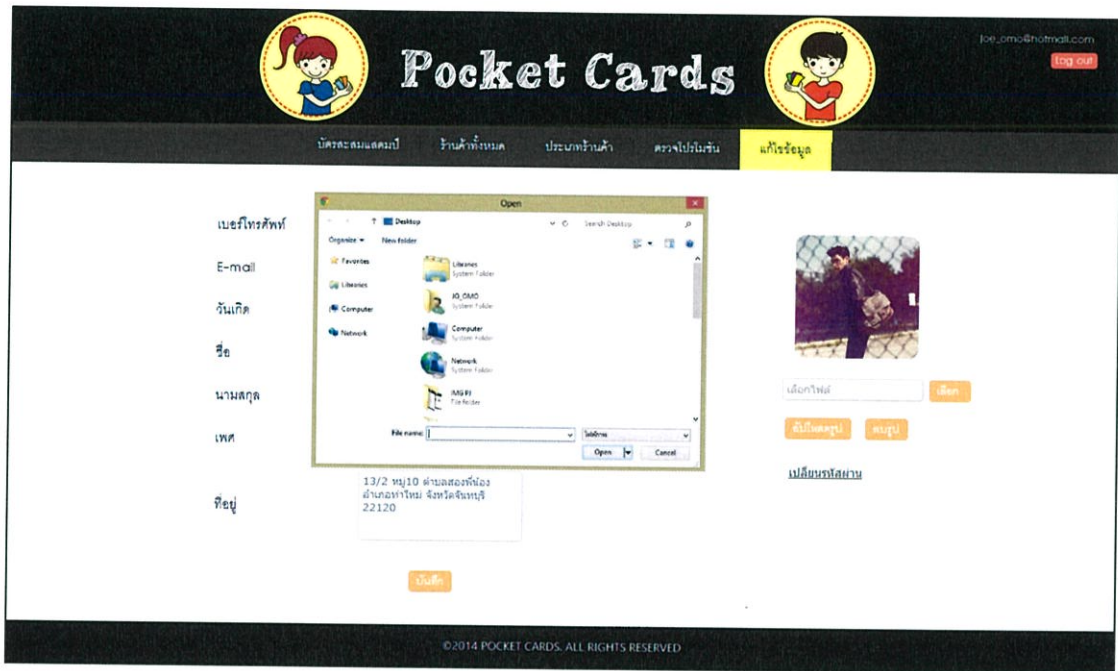
หากต้องการแก้ไขวันเกิด ให้ทำการคลิกที่ช่องกรอกวันเกิดจะปรากฏดังรูปที่ 4.17 หรือหากไม่ต้องการระบุวันเกิด จะสามารถคลิกปุ่ม “ไม่ระบุ”

The screenshot shows the 'Pocket Cards' website registration page. A calendar widget is open over the birth date field. The calendar shows the month of February 1992. The date 24 is selected. Below the calendar, there are buttons for 'Today' and 'Done'. The background form shows the following details:

- เบอร์โทรศัพท์: 1111111111
- E-mail: joe_omo@hotmail.com
- วันเกิด: 24/02/1992
- ชื่อ: Korakot
- นามสกุล: Wirakunnyom
- เพศ: ชาย
- ที่อยู่: Today

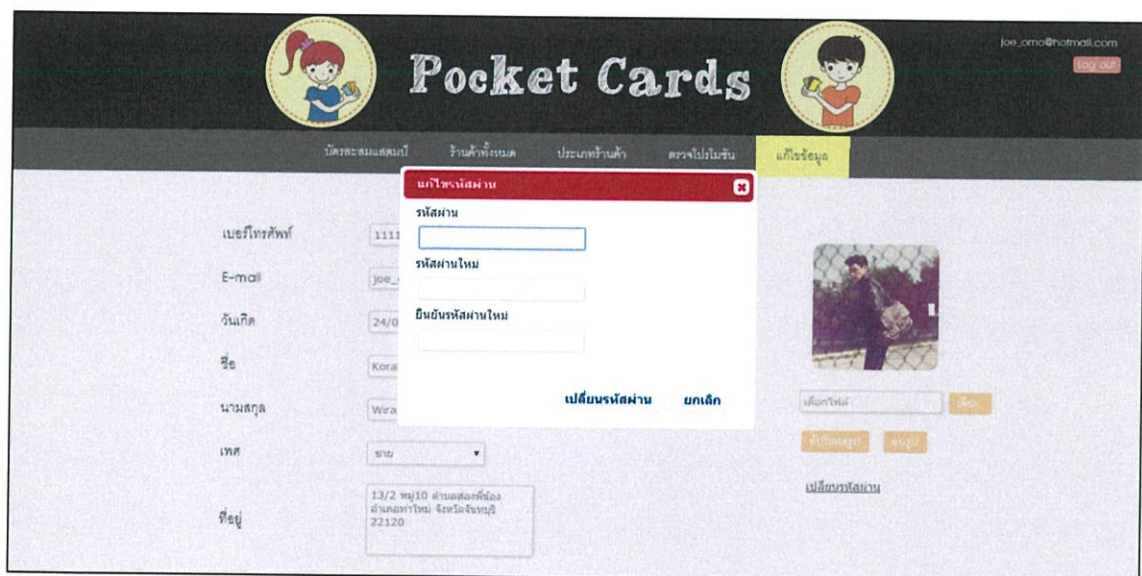
รูปที่ 4.17 หน้าจอแสดงการแก้ไขวันเกิดของลูกค้า

หากต้องการแก้ไขรูป ให้ทำการคลิก“เลือก” เพื่อเลือกไฟล์รูป จากนั้นคลิก “อัปโหลด” เพื่อเปลี่ยนแปลงจากรูปเก่าเป็นรูปใหม่ที่เลือกมา หากต้องการลบรูปจะสามารถคลิกปุ่ม “ลบรูป” ดังรูปที่ 4.18



รูปที่ 4.18 หน้าจอแสดงการแก้ไขรูปลูกค้า

หากต้องการเปลี่ยนรหัสผ่าน ให้ทำการคลิก “เปลี่ยนรหัสผ่าน” จะปรากฏดังรูปที่ 4.19



รูปที่ 4.19 หน้าจอแสดงการเปลี่ยนรหัสผ่านของลูกค้า

เมื่อแก้ไขข้อมูลเสร็จให้ทำการกดปุ่ม “บันทึก” ข้อมูลที่แก้ไขจะถูกจัดเก็บลงในระบบ ดังรูปที่ 4.20

The screenshot shows a web application interface for editing user information. At the top, there is a navigation bar with a logo on the left, a search bar in the center, and a user profile icon on the right. Below the navigation bar, there are several menu items: 'บัตรสะสมแต้ม', 'ร้านค้าทั้งหมด', 'ประเภทร้านค้า', 'วางใจไปรษณีย์', and 'แก้ไขข้อมูล' (highlighted in yellow). The main content area contains a form with the following fields:

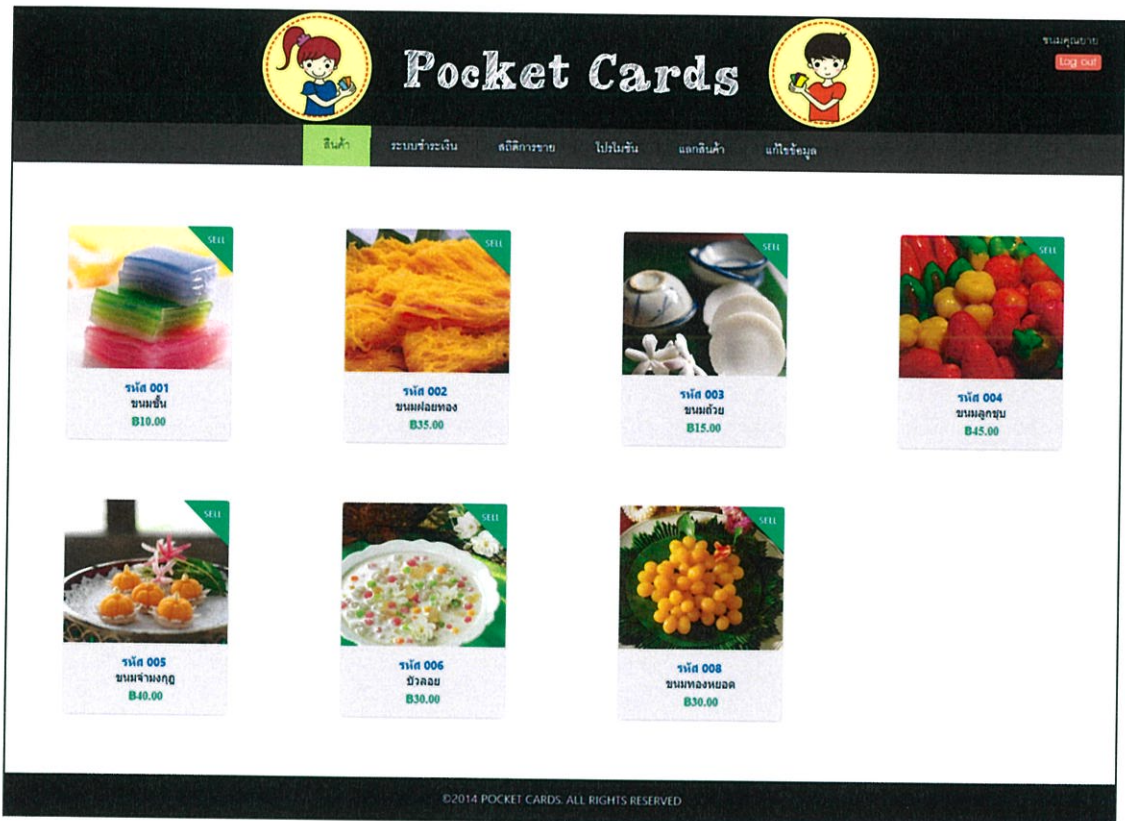
- เบอร์โทรศัพท์: 1111111111 (with an 'แก้ไข' button)
- E-mail: joe_omo@hotmail.com (with an 'แก้ไข' button)
- วันเกิด: 24/02/1449 (with a 'ไม่สนใจ' button)
- ชื่อ: Korakot
- นามสกุล: Wirakunniyom
- เพศ: ชาย (with a dropdown arrow)
- ที่อยู่: 13/2 หมู่10 ตำบลอรทัยใต้ อำเภอท่าใหม่ จังหวัดจันทบุรี 22120

On the right side of the form, there is a profile picture placeholder, a 'แก้ไขโปรไฟล์' button, and two buttons labeled 'ดึงข้อมูล' and 'ลบข้อมูล'. At the bottom of the form, there is a 'บันทึก' button. The footer of the page contains the text '©2014 POCKET CARDS. ALL RIGHTS RESERVED'.

รูปที่ 4.20 หน้าจอแสดงการบันทึกการแก้ไขข้อมูลของลูกค้า

4.1.2 โปรแกรมส่วนของร้านค้า

การเข้าใช้งานภายในระบบร้านค้าจะต้องเป็นสมาชิกของระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมก่อน ซึ่งจะทำการสมัครสมาชิกได้จากหน้าจอหลักของผู้ใช้บริการ หน้าจอหลักในส่วน of ร้านค้าจะประกอบด้วย 6 รายการ คือ สินค้า ระบบชำระเงิน สถิติการขาย โปรโมชั่น แลกสินค้า และแก้ไขข้อมูล



รูปที่ 4.21 หน้าหลักแสดงการเข้าใช้งานของส่วนร้านค้า

4.1.2.1 การสมัครสมาชิก

จากหน้าจอหลักผู้ใช้บริการเข้าสู่รายการสมัครสมาชิก ทำการกรอกข้อมูลที่เป็นต่อระบบ ซึ่งประกอบไปด้วย E-mail รหัสผ่าน ยืนยันรหัสผ่าน ชื่อร้านค้า ประเภทร้านค้า รายละเอียดร้านค้า ที่อยู่ของร้านค้า เบอร์โทรศัพท์ Website โดยตัวอักษรใหญ่และตัวอักษรเล็กจะให้ผลต่างกัน หากทำการสมัครเสร็จถูกต้องระบบจะทำการส่งข้อมูลการสมัครให้ทาง E-mail แต่หากเกิดข้อผิดพลาด ระบบจะแจ้งเตือนข้อผิดพลาดขึ้นมาตรงท้ายของช่อง ดังรูปที่ 4.22



Pocket Cards





E-mail * * ข้อมูลที่ต้องใส่

รหัสผ่าน * * ข้อมูลที่ต้องใส่

ยืนยันรหัสผ่าน *

ชื่อร้านค้า * * ข้อมูลที่ต้องใส่

ประเภทร้านค้า * เลือกประเภท * ข้อมูลที่ต้องใส่

รายละเอียดร้านค้า * * ข้อมูลที่ต้องใส่

ที่อยู่ * * ข้อมูลที่ต้องใส่

เบอร์โทรศัพท์ * * ข้อมูลที่ต้องใส่

Website

รูปประจำตัว * ข้อมูลที่ต้องใส่

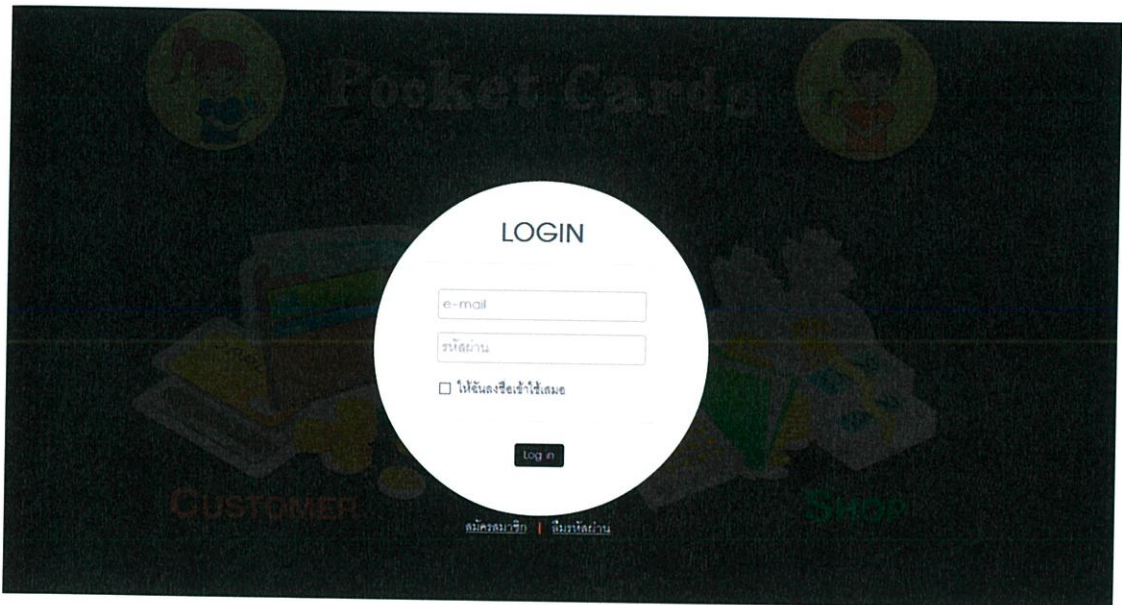


©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.22 หน้าจอแสดงการสมัครสมาชิกของร้านค้า

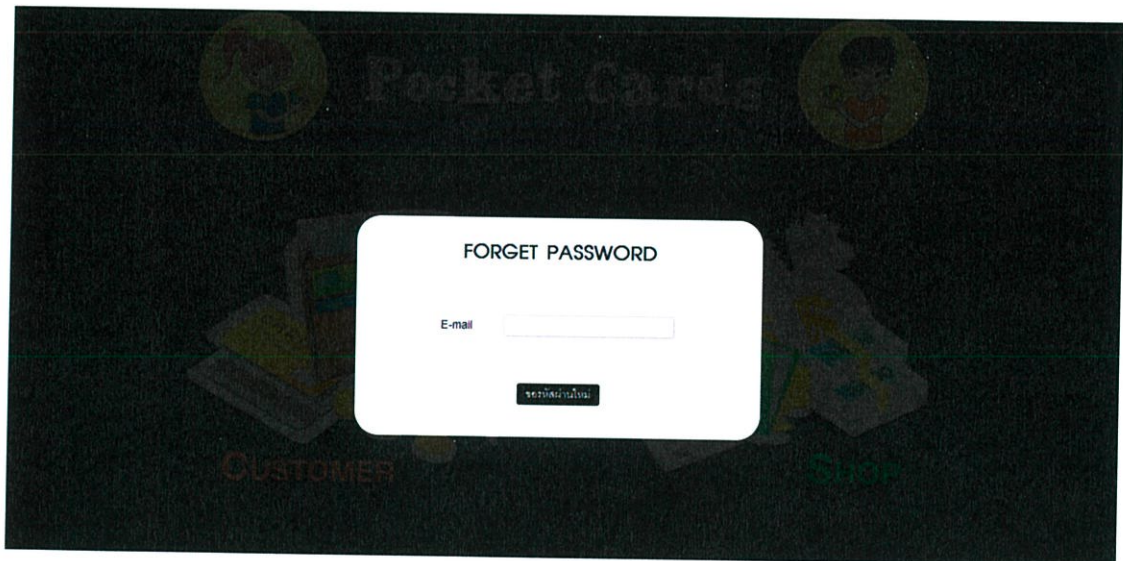
4.1.2.2 การเข้าใช้บริการ(Login)

กรอก E-mail และรหัสผ่านให้ถูกต้องในหน้า “Login” ของร้านค้า หลังจากนั้นกดปุ่ม Login ถ้าหากต้องให้มันลงชื่อเข้าใช้เสมอ ก็ทำการเลือกถูก เพื่อให้สามารถเข้าระบบอีกครั้งได้โดยไม่ต้อง login อีกครั้ง ดังรูปที่ 4.23



รูปที่ 4.23 หน้าจอแสดงการ Log in เข้าสู่ระบบของร้านค้า

หากลืมรหัสผ่าน ให้ทำการกดปุ่ม “ลืมรหัสผ่าน” เพื่อขอรหัสผ่านใหม่ โดยกรอก E-mail เพื่อระบบจะทำการส่งรหัสผ่านใหม่ให้ทาง E-mail ดังรูปที่ 4.24



รูปที่ 4.24 หน้าจอแสดงการขอรหัสผ่านใหม่ โดยกรอก E-mail

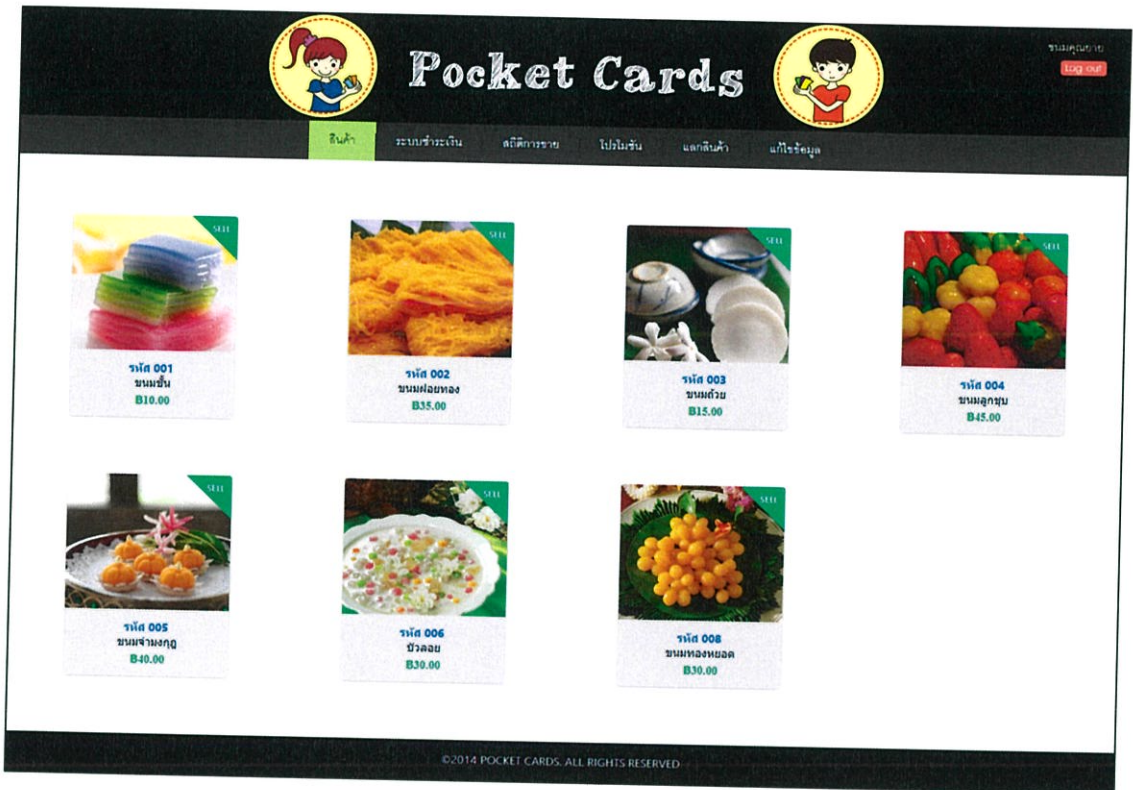
4.1.2.3 การจัดการกับสินค้า

- 1) การเพิ่มรายการสินค้า ทำการคลิก “สินค้า” -> เลือก “เพิ่มรายการสินค้า” จากนั้นกรอกข้อมูลรายละเอียดต่างๆ ได้แก่ รหัสสินค้า ชื่อสินค้า ราคา และใส่รูปภาพสินค้า เมื่อทำการกรอกเสร็จแล้ว ให้คลิก “เพิ่มสินค้า” เพื่อเพิ่มข้อมูลสินค้าลงไปในระบบ ดังรูปที่ 4.25


The screenshot shows the 'Pocket Cards' website interface. At the top, there are two circular icons of children holding cards. The main navigation bar includes 'สินค้า' (Items), 'ระบบชำระเงิน' (Payment System), 'สถิติการขาย' (Sales Statistics), 'ไปรษณีย์' (Post), 'แลกเปลี่ยน' (Exchange), and 'แก้ไขข้อมูล' (Edit Information). The 'สินค้า' menu is highlighted. Below the navigation bar, there is a form to add a new item. The form fields are: 'รหัสสินค้า' (Item Code) with a text input field and a red asterisk indicating it is required; 'ชื่อสินค้า' (Item Name) with a text input field and a red asterisk; 'ราคาสินค้า' (Item Price) with a text input field containing 'พ.ร.บ.ราคา B' and a red asterisk; and 'รูปสินค้า' (Item Image) with a file selection button labeled 'เลือกไฟล์' and a green 'เพิ่มสินค้า' (Add Item) button below it. At the bottom of the form, there is a logo for 'คิงส์คิงส์คิงส์' (Kings Kings Kings) with the phone number '5331519890707' and a copyright notice '©2014 POCKET CARDS. ALL RIGHTS RESERVED'.

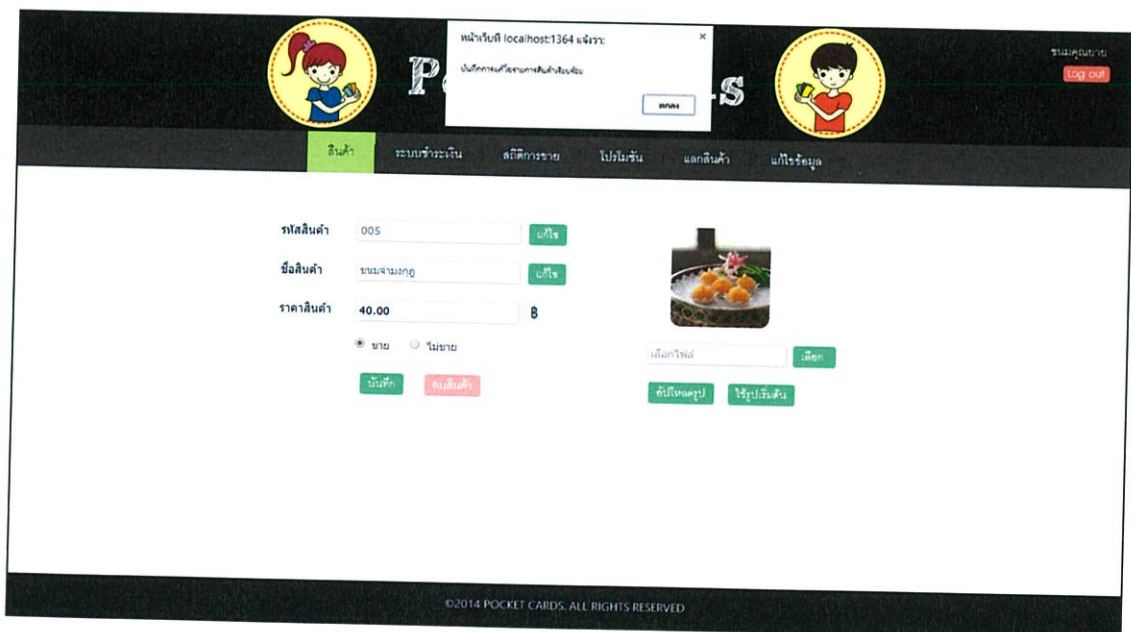
รูปที่ 4.25 หน้าจอแสดงการเพิ่มรายการสินค้า

- 2) รายการสินค้าที่ขาย ทำการคลิก “สินค้า” -> เลือก “รายการสินค้าที่ขาย” จะปรากฏรายการสินค้าที่ขาย ดังรูปที่ 4.26



รูปที่ 4.26 หน้าจอแสดงรายการสินค้าที่ขาย

เมื่อต้องการแก้ไขข้อมูลสินค้า ให้ทำการคลิก  เพื่อแก้ไขข้อมูลของสินค้าได้แก่ รหัสสินค้า ชื่อสินค้า ราคาสินค้า เปลี่ยนแปลงรูปภาพ และสถานะการขาย จากนั้นคลิก “บันทึก” ดังรูปที่ 4.27



รูปที่ 4.27 หน้าจอแสดงการแก้ไขข้อมูลสินค้า

- 3) รายการสินค้าที่ไม่ขาย ให้ทำการคลิก “สินค้า” -> เลือก “รายการสินค้าที่ขาย” จะปรากฏรายการสินค้าที่ไม่ขาย ดังรูปที่ 4.28



รูปที่ 4.28 หน้าจอแสดงรายการสินค้าที่ไม่ขาย

4.1.2.4 ระบบชำระเงิน

เมื่อลูกค้าต้องการซื้อสินค้า ร้านค้าจะกรอกเบอร์โทรศัพท์ของลูกค้า รหัสสินค้า และจำนวนสินค้าที่ซื้อ จากนั้นคลิก “เพิ่มรายการ” เพื่อเพิ่มรายการ สินค้าจะแสดงปรากฏดังรูปที่ 4.29

เว็บไซต์: 0000000000 รหัสสินค้า: [] จำนวนซื้อ: []

ชื่อสินค้า	ราคา/หน่วย	จำนวน	ราคารวม	ลบ	ราคารวม
ขนมชิ้น	10.00	1	10.00	ลบ	275.00
ขนมเปลือกทอง	35.00	1	35.00	ลบ	ส่วนลด 0
ขนมถั่วฝักยาว	15.00	3	45.00	ลบ	ราคารวมสุทธิ 275.00
ขนมลูกชุบ	45.00	1	45.00	ลบ	<input type="button" value="ชำระเงิน"/>
ขนมจางงอู	40.00	2	80.00	ลบ	
บัวลอย	30.00	1	30.00	ลบ	
ขนมทองหยอด	30.00	1	30.00	ลบ	

©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.29 หน้าจอแสดงร้านค้าจัดการการชำระเงินของลูกค้า

หมายเหตุ: ปุ่มลัดในหน้า “ชำระเงิน” ได้แก่

Right arrow key คือ ไปช่องถัดไปทางขวา

Left arrow key คือ ไปช่องถัดไปทางซ้าย

Up arrow key คือ ไปยังช่องเบอร์โทรศัพท์

Bottom arrow key คือ ไปยังปุ่ม “ชำระเงิน”

End key คือ การกดปุ่ม “ชำระเงิน”

ถ้าเลือกอยู่ที่ช่องรหัสสินค้าหรือช่องจำนวนซื้อแล้วกด Enter key จะเป็นการกดปุ่ม “เพิ่มรายการ”

หากลูกค้าต้องการใช้ส่วนลด ให้ทำการคลิกปุ่ม “ใช้ส่วนลด” จะปรากฏดังรูปที่ 4.30

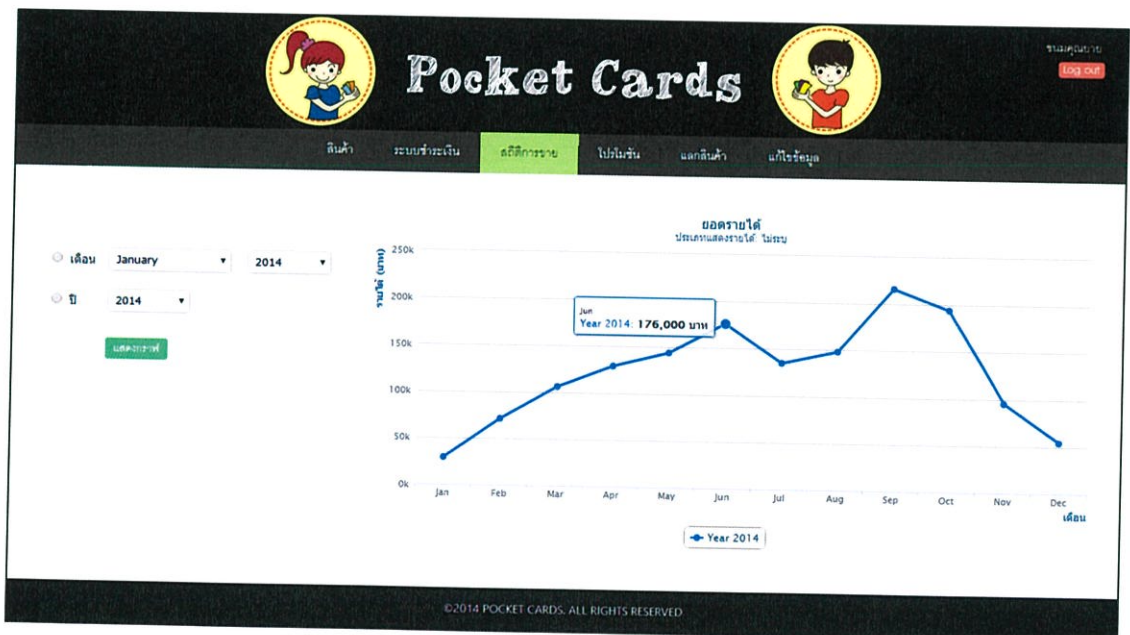


รูปที่ 4.30 หน้าจอแสดงระบบชำระเงินในการใช้ส่วนลด

4.1.2.5 สถิติการขาย

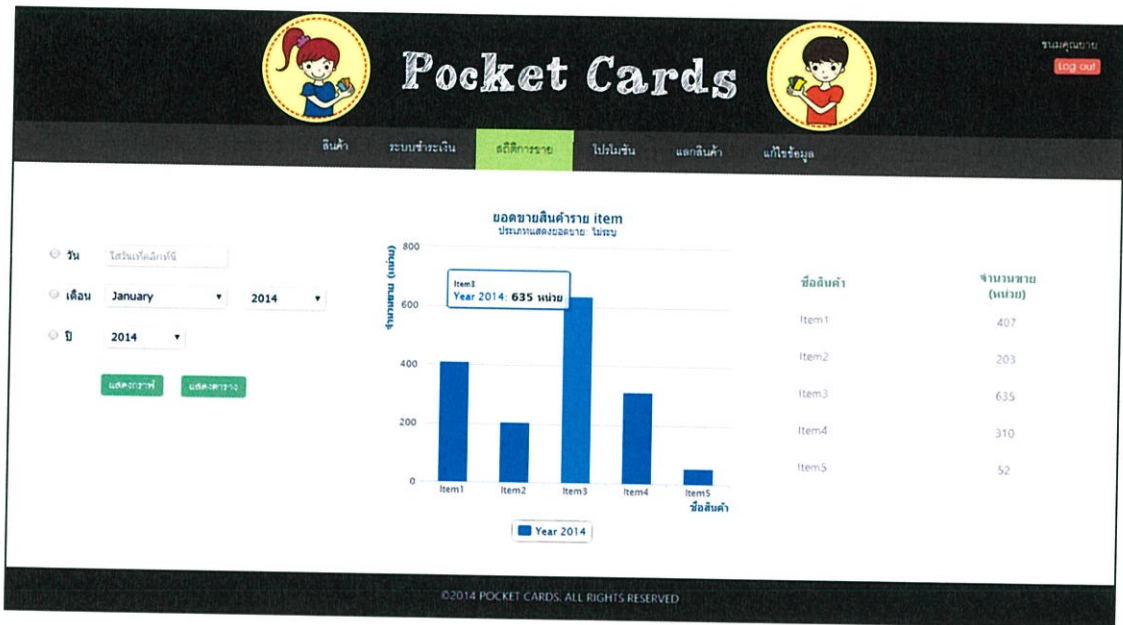
จะประกอบไปด้วย สถิติรายได้และสถิติยอดขาย

1) สถิติรายได้



รูปที่ 4.31 หน้าจอแสดงสถิติรายได้แบบกราฟ

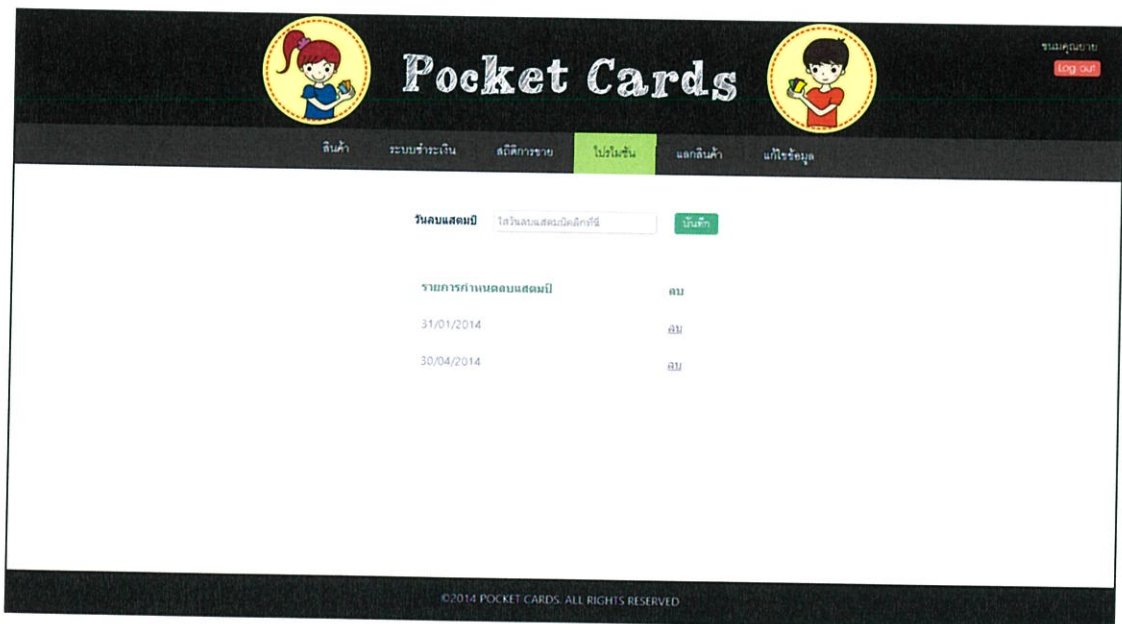
2) สถิติยอดขาย



รูปที่ 4.32 หน้าจอแสดงสถิติยอดขายแบบแผนภูมิ

4.1.2.6 การจัดการกับโปรโมชั่น

- 1) กำหนดลบแสดมปี ร้านค้าสามารถเลือกกำหนดลบแสดมปีจากนั้นคลิกปุ่ม “บันทึก” เพื่อให้ระบบทำการลบแสดมปีทั้งหมดเพื่อเริ่มการสะสมแสดมปีใหม่ ดังรูปที่ 4.33



รูปที่ 4.33 หน้าจอแสดงการกำหนดลบแสดมปี

- 2) ร้านค้าสามารถกำหนดโปรโมชั่นให้แสดมปี โดยสามารถเลือกวันเริ่มต้นโปรโมชั่น และวันสิ้นสุดโปรโมชั่น ซึ่งมีโปรโมชั่นสองแบบ ได้แก่ แบบที่หนึ่ง คือ เมื่อลูกค้าซื้อสินค้าใดสินค้าหนึ่งครบจำนวนหน่วยสินค้าตามที่เรากำหนดโปรโมชั่นไว้ ลูกค้าจะได้รับแสดมปีตามกำหนดจำนวนดวงแสดมปี และแบบที่สอง คือ เมื่อลูกค้าซื้อสินค้าครบจำนวนบาท จะได้รับแสดมปีตามร้านค้ากำหนด จากนั้นคลิก “เพิ่มโปรโมชั่น” จะปรากฏรายการโปรโมชั่นให้แสดมปีตามที่เรากำหนดไว้ข้างล่าง และจะมีการแสดงสถานะการเปิดใช้งานโปรโมชั่น หรือรอเปิดใช้งานโปรโมชั่น ดังรูปที่ 4.34

รายการโปรโมชั่นในแสดมปี	สถานะ	วันเริ่มโปรโมชั่น	วันสิ้นสุดโปรโมชั่น	ลบ
ซื้อครบ 200.00 บาท ได้รับแสดมปีจำนวน 1 ดวง	เปิดใช้งานอยู่	27/01/2014	31/01/2014	
ซื้อครบ 1000.00 บาท ได้รับแสดมปีจำนวน 12 ดวง	เปิดใช้งานอยู่	14/02/2014	14/02/2014	
ซื้อสินค้า ชมมดาก จำนวน 5 หน่วย ได้รับแสดมปีจำนวน 2 ดวง	เปิดใช้งานอยู่	27/01/2014	31/01/2014	
ซื้อครบ 200.00 บาท ได้รับแสดมปีจำนวน 1 ดวง	รอเปิดใช้งาน	01/04/2014	30/04/2014	ลบ
ซื้อสินค้า ชมมดากของลด จำนวน 2 หน่วย ได้รับแสดมปีจำนวน 1 ดวง	รอเปิดใช้งาน	01/04/2014	30/04/2014	ลบ
ซื้อสินค้า ชมมดาก จำนวน 2 หน่วย ได้รับแสดมปีจำนวน 1 ดวง	รอเปิดใช้งาน	01/04/2014	30/04/2014	ลบ

©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.34 หน้าจอแสดมปีโปรโมชั่นให้แสดมปี

- 3) ร้านค้าสามารถกำหนดโปรโมชันใช้แสดมปี โดยสามารถเลือกวันเริ่มต้นโปรโมชัน และวันสิ้นสุดโปรโมชัน ซึ่งมีโปรโมชันสองแบบ ได้แก่ แบบที่หนึ่ง คือ เมื่อลูกค้ามีแสดมปีครบจำนวนดวงตามที่ร้านค้ากำหนดโปรโมชันใช้แสดมปีไว้ ลูกค้าจะสามารถเลือกสินค้าที่ต้องการแลก หรือแบบที่สองคือ เมื่อลูกค้ามีแสดมปีครบจำนวน จะสามารถใช้แสดมปีเพื่อแลกส่วนลดในการซื้อสินค้า จากนั้นคลิก “เพิ่มโปรโมชัน” จะปรากฏรายการโปรโมชันใช้แสดมปีตามที่เรากำหนดไว้ข้างล่าง และจะมีการแสดงสถานะการเปิดใช้งานโปรโมชัน หรือรอเปิดใช้งานโปรโมชัน ดังรูปที่ 4.35

The screenshot shows the 'Pocket Cards' website interface. At the top, there are navigation tabs: 'สินค้า', 'ระบบชำระเงิน', 'สถิติการขาย', 'โปรโมชั่น' (selected), 'แลกสินค้า', and 'แก้ไขข้อมูล'. Below the navigation, there are input fields for 'วันเริ่มต้นโปรโมชั่น' and 'วันสิ้นสุดโปรโมชั่น', both with 'เลือกวัน' (Select Day) buttons. There are two radio button options for card types: 'แสดมปีจำนวน' (with fields for 'ดวง', 'แลกสินค้า', and 'จำนวน หน่วย') and 'แสดมปีจำนวน' (with fields for 'ดวง', 'แลกส่วนลด', and 'บาท'). A green 'เป็นไปไม่จริง' (Not Real) button is below the options. Below the form is a table titled 'รายการโปรโมชั่นใช้แสดมปี' (Active Promotional Cards List).

รายการโปรโมชั่นใช้แสดมปี	สถานะ	วันเริ่มต้นโปรโมชั่น	วันสิ้นสุดโปรโมชั่น	จบ
แสดมปีจำนวน 20 ดวง แลกส่วนลด 50.00 บาท	เปิดใช้งาน	27/01/2014	31/01/2014	
แสดมปีจำนวน 12 ดวง แลกสินค้า ชมรมเพื่อน จำนวน 1 หน่วย	เปิดใช้งาน	27/01/2014	31/01/2014	
แสดมปีจำนวน 10 ดวง แลกสินค้า ชมรมต่าง จำนวน 1 หน่วย	เปิดใช้งาน	13/03/2014	15/03/2014	
แสดมปีจำนวน 15 ดวง แลกส่วนลด 50.00 บาท	รอเปิดใช้งาน	01/04/2014	30/04/2014	ลบ
แสดมปีจำนวน 15 ดวง แลกสินค้า ชมรมลูกทุ่ง จำนวน 1 หน่วย	รอเปิดใช้งาน	01/04/2014	30/04/2014	ลบ
แสดมปีจำนวน 5 ดวง แลกสินค้า ชมรมวิทยากร จำนวน 1 หน่วย	รอเปิดใช้งาน	01/05/2014	31/05/2014	ลบ

At the bottom of the page, it says '©2014 POCKET CARDS. ALL RIGHTS RESERVED'.

รูปที่ 4.35 หน้าจอแสดงโปรโมชันใช้แสดมปี

4.1.2.7 การแลกสินค้า

ถ้าลูกค้าต้องการจะแลกสินค้า ร้านค้าจะกรอกเบอร์โทรศัพท์ลูกค้าและตรวจสอบข้อมูล ถ้าลูกค้ามีแต้มปีสะสมพอจะมีการแลกสินค้าขึ้นให้เลือก เมื่อเลือกแล้วทำการคลิกปุ่ม “แลกสินค้า” ดังรูปที่ 4.36

เบอร์โทรศัพท์: 1111111111

ชื่อ-นามสกุล: Korakot Wirakunniyom

เลือก	รายการแลกสินค้า	วันหมดเขต
<input type="radio"/>	แลกสินค้าจำนวน 12 ดวง แลกสินค้า ขนมเย็บขน จำนวน 1 หนวย	31/01/2014
<input type="radio"/>	แลกสินค้าจำนวน 10 ดวง แลกสินค้า ขนมตาล จำนวน 1 หนวย	15/03/2014
<input type="radio"/>	แลกสินค้าจำนวน 5 ดวง แลกสินค้า ขนมฝอยทอง จำนวน 1 หนวย	31/05/2014
<input type="radio"/>	แลกสินค้าจำนวน 5 ดวง แลกสินค้า ขนมเจมงกุฎ จำนวน 1 หนวย	30/04/2014
<input type="radio"/>	แลกสินค้าจำนวน 10 ดวง แลกสินค้า ขนมกล้วย จำนวน 2 หนวย	30/04/2014

©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.36 หน้าจอแสดงการแลกสินค้า

4.1.2.8 การแก้ไขข้อมูล

การแก้ไขข้อมูลของร้านค้า ทำได้โดยการกรอกหรือการคลิก “แก้ไข” ทำชองเพื่อแก้ไขข้อมูลร้านค้าที่เคยกรอกก่อนหน้าี้ ดังรูปที่ 4.37

Pocket Cards

สมัครสมาชิก

สินค้า ระบบชำระเงิน สถิติการขาย ไปรษณีย์ แลกสินค้า **แก้ไขข้อมูล**

E-mail: joe_omo@hotmail.com **แก้ไข**

ประเภทร้านค้า: อาหารและเครื่องดื่ม **แก้ไข**

ชื่อร้านค้า: ขมดสยาม **แก้ไข**

รายละเอียดร้านค้า: เป็นร้านขายขนมไทยสุดอร่อยไม่ติดอกติดใจใคร่ขาด

ที่อยู่: ตลาดน้ำท่าเสา ลาดกระบัง เขต A ต.ลำไย อ.ลำไย จ.สุพรรณบุรี

เบอร์โทรศัพท์: 08978346839

Website:

บันทึก

เลือกไฟล์ **เลือก**

อัพโหลดรูป **ลบรูป**

[เปลี่ยนรหัสผ่าน](#)

©2014 POCKET CARDS. ALL RIGHTS RESERVED

รูปที่ 4.37 หน้าจอแสดงการแก้ไขข้อมูลของร้านค้า

หากต้องการแก้ไข E-mail ให้ทำการคลิก “แก้ไข” ที่ข้างช่องกรอก E-mail จะปรากฏดังรูปที่ 4.38

Pocket Cards

สมัครสมาชิก

สินค้า ระบบชำระเงิน สถิติการขาย ไปรษณีย์ แลกสินค้า **แก้ไขข้อมูล**

E-mail: joe_omo@hotmail.com **แก้ไขอีเมล**

ประเภทร้านค้า: อาหารและเครื่องดื่ม

ชื่อร้านค้า: ขมดสยาม

รายละเอียดร้านค้า: เป็นร้านขายขนมไทยสุดอร่อยไม่ติดอกติดใจใคร่ขาด

ที่อยู่: ตลาดน้ำท่าเสา ลาดกระบัง เขต A ต.ลำไย อ.ลำไย จ.สุพรรณบุรี

เบอร์โทรศัพท์: 08978346839

เลือกไฟล์ **เลือก**

อัพโหลดรูป **ลบรูป**

[เปลี่ยนรหัสผ่าน](#)

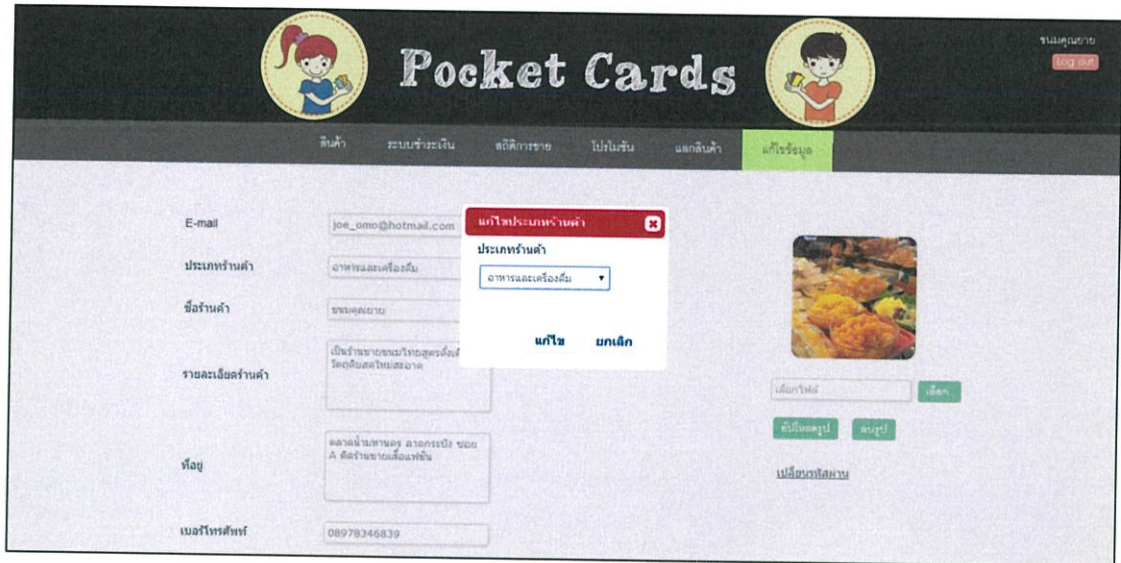
แก้ไขอีเมล

ยืนยันใหม่

แก้ไข **ยกเลิก**

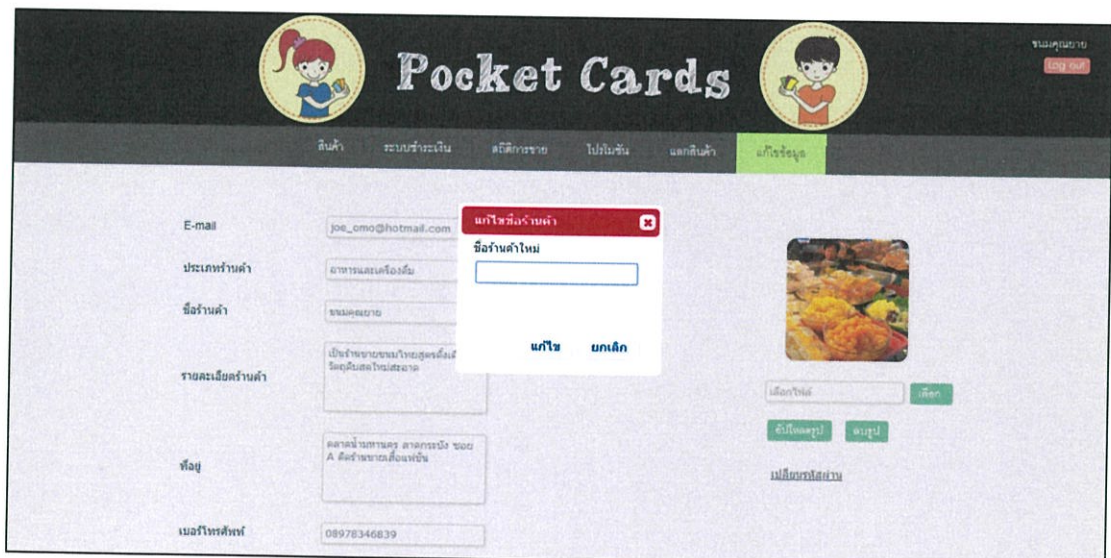
รูปที่ 4.38 หน้าจอแสดงการแก้ไข E-mail ของร้านค้า

หากต้องการแก้ไขประเภทร้านค้า ให้ทำการคลิก “แก้ไข” ทำयช่อง จะปรากฏดังรูปที่ 4.39 แล้วเลือกประเภทร้านค้า จากนั้นทำการคลิก “แก้ไข” เพื่อแก้ไขการแก้ไขประเภทร้านค้า



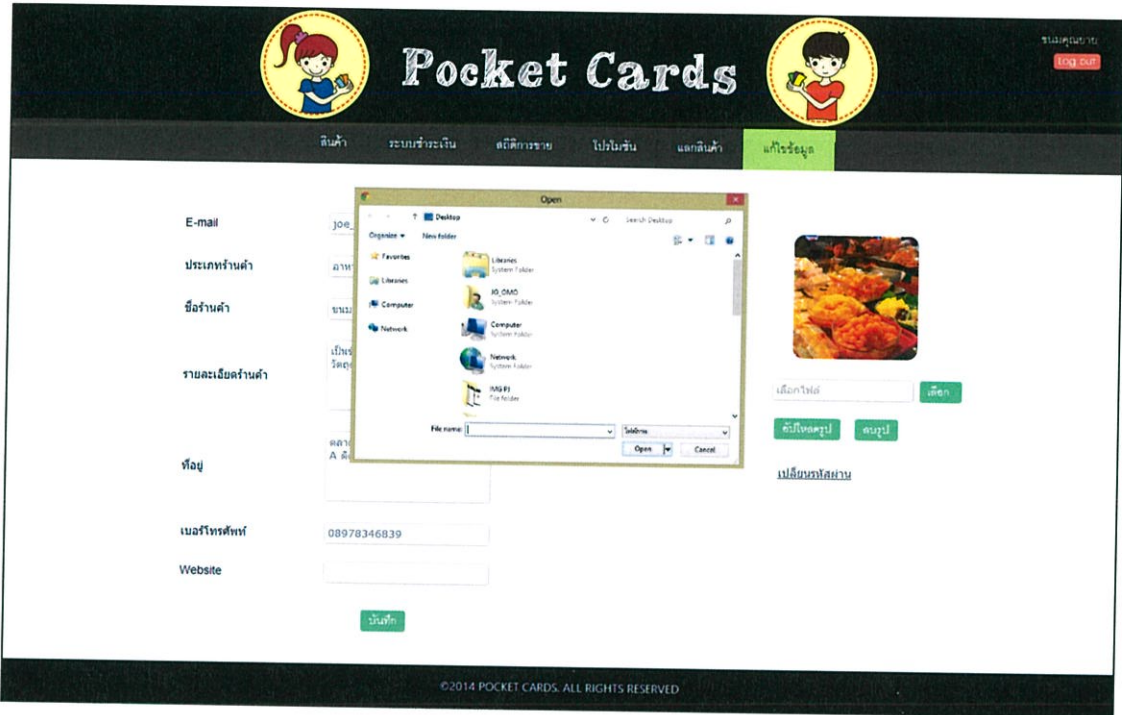
รูปที่ 4.39 หน้าจอแสดงการแก้ไขประเภทร้านค้า

หากต้องการแก้ไขชื่อร้านค้า ให้ทำการคลิก “แก้ไข” ทำยช่อง จะปรากฏดังรูปที่ 4.40 แล้วแก้ไขชื่อร้านค้า จากนั้นทำการคลิก “แก้ไข” เพื่อแก้ไขชื่อร้านค้า



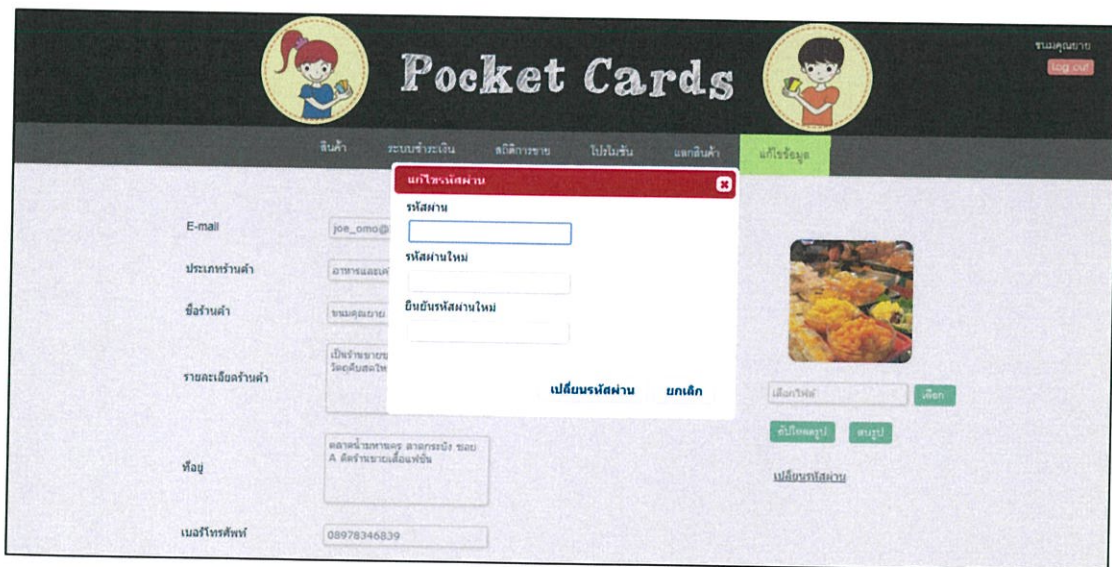
รูปที่ 4.40 หน้าจอแสดงการแก้ไขชื่อร้านค้า

หากต้องการแก้ไขรูป ให้ทำการคลิก "เลือก" เพื่อเลือกไฟล์รูป จากนั้นคลิก "อัปโหลด" เพื่อเปลี่ยนแปลงจากรูปเก่าเป็นรูปใหม่ที่เลือกมา หากต้องการลบรูปจะสามารถคลิกปุ่ม "ลบรูป" ดังรูปที่ 4.41



รูปที่ 4.41 หน้าจอแสดงการแก้ไขรูปร้านค้า

หากต้องการเปลี่ยนรหัสผ่าน ให้ทำการคลิก "เปลี่ยนรหัสผ่าน" จะปรากฏดังรูปที่ 4.42



รูปที่ 4.42 หน้าจอแสดงการเปลี่ยนรหัสผ่านของร้านค้า

เมื่อแก้ไขข้อมูลเสร็จให้ทำการกดปุ่ม “บันทึก” ข้อมูลที่แก้ไขจะถูกจัดเก็บลงในระบบ ดังรูปที่ 4.43

The screenshot shows a web application interface for editing restaurant information. The page has a dark header with navigation links: หน้า, ระบบชำระเงิน, สถิติการขาย, ไปรษณีย์, แลกสินค้า, and **แก้ไขข้อมูล**. A modal window is open at the top center with the title 'หน้าเว็บที่ localhost:1364 แก้ไข' and the text 'แก้ไขข้อมูลร้านค้า'. The main content area contains a form with the following fields:

- E-mail: joe_omo@hotmail.com (with a 'แก้ไข' button)
- ประเภทร้านค้า: อาหารและเครื่องดื่ม (with a 'แก้ไข' button)
- ชื่อร้านค้า: ชนบทคนขาย (with a 'แก้ไข' button)
- รายละเอียดร้านค้า: เป็นร้านขายขนมไทยสุดอร่อยไม่มีใครคิดค้นในเวลา (with a 'แก้ไข' button)
- ที่อยู่: ตลาดน้ำท่ามะขาม ลาดกระบัง ซอย A ตำบลบ้านเขาเมืองพิจิตร (with a 'แก้ไข' button)
- เบอร์โทรศัพท์: 08978346839 (with a 'แก้ไข' button)
- Website: (empty field)

At the bottom of the form is a 'บันทึก' (Save) button. On the right side, there is a preview image of food, a 'เลือกไฟล์' (Select File) button, and buttons for 'แก้ไขข้อมูล' and 'ลบข้อมูล'. Below the preview image is a link 'เปลี่ยนรูปสยาม'. The footer contains the text '©2014 POCKET CARDS. ALL RIGHTS RESERVED'.

รูปที่ 4.43 หน้าจอแสดงการบันทึกการแก้ไขข้อมูลของร้านค้า

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุป

จากการพัฒนาระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อม เพื่อแก้ปัญหาการพกพาบัตรสมาชิก โดยเริ่มจากการศึกษาและรวบรวมข้อมูลต่างๆ ค้นหาข้อมูลทางอินเทอร์เน็ต โดยค้นหาวิธีการพัฒนาระบบด้วยภาษาโปรแกรมและเครื่องมือที่ความอำนวยความสะดวก รวดเร็ว และง่ายต่อการใช้งานบนเทคโนโลยีปัจจุบัน ซึ่งการพัฒนาระบบเป็นไปได้ด้วยดี ระบบสามารถแก้ไขปัญหได้ตามเป้าหมายที่วางไว้ตามการวิเคราะห์และออกแบบระบบ คือ ลูกค้าสามารถสมัครสมาชิก แก้ไขข้อมูลภายในระบบ ดูรายละเอียดร้านค้า ดูรายละเอียดรายการโปรโมชั่น สามารถดูแสดมปีคงเหลือ ตรวจสอบโปรโมชั่นที่สามารถใช้แสดมปีได้ และในส่วนจากร้านค้าสามารถสมัครสมาชิก แก้ไขข้อมูลภายในระบบ เพิ่มและแก้ไขรายการสินค้า กำหนดและแก้ไขโปรโมชั่นในการให้แสดมปี กำหนดและแก้ไขโปรโมชั่นในการใช้แสดมปี กำหนดและลบกำหนด แสดมปี ดูสถิติรายได้และยอดขาย การชำระเงิน และการใช้แสดมปีเป็นส่วนลด จากการพัฒนาระบบตามการวิเคราะห์และออกแบบระบบทำให้ร้านค้าลดต้นทุนในการทำบัตรสมาชิก ลดต้นทุนในการทำระบบจัดการร้านค้าและระบบสมาชิก ทำให้ลูกค้าไม่ต้องพกพาบัตรสมาชิก สะดวกและรวดเร็วในการทราบข้อมูลข่าวสารต่างๆ ที่เกี่ยวกับโปรโมชั่นและสินค้า

ระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมได้ถูกออกแบบและพัฒนาโดยคำนึงถึงความสะดวกต่อการใช้งานของผู้ใช้ สะดวกและไม่ซับซ้อนมาก มีการตรวจสอบข้อบกพร่องฟังก์ชันการทำงานในแต่ละส่วนภายในระบบ ฉะนั้นจึงมีบางส่วนที่ต้องพัฒนาระบบภายหลังจากที่ทำการระบบ เช่น การเพิ่มฟังก์ชันการลิมิตรหัสผ่าน ซึ่งรายละเอียดเหล่านี้จะถูกเพิ่มเติมให้สมบูรณ์เพื่อหาข้อผิดพลาดที่อาจเกิดขึ้นได้ หากตรวจพบก็จะแก้ไขทันที เมื่อเกิดปัญหาขึ้นในระหว่างขั้นตอนของการทดสอบ ทางคณะผู้จัดทำได้ทำการแก้ไขส่วนที่ผิดพลาด เพื่อให้ระบบนั้นสามารถตอบสนองการใช้งานได้อย่างถูกต้องและมีประสิทธิภาพมากขึ้น

ตารางที่ 5.1 ตารางเปรียบเทียบระหว่างระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลาง และขนาดย่อม(PocketCards) และ Application “Stamp” บน Smartphone iOS

ฟังก์ชันการใช้งาน	PocketCards	Stamp
ส่วนของลูกค้า		
- สมัครสมาชิก	✓	✓
- แก้ไขข้อมูล	✓	✗
- ขอรหัสผ่านใหม่	✓	✗
- ดูรายการสินค้า	✓	✗
- ดูบัตรสะสมแต้มปี	✓	✓
- ดูรายละเอียดร้านค้าและรายการโปรโมชั่น	✓	✓
- ตรวจสอบโปรโมชั่นที่สามารถแลกได้	✓	✗
- แจ้งเตือนอัตโนมัติเมื่อบัตรใกล้หมดอายุ	✓	✗
- เลือกประเทศและภาษา	✗	✓
- ให้คะแนนและวิจารณ์	✗	✓
ส่วนของร้านค้า		
- สมัครสมาชิก	✓	✗
- แก้ไขข้อมูล	✓	✗
- ขอรหัสผ่านใหม่	✓	✗
- เพิ่มรายการสินค้า	✓	✗
- ดูรายการสินค้า	✓	✗
- แก้ไขรายการสินค้า	✓	✗
- เพิ่มและแก้ไขโปรโมชั่นในการให้แต้มปี	✓	✗
- เพิ่มและแก้ไขโปรโมชั่นในการใช้แต้มปี	✓	✗
- เพิ่มและลบกำหนดคลับแต้มปี	✓	✗
- ระบบชำระเงิน	✓	✗
- ใช้แต้มปีเป็นส่วนลด	✓	✓
- ดูสถิติรายได้แบบแผนภูมิ	✓	✗
- ดูสถิติยอดขายแบบแผนภูมิและตาราง	✓	✗
- ใช้แต้มปีแลกสินค้า	✓	✓

จากตารางเปรียบเทียบข้างต้นได้ชี้ให้เห็นถึงข้อเปรียบเทียบและข้อแตกต่างของทั้ง 2 ระบบที่ได้นำมาเปรียบเทียบกัน จะเห็นได้ว่าระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมนั้นมีฟังก์ชันการใช้งานที่มากกว่าแอปพลิเคชัน Stamp ทั้งในเรื่องของการดูรายการสินค้า การเพิ่มและแก้ไขรายการสินค้า มีระบบชำระเงิน สามารถที่จะดูสถิติรายได้แบบแผนภูมิและดูสถิติยอดขายได้ทั้งแบบแผนภูมิและแบบตาราง เป็นต้น แต่ทั้งนี้ก็ยังมีส่วนฟังก์ชันที่ระบบของเรายังไม่มี ตัวอย่างเช่น การเลือกประเทศ การเลือกภาษา ระบบการให้คะแนนและวิจารณ์

5.2 ข้อเสนอแนะ

ถึงแม้ระบบการจัดการสมาชิกบนเว็บสำหรับวิสาหกิจขนาดกลางและขนาดย่อมจะสามารถตอบโจทย์ตามเป้าหมายที่วางไว้เป็นอย่างดี แต่ก็มีการทำงานในบางส่วนที่ทางผู้พัฒนาเล็งเห็นว่าสามารถนำไปพัฒนาต่อเพื่อให้มีประสิทธิภาพในการทำงานยิ่งขึ้น อาทิเช่น เพิ่มแอปพลิเคชันบนโทรศัพท์มือถือในส่วนของระบบลูกค้าเพื่อให้ลูกค้าง่ายต่อการใช้งาน เพิ่มระบบสั่งซื้อสินค้าแบบออนไลน์เพื่อสะดวกแก่การขายสินค้าเป็นการเพิ่มโอกาสทางการค้า เพิ่มระบบการให้คะแนนและวิจารณ์สินค้าเพื่อให้ร้านค้าสามารถปรับปรุงสินค้าให้ดีขึ้น เพิ่มระบบการให้คะแนนและวิจารณ์ร้านค้าเพื่อให้ลูกค้าสามารถตัดสินใจในการใช้บริการและให้ร้านค้าสามารถปรับปรุงการให้บริการของร้านได้ เพิ่มระบบรายการสินค้าที่ลดราคาเพื่อดึงดูดลูกค้าและให้ลูกค้าง่ายต่อการเลือกซื้อสินค้าที่ลดราคา เพิ่มระบบแจ้งเตือนอัตโนมัติเมื่อลูกค้ามีแสดมป์สะสมคงเหลือเพียงพอต่อการใช้จ่ายแสดมป์แลกเป็นส่วนลดหรือแลกสินค้าเพื่อให้ลูกค้าสะดวกต่อการตรวจสอบสิทธิการใช้แสดมป์ เป็นต้น

บรรณานุกรม

ASP.NET MVC 3 – View and Razor Engine [online] แหล่งที่มา:

<http://wrach.wordpress.com/2011/09/>

ASP.NET MVC 3 และการ Upload File [online] แหล่งที่มา:

<http://sinsoi.wordpress.com/category/asp-net/>

ASP.NET MVC 3: โครงสร้างของ MVC โดยใช้ Razor [online] แหล่งที่มา:

<http://chavp.wordpress.com/2011/05/29/>

ASP.NET MVC Series I: Foundation of ASP.NET MVC [online] แหล่งที่มา:

<http://greatfriends.biz/webboards/msg.asp?id=123002>

ASP.NET MVC View engine ด้วย Razor [online] แหล่งที่มา:

<http://sinsoi.wordpress.com/tag/engine/>

Complex type [online] แหล่งที่มา: <http://social.msdn.microsoft.com>

Entity Framework 4 Code First และ Data Scaffolding [online] แหล่งที่มา:

http://www.microsoft.com/thailand/msdn/Entity_Framework_4_Code_First.aspx

Entity Framework Development Approaches [online] แหล่งที่มา:

<http://www.ruxcom.com/entity-framework-approaches>

บรรณานุกรม (ต่อ)

How can we inspect the SQL statements that generated from Entity Framework [online]

แหล่งที่มา: <http://greatfriendsbiz.azurewebsites.net/o/how-can-we-inspect-the-sql-statements-that-generated-from-entity-framework/>

Macro Bellinaso. 2551. พัฒนาเว็บไซต์ด้วย ASP.NET 2.0 วิเคราะห์ปัญหา-ออกแบบระบบ-เขียนโปรแกรม. กรุงเทพฯ: ซีเอ็ดยูเคชั่น

MVC WEB FRAMEWORK กับ ASP.NET2 (The Microsoft Entity Framework is an Object Relational Mapping (ORM) Tool) [online] แหล่งที่มา:

<http://www.narisa.com/forums/index.php?app=blog&blogid=27&showentry=1957>

MVC-แนวคิดเกี่ยวกับ MVC [online] แหล่งที่มา: <http://softganz.com/paper/394>

View Engines [online] แหล่งที่มา: http://www.msMvcme.com/2012/09/aspnet-mvc_15.html

ปัญญา ปะสิทธิ์เตสัง. 2546. การเขียนโปรแกรม ASP.NET ด้วย VB.NET และ C#. กรุงเทพฯ: ซีเอ็ดยูเคชั่น

พิรพร หมุนสนิท. 2553. การพัฒนาเว็บไซต์ด้วย XHTML, CSS และ JavaScript. กรุงเทพฯ: เกทีพี

สิทธิชัย รักษาสุข. 2553. พัฒนาเว็บอย่างมืออาชีพด้วย Dreamweaver & jQuery. นนทบุรี: ไอดีซี