

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

๔

ระบบจัดการเอกสารบนเว็บเพจ
Web Page Content Management System

โดย

นางสาวนรีนุช ใจสมานประเสริฐ

รหัส 44067498



H002037

อาจารย์ที่ปรึกษา

ผศ. ดร. ประจวบ วานิชชัชวาล

วัน เดือน ปี.....	27	พ.ค.	2550
เลขทะเบียน.....	02037		
เลขเรียกหนังสือ.....	ศท. ๙๖๗๗๕ ๕๕๔๖		
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."			

11

14/11

รายงานนี้เป็นส่วนหนึ่งของวิชา โครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2546
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	ระบบจัดการเอกสารบนเว็บเพจ
นักศึกษา	นางสาว นรีนุช ใจสมานประเสริฐ
อาจารย์ที่ปรึกษา	ผศ.ดร. ประจวบ วานิชชัชวาล
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

ในปัจจุบันที่ข้อมูลข่าวสารในแต่ละเว็บเพจจำเป็นต้องมีการปรับปรุงเปลี่ยนแปลงเพื่อให้ข้อมูลใหม่หรือทันต่อเหตุการณ์อยู่เสมอ จึงอาจทำให้เกิดความยุ่งยากในการแก้ไขเปลี่ยนแปลงข้อมูลบนเว็บไซต์ที่มีความซับซ้อนหรือมีจำนวนมาก ผู้มีหน้าที่รับผิดชอบข้อมูลบนเว็บไซต์ส่วนใหญ่จึงนิยมใช้ Web Page Content Management System (CMS) เพื่อช่วยในการจัดการ เช่น ควบคุมการเปลี่ยนแปลงของข้อมูล, ควบคุมลำดับการทำงานตาม Workflow, จัดการรูปแบบการนำเสนอ, กำหนดช่วงเวลาที่ทำให้ทำการนำเสนอข้อมูลแต่ละรายการบนเว็บเพจ เป็นต้น โดยการทำงานเหล่านี้ที่ผู้ใช้งานระบบไม่จำเป็นต้องมีความรู้ในด้านการเขียนโปรแกรมภาษาใด ซึ่งเป็นจุดประสงค์ของการจัดทำระบบจัดการเอกสารบนเว็บเพจนี้ขึ้นมา ในบทความนี้จะนำเสนอระบบการจัดการเอกสารบนเว็บเพจที่พัฒนาโดยใช้ ASP.NET เนื่องจากเป็นเทคโนโลยีที่มีการใช้งานเพิ่มมากขึ้นในปัจจุบัน และระบบฐานข้อมูลที่จะนำมาใช้งาน คือ Microsoft SQL Server 2000 ซึ่งผลที่คาดว่าจะได้รับจากระบบที่พัฒนานี้ คือ ระบบการจัดการเอกสารบนเว็บเพจที่ทำงานได้อย่างถูกต้องตามความต้องการของผู้ใช้งาน นอกจากนี้จะทำการพิจารณาถึงข้อจำกัดของระบบที่เกิดขึ้น เพื่อนำไปสู่แนวทางการแก้ไขปรับปรุง และเป็นแนวคิดในการนำไปประยุกต์ใช้งานด้านอื่นๆต่อไป

Title	Content Management System
Student	Miss Nareenuch Jaisamarnprasert
Advisor	Asst. Prof. Prachuab Vanitchatchavan, Ph.D.
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2003

ABSTRACT

Currently, the contents on each website, especially the new websites, need real-time updating in order to serve their clients. The companies that have various categories and contents in their complex website can bring complication as well as delayed updating. A solution to this could be to use "Web Page Content Management System (CMS)" a system with many features. For example, the CMS can manage the workflow of content approval to ensure tasks are completed on time for publishing while maintaining overall content standards. Another advantage of using CMS would be that the website editor need not be fluent in programming. In order to receive the final system for supporting the requirements and increasing the efficiency. This project is developed by ASP.NET which is the powerful technology and has been used more at the present and the database is managed by Microsoft SQL Server 2000. Some limitations maybe found and considered as the next step of further development.

กิตติกรรมประกาศ

โครงการพัฒนาระบบการจัดการข้อมูลบนเว็บเพจ จะไม่สามารถดำเนินการมาจนแล้วเสร็จได้ หากขาดความช่วยเหลือของบุคคลเหล่านี้ ผู้จัดทำจึงใคร่ขอขอบพระคุณเป็นอย่างยิ่ง

ขอขอบพระคุณ บิดา มารดา ที่ให้โอกาสและสนับสนุนทางการศึกษา

ขอขอบพระคุณ ผศ.ดร. ประจวบ วานิชชัชวาล ผู้เป็นอาจารย์ที่ปรึกษา โครงการพัฒนาระบบงานที่กรุณาให้คำปรึกษา แนะนำในด้านต่างๆ

ผู้จัดทำหวังเป็นอย่างยิ่งว่าบทความนี้จะเป็นแนวคิดในการปฏิบัติงานเพื่อสามารถนำไปใช้ประยุกต์กับงานด้านอื่นๆ ได้เป็นอย่างดี

นรินทร์ ใจสมานประเสริฐ

ผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	V
สารบัญภาพ.....	VI
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมา.....	1
1.2 วัตถุประสงค์ในการพัฒนาระบบ.....	2
1.3 ขอบเขตของการพัฒนาระบบ.....	2
1.4 ขั้นตอนการดำเนินการ.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 เทคโนโลยี .NET Framework.....	4
2.2 การนำเสนอ CMS ด้วย .NET Framework.....	8
2.3 Hypertext Transfer Protocol (HTTP).....	14
2.4 Common Gateway Interface (CGI).....	21
3. การวิเคราะห์และออกแบบระบบ.....	23
3.1. โครงสร้างของระบบ.....	23
3.2. การทำงานของระบบ.....	25
3.3. การวิเคราะห์และออกแบบระบบ.....	28
3.4. การออกแบบฐานข้อมูล.....	35
3.5. Data Dictionary.....	37
4. การพัฒนาระบบ.....	32
4.1. การพัฒนาโปรแกรมคอมพิวเตอร์และเครื่องมือที่ใช้.....	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.2. โครงสร้างการทำงานของระบบงาน.....	43
5. บทสรุปและข้อเสนอแนะ.....	56
5.1. ผลการดำเนินการพัฒนาระบบ.....	56
5.2. ประโยชน์ที่ได้รับ.....	56
5.3. ข้อเสนอแนะ.....	57
บรรณานุกรม.....	58
ประวัติผู้เขียน.....	59



สารบัญตาราง

ตารางที่	หน้า
2.1 method ของ HTTP 1.0.....	15
2.2 method ของ HTTP 1.1.....	16
2.3 ค่า HTTP Response ที่ใช้บ่อย.....	19
3.1 ตารางเก็บข้อมูลผู้ใช้งานระบบ.....	37
3.2 ตารางเก็บข้อมูลสิทธิการเข้าถึงข้อมูลในแต่ละหมวดหมู่.....	38
3.3 ตารางเก็บข้อมูลหมวดหมู่ (Categories).....	39
3.4 ตารางเก็บข้อมูลหน้าที่ (Roles).....	39
3.5 ตารางเก็บข้อมูล Content.....	40
3.6 ตารางเก็บข้อมูล Template.....	41
3.7 ตารางเก็บข้อมูลรูปภาพ.....	41
3.8 ตารางเก็บข้อมูล Map รูปภาพให้กับแต่ละ Content.....	42

สารบัญรูป

หน้า

รูปที่

2.1 แพลตฟอร์ม .NET.....	4
2.2 สภาพแวดล้อมของ .NET Framework.....	5
2.3 ขั้นตอนการพัฒนาและใช้งานโปรแกรมใน CLR.....	6
2.4 ความสัมพันธ์ระหว่างผู้ให้บริการ ผู้ใช้บริการ และ UDDI.....	7
2.5 ตัวอย่าง โค้ดของ ASP.NET.....	8
2.6 ตัวอย่าง DataService.asmx.....	9
2.7 การทำงานของ HTTP โปรโตคอล.....	14
2.8 การเชื่อมต่อโดยตรง.....	20
2.9 การเชื่อมต่อผ่านตัวกลาง.....	20
2.10 การทำงานของ CGI.....	21
3.1 โครงสร้างของระบบ CMS.....	23
3.2 ข้อมูลของ Content ในระบบ CMS.....	25
3.3 Workflow ของระบบ CMS.....	26
3.4 Context Diagram ของระบบ CMS	28
3.5 Data Flow Diagram Level 1.....	30
3.6 Data Flow Diagram Level 2 ของ Process 1 (Register).....	31
3.7 Data Flow Diagram Level 2 ของ Process 2 (Login).....	31
3.8 Data Flow Diagram Level 2 ของ Process 3 (Content Information).....	32
3.9 Data Flow Diagram Level 2 ของ Process 4 (Check Content).....	32
3.10 Data Flow Diagram Level 2 ของ Process 5 (Edit User Information).....	33
3.11 Data Flow Diagram Level 2 ของ Process 6 (Deploy Content).....	34
3.12 Data Flow Diagram Level 2 ของ Process 7 (Setup Data).....	34
3.13 Entity – Relationship Diagram.....	36
4.1 หน้าจอสำหรับใส่ข้อมูล UserName และ Password ของผู้เข้าใช้ระบบ.....	45
4.2 หน้าจอสำหรับจัดการข้อมูลสำหรับผู้ใช้งานที่เป็น Editor.....	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
4.3 หน้าจอสำหรับการจัดการข้อมูลสำหรับผู้ใช้งานที่เป็น Approver.....	47
4.4 หน้าจอสำหรับการจัดการข้อมูลสำหรับผู้ใช้งานที่เป็น Administrator.....	48
4.5 หน้าจอแสดงการกำหนดสิทธิการเข้าใช้ในแต่ละหมวดหมู่ให้กับผู้เข้าใช้ระบบ.....	49
4.6 หน้าจอสำหรับการจัดการข้อมูล Content.....	50
4.7 หน้าจอแสดงการนำรูปภาพมาใส่ให้กับ Content.....	51
4.8 หน้าจอแสดงการเลือกรายการที่สนใจขึ้นมาดูในหน้าตรวจสอบ.....	52
4.9 หน้าจอแสดงตัวอย่างการเลือกรายการที่รอการตรวจสอบ.....	53
4.10 หน้าจอแสดงตัวอย่างการเลือกรายการที่รอการนำเสนอ.....	54
4.11 หน้าจอแสดงตัวอย่างการนำเสนอเว็บเพจ.....	55

บทที่ 1

บทนำ

1.1 ความเป็นมา

นับตั้งแต่อินเทอร์เน็ตเข้ามามีบทบาทในการดำเนินธุรกิจ การดำเนินชีวิต และกลายมาเป็นส่วนหนึ่งในชีวิตประจำวัน ทั้งมีส่วนช่วยอำนวยความสะดวกในการทำงานและให้ความรู้ความบันเทิง ดังนั้น ข้อมูลข่าวสารที่ปรากฏบนเว็บเพจจึงจำเป็นที่จะต้องมีการปรับปรุงหรือเปลี่ยนแปลงให้ทันยุคหรือทันต่อเหตุการณ์อยู่เสมอ แต่การปรับปรุงหรือเปลี่ยนแปลงข้อมูลบนเว็บเพจนั้น จำเป็นต้องให้ผู้ที่มีความรู้ทางการเขียน โปรแกรมหรืออย่างน้อยต้องรู้ภาษา HTML จึงจะทำได้ นั่นทำให้เกิดปัญหาหรือเป็นอุปสรรคถ้าในกรณีที่เว็บเพจนั้นๆเป็นเว็บเพจที่นำเสนอข่าวสารที่ต้องทันต่อเหตุการณ์อยู่เสมอ ตัวอย่างเช่น เว็บเพจที่นำเสนอข่าวปัจจุบันทันด่วนเสมือนเป็นหนังสือพิมพ์บนอินเทอร์เน็ต ผู้ที่มีเนื้อหาของข่าวที่ต้องนำเสนอก็คือนักข่าว ดังนั้น ถ้าต้องการนำเสนอข่าวสารแบบปัจจุบันทันด่วน ผู้ที่จะทำการเพิ่มเติมหรือเปลี่ยนแปลงแก้ไขข่าวบนเว็บเพจก็ควรจะเป็นนักข่าวหรือผู้ที่มีหน้าที่รับผิดชอบในข้อมูลข่าวสารนั้น ซึ่งสามารถนำข้อมูลไปลงบนเว็บเพจได้โดยไม่ต้องคำนึงว่าจะต้องเรียนรู้วิธีการเขียน โปรแกรมก่อน เพียงแต่นำข้อมูลไปใส่ไว้เท่านั้นก็สามารถที่จะนำเสนอข้อมูลให้ผู้อื่นดูได้

นอกจากความต้องการในเรื่องการอำนวยความสะดวกในด้านการจัดการข้อมูลข่าวสารแล้ว ส่วนที่สำคัญอีกส่วนหนึ่งที่ต้องคำนึงถึงก็คือ ความถูกต้องของข้อมูลซึ่งควรที่จะทำการตรวจสอบความถูกต้องก่อนนำเสนอบนเว็บเพจ เพื่อลดความผิดพลาดที่อาจเกิดขึ้นได้เสมอ

จากปัญหาและความต้องการดังกล่าวข้างต้น จึงทำให้เกิดแนวความคิดที่จะทำการพัฒนาระบบเพื่ออำนวยความสะดวกและแก้ปัญหาหรืออุปสรรคที่เกิดขึ้น โดยระบบที่พัฒนาขึ้นมาี้ คือ Web Page Content Management System (CMS) ซึ่งกล่าวได้ว่าเป็นระบบที่ช่วยในการจัดการข้อมูลข่าวสาร โดยที่ผู้ใช้งานสามารถสร้าง ควคุม และนำเสนอข้อมูลบนเว็บไซค์ได้โดยตรงผ่านทาง Browser-based Interface ทั้งแบบ Real-Time และกำหนดเป็นตารางเวลา โดยไม่จำเป็นต้องมีความรู้ทางการเขียน โปรแกรมและภาษา HTML เลยก็ได้ เพียงแต่ใช้งานคอมพิวเตอร์และอินเทอร์เน็ต เป็น และทำการศึกษาเรียนรู้วิธีการใช้งานระบบ CMS เท่านั้น นอกจากนี้ส่วนที่สำคัญอีกส่วนหนึ่งที่จำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับระบบนี้ก็คือ การควบคุม Workflow การทำงาน ซึ่งช่วยให้ข้อมูลที่น่าเสนอบนเว็บ-เพจมีความถูกต้อง โดยผ่านการตรวจสอบก่อนที่จะอนุญาตให้นำเสนอข่าวสารนั้นๆ

1.2 วัตถุประสงค์ในการพัฒนาระบบ

- 1.2.1 เป็นระบบที่พัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับผู้ใช้งานในการปรับปรุงหรือเปลี่ยนแปลงข้อมูลข่าวสารบนเว็บเพจ
- 1.2.2 เพื่อควบคุมการทำงานให้เป็นไปตามลำดับขั้นตอน และสามารถทำการตรวจสอบข้อมูลเพื่อลดความผิดพลาดที่อาจเกิดขึ้นได้ในการทำงาน
- 1.2.3 เพื่อให้ข้อมูลที่น่าเสนอบนเว็บเพจมีความถูกต้อง
- 1.2.4 เพื่อเพิ่มประสิทธิภาพในการปรับปรุงหรือเปลี่ยนแปลงข้อมูลบนเว็บเพจ

1.3 ขอบเขตของโครงการ

เมื่อได้ข้อมูลและรายละเอียดจากการศึกษามาวิเคราะห์แล้ว สามารถกำหนดเป็นขั้นตอนในการดำเนินการต่างๆเพื่อให้ได้มาซึ่งระบบที่รองรับการทำงานและสามารถทำงานได้จริงตรงตามวัตถุประสงค์ ซึ่งระบบที่พัฒนามานี้ มีขอบเขตของการพัฒนาระบบ ดังต่อไปนี้

- 1.3.1 การทำงานของระบบจะเป็นในลักษณะของเว็บแอปพลิเคชัน โดยที่ผู้ใช้งานสามารถใช้งานระบบได้ผ่านทางบราวเซอร์
- 1.3.2 ระบบสามารถให้ผู้ใช้งานสามารถทำการเพิ่ม, ลบ และแก้ไขข้อมูลที่ตนเองรับผิดชอบอยู่
- 1.3.3 ระบบสามารถให้ผู้ใช้งานทำการตรวจสอบข้อมูลก่อนที่จะนำข้อมูลที่ผ่านการตรวจสอบแล้วนั้นขึ้นแสดงบนเว็บเพจ
- 1.3.4 ผู้ใช้งานสามารถค้นหาข้อมูลรายการที่ต้องการและติดตามข้อมูลที่ตนเองรับผิดชอบอยู่ได้
- 1.3.5 ระบบสามารถให้ผู้ใช้งานที่เป็นผู้ดูแลระบบสำหรับทำการเพิ่ม, ลบ และแก้ไขข้อมูลของผู้ใช้งานในระบบได้ รวมถึงทำการกำหนดสิทธิหรือสถานะหรือขอบเขตการใช้งานของผู้ใช้งานแต่ละคนด้วย
- 1.3.6 ระบบสามารถให้ผู้ใช้งานจัดการข้อมูลที่จะนำเสนอบนเว็บเพจรวมไปถึงรูปแบบของเว็บเพจ

1.4 ขั้นตอนการดำเนินการโครงการ

- 1.4.1 ศึกษาขั้นตอนการทำงานและสำรวจความต้องการระบบของผู้ใช้งาน โดยการสัมภาษณ์ สํารวจ และการสังเกตการณ์การทำงาน
- 1.4.2 นำข้อมูลที่ได้มาทำการวิเคราะห์ถึงปัญหาและความต้องการที่สามารถตอบสนองได้
- 1.4.3 ทำการออกแบบระบบงาน โดยออกแบบขั้นตอนการทำงาน, ออกแบบแอปพลิเคชัน, ออกแบบหน้าจอให้เหมาะสมสำหรับใช้งาน และออกแบบฐานข้อมูล ให้สามารถตอบสนองความต้องการของผู้ใช้งาน
- 1.4.4 ทำการพัฒนาระบบต้นแบบขึ้นมา ตามที่ได้ออกแบบไว้ในหัวข้อ 1.5.3 และทดลองให้ ผู้ใช้งานได้ใช้งานจริง
- 1.4.5 ทำการสรุปผลจากข้อคิดเห็นและคำแนะนำในการทดลองใช้งานระบบ ตลอดจน การจัดทำเอกสารประกอบการพัฒนาระบบงาน

1.5 ขั้นตอนการดำเนินการโครงการ

- 1.5.1 ศึกษาขั้นตอนการทำงานว่ามีการดำเนินงานอย่างไรและสำรวจความต้องการระบบของ ผู้ใช้งาน โดยการสัมภาษณ์ สํารวจ และการสังเกตการณ์การทำงาน เพื่อใช้เป็นแนวทาง ในการพัฒนาระบบ
- 1.5.2 นำข้อมูลที่ได้มาทำการวิเคราะห์ถึงปัญหาและความต้องการที่สามารถตอบสนองได้
- 1.5.3 ศึกษาเทคโนโลยีต่างๆที่เกี่ยวข้องเพื่อที่จะเลือกเทคโนโลยีที่เหมาะสมและนำมาใช้ในการ พัฒนาระบบที่ทำงานบนเครือข่ายอินเทอร์เน็ต
- 1.5.4 ทำการออกแบบระบบงาน โดยออกแบบขั้นตอนการทำงาน, ออกแบบแอปพลิเคชัน, ออกแบบหน้าจอให้เหมาะสมสำหรับใช้งาน และออกแบบฐานข้อมูล ให้สามารถตอบสนองความต้องการของผู้ใช้งาน
- 1.5.5 ทำการพัฒนาระบบต้นแบบขึ้นมา ตามที่ได้วิเคราะห์และออกแบบไว้ใน และทดลอง ให้ผู้ใช้งานได้ใช้งานจริง
- 1.5.6 ทำการสรุปผลจากข้อคิดเห็นและคำแนะนำในการทดลองใช้งานระบบ ตลอดจน การจัดทำเอกสารประกอบการพัฒนาระบบงาน

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1.6.1 ช่วยอำนวยความสะดวกในการจัดการข้อมูลที่จะนำเสนอบนเว็บเพจ และทำการปรับปรุงและเปลี่ยนแปลงข้อมูลได้รวดเร็วยิ่งขึ้น
- 1.6.2 ช่วยให้การทำงานเป็นไปตามขั้นตอน ซึ่งทำให้ไม่มีข้อมูลใดถูกละเลยหรือไม่ได้รับการตรวจสอบความถูกต้องก่อนนำไปใช้งาน
- 1.6.3 ช่วยลดความผิดพลาดของข้อมูลที่จะนำเสนอบนเว็บเพจ
- 1.6.4 ช่วยลดการทำงานของผู้พัฒนาเว็บไซต์ซึ่งต้องคอยปรับปรุงหรือเปลี่ยนแปลงข้อมูลให้กับทุกฝ่าย ซึ่งงานในการปรับปรุงข้อมูลนี้จะกระจายไปให้กับผู้รับผิดชอบข้อมูลเหล่านั้นแทน
- 1.6.5 ช่วยให้เว็บไซต์มีข้อมูลที่ทันสมัยอยู่เสมอ



บทที่ 2

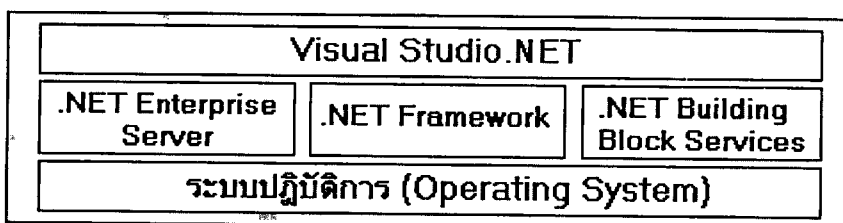
เทคโนโลยีที่เกี่ยวข้อง

2.1 เทคโนโลยี .NET Framework

ก่อนที่จะมีเทคโนโลยี .NET นั้น ทางไมโครซอฟท์มี Windows DNA (Windows Distributed Internet Architecture) ที่ใช้ในการพัฒนาแอปพลิเคชันแบบ 3-Tiers หรือ Distributed Application ซึ่งมี COM (Component Object Model) เป็นเทคโนโลยีที่สำคัญและมี ASP เป็นภาษาสำหรับสร้างไดนามิกเว็บซึ่งจะทำงานอยู่บนเว็บเซิร์ฟเวอร์ IIS ของไมโครซอฟท์ ปัญหาที่พบขณะนั้นก็คือ การสร้างและใช้งาน COM Component ก่อนข้างยุ่งยาก คือต้องมีการลงทะเบียน (Register) ก่อนใช้และหากจะนำออกจากเครื่องก็ต้องมีการยกเลิกการลงทะเบียน (Unregister) COM Component ที่มีหลายเวอร์ชันและการทำงานร่วมกันระหว่าง COM Component หลาย Component ภายในคอมพิวเตอร์เครื่องเดียวกันก็อาจเกิดปัญหาที่ทำให้ไม่สามารถใช้งานแอปพลิเคชันได้ด้วย

นอกจากนี้ ในปัจจุบันภาษามาตรฐานอย่าง XML เป็นที่ยอมรับและใช้งานกันอย่างแพร่หลาย เพราะเป็นภาษาที่มีรูปแบบมาตรฐานในการเก็บข้อมูลเพื่อการแลกเปลี่ยนบนเครือข่าย ไมโครซอฟท์จึงนำเสนอเทคโนโลยี .NET ขึ้นมา โดยใช้ภาษา XML เป็นมาตรฐานในการแลกเปลี่ยนข้อมูลด้วย และไม่ว่าผู้พัฒนาระบบจะพัฒนาซอฟต์แวร์ด้วยภาษาใดก็ตาม ก็สามารถทำงานร่วมกันได้บน .NET แพลตฟอร์ม ซึ่งเป็นข้อดีที่สำคัญของเทคโนโลยี .NET เมื่อใช้ภาษา XML การเรียกใช้ COM Component ข้ามเครือข่ายก็สามารถทำได้ง่าย โดยใน .Net แพลตฟอร์มนี้จะอนุญาตให้ผู้พัฒนาระบบพัฒนา COM Component ไปเป็น Web Service ที่สามารถเรียกใช้งานได้ผ่าน โปรโตคอล HTTP โดยใช้โปรโตคอล SOAP (Simple Object Access Protocol)

ในแพลตฟอร์ม .NET แบ่งออกได้เป็น 3 ชั้น[1] ดังรูปที่ 2.1



รูปที่ 2.1 แพลตฟอร์ม .NET

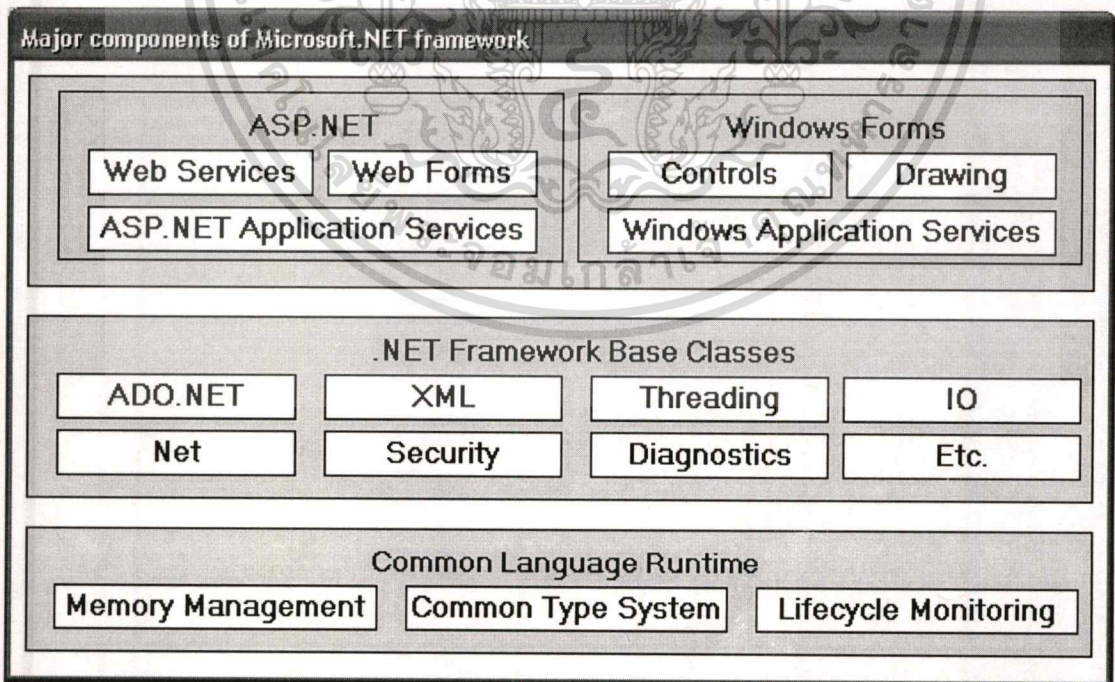
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นล่างสุดคือระบบปฏิบัติการ (Operating System) ในเครื่องเซิร์ฟเวอร์ เช่น Windows NT, Windows 2000 ส่วนในชั้นบนสุดคือ Visual Studio .NET ซึ่งเป็นเครื่องมือที่ช่วยในการพัฒนาโปรแกรมหรือชั้นที่อยู่ตรงกลางจะประกอบด้วย 3 ส่วน ดังนี้

- .NET Enterprise Server คือผลิตภัณฑ์ต่างๆที่ออกแบบมาเพื่อรองรับการพัฒนาแอปพลิเคชันในองค์กร เช่น SQL Server 2000, Exchange 200 เป็นต้น
- .NET Building Block Services คือบริการ Web Services ที่ไมโครซอฟท์สร้างขึ้นมาเพื่อให้ นักพัฒนาใช้ เช่น Microsoft Passport

.NET Framework

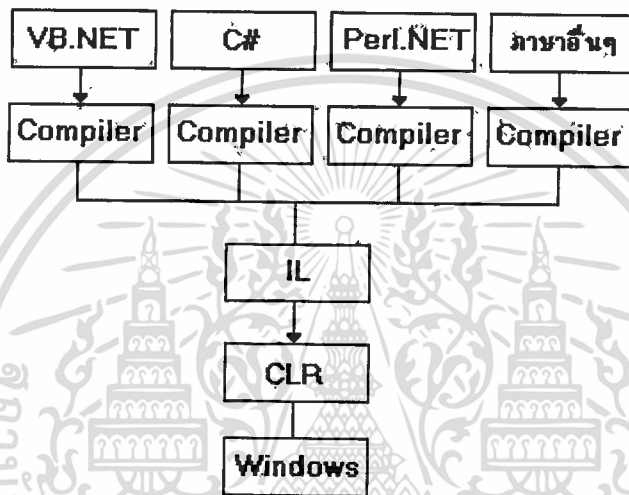
.NET Framework เป็นส่วนที่ผู้พัฒนาระบบเกี่ยวข้องกับมากที่สุด เพราะเป็นสภาพแวดล้อมที่เหมาะสมสำหรับการสร้าง, การนำไปใช้ และรับบริการ XML เว็บ และแอปพลิเคชันอื่นๆ ภายใน .NET Framework แบ่งออกเป็นระดับต่างๆตามลักษณะของการพัฒนาซอฟต์แวร์คือตั้งแต่ระดับระบบปฏิบัติการ (Operating System), ระดับ Component ที่ช่วยในการพัฒนาซอฟต์แวร์ คือ คลาสพื้นฐาน (Base Class Library) และระดับการแสดงผล อย่าง Web Form, Windows Form และ Web Service ดังรูปที่ 2.2



รูปที่ 2.2 สภาพแวดล้อมของ .NET Framework

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Common Language Runtime (CLR) เป็นสถานะแวดล้อมแบบ Runtime ในการจัดการโค้ดที่ได้ถูก Compile แล้ว ซึ่งเปรียบได้กับ Virtual Machine สำหรับภาษาจาวา ซึ่ง CLR สามารถรองรับการพัฒนาโปรแกรมได้หลายภาษา โดยที่ภาษาเหล่านั้นต้องถูก Compile ให้อยู่ในรูปแบบหนึ่งๆ เรียกว่า Intermediate Language (IL) ซึ่งเป็นรูปแบบเฉพาะที่ทำงานใน CLR ได้ ดังรูปที่ 2.3 ข้อจำกัดของ CLR คือทำงานได้เฉพาะบนวินโดวส์แพลตฟอร์ม



รูปที่ 2.3 ขั้นตอนการพัฒนาและใช้งานโปรแกรมใน CLR

ส่วนถัดมาคือ Base Class Library คลาสที่รวมเข้าด้วยกันนั้น ก่อให้เกิดรูปแบบที่แน่นอนในการเข้าถึงฟังก์ชันต่างๆ ของแพลตฟอร์ม จึงช่วยลดความจำเป็นในการเรียนรู้และควบคุมสถาปัตยกรรม API ที่แตกต่างกัน ซึ่งใช้ในการเขียนแอปพลิเคชัน รวมถึงการจัดการเรียกใช้ข้อมูล เช่น เรียกใช้ข้อมูลจากฐานข้อมูล หรือข้อมูลที่อยู่ในรูปของ XML กลุ่มของคลาสที่ใช้กันทั่วไป เช่น คลาส String, I/O, Security, ADO.NET, XML เป็นต้น

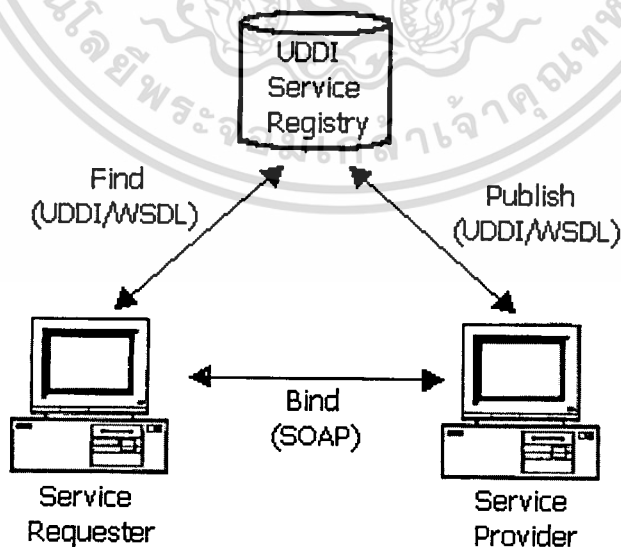
ASP.NET ถูกสร้างอยู่บนพื้นฐานคลาสของ .NET Framework เพื่อเป็น โมเดลแอปพลิเคชันบนเว็บในรูปแบบของชุดคอนโทรลและโครงสร้างพื้นฐานที่ง่ายต่อการสร้างเว็บแอปพลิเคชันและบริการ XML Web ผู้พัฒนาระบบสามารถเรียกใช้ชุดคอนโทรล ASP.NET ซึ่งมีอินเทอร์เฟซพื้นฐานของ HTML สำหรับติดต่อกับผู้ใช้เช่น Text Box, Drop Down Menu และอื่นๆ ซึ่งแม้ผลที่ได้รับจะเป็นเพียงอินเทอร์เฟซในรูปแบบของ HTML บนบราวเซอร์ แต่ในความจริงแล้ว คอนโทรลเหล่านี้ได้ทำงานอยู่บนเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADO.NET เป็น API ที่พัฒนามาจาก ADO เพื่อใช้ติดต่อกับฐานข้อมูลสำหรับเว็บแอปพลิเคชัน ซึ่งในแพลตฟอร์ม .NET ก็คือ Web Form และ Web Service โดยมี XML เป็นตัวกลางในการแลกเปลี่ยนข้อมูล

ส่วนบนสุดคือส่วนที่เป็นแอปพลิเคชัน มีอยู่ 3 แบบด้วยกันคือ Web Services, Web Form และ Windows Form

- Web Form คือการพัฒนาเว็บเพจแบบใหม่ สามารถสร้าง User Interface ได้อย่างง่ายดาย โดยใช้เครื่องมือที่มีอยู่ใน Visual Studio.NET
- Windows Form คือการพัฒนาโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ เปรียบได้กับการพัฒนาโปรแกรมด้วย Visual Basic แบบเดิม
- Web Services คือการสร้าง COM Component หรือโปรแกรมเพื่อให้บริการผ่านโปรโตคอล SOAP (Simple Object Access Protocol) และโปรโตคอล HTTP โดยมี WSDL (Web Services Description Language) เป็นเอกสารที่ใช้อธิบายคุณสมบัติของบริการที่มี ซึ่งผู้ที่ต้องการเรียกใช้งานสามารถค้นหาบริการเหล่านี้ได้ผ่าน UDDI (Universal Description, Discovery and Integration) ผู้ใช้บริการไม่จำเป็นต้องรู้ที่อยู่จริงของ COM Component และโปรแกรม เพียงแค่เรียกใช้งานให้ถูกต้องตามที่อธิบายไว้ในเอกสาร WSDL ความสัมพันธ์ระหว่างผู้ให้บริการ ผู้ใช้บริการ และ UDDI แสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 ความสัมพันธ์ระหว่างผู้ให้บริการ ผู้ใช้บริการ และ UDDI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การนำเสนอ CMS ด้วย .NET Framework

ส่วนสำคัญใน .NET Framework ที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชันของ CMS คือ Web Form และ Web Service เนื่องจาก CMS เป็นระบบที่ใช้งานผ่านทาง Browser ซึ่งเครื่องมือที่ใช้พัฒนาแอปพลิเคชันที่เป็น Web Form และ Web service ก็คือ Visual Studio.NET[2] ภาษาที่ใช้ในการพัฒนาจะเลือกใช้ภาษาใดก็ได้ โดยมี ASP.NET เป็นเครื่องมือที่ช่วยสร้างเว็บเพจ สาเหตุที่ใช้ Visual Stdio.NET เนื่องจาก

- ใน Visual Studio.NET นำเสนอรูปแบบการเขียนโค้ด, การใช้งาน และสิ่งอำนวยความสะดวกในการใช้งานที่มีเพิ่มขึ้น ช่วยให้สามารถพัฒนาระบบได้เร็วขึ้นและดียิ่งขึ้น
- สามารถคอมไพล์แอปพลิเคชันที่สร้างขึ้นมาเป็นคลาสใน .NET และคลาสที่ได้สามารถ Inherit ไปใช้งานได้
- ใช้งานร่วมกับภาษาอื่นได้เช่น C#, Visual Basic.NET และ Jscript.NET เป็นต้น
- สามารถทำงานร่วมกับ XML เพื่อใช้ในการแลกเปลี่ยนข้อมูล
- เรียกใช้ Web Service หรือพัฒนาแอปพลิเคชันที่สร้างขึ้นให้เป็น Web Service ได้

2.2.1 การสร้าง Web Form

ตัวอย่างโค้ดที่ได้จากการสร้าง Web Form ด้วย ASP.NET จะเป็นดังรูปที่ 2.5 ซึ่งไฟล์ที่ได้จะมีนามสกุลเป็น .aspx

```
<%@ Page Language="VB" %>
<script runat="server">
  Sub Button1_Click(sender As Object, e As EventArgs)
    Label1.Text = "Hello, " & TextBox1.Text &
      "! Welcome to the ASP.NET Web Project"
  End Sub
</script>
<html>
<body>
  <form runat="server">
    <p><asp:Label id="Label1" runat="server">Label</asp:Label></p>
    <p><asp:TextBox id="TextBox1" runat="server"></asp:TextBox></p>
```

```

<p><asp:Button id="Button1" onclick="Button1_Click" runat="server" Text="Button">
  </asp:Button></p>
</form>
</body>
</html>

```

รูปที่ 2.5 ตัวอย่างโค้ดของ ASP.NET

จากตัวอย่างโค้ดจะเห็นได้ว่ามีส่วนที่ให้ระบุชนิดของภาษาที่ใช้ ซึ่งภาษาที่เป็น Default คือ “VB” ถ้าต้องการใช้ภาษาอื่นก็ให้ระบุตรง Language ไปเป็นภาษาที่ต้องการ ASP.NET จะมี Server-Side Control เปลี่ยนการทำงานของตัวคอนโทรลจากปกติที่ทำงานบนเครื่องไคลเอ็นต์มาทำงานบนเซิร์ฟเวอร์แทน และสามารถตรวจสอบความถูกต้องของข้อมูลได้ โดยผู้พัฒนาระบบไม่ต้องเขียนสคริปต์มาตรวจสอบความถูกต้องของข้อมูลเอง รูปแบบการเขียนก็เหมือนกับ HTML แต่จะระบุด้วย <asp:> เช่น

```

<asp:Label id="Label1" runat="server">Label
</asp:Label>
<asp:TextBox id="TextBox1" runat="server">
</asp:TextBox>

```

สำหรับการใช้ Web Form สร้างแอปพลิเคชันใน CMS ก็เพียงแค่สร้างเว็บเพจแต่ละหน้าด้วย ASP.NET และติดต่อกับฐานข้อมูลโดยใช้ API ที่เป็น ADO.NET หรือสามารถเรียกใช้ Web Service ที่นำเสนอข้อมูลด้วย XML

2.2.2 การสร้าง Web Service

การสร้าง Web Service สามารถสร้างโดยใช้ Text Editor หรือสร้างโดยใช้ Visual Studio.NET ก็ได้ รูปที่ 2.6 เป็นตัวอย่างการสร้าง Web Service เพื่อนำข้อมูลจากฐานข้อมูลมาแสดงบนเว็บเพจ

```

<%@ WebService Language="VB" Class="DataService" %>
Imports SystemImports System.Data
Imports System.Data.SqlClient

```

```

Imports System.Web.Services

Public Class DataService : Inherits WebService

  <WebMethod(>

    Public Function GetTitleAuthors() As DataSet

      Dim MyConnection As SqlConnection = New
        SqlConnection("server=(local)\NetSDK;
          Database=pubs;Trusted_Connection=yes")

      Dim MyCommand1 As SqlDataAdapter = New
        SqlDataAdapter("select * from Authors",
          MyConnection)

      Dim MyCommand2 As SqlDataAdapter = New
        SqlDataAdapter("select * from Titles", MyConnection)

      Dim DS As New DataSet
      MyCommand1.Fill(DS, "Authors")
      MyCommand2.Fill(DS, "Titles")

      Return DS

    End Function

  </WebMethod(>

  Public Function PutTitleAuthors(DS As DataSet) As Integer

    Return DS.Tables(0).Rows.Count

  End FunctionEnd Class

```

รูปที่ 2.6 ตัวอย่าง DataService.asmx

ไฟล์ที่จะทำเป็น Web Service นั้น จะต้องประกาศฟังก์ชันภายในไฟล์เป็น Public ด้วย จึงจะสามารถเรียกใช้งานผ่านเว็บได้ และไฟล์ที่ได้จะบันทึกเป็นไฟล์นามสกุล .asmx ไว้ที่ Web Server ซึ่งบนเครื่องที่เป็น Web Server จะต้องมีการติดตั้ง .NET Framework SDK อยู่ด้วย

ไฟล์สำคัญอีกไฟล์หนึ่งที่ได้จากการสร้าง Web Service เป็นเอกสาร XML คือ ไฟล์ WSDL ส่วนที่สำคัญในเอกสาร WSDL ที่ผู้ต้องการเรียกใช้บริการควรรู้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนที่บอกชื่อ Web Service ซึ่งในเว็บเซิร์ฟเวอร์หนึ่งๆจะมี Web Service จำนวนกี่บริการก็ได้ จากตัวอย่างจะได้บริการชื่อ DataService ในเอกสาร WSDL ก็จะระบุไว้เป็น `<service name = "DataService">` ตามรูปแบบเอกสาร XML
- ชื่อไฟล์และตำแหน่งไคเรกทอรีที่เก็บไฟล์ที่ทำหน้าที่เป็น SOAP Listener เช่น `<soap:address location= "http://localhost/DataService.asmx" />`
- ชื่อเมธอดของ Web Service เช่น จากตัวอย่าง มีเมธอดชื่อ GetTitleAuthors และ PutTitleAuthors

```
<operation name="GetTitleAuthors">
```

```
<operation name="PutTitleAuthors">
```

- ข้อมูลเกี่ยวกับพารามิเตอร์ของเมธอดและค่าที่ส่งกลับ

```
<message name="GetTitleAuthorsSoapIn">
```

```
<part name = "parameters" element="s0:GetTitleAuthors"/>
```

```
</message>
```

```
<message name="GetTitleAuthorsSoapOut">
```

```
<part name="parameters" element="s0:GetTitleAuthorsResponse" />
```

```
</message>
```

กรณีที่ทำ CMS ให้เป็น Web Service จึงค่อนข้างยุ่งยากกว่าการสร้างเป็น Web Form ธรรมดา แต่มีข้อดีคือ ช่วยลดระยะเวลาในการพัฒนาระบบถ้า Web Service ที่สร้างขึ้นสามารถให้บริการที่ส่วนอื่นภายในระบบหรือผู้อื่นนำไปใช้งานต่อได้ ตัวอย่างเช่น ถ้าภายในระบบ CMS มีหลายเว็บเพจที่จำเป็นต้องใช้การคำนวณ ก็สามารถเรียกใช้บริการ Web Service ที่ให้บริการคำนวณได้โดยแทนที่จะสร้างฟังก์ชันเพื่อการคำนวณขึ้นเองเพื่อให้แต่ละเว็บเพจเรียกใช้งาน หรือ เขียนโปรแกรมเพื่อการคำนวณไว้ในทุกๆเว็บเพจ ข้อดีอีกอย่างคือสามารถเรียกใช้งานผ่านเว็บได้โดยไม่ต้องคำนึงถึงแพลตฟอร์ม, ภาษา และสามารถทำงานผ่านระบบที่มีไฟร์วอลล์ได้ (Firewall) เพราะมีโปรโตคอล SOAP ที่ทำงานอยู่กับโปรโตคอล HTTP ซึ่งไฟร์วอลล์จะเปิดให้การสื่อสารด้วยโปรโตคอล HTTP ผ่านได้อย่างสะดวกอยู่แล้ว

สาเหตุที่นำเทคโนโลยี .Net Framework มาใช้พัฒนาระบบนี้ เนื่องจาก

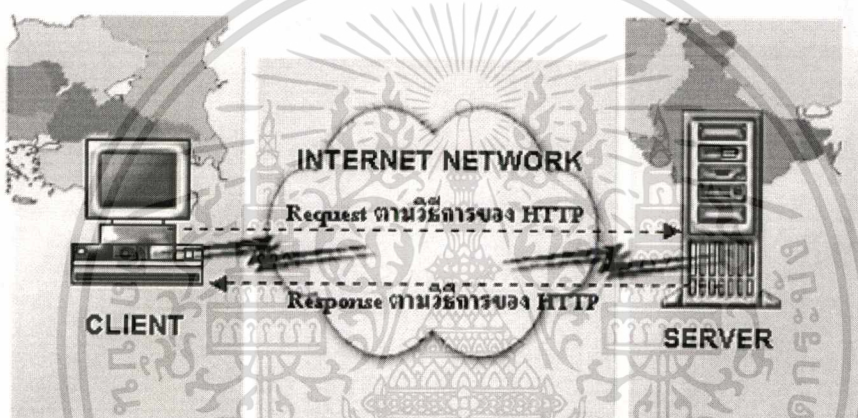
- เป็นเทคโนโลยีที่ใหม่ที่ควรศึกษาไว้
- สนับสนุนการพัฒนาแบบอ็อบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เชื่อมต่อกับระบบอื่นในองค์กรได้โดยไม่ต้องคำนึงถึงภาษาที่ใช้ในการเขียน โปรแกรม เนื่องจาก .NET Framework สามารถใช้ภาษาโปรแกรมใดก็ได้ และใช้กับแอปพลิเคชันที่เขียนขึ้นด้วยภาษาโปรแกรมใดๆ เพื่อรวมเข้าด้วยกันได้
- กรณีที่ผู้พัฒนาระบบนี้ไม่เกี่ยวข้องกับระบบหรือออกจากองค์กรไปแล้วก็ยังสามารถพัฒนาต่อเป็นเฟสถัดไปของระบบ โดยที่ผู้ที่มาพัฒนาต่อสามารถใช้ภาษาที่ตนเองถนัดได้
- เนื่องจาก .NET Framework เน้นการออกแบบโดยใช้ Component ดังนั้นในการพัฒนาจึงให้ความสนใจกับการเขียน business logic ได้อย่างเต็มที่ โดยไม่จำเป็นต้องเขียนคำสั่งของ IDL หรือ Registry ตัวอย่างเช่น ASP.NET ได้รวบรวมคอนโทรลต่างๆ ซึ่งเป็นการทำงานที่โปรแกรมเมอร์ต้องเขียนบ่อย ทำให้ใช้ในการพัฒนาระบบน้อยลง
- .NET Framework ถูกสร้างขึ้นมาเพื่อการนำเสนอซอฟต์แวร์ในรูปแบบของบริการ จึงถูกสร้างขึ้นมาบนมาตรฐานร่วมกันระหว่าง XML และ SOAP ดังนั้น ด้วยการเพิ่มการเรียก เมธอดเข้าไป .NET Framework ก็สามารถแปลงให้กลายเป็นบริการ XML Web ได้อย่างเต็มรูปแบบ ซึ่งเหมาะสำหรับการพัฒนา CMS ที่ต้องใช้ XML เพื่อช่วยในการนำเสนอข้อมูลแลกเปลี่ยนข้อมูลบนเว็บ[3]
- รันแอปพลิเคชันที่น่าเชื่อถือได้มากขึ้น โดย .NET Framework ได้รวมเทคโนโลยี เพื่อให้แอปพลิเคชันมีความเชื่อถือได้มากยิ่งขึ้น ตัวอย่างเช่น หน่วยความจำ, เทรดและ โปรเซส จะถูกจัดการโดย .NET Framework เพื่อให้แน่ใจได้ว่า จะไม่มีเหตุการณ์การรั่วไหลของหน่วยความจำเกิดขึ้น ASP.NET จะติดตามการทำงานของเว็บแอปพลิเคชันที่ทำงานอยู่ และสามารถเริ่มทำงานแอปพลิเคชันขึ้นใหม่ได้โดยอัตโนมัติภายในช่วงเวลาที่กำหนดไว้
- ประสิทธิภาพที่ดีขึ้น .NET Framework จะช่วยปรับปรุงประสิทธิภาพของเว็บแอปพลิเคชันให้ดีขึ้น โดย ASP.NET ได้รวมคุณสมบัติในการคอมไพล์และการทำแคชไว้ ซึ่งช่วยปรับปรุงประสิทธิภาพขึ้น 2-3 เท่าเมื่อเทียบกับแอปพลิเคชัน Active Server Pages เดิมที่มีอยู่

2.3 Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol คือโปรโตคอลที่ใช้สำหรับแลกเปลี่ยนข้อมูลระหว่างเว็บเซิร์ฟเวอร์และเว็บเบราว์เซอร์ผ่านทางเครือข่ายอินเทอร์เน็ต การทำงานของ HTTP จะเป็นลักษณะของการขอ Request และได้ Response ตอบกลับมา โดยฝ่ายไคลเอนต์ ซึ่งในกรณีนี้คือเว็บเบราว์เซอร์จะเป็นฝ่ายขอ Request ไปยังฝ่ายเว็บเซิร์ฟเวอร์หลังจากเว็บเซิร์ฟเวอร์ได้รับ Request มาแล้วก็จะทำการ Response ตอบกลับไปยังเว็บเบราว์เซอร์ การทำงานของ HTTP จะทำงานอยู่บน TCP โดยมีหมายเลขมาตรฐานของ Port ที่ใช้คือหมายเลข 80 โดยมีการทำงานของ HTTP โปรโตคอลดังรูปที่ 2.1



รูปที่ 2.7 การทำงานของ HTTP โปรโตคอล

HTTP ได้มีการกำหนดลักษณะการทำงานรวมถึงรายละเอียดต่างๆ ไว้ใน RFC 2068 และได้มีการพัฒนาแก้ไขเปลี่ยนแปลงใหม่อีกครั้งดังรายละเอียดใน RFC 2616 ซึ่งในปัจจุบันเวอร์ชันล่าสุดของ HTTP คือเวอร์ชัน 1.1 สำหรับรายละเอียดภายใน RFC สามารถหาข้อมูลเพิ่มเติมได้ที่ <http://www.w3c.org/Protocols/>

ใน HTTP ฝ่ายไคลเอนต์ เท่านั้นที่สามารถเป็นฝ่ายเริ่มต้นการติดต่อขอ Request ไปยังฝ่าย Server ได้โดยฝ่าย Server ไม่สามารถเป็นฝ่ายที่จะขอเริ่มการติดต่อไปยังไคลเอนต์ ได้ไม่ว่าจะในกรณีใดๆก็ตาม ส่วนในการขอยุติการติดต่อสื่อสารของ HTTP นั้นสามารถกระทำได้ทั้งไคลเอนต์ และ server ยกตัวอย่างเช่น ไคลเอนต์ ซึ่งในกรณีนี้คือเว็บเบราว์เซอร์สามารถขอยุติการติดต่อสื่อสารได้โดยการกดปุ่ม Stop ที่โปรแกรมเว็บเบราว์เซอร์ หรือ Server สามารถขอยุติการติดต่อสื่อสารได้โดยการ Stop เว็บเซิร์ฟเวอร์

2.3.1 HTTP Requests

HTTP Transaction เริ่มต้นการทำงานขึ้นโดยเว็บเบราว์เซอร์ขอ Request ไปยังเว็บเซิร์ฟเวอร์และจะสิ้นสุด HTTP Transaction เมื่อเว็บเซิร์ฟเวอร์ส่ง Response ตอบกลับมายังเว็บเบราว์เซอร์

HTTP Request ประกอบไปด้วยองค์ประกอบ 3 ส่วนด้วยกันคือ

2.3.1.1 Method – URI – Protocol/Version

ในแต่ละ HTTP Request สามารถที่จะใช้ Method ใดๆก็ได้ ดังที่ระบุไว้ในมาตรฐานของ HTTP ใน HTTP 1.0 Method ที่สามารถใช้ได้จะมีเพียง 3 Method ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 Method ของ HTTP 1.0

Method	คำอธิบาย
GET	GET เป็น Method ที่ใช้บ่อยที่สุดใน HTTP หน้าแรกของ Method นี้ใช้เพื่อดึงข้อมูลจาก URL ที่ระบุมาหลัง Method นี้ ในกรณีที่ URL ที่ระบุมาเป็น CGI Script เช่น php หรือ ASP ข้อมูลที่ได้มาก็คือข้อมูลที่ได้รับมาจากการทำงานของ CGI Script นั้นๆ
HEAD	Method HEAD ให้ข้อมูลกลับมาคล้ายๆ Method GET จะต่างกันเพียง Method HEAD จะได้ผลลัพธ์กลับมาแค่ HTTP Header เท่านั้น ไม่มีส่วนของ Body
POST	จะคล้ายๆ Method GET แต่ Method POST นี้จะใช้ในการส่งข้อมูลในรูปของ Block ซึ่งได้มาจาก HTML Form ซึ่งจะทำให้สามารถส่งข้อมูลที่มีขนาดใหญ่กว่า Method GET ไปยังเว็บเซิร์ฟเวอร์ได้

สำหรับ HTTP เวอร์ชัน 1.1 จะมี Method เพิ่มขึ้นมากกว่า Method ของ HTTP เวอร์ชัน 1.0 ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 Method ของ HTTP 1.1

Method	คำอธิบาย
GET	GET ใช้เพื่อดึงข้อมูลจาก URL ที่ระบุมาหลัง Method นี้
HEAD	Method HEAD ใช้เพื่อขอข้อมูลของ HTTP Header
POST	Method POST จะใช้ในการส่งข้อมูลในรูปของ Block ซึ่งได้มาจาก HTML Form ซึ่งจะทำให้สามารถส่งข้อมูลที่มีขนาดใหญ่มากกว่า Method GET ไปยังเว็บเซิร์ฟเวอร์ได้
OPTIONS	Method OPTIONS ใช้เพื่อสอบถามเว็บเซิร์ฟเวอร์ Resource ที่เว็บเซิร์ฟเวอร์สามารถให้บริการได้
PUT	Method PUT จะใช้ร่วมกับ Method GET โดยจะใช้สำหรับส่งข้อมูลที่ระบุจาก URL ขึ้นไปยังเว็บเซิร์ฟเวอร์คล้ายกับคำสั่ง PUT ของ FTP โดยมากจะใช้ Method PUT ในการ Upload File ขึ้นไปยังเว็บเซิร์ฟเวอร์
DELETE	Method DELETE ใช้สำหรับลบเอกสารออกจากเว็บเซิร์ฟเวอร์ โดยเอกสารที่จะลบจะถูกระบุอยู่ใน URI ที่ส่งมาตามหลัง Method นี้
TRACE	Method TRACE จะใช้ในลักษณะคล้ายกับคำสั่ง Traceroute ของ UNIX จะมีประโยชน์มากสำหรับใช้ในการวิเคราะห์ปัญหาที่อาจเกิดขึ้นของการใช้งานเว็บในเครือข่ายที่ซับซ้อน

ตัวอย่างของ HTTP request แสดงได้ดังนี้

GET /http/form.php HTTP/1.1

Accept: text/plain; text/html

Accept-Language: th

Connection: Keep-Alive

Host: www.sawasdee.com

Referer: http://www.sawasdee.com/http/index.html

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)

Content-Length: 25

Content-Type: application/x-www-form-urlencoded

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Accept-Encoding: gzip, deflate

Name=Nareenuch&Email=Secret

จะเห็นว่า Method – URI – Protocol/Version ซึ่งเป็นสิ่งที่ใช้ในการขอ Request จะอยู่ในบรรทัดแรกของ Request

GET /http/form.php HTTP/1.1

โดย GET คือ Method ของการ request, /http/form.php คือ URI และ HTTP/1.1 คือส่วนของ Protocol/Version

URI คือส่วนที่บอกถึงตำแหน่งของ resource บนเซิร์ฟเวอร์ และจะเริ่มต้นด้วยเครื่องหมาย / สำหรับประเภทของ URI สามารถหารายละเอียดเพิ่มเติมได้ใน RFC 2396 (<http://www.ietf.org/rfc/rfc2396.txt>)

Protocol Version คือเวอร์ชันของ HTTP Protocol ที่ใช้ในการติดต่อกับเซิร์ฟเวอร์

2.3.1.2 Request Header

Request Header ประกอบด้วยข้อมูลเกี่ยวกับสถานะแวดล้อมของไคลเอ็นต์ และส่วนของ Body ของ Request ยกตัวอย่างเช่น บราวเซอร์ของไคลเอ็นต์ ว่าใช้อะไรเป็นเว็บเบราว์เซอร์ หรือ ภาษาที่ไคลเอ็นต์ ใช้ว่าเป็นภาษาใด และยังประกอบไปด้วยจำนวนความยาวของ Entity Body ที่ถูกส่งตามมาด้วย โดย Request Header จะถูกแบ่งออกจาก Entity body โดย Carriage Return และ Linefeed (CRLF)

2.3.1.3 Entity Body

Entity Body คือส่วนที่ตามหลัง Request Header มา โดยมี CRLF เป็นตัวคั่นกลางระหว่าง request header กับ Entity Body โดยในส่วนของ Entity Body นี้จะใช้ในการส่งข้อมูลกลับไปยังเว็บเซิร์ฟเวอร์ ในตัวอย่างของ HTTP Request ในหัวข้อที่ 2.2.1 Entity Body คือ Name=Nareenuch&Email=Secret

2.3.2 HTTP Responses

HTTP Response จะเป็นส่วนที่เว็บเซิร์ฟเวอร์ตอบกลับมาหลังจากได้ Request จาก ไคลเอนต์ จะคล้ายๆกับ HTTP Request คือ HTTP Response จะประกอบไปด้วยส่วนประกอบ 3 ส่วน ได้แก่

- Protocol - Status Code – Description
- Response Headers
- Entity Body

ตัวอย่างของ HTTP Response มีดังนี้

HTTP/1.1 200 OK

Date: Tue, 10 Dec 2002 11:42:33 GMT

Server: Apache/1.3.27 (Unix) PHP/4.2.3

Last-Modified: Thu, 19 Dec 2002 15:53:37 GMT

ETag: "40a2d-65-3e01eb81"

Accept-Ranges: bytes

Content-Length: 101

Connection: close

Content-Type: text/html

<html>

<head>

<title>CMS Project</title>

</head>

<body>

CMS Project Detail

</body>

</html>

บรรทัดแรกของ Response Header จะคล้ายกับ Request Header ก็จะบอกว่าใช้ HTTP เวอร์ชัน 1.1, สถานะของ Request สมบูรณ์ (200 คือค่าของ HTTP ที่บอกว่า Success) และบอกว่าทุกอย่างทำงานได้ปกติ (OK) หลังจากนั้นก็จะตามมาด้วย Response Header และตามมาด้วย Entity เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Body ซึ่งก็คือ HTML โดยในส่วนของ Response Header กับ Entity Body จะถูกแบ่งออกจากกัน โดย Carriage Return/Linefeed (CRLF) สำหรับค่าของ HTTP Response Code ที่ใช้บ่อยสามารถแสดงได้ดังตารางที่ 2.3

ตารางที่ 2.3 ค่า HTTP Response ที่ใช้บ่อย

Status Code	คำอธิบาย
200	(OK)
301	(Moved Permanently) ใช้บอกให้ไคลเอ็นต์ Link ไปยัง URL ใหม่
400	(Bad Request)
401	(Unauthorized)
403	(Forbidden) ในกรณีที่เซิร์ฟเวอร์ถาม User และ Password ไปยังไคลเอ็นต์แล้วไคลเอ็นต์ ตอบผิดเซิร์ฟเวอร์จะตอบ Response Code นี้กลับมา
404	(Not Found) ใช้เมื่อเอกสารที่ Request ไปใน URL ไม่ปรากฏอยู่ในเซิร์ฟเวอร์
500	(Internal Error) ถ้าเซิร์ฟเวอร์มีข้อผิดพลาดเกิดขึ้นเช่น CGI Script บนเซิร์ฟเวอร์ทำงานผิดพลาด เซิร์ฟเวอร์จะตอบ Response Code นี้กลับมา
502	(Timed out)
503	(Service Unavailable) เมื่อเว็บเซิร์ฟเวอร์รับโหลดหนักๆและไม่สามารถประมวลผลต่อไปได้ เซิร์ฟเวอร์จะตอบ Response Code นี้กลับมา

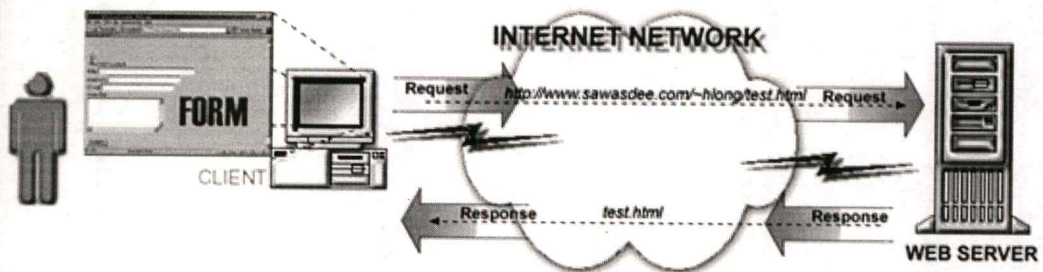
2.3.3 การเชื่อมต่อของไคลเอ็นต์ และ เซิร์ฟเวอร์ ใน HTTP

ในส่วนของเครือข่ายหรือ Network ในระบบเครือข่ายเวิร์ลไวด์เว็บ มีการเชื่อมต่อระหว่างไคลเอ็นต์ และเซิร์ฟเวอร์ ได้ 2 ลักษณะ คือ

2.3.3.1 การเชื่อมต่อโดยตรง

ลักษณะการทำงานคือ ไคลเอ็นต์ จะติดต่อกับเซิร์ฟเวอร์ หรือผู้ให้บริการ โดยตรง ซึ่งส่วนของ ไคลเอ็นต์ จะเรียกว่า User Agent โดยมีบราวเซอร์ทำหน้าที่นี้ให้ และส่วนเซิร์ฟเวอร์ จะเรียกว่า “Origin” จะทำงานกับบราวเซอร์หรือ User Agent นี้โดยตรง ดังแสดงในรูปที่ 2.2

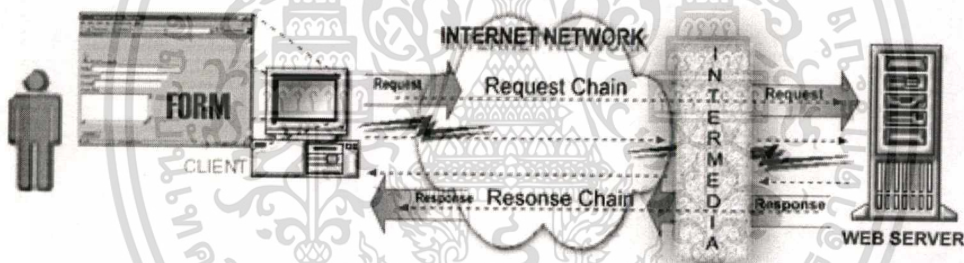
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การเชื่อมต่อโดยตรง

2.3.3.2 การเชื่อมต่อผ่านตัวกลาง

ลักษณะการติดต่อแบบนี้ ส่วนของ User Agent ไม่สามารถติดต่อกับ Origin ได้โดยตรง นั่นคือ ต้องติดต่อผ่านตัวกลางทุกครั้งที่มีการร้องขอบริการ และการตอบสนองก็จะต้องผ่านตัวกลางเช่นกัน ดังนั้น การร้องขอ หรือการตอบสนอง จะมีลักษณะเป็นเหมือนลูกโซ่ โยงผ่านเป็นช่วงๆ เรียกว่า Request Chain / Response Chain



รูปที่ 2.9 การเชื่อมต่อผ่านตัวกลาง

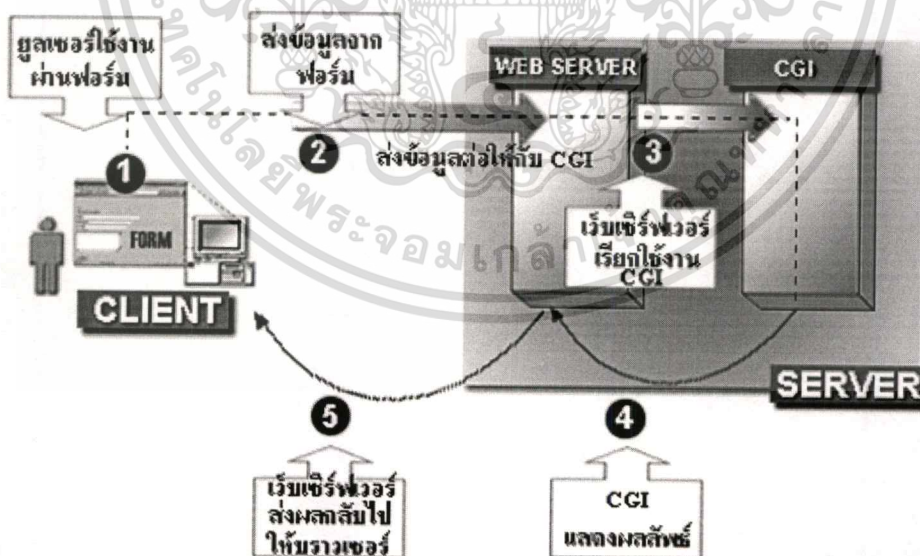
ประเภทของตัวกลาง (Intermedia) ตามข้อกำหนดของ HTTP มี 3 ประเภทคือ

- Tunnel ทำหน้าที่เชื่อมต่อเท่านั้น ตัวกลางนี้จะไม่มีหน้าที่หรืออำนาจในการเปลี่ยนข้อมูลที่วิ่งผ่าน
- Proxy ส่วนนี้สามารถปรับปรุงรายละเอียด มีการประยุกต์ใช้งานได้ทั้ง 2 ส่วนคือ โคลเ็นต์ และเซิร์ฟเวอร์
- Gateway ส่วนนี้มักทำหน้าที่เชื่อมต่อ ในกรณีที่ไม่สามารถติดต่อหรือใช้งานเชื่อมต่อ กับตัว เว็บเซิร์ฟเวอร์ได้โดยตรง

ตัวกลางที่เรามักพบบ่อยในการใช้งานคือ Proxy ซึ่งมักนำมาติดตั้งเพื่อทำหน้าที่เป็น Cache Server หรือเป็น FireWall

2.4 Common Gateway Interface (CGI)

เมื่อเกิดระบบเครือข่ายเวิร์ลไวด์เว็บใช้งานจนเป็นที่นิยมดังเช่นปัจจุบัน หลายๆเว็บไซต์ เริ่มต้องการนำเสนอข้อมูลภายในองค์กร ที่เคยใช้งานกับโปรแกรมประยุกต์ของตนภายในองค์กร ผ่านเว็บเพจ หรือ โฮมเพจ (HomePage) ของตน จึงเกิดปัญหาว่าจะสามารถทำ อย่างไร ทั้งนี้เพราะทั้งสองแอปพลิเคชัน อยู่คนละส่วนกัน และวิธีการทำงานก็แตกต่างกันอย่างสิ้นเชิง ทางออกคือการพัฒนาแอปพลิเคชัน ในลักษณะเหมือนโปรแกรมประยุกต์ที่องค์กรใช้งาน อยู่ โดยอาศัยหลักการของ CGI ในการพัฒนา แต่ณี่ยังเป็นเพียงแค่จุดเริ่มต้น ของความต้องการ เท่านั้น เพราะปัจจุบันเราจะเห็นว่า มีแอปพลิเคชันหลากหลายรูปแบบบนระบบเว็บ เช่น การให้บริการค้นหางาน, การให้บริการความช่วยเหลือแบบออนไลน์ เป็นต้น ซึ่งแอปพลิเคชันเหล่านี้เกิดจากความต้องการที่หลากหลาย และต่างความคิด รวมไปถึง วิสัยทัศน์ของแต่ละคนในการที่คิดประยุกต์ และร่วมสร้างกิจกรรมต่างๆ บนระบบเว็บ จนทำให้ การใช้งานระบบเว็บนี้กลายเป็นส่วนสำคัญหลักของเครือข่ายอินเทอร์เน็ตในปัจจุบันไปเสียแล้ว และด้วยความสามารถของหลักการ CGI นี้เองทำให้หลายๆ องค์กรต้องการนำมาประยุกต์ใช้ ในองค์กรจนเกิดคำที่ว่า “แอปพลิเคชันในอนาคต คือแอปพลิเคชันที่ใช้งานผ่านบราวเซอร์ หรือ ใช้งานภายใต้พื้นฐานเว็บ (Web - based หรือเรียกว่า Web - based Application)”



รูปที่ 2.10 การทำงานของ CGI

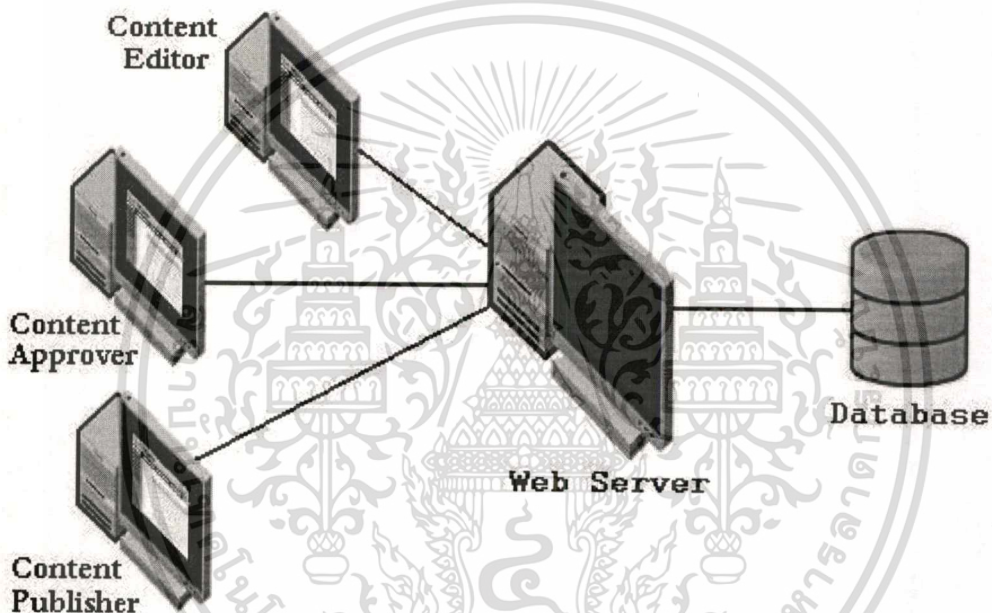
จากรูปที่ 2.10 เราจะมาให้ความหมายว่า อะไรคือ CGI จริงๆ แล้ว CGI ก็คือ หลักการหรือวิธีการของการพัฒนาแอปพลิเคชัน ที่ทำหน้าที่เหมือนประตู (Gateway) เชื่อมโยงการติดต่อ กับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานอื่นๆ เพื่อให้เกิดการทำงานที่หลากหลายในการใช้งาน โดยอาศัยพื้นฐานของระบบ เว็บ หรือจะกล่าวได้ว่าทำงานควบคู่กับเว็บเซิร์ฟเวอร์ เพราะบราวเซอร์ไม่สามารถติดต่อส่วนอื่นๆ โดยตรงได้ เช่น จะติดต่อกับฐานข้อมูล เป็นต้น จำเป็นต้องติดต่อผ่านเว็บเซิร์ฟเวอร์ ไปยังส่วน ของ CGI โดยเรามักเรียกว่า “CGI โปรแกรม” หรือ “CGI แอปพลิเคชัน” หรือ “เว็บแอปพลิเคชัน” ก็ได้ ด้วยเหตุนี้เองเราจึงเห็นว่าจริงๆ แล้ว CGI แอปพลิเคชัน หรือ แอปพลิเคชัน ที่พัฒนาตาม แนวทาง CGI เป็นแอปพลิเคชันประเภท เซิร์ฟเวอร์ แอปพลิเคชัน (Server Application) หรือ แอปพลิเคชันที่ ทำงานอยู่ที่ฝั่งเซิร์ฟเวอร์ โดยมีส่วนที่ทำหน้าที่ติดต่อกับผู้ขอใช้บริการ หรือ ไคลเอนต์ (Client) คือ เว็บเซิร์ฟเวอร์ และไคลเอนต์ใช้งานผ่านเว็บบราวเซอร์ ข้อดีของเซิร์ฟเวอร์ แอปพลิเคชันที่เห็นได้ ชัดคือ การปรับปรุงหรือเปลี่ยนเวอร์ชันจะทำได้ง่าย โดยไม่ต้องแจกจ่าย ให้ผู้ใช้งานทุกครั้งแต่ สามารถดูแลปรับปรุงได้ที่เซิร์ฟเวอร์โดยตรง พอมีวิธีการของ CGI เกิดขึ้น ปัจจุบันเราจึงได้เห็นรูปแบบของโฮมเพจที่เปลี่ยนไปจากสมัยที่เริ่มต้นของอินเทอร์เน็ตที่เคยเป็นแค่ “Static Hypermedia Document” คือเอกสารที่แสดงโดยไม่มีการเปลี่ยนแปลง ไปเป็นเอกสาร ที่สามารถเปลี่ยนแปลงรูปแบบได้ ตลอดจนเห็นเป็นโฮมเพจ ที่สามารถโต้ตอบหรือเป็น อินเตอร์แอคทีฟ (Interactive) เหมือนส่วนของอินเตอร์เฟซ (Interface) ของ CGI แอปพลิเคชัน ที่แปรเปลี่ยนตลอดเหมือนกับการ ใช้งานโปรแกรมประยุกต์นั่นเอง

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 โครงสร้างของระบบ



รูปที่ 3.1 โครงสร้างของระบบ CMS

จากรูปที่ 3.1 จะเห็นว่า ผู้ที่เกี่ยวข้องกับระบบนี้จะแบ่งออกเป็น 4 กลุ่ม ซึ่งจากรูปแสดงโครงสร้างของระบบสามารถอธิบายรายละเอียดในหน้าที่ของแต่ละกลุ่มได้ดังนี้

1) Content Editor เป็นผู้ที่น่าข้อมูล Content เข้าสู่ระบบ คอยทำการเพิ่ม, ลบ และแก้ไขข้อมูลแต่ละรายการ โดยผู้ที่สามารถนำข้อมูลเข้าระบบได้ จะต้องมีชื่อหรือหมายเลขประจำตัวและรหัสผ่านสำหรับการเข้าสู่ระบบ ซึ่งจะทำการเข้าสู่ระบบจากเครื่องคอมพิวเตอร์จากที่ใดก็ได้ นอกจกานำข้อมูลเข้าแล้ว ยังมีรายละเอียดอื่นๆอีก เช่น สามารถเรียกดู ข้อมูลของตนเองได้, ทำการแก้ไขหรือลบข้อมูลได้ และทำการกำหนดสถานะให้กับ Content แต่ละรายการ เป็นต้น ทั้งนี้ Content Editor แต่ละคนจะสามารถทำการแก้ไขหรือทำงานกับข้อมูลของตนเองได้เท่านั้น ไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเข้าไปแก้ไขข้อมูลของ Content Editor คนอื่นได้ เมื่อข้อมูล Content รายการที่เสร็จเรียบร้อยแล้ว จะถูกทำการยืนยัน (Submit) เพื่อส่งให้ Content Approver ทำการตรวจสอบ จะเป็น Content ล่าสุดหรือใหม่สุด และ Content ที่ได้รับการยืนยันแล้วจะไม่สามารถถูกทำการแก้ไขหรือลบได้อีก นอกจากนี้จะถูกส่งคืนกลับมาจาก Content Approver เพื่อให้ทำการแก้ไข

นอกจากนี้ ข้อมูลที่เกี่ยวข้องซึ่ง Content Editor จะต้องระบุให้กับ Content แต่ละรายการด้วย คือ การกำหนดคุณสมบัติเบื้องต้นให้กับข้อมูลแต่ละรายการ เช่น กำหนดช่วงเวลาที่จะนำขึ้นแสดงบนเว็บเพจ, กำหนดสถานะให้กับรายการนั้นๆ และกำหนดว่ารายการไหนควรอยู่ในหัวเรื่องใด ซึ่งเป็นการจัดหมวดหมู่ให้กับข้อมูล เป็นต้น

2) Content Approver เป็นผู้ทำการตรวจสอบ (Approve) ข้อมูลรายการที่ Content Editor นำเข้าสู่ระบบ โดยระบบจะมีการจัดการแจ้งให้ Content Approver ทราบทางอีเมลว่ามีรายการใดที่ต้องตรวจสอบบ้าง สิ่งที่ Content Approver เห็น จะเป็นรายการที่แสดงตามลักษณะ Template ที่กำหนดไว้

- ถ้าผ่านการอนุมัติ รายการนั้นก็ส่งต่อไปยัง Content Publisher เพื่อกำหนดวันเวลาที่จะนำเสนอรายการนั้นบนเว็บเพจ
- ถ้าไม่ผ่านการอนุมัติ ก็จะแจ้งไปยัง Content Editor เพื่อให้ทำการแก้ไขหรือลบข้อมูลรายการนั้นตามที่ Content Approver แจ้งมาในรายละเอียดแนบ หรือผู้ตรวจสอบจะทำการยกเลิก Content นั้นได้เลย นั่นคือข้อมูล Content รายการนั้นจะไม่มี การแก้ไข โดยจะลบทิ้งหรือเก็บเป็นข้อมูลประวัติไว้ในฐานข้อมูล ทั้งนี้ ขึ้นอยู่กับ Content Approver จะเป็นผู้พิจารณากำหนด

นอกจากจะทำหน้าที่ในการตรวจสอบข้อมูลแล้ว ผู้ที่เป็น Content Approver ยังสามารถทำการแก้ไขเปลี่ยนแปลงข้อมูลที่อยู่ในความรับผิดชอบของตนเอง หรือทำหน้าที่ได้เช่นเดียวกับ Content Editor คือทำการสร้างข้อมูล Content ขึ้นมาเอง ซึ่งหมายความว่า สิทธิในการใช้งานระบบของ Content Approver จะมีมากกว่า Content Editor นั่นเอง

3) Content Publisher เป็นผู้ดูแลในขั้นตอนสุดท้ายก่อนนำข้อมูลขึ้นแสดงบนเว็บ โดยตรวจสอบความถูกต้องของการกำหนดตารางเวลาของข้อมูลแต่ละรายการในแต่ละหมวดหมู่ และเป็นผู้จัดการดูแลหมวดหมู่และทำการแบ่งประเภทหมวดหมู่ของข้อมูลไว้ให้

นอกจากนี้ จะมีหน้าที่กำหนดรูปแบบการนำเสนอหรือ Template เพราะข้อมูลที่แสดงบนเว็บเพจจะแสดงตามรูปแบบที่ Template Designer ออกแบบมา ซึ่งขณะที่ผู้ใช้งานคนอื่นทำงานก็จะ

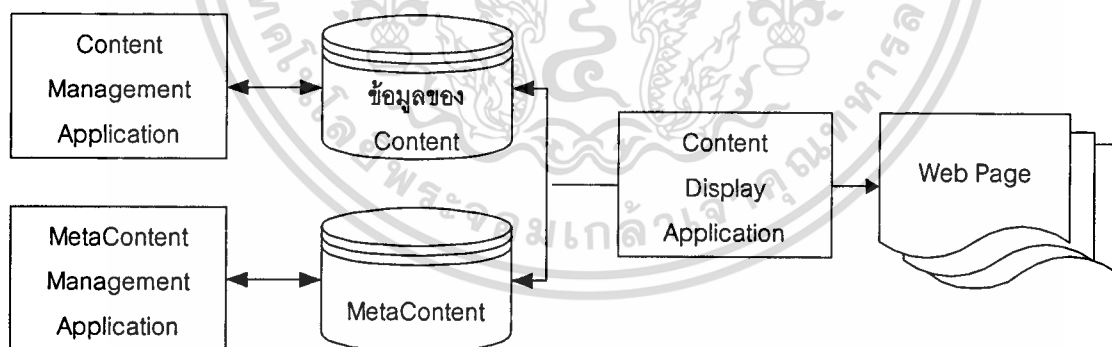
เห็นรูปแบบตามที่ถูกกำหนดไว้ตรงกัน และเมื่อต้องการให้ทำการแก้ไข Content Approver จะทำการแจ้งกลับไปยัง Template Designer ทางอีเมล

4) Template Designer เป็นผู้ออกแบบ Template หรือรูปแบบที่นำเสนอ ซึ่งก็คือของเว็บเพจนั่นเอง โดยจะต้องมีการทำงานร่วมกันหรือเกี่ยวข้องกับผู้พัฒนาเว็บ นอกจากนี้ อาจต้องจัดการแก้ไขรูปภาพที่จะนำขึ้นแสดงบนแต่ละหน้าของ Content เมื่อได้รับแจ้งให้แก้ไขด้วย ซึ่งส่วนนี้จะอยู่นอกเหนือระบบที่จัดทำขึ้น

เมื่อพิจารณาข้อมูลหลักที่สำคัญในการสร้าง Content ภายในระบบ สามารถสรุปได้ดังนี้

ข้อมูล Content 1 รายการจะประกอบด้วยข้อมูล 2 ส่วน คือ ข้อมูลที่เป็นเนื้อหาของ Content และข้อมูลเกี่ยวกับคุณสมบัติของ Content นั้น หรืออาจเรียกได้ว่า MetaContent

MetaContent เป็นข้อมูลที่บอกรายละเอียดเกี่ยวกับตัว Content นั้นๆ เช่น สถานะปัจจุบันของ Content, รูปแบบการนำเสนอ, หัวข้อเรื่องของ Content นั้นว่าอยู่ภายใต้หัวข้อเรื่องใดหรือคอลัมน์ใด เป็นต้น ดังนั้น ภายในระบบจะต้องประกอบด้วยแอปพลิเคชันที่ทำการจัดการข้อมูล 2 ส่วนนี้ และเมื่อจะนำเสนอข้อมูลรายการใด ก็ให้นำข้อมูล 2 ส่วนนี้มาประกอบเข้าด้วยกัน โดยมีแอปพลิเคชันที่จัดการให้ ดังแสดงในรูปที่ 3.2

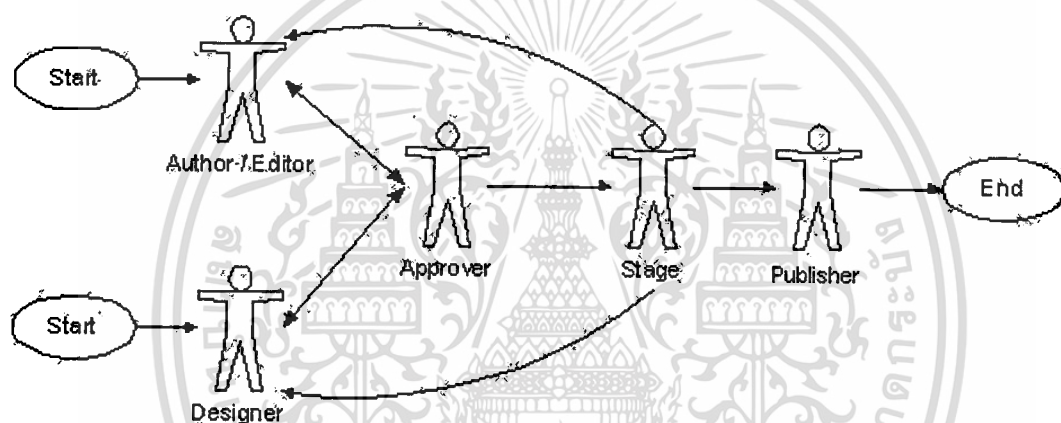


รูปที่ 3.2 ข้อมูลของ Content ในระบบ CMS

3.2 การทำงานของระบบ

เนื่องจาก Content ทุกรายการที่กำลังทำงานอยู่จะต้องมีสถานะปัจจุบันเพื่อใช้สำหรับบอกให้รู้ว่าขณะนี้ ข้อมูลแต่ละรายการอยู่ในขั้นตอนใด เช่น อยู่ในขั้นตอนรอการอนุมัติ หรืออยู่ในขั้นตอนถูกส่งกลับและแจ้งไปยังผู้รับผิดชอบเพื่อให้ทำการแก้ไข เป็นต้น จึงจำเป็นที่จะต้องนำ Workflow เข้ามาช่วยจัดการควบคุมขั้นตอนการทำงาน เอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Workflow สำหรับระบบนี้ก็คือ การจัดงานต่างๆที่จะต้องทำในการสร้าง Content ให้เป็นลำดับขั้นตอนแบบ Sequential สำหรับงานที่ต้องมีการทำเป็นลำดับขั้นตอนก่อนหลังหรือแบบ Parallel สำหรับงานที่สามารถทำไปพร้อมกันหรือทำคู่ขนานกันไปได้ เช่น Content Author และ Content Editor จะทำการสร้างและแก้ไข Content ในขณะที่ Designer กำลังทำการสร้าง Template หรือรูปแบบของหน้าเว็บเพจปัจจุบัน ซึ่งทั้ง 2 งานนี้สามารถทำไปพร้อมกันได้ โดยไม่ต้องรอให้งานใดงานหนึ่งเสร็จก่อน ส่วน Content Approver จะสามารถทำงานได้ก็เมื่อได้ Content และ Template แล้ว ภายใน Workflow มีส่วนประกอบดังรูปที่ 3.3



รูปที่ 3.3 Workflow ของระบบ CMS

- ขั้นตอนการสร้าง Content คือส่วนของผู้ใช้งานที่รับผิดชอบทำการสร้างและทำการแก้ไขเปลี่ยนแปลงข้อมูล Content ซึ่งจะถูกแก้ไขโดยผู้รับผิดชอบหรือผู้ที่มีสิทธิเข้าไปทำการแก้ไข เมื่อได้รายการ Content ที่เสร็จสมบูรณ์แล้วก็จะแจ้งไปยัง Content Approver เพื่อให้ Content Approver ทำการตรวจสอบ ในการแก้ไขก่อนที่จะมีการยืนยันรายการนั้นเพื่อส่งไปยัง Content Approver โดยเมื่อมีการแก้ไขข้อมูลและทำการบันทึกข้อมูลรายการนั้นเข้าระบบจะกำหนดเป็นเวอร์ชันถัดไปให้โดยอัตโนมัติ ดังนั้น แต่ละ Content อาจมีได้หลายเวอร์ชัน ซึ่งเมื่อรายการใดถูกทำการยืนยันแล้วจะใช้เป็นเวอร์ชันล่าสุด ซึ่งจะสามารถนำมาใช้งานได้
- ขั้นตอนการออกแบบ Template สำหรับรูปแบบการนำเสนอเว็บเพจ คือส่วนของผู้ใช้งานที่ทำการสร้าง Template จากรูปจะเห็นได้ว่าขั้นตอนนี้จะสามารถทำคู่ขนานไปกับขั้นตอนการสร้าง Content

- Content หรือ Template ที่ถูกยืนยันแล้ว จะถูกส่งมายัง Content Approver เพื่อรอการตรวจสอบ และจะไม่สามารถถูกทำการแก้ไขหรือลบทิ้งได้ ไม่ว่าผู้ที่เข้าสู่ระบบจะเป็นเจ้าของ Content นั้นหรือเป็นผู้ที่ทำการตรวจสอบ
- ขั้นตอนการตรวจสอบ Content และ Template เป็นหน้าที่ของ Content Approver ที่จะทำการตรวจสอบก่อนนำ Content ขึ้น หรือส่งกลับไปยัง Content Editor หรือ Content Designer เพื่อให้ทำการแก้ไข
- หลังจากผ่านการตรวจสอบแล้วก็จะเข้าสู่ Stage ซึ่งเป็นขั้นตอนการนำเสนอและเป็นการบอกให้รู้ว่ามี Content รายการใดบ้างที่พร้อมนำขึ้นแสดงบนเว็บเพจได้
- ขั้นตอนการนำเสนอ คือส่วนของ Content Publisher ที่จะนำ Content ที่เสร็จสมบูรณ์แล้วขึ้นแสดงบนเว็บเพจ โดยตรวจสอบความถูกต้องของการกำหนดช่วงเวลาและเวลาของแต่ละรายการที่ทางผู้รับผิดชอบจะมาอีกครึ่งหนึ่ง และสามารถทำการแก้ไขช่วงเวลาได้ด้วย ซึ่งระบบจะต้องมีการตรวจสอบว่า แต่ละรายการจะต้องไม่มีช่วงเวลาที่นำเสนอซ้ำซ้อนกันในคอลัมน์หรือหัวข้อเรื่องเดียวกัน แต่อาจมีช่วงวันและเวลาที่ไม่มีรายการใดๆเลยก็ได้ ซึ่งระบบจะกำหนดให้มีการแจ้งเตือนไปยัง Content Publisher เมื่อไม่มีการนำเสนอข้อมูล Content รายการใดๆบนเว็บเพจว่ามีในคอลัมน์หรือหัวข้อเรื่องใดบ้าง
- Content แต่ละรายการที่ครบวงจรการทำงานตาม Workflow แล้วจะถูกเก็บไว้ในฐานข้อมูล โดยไม่มีการลบทิ้ง เพื่อประโยชน์ในการค้นหาหรือเรียกขึ้นมาดูในอนาคตได้ ซึ่งข้อมูลในแต่ละเวอร์ชันของรายการหนึ่งๆจะถูกเก็บไว้ทั้งหมด

ข้อดีของการมี Workflow คือ

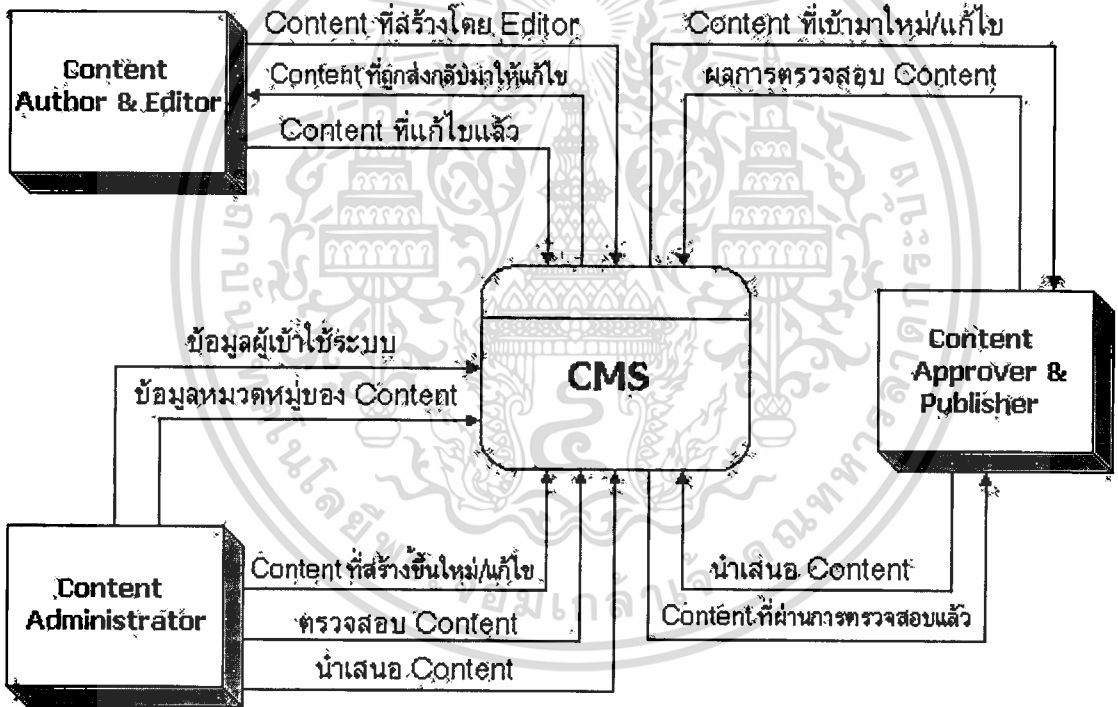
- ทำให้การสร้างรายการ Content ให้เป็นไปตามลำดับขั้นตอน ช่วยให้ไม่มีขั้นตอนใดถูกลืมหรือข้ามไป และไม่มี Content รายการใดที่ถูกลืม หรือทิ้งค้างไว้นานๆ
- เป็นการแบ่งหน้าที่อย่างชัดเจน ซึ่งช่วยให้ผู้ใช้งานสามารถรู้ว่าจะงานในส่วนที่รับผิดชอบอยู่มีอะไรบ้าง และดำเนินงานตามหน้าที่ของตนได้อย่างเป็นระบบ
- ลดเวลาในการทำงาน (Cycle Time) ของแต่ละ Content เพราะมีการแจ้งผู้รับผิดชอบงานในทุกขั้นตอนที่ Content ถูกส่งไปให้ทราบว่ามีรายการ Content ที่รอการดำเนินงานอยู่ เพราะฉะนั้น
- ลดการทำงานซ้ำซ้อน และลดความผิดพลาดในการทำงาน เพราะทุก Content จะถูกทำตามขั้นตอนของ Workflow ตั้งแต่เริ่มสร้างจนถึงนำเสนอบนเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถติดตามดูได้ว่า Content แต่ละรายการที่สร้างขึ้นถูกดำเนินการไปถึงขั้นตอนไหนแล้วบ้าง เช่น รอการตรวจสอบหรือผ่านการตรวจสอบแล้ว
- สามารถปรับเปลี่ยน Workflow ได้ตามความต้องการ เพื่อให้สอดคล้องกับลักษณะงานและธุรกิจ เช่น อาจเพิ่มให้มีขั้นตอนการทดสอบก่อนที่จะถึงขั้นตอนการ Publish หรือแยกงานระหว่าง Author และ Editor ออกจากกันเป็นคนละขั้นตอนหรือตัดงานที่ไม่จำเป็นทิ้งไป

3.3 การวิเคราะห์และออกแบบระบบ

เมื่อทำการวิเคราะห์ที่ศึกษาการทำงานของระบบแล้ว สามารถแสดงผลการวิเคราะห์ได้ดังต่อไปนี้



รูปที่ 3.4 Context Diagram ของระบบ CMS

3.3.1 Context Diagram

จากรูปจะเห็นว่า มีผู้ที่เกี่ยวข้องกับระบบทั้งหมด 4 ประเภท ซึ่งแบ่งตามหน้าที่การทำงานดังที่ได้อธิบายไว้ก่อนหน้านี้ ข้อมูลที่เข้าและออกจากแต่ละคนจะแตกต่างกัน โดยข้อมูลส่วนใหญ่จะมาจาก Content Editor ซึ่งส่งข้อมูลเข้ามาในระบบ และรอรับผลการตรวจสอบจาก Content Approver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ได้มาจาก Template Designer ซึ่งจะเก็บไว้เป็น Template สำหรับแสดงผล จะมีส่วนที่เป็นโค้ดหรือแท็กที่ใช้ในการจัดรูปแบบของ Content ลง Template และส่วนของรูปภาพที่มาจากเจ้าของ Content นั้นๆ ส่งมาให้ทำการปรับแต่ง เพื่อให้แสดงผลบนเว็บเพจได้อย่างถูกต้อง ข้อมูลจาก Template Designer จะถูกเก็บไว้บนเซิร์ฟเวอร์ แต่ไม่ได้เก็บลงฐานข้อมูล

แอปพลิเคชันในส่วนของ Content Management Application, Metacontent Management Application และ Content Display Application ที่แสดงในรูปแบบที่ 2 จะอยู่ในระบบ CMS ทั้งหมด

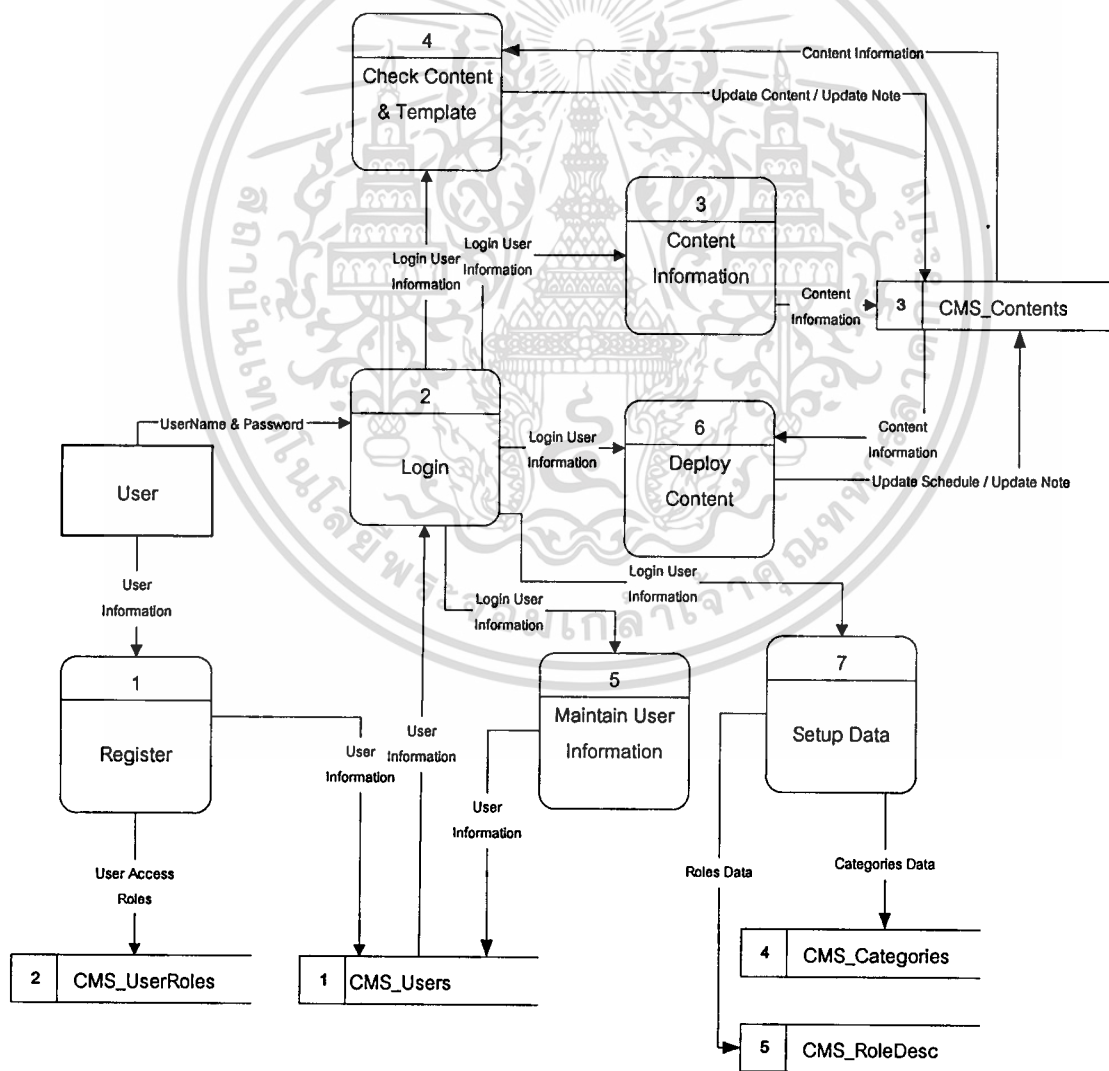
3.3.2 DataFlow Diagram

แสดงส่วนประกอบย่อยภายในระบบ CMS เพื่อให้เห็นชัดเจนยิ่งขึ้น โดยแสดงส่วนหลักๆ ของระบบดังนี้

- การบันทึกข้อมูลผู้ใช้งานระบบ คือ จัดการข้อมูลของผู้ใช้งานระบบ, User name และ Pasword สำหรับการเข้าใช้งานระบบ และกำหนดหน้าที่ของผู้ใช้งานระบบให้ โดยผู้ใช้งานระบบที่เป็นผู้ดูแลระบบจะเป็นผู้กำหนดให้ ซึ่งผู้ใช้งานคนหนึ่งจะถูกระบุหน้าที่ให้ได้เพียงหน้าที่เดียว เช่น ผู้ใช้ระบบที่ถูกระบุให้ทำหน้าที่เป็น Content Editor จะสามารถทำการบันทึกและแก้ไขข้อมูลรายการของตนเองได้เท่านั้น และไม่สามารถทำหน้าที่ตรวจสอบข้อมูลได้ การจัดเก็บว่าผู้ใช้งานคนใดทำหน้าที่ใดได้บ้างในอีกตารางแยกจากรายการที่เก็บข้อมูลเกี่ยวกับผู้ใช้งานระบบ
- การเข้าสู่ระบบ คือ ส่วนของตรวจสอบการเข้าใช้ระบบว่าผู้ที่ทำการล็อกอินเข้ามาสามารถใช้งานในฟังก์ชันการทำงานใดได้บ้าง เช่น ผู้ที่ทำหน้าที่เป็น Content Editor จะสามารถทำงานในหน้าจอการจัดการข้อมูล Content โดยสามารถบันทึกข้อมูลหรือแก้ไข Content และสามารถบันทึกผลการตรวจสอบได้ หรือผู้ที่ทำหน้าที่เป็น Content Approver จะสามารถทำงานได้ทั้งในหน้าจอการจัดการข้อมูล Content โดยสามารถบันทึกข้อมูลหรือแก้ไข Content และสามารถบันทึกผลการตรวจสอบได้
- การบันทึกหรือแก้ไขข้อมูล Content คือ ส่วนของการจัดการ Content ซึ่งผู้ที่เข้าสู่ระบบมาจะสามารถทำงานใดได้บ้างในส่วนนี้ก็จะขึ้นอยู่กับหน้าที่ของผู้ใช้ระบบคนนั้น ถ้าผู้ใช้งานระบบเป็น Content Editor จะสามารถทำการสร้าง, บันทึก และแก้ไข Content ได้ รวมทั้งสามารถบันทึกหมายเหตุไว้ในตารางเก็บหมายเหตุของ Content
- การตรวจสอบ Content และ Template ถ้าผู้ใช้งานระบบเป็น Content Approver ก็จะทำกรตรวจสอบ Content และแจ้งผลการตรวจสอบโดยบันทึกลงตารางที่เก็บข้อมูล Content นั้นหรือถ้ามีหมายเหตุก็จะบันทึกลงตารางที่เก็บหมายเหตุ พร้อมทั้งตรวจสอบ

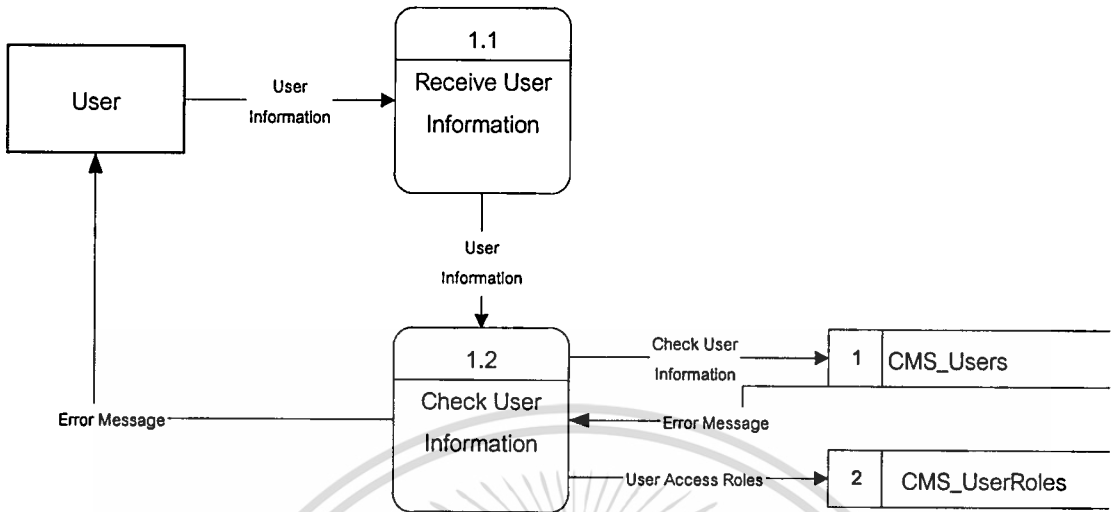
Template ว่าต้องการให้แก้ไขหรือไม่และส่งผลการตรวจสอบไปยัง Template Designer ทางเมล

- การจัดการ Content คือการเตรียม Content เพื่อนำเสนอบนเว็บเพจ โดยผู้ที่ใช้งานระบบจะเป็น Content Publisher ซึ่งจะทำการศึกษาตรวจสอบการจัดตารางกำหนดวันเวลาที่นำเสนอ Content และตรวจสอบรายการ Content ลงตามหมวดหมู่ที่ควรอยู่ โดยแอปพลิเคชันจะมีการตรวจสอบว่าช่วงวันที่และเวลาที่มีการซ้อนกันระหว่างแต่ละรายการหรือไม่ และช่วงวันและเวลาใดที่ไม่มี Content อยู่ในช่วงนั้น ซึ่งจะมีการแจ้งเตือนไปยัง Content Publisher ให้รับทราบ

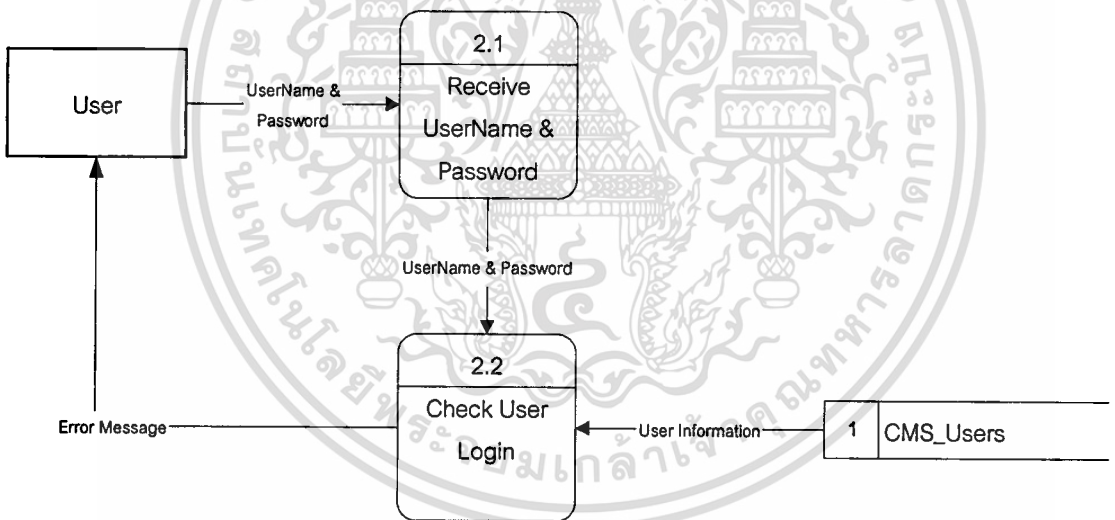


รูปที่ 3.5 Dataflow Diagram Level 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

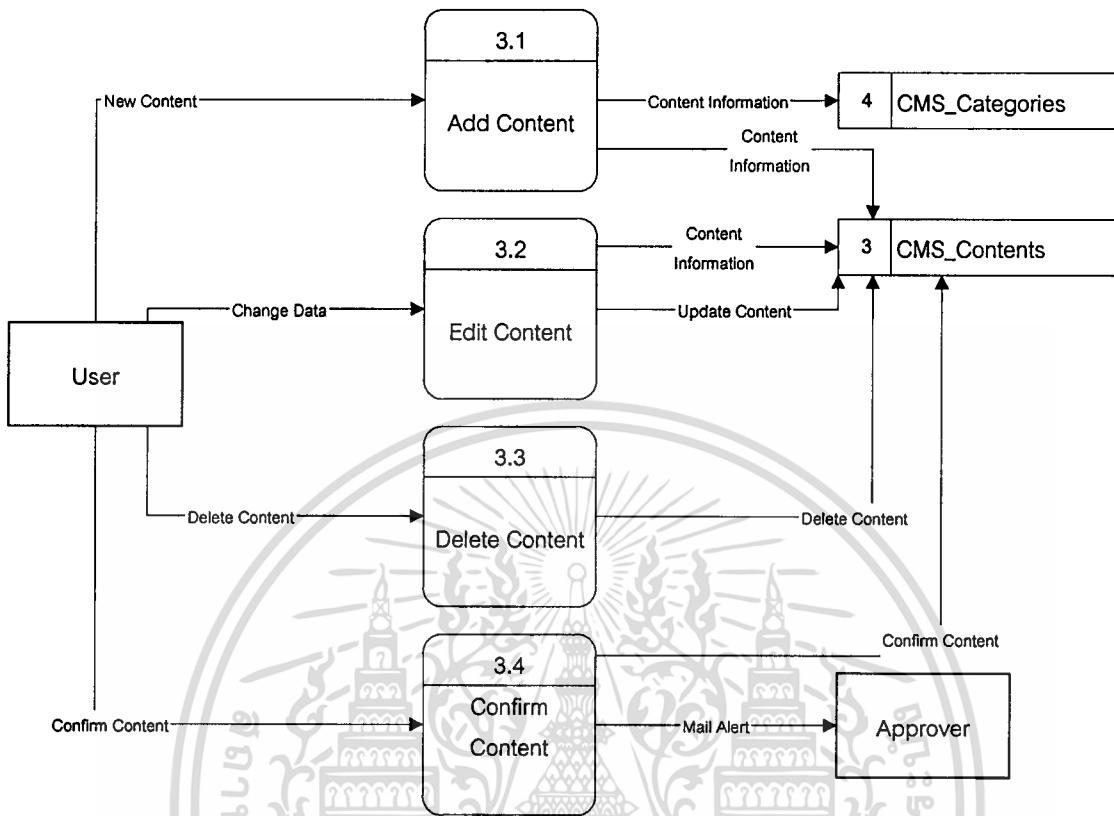


รูปที่ 3.6 Dataflow Diagram Level 2 ของ Process 1 (Register)

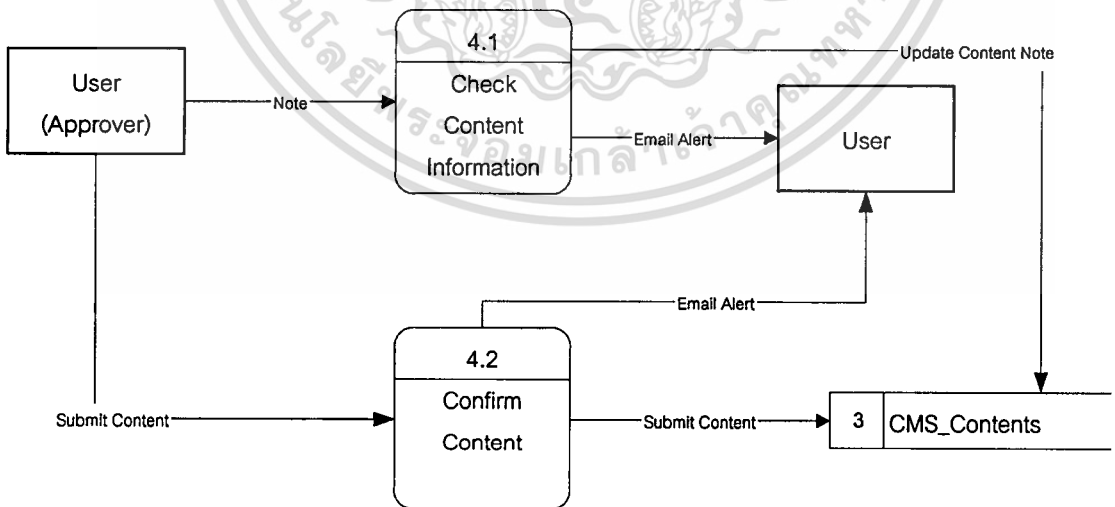


รูปที่ 3.7 Dataflow Diagram Level 2 ของ Process 2 (Login)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

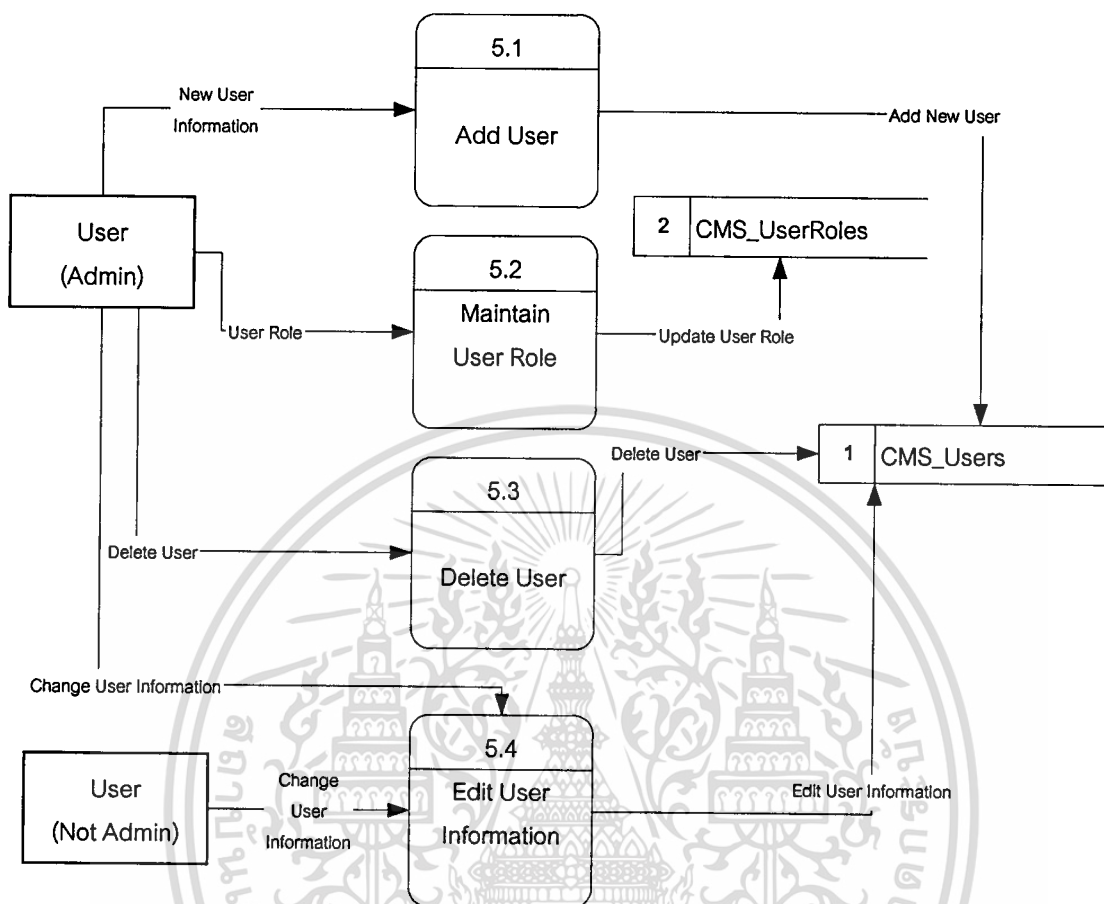


รูปที่ 3.8 Dataflow Diagram Level 2 ของ Process 3 (Content Information)

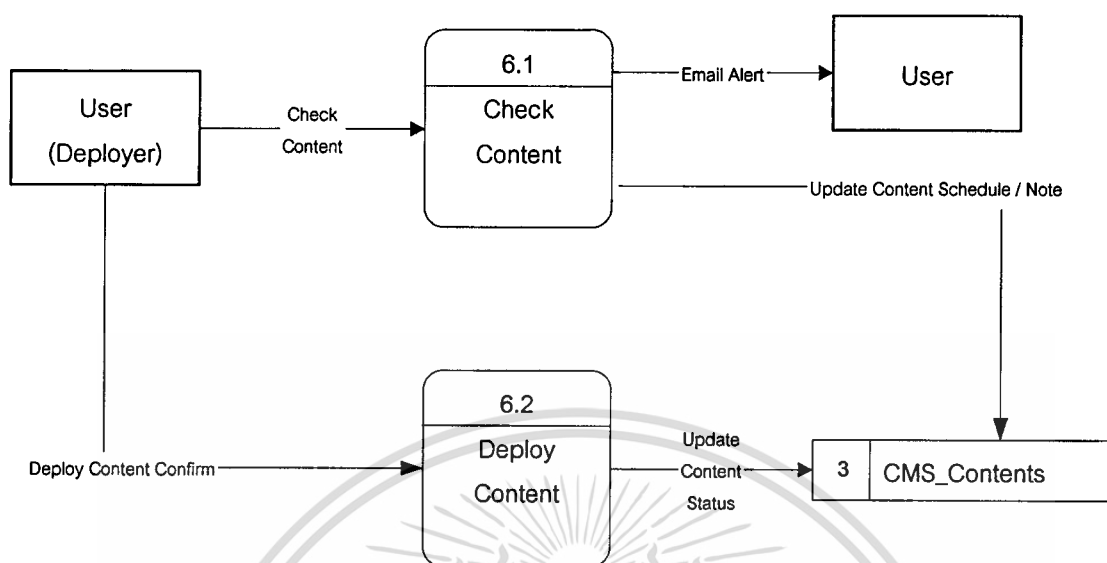


รูปที่ 3.9 Dataflow Diagram Level 2 ของ Process 4 (Check Content)

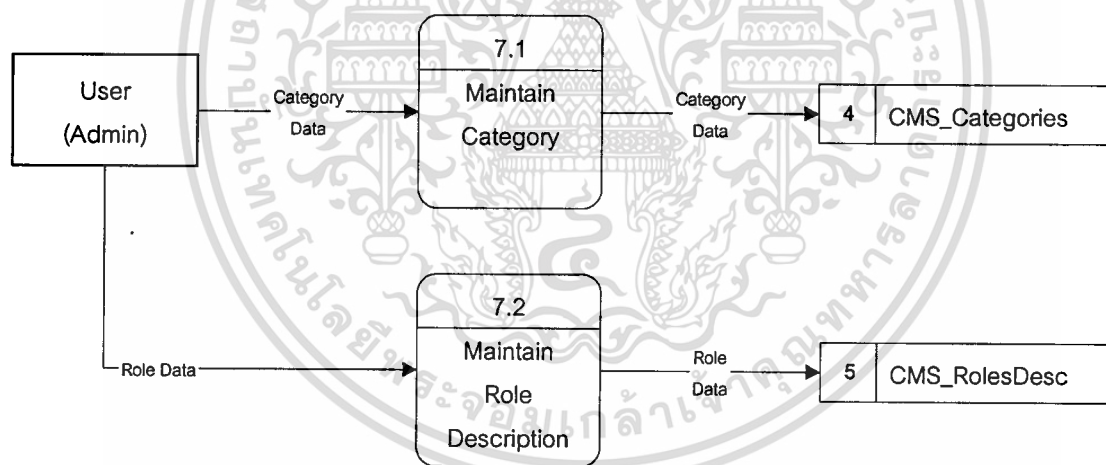
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Dataflow Diagram Level 2 ของ Process 5 (Edit User Information)



รูปที่ 3.11 Dataflow Diagram Level 2 ของ Process 6 (Publish Content)



รูปที่ 3.12 Dataflow Diagram Level 2 ของ Process 7 (Setup Data)

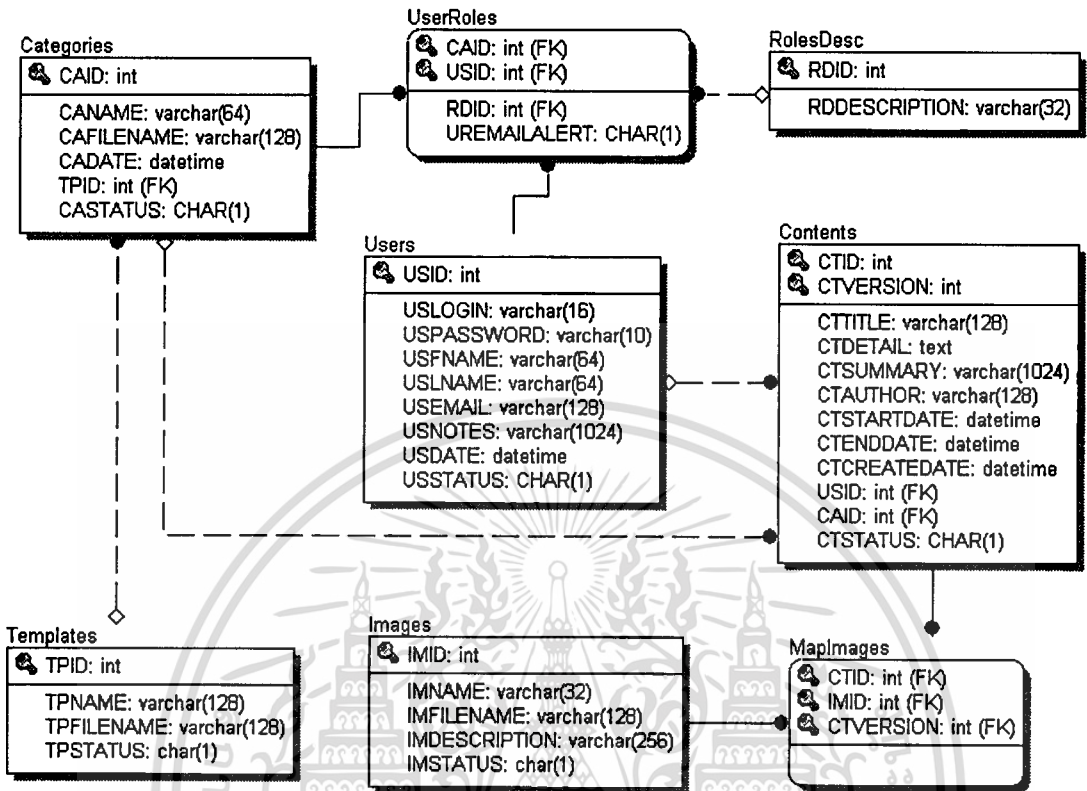
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลของระบบจะพิจารณาจากข้อมูลต่างๆที่ได้ศึกษารายละเอียดการทำงานจากระบบ จากนั้นจะทำการสร้างความสัมพันธ์ระหว่างตารางต่างๆ ซึ่งสามารถแสดงได้ด้วย Entity Relationship Diagram (E-R Diagram) ดังรูปที่

รายละเอียดของแต่ละตารางมีดังนี้

- ตาราง Users เก็บข้อมูลเกี่ยวกับผู้ใช้ระบบที่ได้ทำการลงทะเบียนไว้ เพื่อใช้ในการล็อกอินเข้าใช้งานระบบ โดยผู้ดูแลระบบจะเป็นคนจัดการ ส่วนข้อมูลของผู้ใช้ระบบที่ไม่ใช่ผู้ดูแลระบบจะสามารถทำการแก้ไขได้เฉพาะข้อมูลของตนเอง
- ตาราง UserRoles เก็บข้อมูลความสัมพันธ์ระหว่างผู้ใช้ระบบกับหมวดหมู่ที่มีในระบบ เช่น ระบุว่าผู้ใช้ระบบรหัส 1 จะสามารถทำงานได้เฉพาะในหมวดหมู่ที่มีรหัส 1 ในสถานะของ Editor โดยข้อมูลในตารางนี้จะมีผู้ดูแลระบบเป็นผู้รับผิดชอบ
- ตาราง Categories เก็บข้อมูลเกี่ยวกับหมวดหมู่ของข้อมูล เพื่อใช้ในการจัดเก็บ Content ให้เป็นระเบียบตามหมวดหมู่ ซึ่งสามารถจะระบุให้หมวดหมู่ใดแสดงบนเว็บเพจหรือไม่ให้แสดงบนเว็บเพจก็ได้
- ตาราง RoleDesc เก็บคำอธิบายหน้าที่หรือสถานะ เช่น RDID=1 คือ Admin , RDID=2 คือ Editor เป็นต้น
- ตาราง Contents เก็บข้อมูลรายละเอียดภายในของ Content แต่ละรายการ ซึ่งรายการหนึ่งจะสามารถมีได้หลายเวอร์ชัน ตามจำนวนการแก้ไข โดยทางระบบจะทำการเพิ่มเวอร์ชันให้โดยอัตโนมัติเมื่อมีการ
- ตาราง Templates เก็บข้อมูลรูปแบบของ Content แต่ละรายการตามที่ผู้ใช้ระบบกำหนด โดยจะเก็บข้อมูลเป็นไฟล์ HTML และระบุชื่อไฟล์และไคเร็กทอรีที่เก็บไฟล์ไว้ในตารางนี้
- ตาราง Images เก็บข้อมูลรูปภาพที่ใช้ในแต่ละ Content โดยจะเก็บเป็นชื่อไฟล์และไคเร็กทอรีที่เก็บไฟล์รูปภาพนั้นๆ
- ตาราง MapImages เก็บข้อมูลความสัมพันธ์ระหว่างรูปภาพกับ Content เพื่อบอกให้รู้ว่าภาพใดถูกนำไปใช้ใน Content ใดบ้าง



รูปที่ 3.13 Entity – Relationship Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Data Dictionary

จากการวิเคราะห์และออกแบบระบบฐานข้อมูลโดยวิธีใช้ Entity Relational Model สามารถนำไปสร้างเป็นตารางข้อมูล ซึ่งมีตารางข้อมูลที่ใช้ในระบบดังนี้

ตารางที่ 3.1 ตารางเก็บข้อมูลผู้ใช้ระบบ

ชื่อตาราง Users

ความหมาย เก็บรายละเอียดข้อมูลของผู้เข้าใช้ระบบ

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
USID	รหัสผู้ใช้ระบบ	Int(4)	PK	
USNAME	ชื่อที่ใช้ในการเข้าระบบ	Varchar(16)		
USFNAME	ชื่อของผู้ใช้ระบบ	Varchar(64)		
USLNAME	ชื่อสกุลของผู้ใช้ระบบ	Varchar(64)		
USPASSWORD	รหัสผ่านที่ใช้ในการเข้าระบบ	Varchar(10)		
USEMAIL	อีเมลของผู้ใช้ระบบ	Varchar(128)		
USNOTES	บันทึกข้อมูลอื่นๆของผู้ใช้ระบบ	Varchar(1024)		
USDATE	วันที่สร้างรายการ	Datetime		
USSTATUS	สถานะของผู้ใช้ระบบ (0=Inactive, 1=Active)	Char(1)		

ตารางที่ 3.2 ตารางเก็บข้อมูลสิทธิการเข้าถึงข้อมูลในแต่ละหมวดหมู่ (Categories)

ชื่อตาราง

UserRoles

ความหมาย

เก็บข้อมูลสิทธิของผู้ใช้ระบบในการเข้าถึงข้อมูลในแต่ละหมวดหมู่และสถานะของผู้ใช้ระบบในแต่ละหมวดหมู่ (Categories)

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
CAID	รหัสหมวดหมู่	Int(4)	PK	CMS_Categories
USID	รหัสผู้ใช้ระบบ	Int(4)	PK	CMS_Users
RDID	รหัสหน้าที่ของผู้ใช้ระบบ	Int(4)	FK	CMS_Templates
UREMAILALERT	สถานะการแจ้งเตือนทางอีเมล (0=ไม่แจ้งเตือน, 1=แจ้งเตือน)	Char(1)		

ตารางที่ 3.3 ตารางเก็บข้อมูลหมวดหมู่ (Categories)

ชื่อตาราง

Categories

ความหมาย

เก็บรายละเอียดข้อมูลของแต่ละหมวดหมู่ (Categories)

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
CAID	รหัสหมวดหมู่	Int(4)	PK	
CANAME	ชื่อหมวดหมู่	Varchar(64)		
CASTATUS	สถานะของหมวดหมู่ (0=Invisible, 1=Visible)	Char(1)		
CAFILENAME	ชื่อไฟล์หมวดหมู่	Varchar(128)		
TPID	รหัส Template	Int(4)	FK	CMS_Templates
CADATE	วันที่สร้างหมวดหมู่	Datetime		

ตารางที่ 3.4 ตารางเก็บข้อมูลหน้าที่ (Roles)

ชื่อตาราง

RolesDesc

ความหมาย

เก็บรายละเอียดข้อมูลความหมายของแต่ละหน้าที่ (Roles)

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
RDID	รหัสหน้าที่รับผิดชอบ	Int(4)	PK	
RDDESCRIPTION	คำอธิบายหน้าที่หรือ สถานะผู้ใช้ระบบ	Varchar(32)		

ตารางที่ 3.5 ตารางเก็บข้อมูล Content

ชื่อตาราง

Contents

ความหมาย

เก็บรายละเอียดข้อมูลของ Content แต่ละรายการ

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
CTID	รหัส Content	Int(4)	PK	
CTVERSION	หมายเลขเวอร์ชันของ Content	Int(4)	PK	
CTTITLE	ชื่อหัวข้อ Content	Varchar(128)		
CTDETAIL	ข้อมูล Content	Text(16)		
CTSUMMARY	สรุป Content	Varchar(1027)		
CTSTATUS	สถานะของ Content (0=Inactive, 1=Active)	Char(1)		
CTAUTHOR	ชื่อผู้แต่ง	Varchar(128)		
CTSTARTDATE	วันที่เริ่มต้นการนำเสนอ บนเว็บเพจ	Datetime		
CTENDDATE	วันที่สิ้นสุดการนำเสนอ บนเว็บเพจ	Datetime		
CTCREATEDATE	วันที่สร้าง Content	Datetime		
USID	รหัสผู้ทำการแก้ไขข้อมูล	Int(4)	FK	CMS_Users
CAID	รหัสหมวดหมู่ของ Content	Int(4)	FK	CMS_Categories

ตารางที่ 3.6 ตารางเก็บข้อมูล Template

ชื่อตาราง Templates

ความหมาย เก็บรายละเอียดของไฟล์ Template ของข้อมูล

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
TPID	รหัส Template	Int(4)	PK	
TPNAME	ชื่อ Template	Varchar(128)		
TPFILENAME	ชื่อไฟล์ของข้อมูล Template	Varchar(128)		
TPSTATUS	สถานะของ Template	Char(1)		

ตารางที่ 3.7 ตารางเก็บข้อมูลรูปภาพ

ชื่อตาราง Images

ความหมาย เก็บรายละเอียดข้อมูลเกี่ยวกับรูปภาพที่ใช้ใน Content

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
IMID	รหัสของรูปภาพ	Int(4)	PK	
IMNAME	ชื่อรูปภาพ	Varchar(32)		
IMFILENAME	ชื่อไฟล์รูปภาพ	Varchar(128)		
IMSTATUS	สถานะของรูปภาพ (0=Invisible, 1=Visible)	char(1)		
IMDESCRIPTION	ชื่อของแต่ละหน้าที่หรือ สถานะผู้ใช้งานระบบ	Varchar(256)		

ตารางที่ 3.8 ตารางเก็บข้อมูล Map รูปภาพให้กับแต่ละ Content

ชื่อตาราง

MapImages

ความหมาย

ตารางเก็บข้อมูล Map รูปภาพให้กับแต่ละ Content

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
CTID	รหัส Content	Int(4)	FK	CMS_Contents
IMID	รหัสรูปภาพ	Int(4)	FK	CMS_Images
CTVERSION	หมายเลขเวอร์ชันของ Content	Int(4)	FK	CMS_Contents



บทที่ 4

การพัฒนาระบบ

หลังจากได้มีการศึกษาข้อมูลและออกแบบระบบการทำงานของระบบงานเป็นที่เรียบร้อยแล้ว จึงทำการพัฒนาระบบให้สามารถนำไปใช้งานได้จริง

4.1 การพัฒนาโปรแกรมคอมพิวเตอร์และเครื่องมือที่ใช้

โดยภาพรวมในการทำงานของระบบงานทั้งหมด จะประกอบไปด้วย เว็บไคลเอนต์ เว็บเซิร์ฟเวอร์ แอปพลิเคชันเซิร์ฟเวอร์ และดาต้าเบสเซิร์ฟเวอร์ ในการพัฒนาระบบ ต้องมีการติดตั้งเครื่องมือต่างๆเพื่อให้สามารถทำการพัฒนาระบบงานได้ ซึ่งประกอบด้วย

- เว็บไคลเอนต์ ซึ่งไม่ได้ทำการพัฒนา แต่สามารถใช้งานได้จากโปรแกรมประเภทเว็บเบราว์เซอร์ต่างๆไป เช่น Internet Explorer
- เว็บเซิร์ฟเวอร์ โดยการใช้ระบบปฏิบัติการ Windows XP ทำหน้าที่เป็นตัวให้บริการกับเว็บไคลเอนต์
- เซิร์ฟเวอร์แอปพลิเคชัน ใช้โปรแกรม Visual Studio .NET เป็นเครื่องมือในการพัฒนาระบบ รวมถึงส่วนระบบรักษาความปลอดภัยของการใช้งานระบบ
- ดาต้าเบสเซิร์ฟเวอร์ ใช้ SQL Server 2000 เพื่อเก็บข้อมูลในระบบฐานข้อมูลที่ให้บริการผ่านเครือข่ายระบบอินเทอร์เน็ต

4.2 โครงสร้างการทำงานของระบบงาน

จากการวิเคราะห์การทำงานของระบบงาน สามารถมองการทำงานของระบบที่ทำการพัฒนานี้เป็นส่วนๆแบ่งตามหน้าที่ความรับผิดชอบของผู้ใช้งาน โดยสามารถแบ่งออกได้เป็น 3 ส่วน ดังนี้

- 4.2.1 ส่วนการทำงานของผู้จัดการข้อมูล เป็นส่วนที่เป็นหลักของระบบ เพราะเป็นส่วนของการจัดการข้อมูลที่จะนำเสนอบนเว็บเพจ โดยจะทำงานผ่านโปรแกรมบนเว็บเบราว์เซอร์ มีความสามารถรองรับการทำงาน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

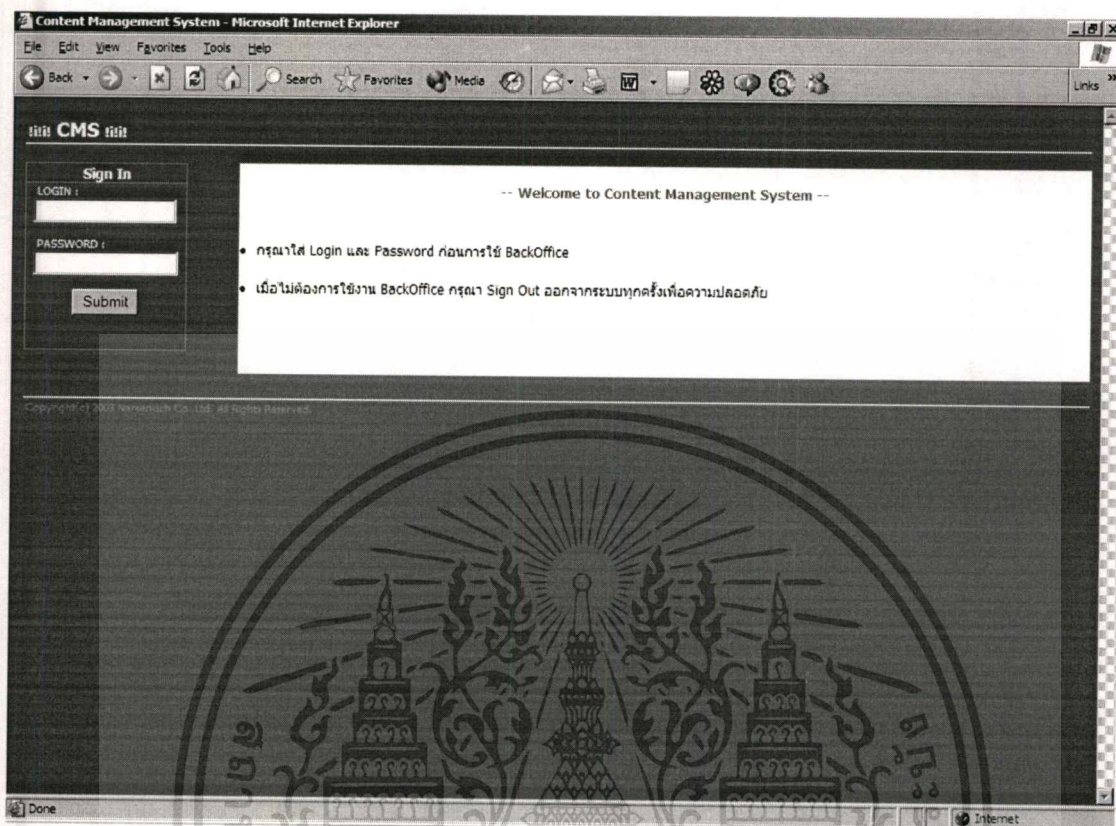
- สามารถแสดงรายการข้อมูลที่จะนำเสนอบนเว็บเพจ พร้อมแสดงขั้นตอนการทำงานที่สามารถทำได้ของแต่ละรายการ ซึ่งข้อมูลแต่ละรายการจะสามารถถูกทำการแก้ไขเปลี่ยนแปลงหรือทำการยืนยันเมื่อข้อมูลรายการนั้นถูกต้องหรือสมบูรณ์แล้ว โดยจะสามารถทำอะไรกับรายการนั้นได้บ้างนั้นก็ขึ้นอยู่กับสถานะของรายการนั้นและสถานะของผู้ใช้ระบบผู้นั้นด้วย ซึ่งสถานะของผู้ใช้ระบบจะถูกตรวจสอบตั้งแต่ในขั้นตอนเข้าสู่ระบบอยู่แล้ว
- สามารถทำการเพิ่มรายการข้อมูลที่จะนำเสนอบนเว็บเพจ หรือแก้ไขเปลี่ยนแปลง หรือยืนยันข้อมูลที่สมบูรณ์แล้วไปปรับการอนุมัติต่อไป โดยผู้ใช้ระบบจะต้องเป็นเจ้าของข้อมูลรายการนั้นหรือเป็นผู้ที่มีสิทธิพอที่จะทำการแก้ไขเปลี่ยนแปลงหรือยืนยันข้อมูลนั้น

4.2.2 ส่วนการทำงานของผู้ตรวจสอบและจัดการการนำเสนอข้อมูลบนเว็บไซต์ เป็นส่วนของการตรวจสอบความถูกต้องของข้อมูลแต่ละรายการที่ผ่านการยืนยันความถูกต้องสมบูรณ์แล้วจากผู้ที่จัดการข้อมูลรายการนั้น ซึ่งจะส่งมาให้ผู้ตรวจสอบทำการอนุมัติการนำเสนอหรือส่งกลับให้ทำการแก้ไขใหม่ โดยสามารถมีข้อความอธิบายจากผู้ตรวจสอบแนบไปกับรายการนั้นด้วย

4.2.3 ส่วนการทำงานของผูดูแลระบบ เป็นส่วนของการจัดการและการจัดการข้อมูลผู้ใช้ระบบ และข้อมูลเบื้องต้นของระบบ เช่น ข้อมูลการกำหนดหมวดหมู่ให้กับรายการที่จะนำเสนอบนเว็บเพจ ซึ่งในการนำเสนอก็จะแสดงหัวข้อเรื่องของแต่ละรายการเป็นหมวดหมู่ด้วย

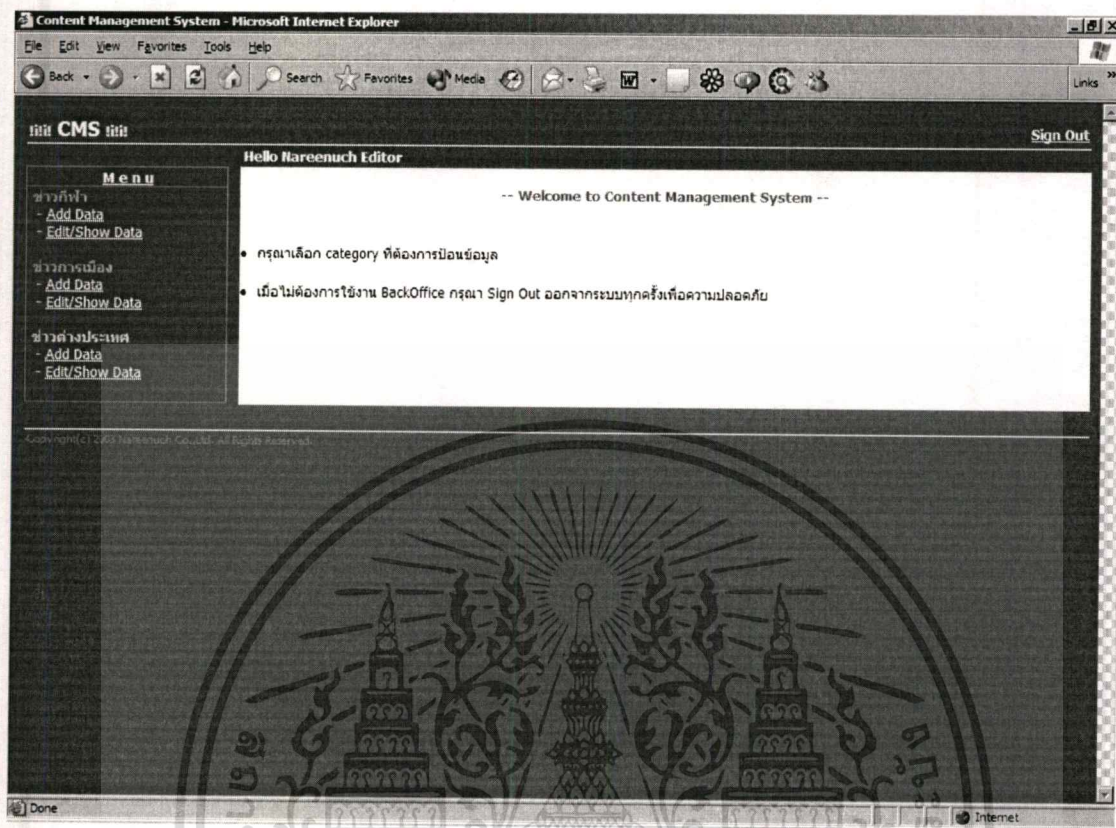
4.3 รายละเอียดของการพัฒนาระบบ

จากการพัฒนาระบบด้วยโปรแกรม Visual Studio .NET ซึ่งมีเครื่องมือที่ช่วยในการพัฒนาระบบและองค์ประกอบของระบบงาน เพื่อการออกแบบหน้าจอการทำงาน ทำให้ได้ระบบที่มีหน้าจการทำงานดังต่อไปนี้



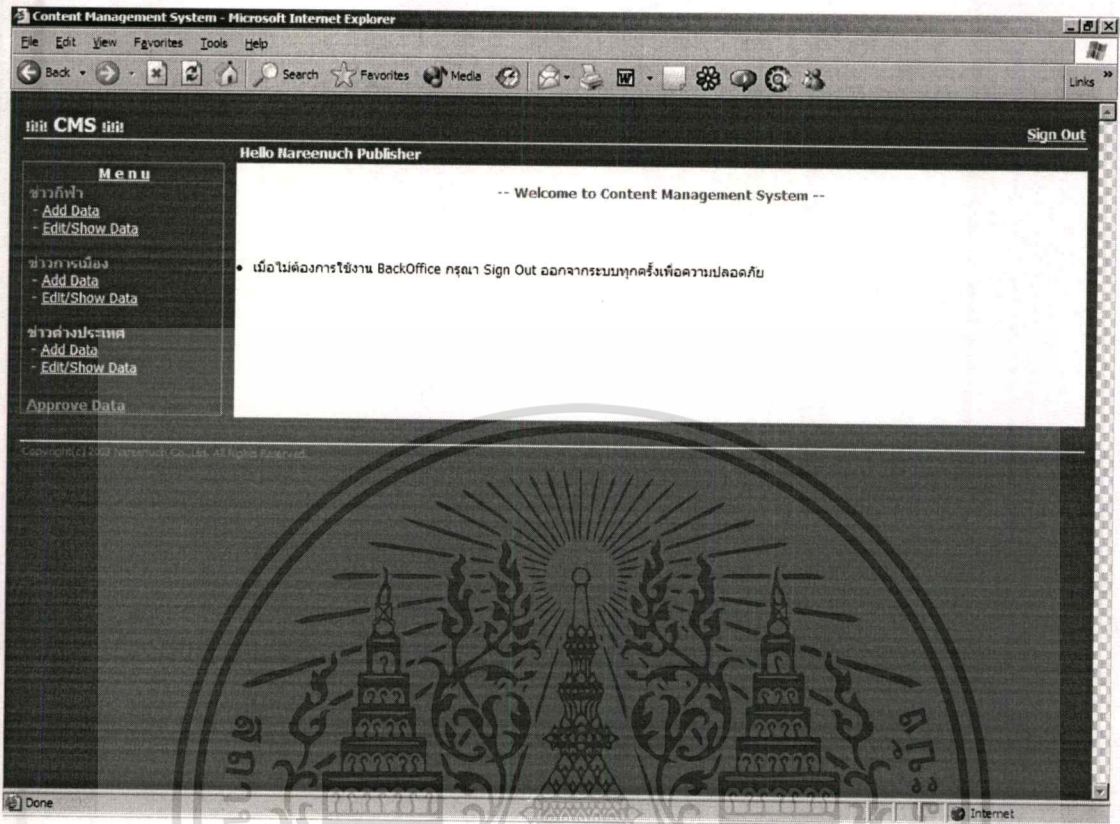
รูปที่ 4.1 หน้าจอสำหรับใส่ข้อมูล UserName และ Password ของผู้เข้าใช้ระบบ

เมื่อทำการล็อกอินเข้าสู่ระบบแล้วจะเข้าไปสู่หน้าจอใดหน้าจอหนึ่งในรูปที่ 4.2 ถึงรูปที่ 4.4 โดยจะขึ้นอยู่กับสถานะของผู้เข้าใช้ระบบที่ถูกตรวจสอบพบในตอนที่ทำกรล็อกอิน ซึ่งสถานะของผู้เข้าใช้ระบบจะเป็นตัวบอกว่าผู้ใช้ระบบสามารถเข้ามาทำอะไรในระบบได้บ้าง



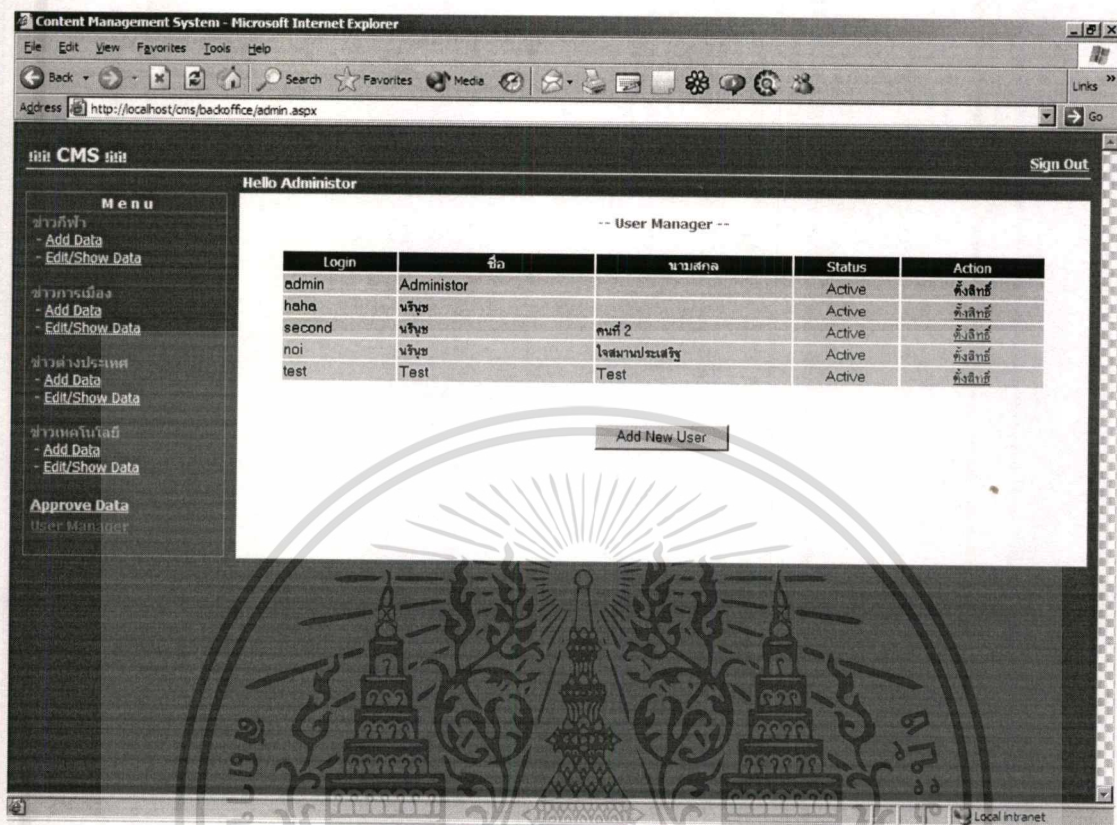
รูปที่ 4.2 หน้าจอสำหรับจัดการข้อมูลสำหรับผู้ใช้งานที่เป็น Editor

หน้าจอสำหรับจัดการข้อมูลสำหรับผู้ใช้งานที่เป็น Editor จะมีเมนูสำหรับใช้งานแยกตามหมวดหมู่ที่ผู้ใช้งานคนนั้นมีสิทธิเข้ามาทำได้ ซึ่งถ้าหมวดหมู่ใดที่ผู้ใช้งานไม่มีสิทธิเข้าไปทำงานได้ก็จะไม่อนุญาตให้เข้าไปทำอะไร ในแต่ละหมวดหมู่ก็จะมีเมนูสำหรับการเพิ่มข้อมูล (เมนู Add Data) และเมนูสำหรับการแก้ไขหรือแสดงข้อมูล (Edit/Show Data)



รูปที่ 4.3 หน้าจอสำหรับการจัดการข้อมูลสำหรับผู้ใช้งานที่เป็น Approver

หน้าจอสำหรับจัดการข้อมูลสำหรับผู้ใช้งานที่เป็น Publisher จะมีเมนูสำหรับใช้งานแยกตามหมวดหมู่ที่ผู้ใช้งานคนนั้นมีสิทธิเข้ามาทำได้ ซึ่งถ้าหมวดหมู่ใดที่ผู้ใช้งานไม่มีสิทธิเข้าไปทำงานได้ ก็จะไม่อนุญาตให้เข้าไปทำอะไร ในแต่ละหมวดหมู่ก็จะมีเมนูสำหรับการเพิ่มข้อมูล (เมนู Add Data), เมนูสำหรับการแก้ไขหรือแสดงข้อมูล (Edit/Show Data) และตรวจสอบข้อมูล (เมนู Approve Data)

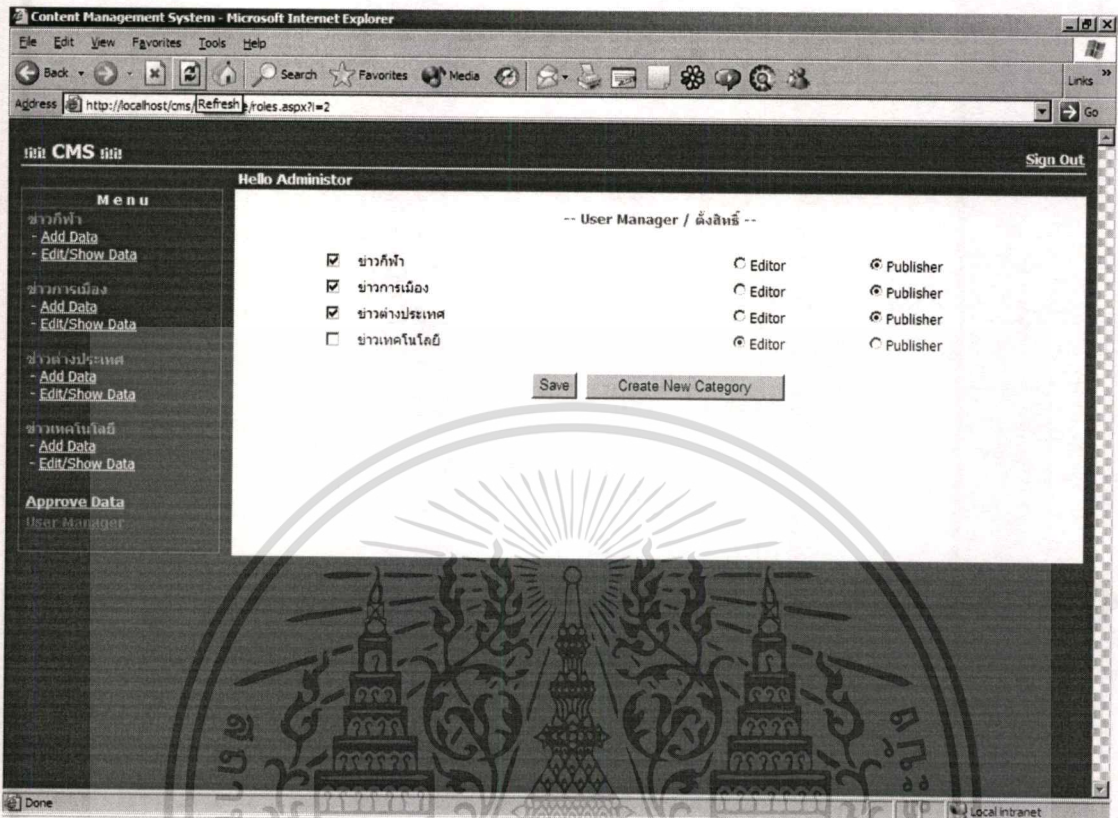


รูปที่ 4.4 หน้าจอสำหรับการจัดการข้อมูลสำหรับผู้ใช้งานที่เป็น Administrator

หน้าจอสำหรับจัดการข้อมูลสำหรับผู้ใช้งานที่เป็นผู้ดูแลระบบ (Administrator) ซึ่งจะมีสิทธิทำได้ทุกอย่าง โดยมีเมนูทั้งหมดดังนี้

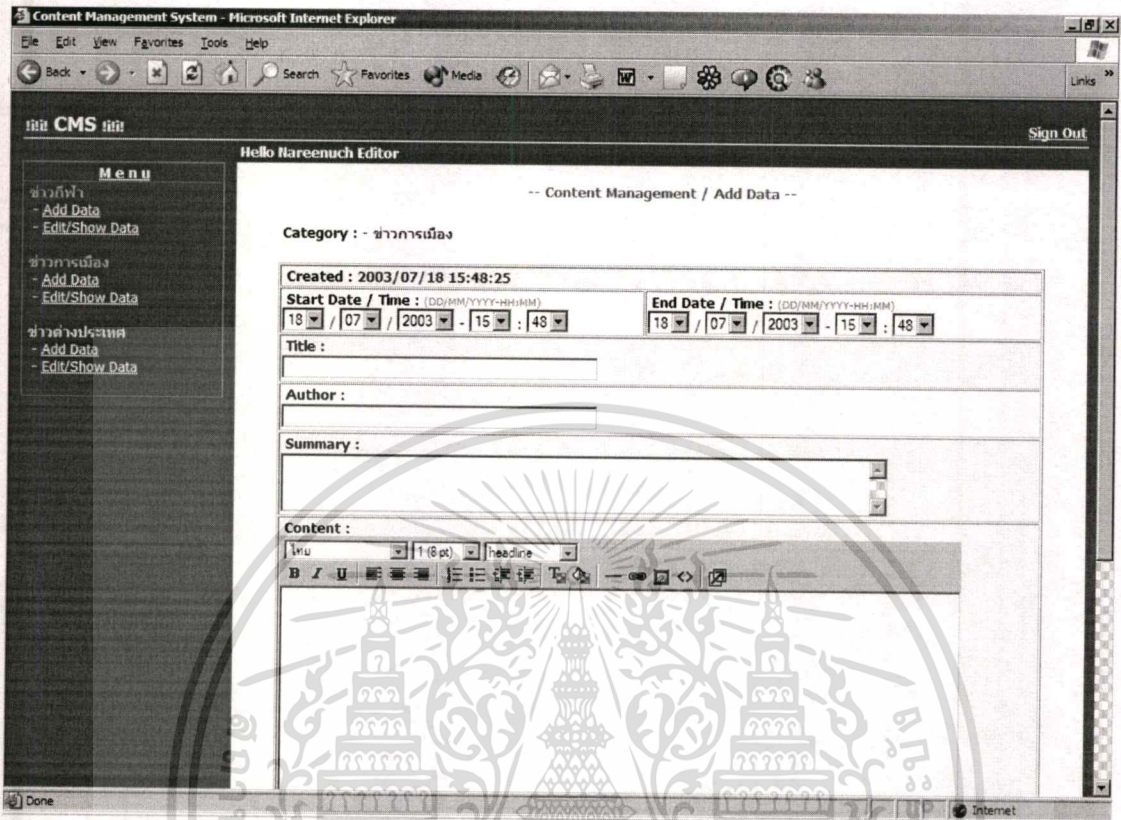
- เมนูสำหรับจัดการข้อมูลผู้เข้าใช้ระบบ (เมนู User Manager)
- เมนูสำหรับจัดการข้อมูลหมวดหมู่ (เมนู Category Manager)
- เมนูสำหรับจัดการข้อมูล Content ได้แก่ เมนูสำหรับเพิ่มข้อมูล (เมนู Add Data), เมนูสำหรับทำการแก้ไขหรือแสดงข้อมูล (เมนู Edit/Show Data)
- เมนูสำหรับทำการตรวจสอบข้อมูลก่อนทำการอนุมัติ (เมนู Approve Data)
- เมนูสำหรับจัดการข้อมูลผู้เข้าใช้ระบบ (เมนู User Manager)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

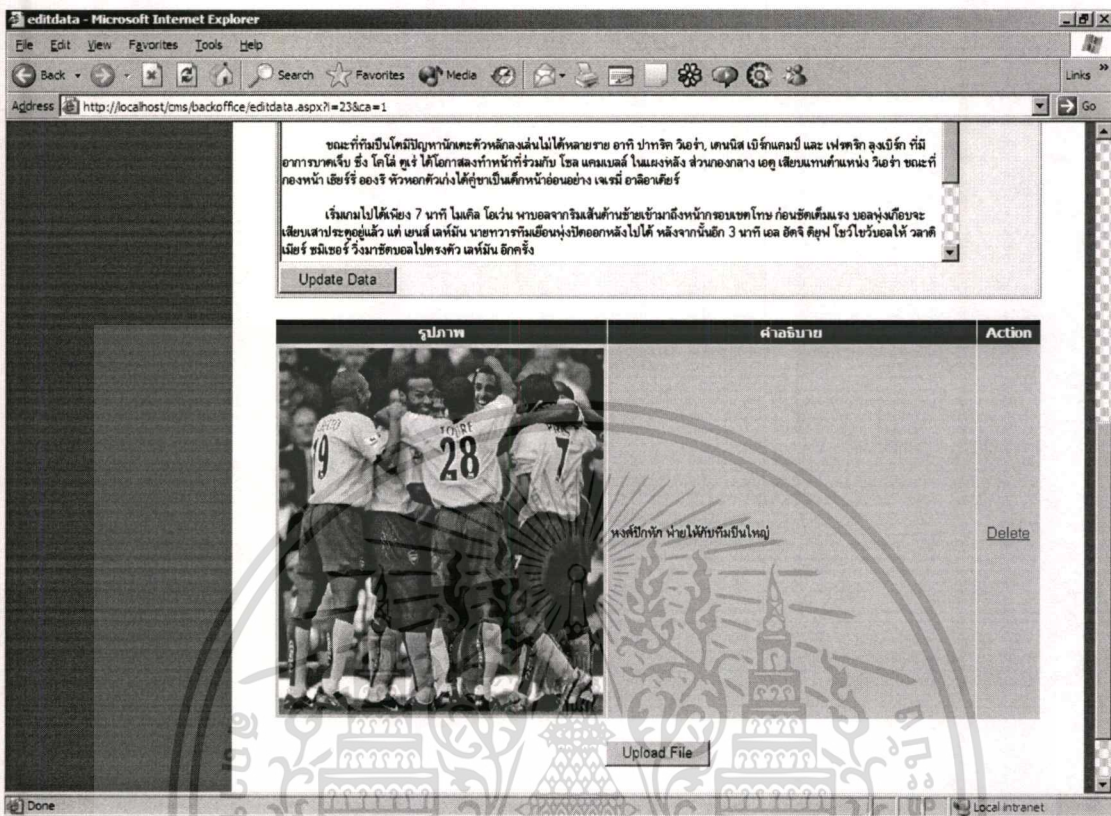


รูปที่ 4.5 หน้าจอแสดงการกำหนดสิทธิ์การเข้าใช้ในแต่ละหมวดหมู่ให้กับผู้เข้าใช้ระบบ

หน้าจอสำหรับจัดการข้อมูลสำหรับผู้ใช้งานที่เป็นผู้ดูแลระบบ (Administrator) ซึ่งสามารถระบุว่าจะให้ผู้เข้าใช้ระบบสามารถเข้าใช้งานในหมวดหมู่ใดได้บ้างและระบุสถานะการเข้าใช้ในแต่ละหมวดหมู่

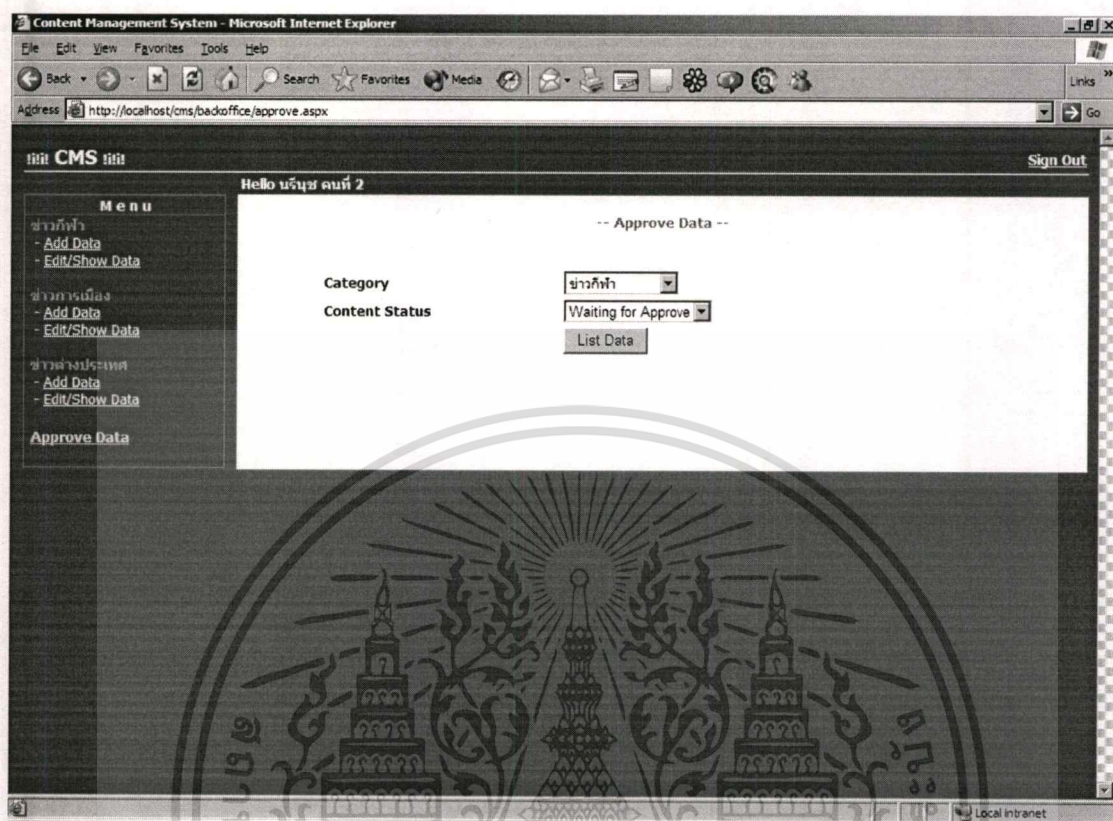


รูปที่ 4.6 หน้าจอสำหรับการจัดการข้อมูล Content



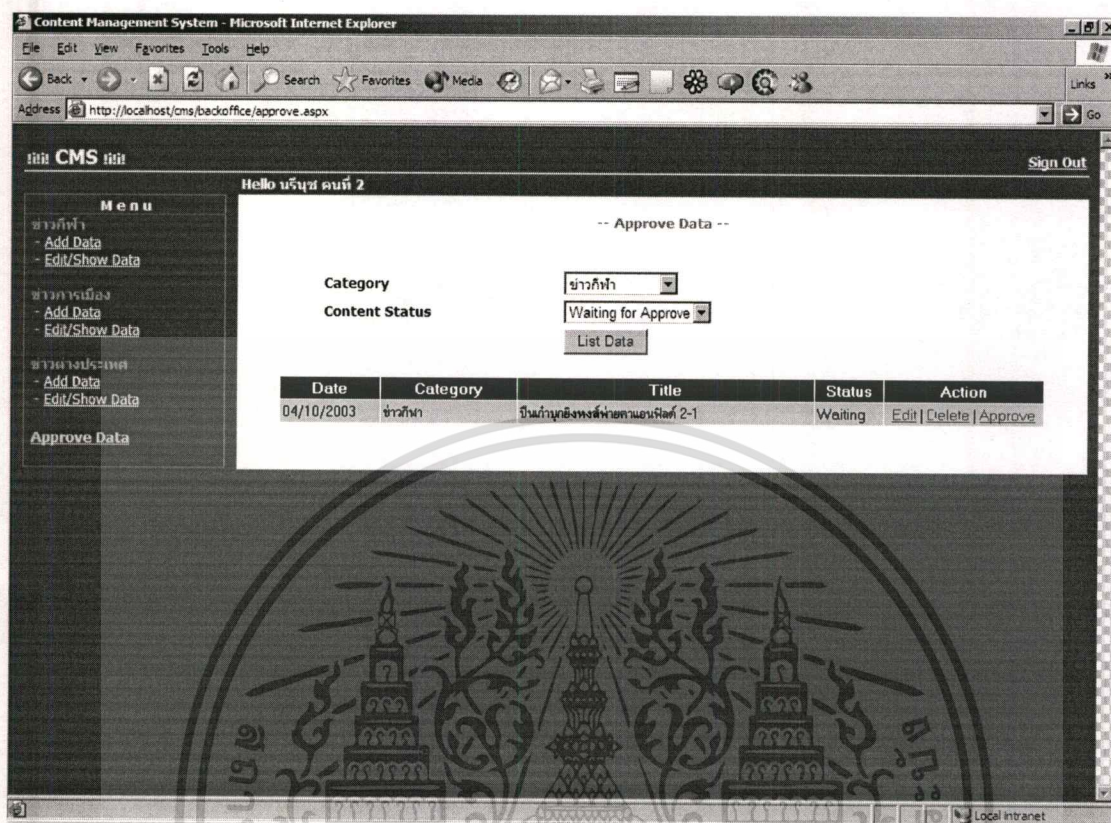
รูปที่ 4.7 หน้าจอแสดงการนำรูปภาพมาใส่ให้กับ Content

เมื่อจำเป็นจะต้องมีรูปสำหรับประกอบคำบรรยายในรายการใด ก็สามารถนำรูปเข้ามาได้ ดังรูปที่ 4.7



รูปที่ 4.8 หน้าจอแสดงการเลือกรายการที่สนใจขึ้นมาดูในหน้าตรวจสอบ

หลังจากเลือกประเภทหมวดหมู่และสถานะของข้อมูลที่สนใจ คือ สถานะรอการตรวจสอบ และสถานะรอการถูกนำขึ้นแสดงบนเว็บเพจ แล้วกดปุ่ม “List Data” ก็จะปรากฏรายการตาม que เลือกขึ้นมา โดยแสดงข้อมูลเรียงตามวันที่



รูปที่ 4.9 หน้าจอแสดงตัวอย่างการเลือกรายการที่รอการตรวจสอบ

หลังจากเลือกรายการที่ต้องการตรวจสอบแล้วจะปรากฏข้อมูลดังรูปที่ 4.9 นี้

- กรณีที่ผู้ตรวจสอบต้องแก้ไขก็จะเลือกที่ Edit
- กรณีที่ผู้ตรวจสอบต้องการลบรายการนั้นทิ้งก็จะเลือกที่ Delete
- กรณีที่ต้องการอนุมัติรายการนั้นเพื่อนำเสนอบนเว็บเพจต่อไป จะสามารถเลือกที่ Approve รายการนั้นก็จะถูกเปลี่ยนสถานะให้เป็นรายการที่รอนำเสนอ

Content Management System - Microsoft Internet Explorer

Address: http://localhost/cms/backoffice/approve.aspx

CMS อนุมัติ Sign Out

Hello Administrator

-- Approve Data --

Category: ข่าวกีฬา

Content Status: Published

List Data

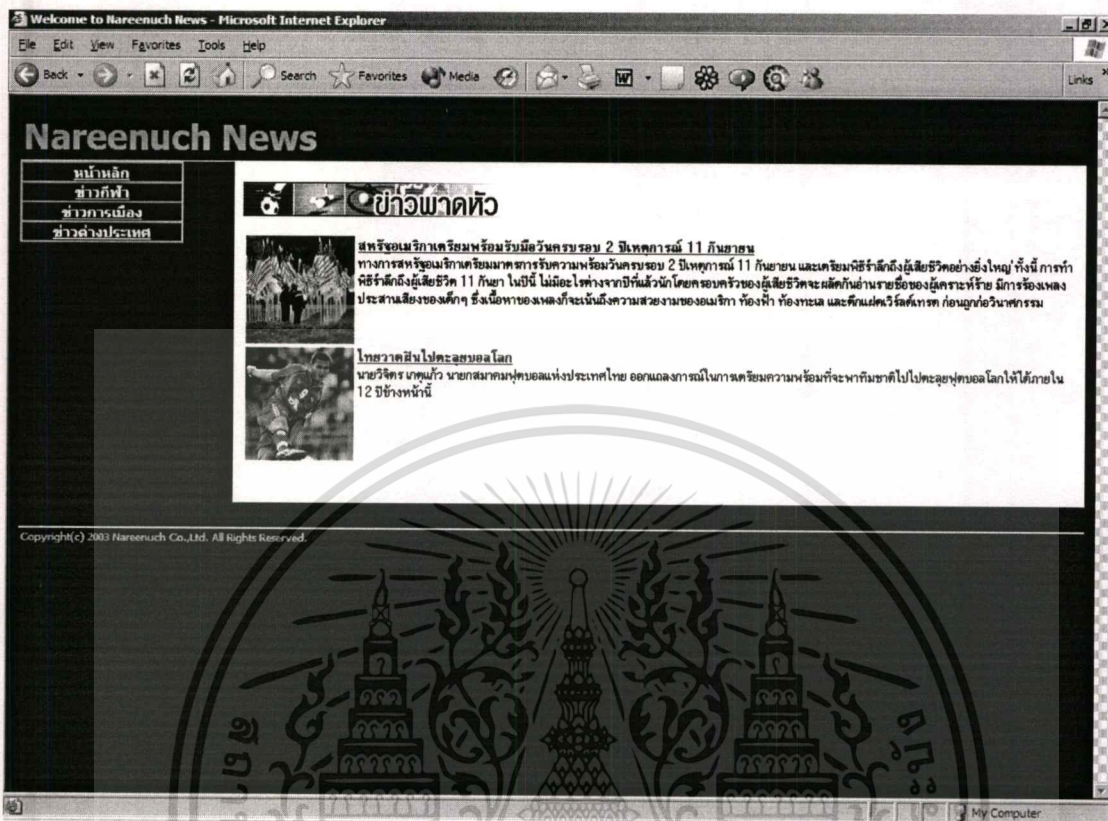
Date	Category	Title	Status	Action
04/10/2003	ข่าวกีฬา	ชิงแชมป์ชิงหงส์พ่ายคนนอกศึก 2-1	Published	Edit Delete Approve
03/10/2003	ข่าวกีฬา	"เจ้าบอล"พอร์ซุสเซ่นเข้าทีม 8 คนแชมป์	Published	Edit Delete Approve
03/10/2003	ข่าวกีฬา	มหัศจรรย์โยหนูโรดได้แชมป์	Published	Edit Delete Approve

Done Local intranet

รูปที่ 4.10 หน้าจอแสดงตัวอย่างการเลือกรายการที่รอการนำเสนอ

หลังจากเลือกรายการที่ต้องการตรวจสอบแล้วจะปรากฏข้อมูลดังรูปที่ 4.10 นี้

- กรณีที่ผู้ตรวจสอบต้องแก้ไขก็จะเลือกที่ Edit
- กรณีที่ผู้ตรวจสอบต้องการลบรายการนั้นทิ้งก็จะเลือกที่ Delete
- กรณีที่ต้องการอนุมัติรายการนั้นเพื่อนำเสนอบนเว็บเพจจะสามารถเลือกที่ Approve รายการนั้นก็จะถูกนำขึ้นแสดงบนเว็บเพจตามวันและเวลาที่ระบุไว้



รูปที่ 4.11 หน้าจอแสดงตัวอย่างการนำเสนอเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 ผลการดำเนินการพัฒนาระบบ

จากการศึกษาและพัฒนาระบบงาน เพื่อให้ได้ระบบจัดการเอกสารบนเว็บเพจที่สามารถจัดการและควบคุมความถูกต้องของข้อมูลตามขั้นตอน สามารถสรุปผลการใช้งานระบบได้ดังนี้

- 5.1.1 ส่วนการจัดการข้อมูล สามารถเรียกดู, เพิ่ม, ลบ และแก้ไข ข้อมูล และสามารถยืนยันความถูกต้องของข้อมูลเพื่อส่งไปยังขั้นตอนการอนุมัติ โดยขึ้นอยู่กับสถานะของผู้เข้าใช้ระบบซึ่งเป็นตัวกำหนดสิทธิและหน้าที่ของการเข้าใช้งาน
- 5.1.2 ส่วนของการตรวจสอบข้อมูลและจัดการนำเสนอข้อมูลบนเว็บเพจ สามารถตรวจสอบข้อมูลแต่ละรายการและทำการอนุมัติเพื่อนำเสนอบนเว็บเพจต่อไปหรือส่งคืนรายการที่ต้องมีแก้ไขกลับไปให้ทำการแก้ไข และสำหรับข้อมูลรายการที่ผ่านการอนุมัติแล้วสามารถนำมาจัดการในเรื่องการนำเสนอบนเว็บเพจได้
- 5.1.3 ส่วนของการจัดการข้อมูลเบื้องต้น สามารถจัดการข้อมูลผู้เข้าใช้ระบบและกำหนดสิทธิหรือสถานะให้กับผู้ใช้ระบบแต่ละคน และสามารถจัดการข้อมูลหมวดหมู่สำหรับการจัดหมวดหมู่ให้กับข้อมูลแต่ละรายการ
- 5.1.4 ส่วนของการจัดการรูปแบบการนำเสนอ สามารถจัดการรูปแบบการนำเสนอเว็บเพจได้

5.2 ประโยชน์ที่ได้รับ

- 5.2.1 ได้รับความรู้ในเทคโนโลยีใหม่ และเกิดทักษะ และความเข้าใจในการนำเทคโนโลยีใหม่มาประยุกต์ใช้ในการพัฒนาระบบ
- 5.2.2 ได้รับความรู้และทักษะในการพัฒนาระบบ และได้ระบบจัดการเอกสารบนเว็บเพจที่สามารถใช้งานได้ตามต้องการ
- 5.2.3 สามารถนำระบบที่ได้ไปทำการพัฒนาหรือประยุกต์ให้เข้ากับการจัดการเอกสารบนเว็บเพจที่แตกต่างจากนี้ได้อีก เช่น อาจกำหนดให้มีขั้นตอนอื่นๆเพิ่มเติม หรือเปลี่ยนแปลง Workflow ของระบบให้เหมาะสมกับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับผู้ที่ต้องการจะนำระบบไปศึกษาหรือนำไปพัฒนาต่อไปในอนาคต

- 5.3.1 สามารถขยายความสามารถของระบบ ให้สามารถจัดการข้อมูลในรูปแบบที่ต้องการได้มากขึ้น หรือเพิ่มฟังก์ชันการทำงานในบางขั้นตอน เช่น ขยายความสามารถของระบบ ให้สามารถเปรียบเทียบความแตกต่างของข้อมูลในแต่ละเวอร์ชันได้, เพิ่มความสามารถในการตรวจสอบคำที่พิมพ์ผิด, เพิ่มหรือลดขั้นตอนการทำงานใน Workflow ตามความเหมาะสมได้ เป็นต้น
- 5.3.2 ควรเพิ่มความสามารถในการค้นหาข้อมูลรายการเก่าได้
- 5.3.3 ถ้าใช้งานระบบในระยะยาว ควรคำนึงถึงการออกแบบให้ระบบสามารถรองรับการใช้ในสถานะที่ข้อมูลเป็นจำนวนมาก เนื่องจากเมื่อใช้งานระบบไประยะหนึ่งแล้ว จะพบว่ามีการสะสมของข้อมูลเป็นจำนวนเพิ่มมากขึ้นไปเรื่อยๆ ซึ่งอาจทำให้การทำงานของระบบช้าลง



บรรณานุกรม

B. Surjanto, N.Ritter and H.Loesser, 2000, "XML Content Management based on Object-Relational".

James Conard, Patrick dengler, Brian Francis, Jay Glynn, Burton Harvey, Billy Hollis, Rama Ramachandran, John Schenken, Scott Short and Chris Ullman, January 2001, "Introducing .NET".

Microsoft.NET Framework. <http://msdn.microsoft.com/netframework/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวนรีนุช ใจสมานประเสริฐ
วัน-เดือน-ปี เกิด	2 ตุลาคม 2517
สถานที่เกิด	กรุงเทพฯ
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตร์บัณฑิต(วิทยาการคอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยหอการค้าไทย
ปีการศึกษาที่สำเร็จการศึกษา	2539
ตำแหน่งหน้าที่ปัจจุบัน	Programmer Analyst
สถานที่ทำงาน	บริษัท โปรเกรสซอฟต์แวร์ จำกัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้