

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ตารางนัดหมายบนเว็บ
Web Based Schedule

โดย

นาย สมทบ กุลมงคล

รหัส 43067078

อาจารย์ที่ปรึกษา

ดร. จันท์บูรณ์ สถิตวิริยวงศ์



H001924

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2545

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วัน เดือน ปี.....	19 ธ.ค. 2550
เลขทะเบียน.....	H001924
เลขเรียกหนังสือ.....	วิท. ๙๒๔๙๓ ZE45
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ตารางนัดหมายบนเว็บ
นักศึกษา	นาย สมทบ กุลมงคล
อาจารย์ที่ปรึกษา	ดร. จันทร์บุรณธ์ สติตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

โครงการนี้เป็นการออกแบบและพัฒนาโปรแกรมตารางนัดหมายบนเว็บ เพื่อเป็นต้นแบบที่จะนำไปพัฒนาต่อไปให้สามารถใช้งานได้จริง การเรียกใช้งานจะใช้งานผ่านทางเว็บเบราว์เซอร์ที่เครื่องคอมพิวเตอร์ของผู้ใช้แต่ละคน โดยผู้ใช้สามารถที่จะนัดหมายครั้งละหนึ่งคนหรือครั้งละหลายคนพร้อมกันก็ได้

Title	Web Based Schedule
Student	Mr. Somtob Kulmongkol
Advisor	Dr. Chanboon Sathitwiriawong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

Abstract

This project is designing and developing program “ Web Based Schedule ”. It is an original that can develop in the future. For using , this program can use pass web browser at a computer of each user. The user can make an appointment with the only one or many people concurrently.

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
สารบัญ.....	III
สารบัญตาราง.....	V
สารบัญภาพ.....	VI
บทที่	
1. บทนำ.....	1
1.1 ที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ลักษณะและขอบเขตของโครงการ.....	1
1.4 ประโยชน์ที่ได้รับจากโครงการ.....	3
2. ทฤษฎีพื้นฐานเบื้องต้นเกี่ยวกับฐานข้อมูล.....	4
2.1 การแบ่งระดับข้อมูล.....	4
2.2 อินสแตนและเค้าร่าง (Instance and Schema).....	4
2.3 ภาษาสำหรับการใช้ข้อมูล.....	5
2.4 การใช้ภาษาสำหรับการ ใช้ข้อมูล.....	5
2.5 การจัดเก็บข้อมูล.....	5
2.6 ภาษาเอสคิวแอล (SQL).....	6
2.7 รูปแบบบรรทัดฐาน (Normal form).....	11
2.8 สถาปัตยกรรมฐานข้อมูลแบบไคลเอนต์เซิร์ฟเวอร์.....	13
3. ความรู้เบื้องต้นเกี่ยวกับเว็บเพจ.....	19
3.1 ลักษณะของเว็บเพจแบบสถิต (Static Web Pages).....	19
3.2 ลักษณะของเว็บเพจแบบพลวัต (Dynamic Web Pages).....	20
3.3 แอ็กทีฟเซิร์ฟเวอร์เพจส์ คืออะไร?.....	24
3.4 แอ็กทีฟเซิร์ฟเวอร์เพจส์ ทำงานอย่างไร?.....	25

3.5 ภาษาสคริปต์กับเอกทีพีเซิร์ฟเวอร์เพจส์.....	27
4. การออกแบบโปรแกรม.....	30
4.1 แนวความคิดในการออกแบบ.....	30
4.2 การออกแบบขั้นตอนการทำงานของระบบ.....	31
4.3 การออกแบบฐานข้อมูล.....	35
4.4 การออกแบบโปรแกรมฝั่งผู้ใช้.....	40
4.5 การออกแบบโปรแกรมฝั่งผู้ดูแลระบบ.....	52
5. สรุปผล.....	55
6. บรรณานุกรม.....	56
7. ประวัติผู้เขียน.....	57



สารบัญตาราง

หน้า

ตารางที่

4.1	รายละเอียดของตาราง TabAppoint.....	35
4.2	รายละเอียดของตาราง TabForget.....	36
4.3	รายละเอียดของตาราง TabMail.....	37
4.4	รายละเอียดของตาราง TabPermit.....	38
4.5	รายละเอียดของตาราง TabUserInfo.....	38
4.6	รายละเอียดของตาราง TabWork.....	39



สารบัญภาพ

หน้า

รูปที่

2.1	แอปพลิเคชันแบบวันเทียร์.....	14
2.2	แอปพลิเคชันแบบทูเทียร์.....	15
2.3	แอปพลิเคชันแบบทรีเทียร์.....	16
3.1	ลักษณะการทำงานของเว็บเพจแบบสถิต.....	19
3.2	การทำงานของซีจีไอและไอเอสเอพีไอ.....	21
3.3	การทำงานของซีจีไอสคริปต์และไอคิซี.....	22
3.4	ลักษณะการทำงานของเอกซีพีเซิร์ฟเวอร์เพจส์.....	24
4.1	การออกแบบโปรแกรม.....	30
4.2	ผู้ใช้และผู้ดูแลระบบในโปรแกรม.....	31
4.3	Context Diagram.....	32
4.4(ก)	Data Flow Diagram.....	33
4.4(ข)	Data Flow Diagram.....	34
4.5	ความสัมพันธ์ของตาราง.....	35
4.6	การใช้งานของผู้ใช้.....	40
4.7	หน้าจอหลัก.....	41
4.8	ขั้นตอนการลงทะเบียนใหม่.....	41
4.9	หน้าจอแบบฟอร์มการลงทะเบียนใหม่.....	42
4.10	ขั้นตอนการเปลี่ยนรหัสผ่าน.....	42
4.11	หน้าจอแบบฟอร์มการเปลี่ยนรหัสผ่าน.....	43
4.12	ขั้นตอนการขอรหัสผ่านในกรณีลืมรหัสผ่าน.....	43
4.13	หน้าจอแบบฟอร์มการขอรหัสผ่านในกรณีลืมรหัสผ่าน.....	44
4.14	ขั้นตอนของตารางเวลา.....	45
4.15	หน้าจอตารางเวลา.....	45
4.16	ขั้นตอนการกำหนดคสิทธิ์.....	46
4.17	หน้าจอการกำหนดคสิทธิ์.....	47

4.18	หน้าจ้อข้อความ.....	48
4.19	ขั้นตอนการนัดหมาย.....	49
4.20	หน้าจ้อการนัดหมาย.....	50
4.21	หน้าจ้อการกำหนดคุณสมบัติ.....	51
4.22	การใช้งานของผู้ดูแลระบบ.....	52
4.23	ขั้นตอนการตอบรับผู้ใช้ที่ลงทะเบียนใหม่.....	53
4.24	ขั้นตอนการตอบรับผู้ใช้ที่ลืมรหัสผ่าน.....	53
4.25	ขั้นตอนการระงับสิทธิ์และตัดสิทธิ์ผู้ใช้.....	54



บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

ปัจจุบันการพบปะพูดคุยหรือทำกิจกรรมร่วมกันในเรื่องต่างๆของกลุ่มบุคคลนั้น จำเป็นต้องมีการนัดหมายก่อนทุกครั้ง ซึ่งการที่จะนัดหมายใครสักคนหนึ่งบางครั้งอาจเป็นไปได้ด้วยความยากลำบาก โดยเฉพาะการนัดหมายหลายๆคนในเวลาเดียวกัน เช่น โทรศัพท์ไปนัดหมายแล้วแต่ไม่เจอตัวหรือบางครั้งใช้การทักไลน์ไว้ซึ่งก็อาจจะไม่ถึงผู้รับได้ เหตุการณ์เหล่านี้ล้วนแต่เป็นการนัดหมายที่ไม่ประสบความสำเร็จ ซึ่งโอกาสที่จะเกิดการนัดหมายที่ไม่สำเร็จนี้ก็มีสูง ดังนั้นจึงเป็นดีถ้าหากมีโปรแกรมที่ทำหน้าที่เป็นสื่อกลางนัดหมายบุคคลต่างๆ เพื่อลดปัญหาซึ่งอาจเกิดขึ้นได้เหล่านี้ และจะช่วยให้การนัดหมายประสบความสำเร็จมากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการทำงานร่วมกันของเซิร์ฟเวอร์, ไลอเนค และ ฐานข้อมูล
2. เพื่อศึกษาการเขียนโปรแกรมแบบ ไลอเนคเซิร์ฟเวอร์ติดต่อกับฐานข้อมูลผ่านเว็บเบราว์เซอร์
3. เพื่อพัฒนาต้นแบบของ โปรแกรมนัดหมายบนเว็บ ที่สามารถนำมาใช้งานและพัฒนาต่อในอนาคตได้

1.3 ลักษณะและขอบเขตของโครงการ

โครงการนี้ได้ถูกพัฒนาขึ้นเพื่อใช้ในการจัดการทางด้านกรนัดหมาย รวมไปถึงการทำหน้าที่เป็นสมุดบันทึกกิจกรรมต่างๆ โดยที่ผู้ใช้สามารถเรียกใช้งานผ่านทางเว็บเบราว์เซอร์ที่เครื่องของผู้ใช้แต่ละคนได้ตลอดเวลา โดยสามารถแบ่งรายละเอียดของโปรแกรมได้ 2 ส่วนคือ

1. ส่วนตัวโปรแกรม

ใช้เทคโนโลยีแอคทีฟเซิร์ฟเวอร์เพจส์ (Active Server Pages) ของบริษัทไมโครซอฟท์ ในการพัฒนา

1.1 ส่วนตัวโปรแกรมฝั่งผู้ใช้

- การลงทะเบียนใหม่
- การเปลี่ยนรหัสผ่าน
- การขอรหัสผ่านในกรณีลืมรหัสผ่าน
- การส่งข้อความถึงผู้ดูแลระบบ
- ตารางเวลา
- การกำหนดสิทธิ์
- การสร้างและลบกลุ่ม
- การส่งและอ่านข้อความ
- การนัดหมาย
- การกำหนดคุณสมบัติ
- การออกจากระบบ

1.2 ส่วนตัวโปรแกรมฝั่งผู้ดูแลระบบ

- การตอบรับผู้ใช้ที่ลงทะเบียนใหม่
- การตอบรับผู้ใช้ที่ลืมรหัสผ่าน
- การตอบรับข้อความที่ผู้ใช้ส่งมาถึง
- การประกาศข้อความ
- การระงับสิทธิ์และตัดสิทธิ์ผู้ใช้
- การเพิ่มและลบสถานที่
- การเพิ่มและลบตำแหน่ง
- การเปลี่ยนตำแหน่งของผู้ใช้
- การออกจากระบบ

2. ส่วนฐานข้อมูล

ใช้ฐานข้อมูลเป็น ไมโครซอฟท์แอคเซส (Microsoft Access)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่ได้รับจากโครงการ

- ช่วยให้การนัดหมายสะดวกและมีประสิทธิภาพมากขึ้น ลดโอกาสการนัดหมายที่ไม่ประสบความสำเร็จให้น้อยลง
- ทำหน้าที่เป็นสมุดเวลาส่วนตัวได้ ช่วยให้สามารถตรวจดูกิจกรรมและการนัดหมายที่มี ทำให้ไม่ลืมการนัดหมายต่างๆ
- เป็นต้นแบบในการพัฒนาโปรแกรมนัดหมายบนเว็บต่อไปในอนาคตได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐานเบื้องต้นเกี่ยวกับฐานข้อมูล*

2.1 การแบ่งระดับข้อมูล

2.1.1 ระดับภายใน เป็นระดับที่ต่ำที่สุดอันได้แก่ระดับของการจัดเก็บข้อมูลจริงๆซึ่งเนื้อหาในส่วนนี้เกี่ยวข้องกับโครงสร้างของข้อมูล

2.1.2 ระดับหลักการ เป็นระดับที่อยู่ถัดขึ้นมาอันได้แก่ระดับของการมองเอนทิตี (Entity) และความสัมพันธ์ระหว่างเอนทิตีทั้งหมดรวมทั้งกฎเกณฑ์ต่างๆที่เกี่ยวกับข้อมูลและผู้ที่มิสิทธิ์จะใช้ ข้อมูลในระดับนี้จะอยู่ในความสนใจของ DBA เพราะจะต้องเป็นผู้ออกแบบและควบคุมการใช้งานข้อมูล

2.1.3 ระดับภายนอก เป็นระดับที่อยู่สูงที่สุดอันเป็นระดับข้อมูลที่จะมองเห็นจากการใช้งานของผู้ใช้

2.2 อินสแตนซ์และเค้าร่าง (Instance and Schema)

การเรียกใช้และแก้ไขข้อมูล จะทำให้ค่าของข้อมูลในระบบมีการเปลี่ยนแปลงตลอดเวลา เราเรียกข้อมูลในฐานข้อมูลในขณะใดขณะหนึ่งว่า อินสแตนซ์ (Instance) ของฐานข้อมูล ส่วนเค้าร่างที่ได้จากการออกแบบฐานข้อมูลโดยส่วนรวมนั้นเรียกว่า สคีมา (Schema) ซึ่งตามปกติแล้วไม่ควรมีการเปลี่ยนแปลงหรือหากมีก็ควรจะไม่บ่อยนัก การแบ่งโครงสร้างสามารถแบ่งออกเป็นหลายๆระดับคล้ายๆกับของข้อมูลการกำหนดเค้าร่างของฐานข้อมูลทั้ง 3 ระดับ กระทำได้โดยการใช้ภาษาสำหรับนิยามข้อมูล (Data Definition Language : DDL) ผลจากการคอมไพล์ประโยคที่เขียนด้วย DDL จะทำให้เกิดตารางที่จะจัดเก็บข้อมูล ในไฟล์ชนิดหนึ่งที่เรียกว่า ดาต้าดิคชันนารี (Data Dictionary) ซึ่งจะเก็บข้อมูลที่เกี่ยวข้องกับโครงสร้างที่ได้จากการออกแบบฐานข้อมูลนั้นๆ

- * อ้างอิงจาก 1. การใช้งานฐานข้อมูลเชิงสัมพันธ์ Microsoft SQL Server โดย บัณฑิต จามรภูติ
2. เรียนรู้การทำงานของไคลเอ็นต์เซิร์ฟเวอร์ เรียบเรียงโดย นรุตม์ วรรณเลิศลักษณ์

2.3 ภาษาสำหรับการใช้ข้อมูล

การใช้ข้อมูลในระบบฐานข้อมูลแบ่งออกเป็น 3 ประเภทคือ

2.3.1 การเรียกข้อมูล

2.3.2 การเพิ่มข้อมูลใหม่เข้าไป

2.3.3 การลบข้อมูล

การใช้ข้อมูลในระดับของผู้ใช้นั้นก็สามารถกระทำได้โดยผ่านทางดีบีเอ็มเอส (DBMS) โดยการใช้ภาษาสำหรับการใช้ข้อมูล (Data Manipulation Language : DML)

2.4 การใช้ภาษาสำหรับการใช้ข้อมูล

การใช้ภาษาสำหรับการใช้ข้อมูล สามารถแบ่งออกได้เป็น 2 ชนิด

2.4.1 Procedural ในการใช้ DML แบบนี้เป็นหน้าที่ของผู้ใช้ที่จะต้องระบุว่าการข้อมูลอะไร และจะเอาข้อมูลนั้นมาได้ด้วยวิธีการใด คล้ายๆกับการเขียนโปรแกรมด้วยภาษาชั้นสูง

2.4.2 Nonprocedural สำหรับ DML แบบนี้ ผู้ใช้เพียงระบุว่าการข้อมูลอะไร โดยไม่ต้องบอกวิธีการเลย เป็นลักษณะคล้ายๆกับภาษาธรรมชาติ ซึ่งจะมีความง่ายกว่ามาก การจัดฐานข้อมูลได้นั้นต้องมีผู้บริหารฐานข้อมูล ซึ่งจะเรียกว่า DBA เป็นผู้ที่ทำหน้าที่ควบคุม

2.5 การจัดเก็บข้อมูล

การจัดเก็บข้อมูลแบ่งออกเป็น 3 ชนิด

2.5.1 การจัดเก็บข้อมูลสำหรับความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ยกตัวอย่างเช่น พนักงานขายแต่ละคนมีสิทธิ์ขายของให้แก่ลูกค้าเพียงคนเดียวเท่านั้น ซึ่งเป็นการแสดงถึงความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างพนักงานขายและลูกค้า

2.5.2 การจัดเก็บข้อมูลสำหรับความสัมพันธ์แบบหนึ่งต่อกลุ่ม

ยกตัวอย่างเช่น พนักงานขายแต่ละคนติดต่อกับลูกค้าได้มากกว่า 1 คน แต่ลูกค้าแต่ละคนต้องซื้อของจากพนักงานขายเพียงคนเดียวเท่านั้น ซึ่งเป็นการแสดงถึงความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างพนักงานขายและลูกค้า

2.5.3 การจัดเก็บข้อมูลสำหรับความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ยกตัวอย่างเช่น ลูกค้าสามารถสั่งซื้อสินค้าได้มากกว่า 1 อย่างในการซื้อแต่ละครั้ง และในทำนองเดียวกันก็อนุญาตให้สินค้า 1 อย่าง ปรากฏอยู่ในรายการสั่งซื้อได้มากกว่า 1 ครั้ง ซึ่งเป็นการแสดงถึงความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างพนักงานขายและลูกค้า

2.6 ภาษาเอสคิวแอล (SQL)

ภาษาสำหรับการใช้ข้อมูลที่เป็นที่นิยมก็คือ ภาษาเอสคิวแอล (SQL) คุณสมบัติของภาษานี้ก็คือ สามารถเรียกดูข้อมูลตามที่เราระบุในความต้องการได้ สามารถใช้ในการเปลี่ยนแปลงเพิ่มเติมและลบข้อมูลออกจากระบบ นอกจากนี้ยังสามารถใช้ในการสร้างระบบฐานข้อมูลด้วย ดังนั้นเราจึงสามารถพูดได้ว่าภาษาสำหรับการใช้ข้อมูลนี้เป็นภาษาสำหรับนิยามข้อมูล

2.6.1 การเรียกดูข้อมูล

ประโยคคำสั่งสำหรับการเรียกดูข้อมูลได้แก่ คำสั่ง Select ซึ่งมีรูปแบบของคำสั่งนี้

SELECT ... FROM ... WHERE ... ORDER BY ...

ซึ่งผู้ใช้สามารถระบุส่วนต่างๆ ในคำสั่งได้ดังนี้

- ส่วนหลัง SELECT ใช้ระบุ คอลัมน์หรือกลุ่มของคอลัมน์ที่เราต้องการดูข้อมูล
- ส่วนหลัง FROM ใช้ระบุ ชื่อของตารางที่เราต้องการดูข้อมูล
- ส่วนหลัง WHERE ใช้ระบุ เงื่อนไขของข้อมูลที่เราสนใจซึ่งอาจจะมีหรือไม่มีก็ได้
- ส่วนหลัง ORDER BY ใช้ระบุ วิธีการเรียงลำดับการแสดงผลข้อมูล ซึ่งอาจจะมีหรือไม่มีก็ได้

ตัวอย่างของประโยคคำสั่งก็คือ

SELECT * FROM Tablename

ซึ่งจะแสดงข้อมูลทั้งหมดจากตารางที่ระบุออกมา ขอให้สังเกตว่าเราใส่เครื่องหมาย * ไว้ข้างหลัง SELECT หมายความว่า ให้เรียกดูคอลัมน์นั่นเอง ถ้าเราต้องการระบุเงื่อนไขของข้อมูลเฉพาะแถวที่ต้องการเรียกดู ก็ใส่เครื่องหมายไว้ข้างหลังคำว่า WHERE เช่น

SELECT * FROM Tablename WHERE Condition

ซึ่งอาจมีมากกว่า 1 เงื่อนไขก็ได้โดยการเชื่อมด้วย AND หรือ OR ตัวอย่างเช่น เรามีตารางลูกค้าที่ประกอบด้วยคอลัมน์ 6 คอลัมน์เช่น

**CUSTOMER (CUSTOMER_NUMBER , NAME , ADDRESS , CURRENT_BALANCE ,
CREDIT_LIMIT , SLSREP_NUMBER)**

ถ้าเราสนใจจะดูข้อมูลเฉพาะเพียงบางคอลัมน์ก็สามารถทำได้ โดยระบุชื่อคอลัมน์ไว้ข้างหลัง SELECT โดยคั่นระหว่างชื่อคอลัมน์ด้วยเครื่องหมาย “,” เช่น

**SELECT CUSTOMER_NAME , CREDIT_LIMIT , ADDRESS FROM CUSTOMER
WHERE SLSREP_NUMBER = 6**

ในลักษณะนี้จะได้คอลัมน์ CUSTOMER_NAME , CREDIT_LIMIT , ADDRESS จากตาราง CUSTOMER ซึ่งมีเงื่อนไข SLSREP_NUMBER = 6 แสดงออกมา คำสั่ง SELECT นี้ยังสามารถกำหนดให้มีการเรียงลำดับผลลัพธ์ได้ตามค่าที่อยู่ในคอลัมน์ที่ระบุโดยการใส่ ORDER BY เช่น

**SELECT CUSTOMER_NAME , CREDIT_LIMIT , ADDRESS FROM CUSTOMER
WHERE SLSREP_NUMBER = 6
ORDER BY CREDIT_LIMIT , CURRENT_BALANCE , CUSTOMER_NAME**

ผลลัพธ์ที่ออกมานั้นก็จะเป็นคอลัมน์ CUSTOMER_NAME , CREDIT_LIMIT , ADDRESS จากตาราง CUSTOMER และเรียงลำดับ CREDIT_LIMIT , CUSTOMER_NAME

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 การเพิ่มเติมข้อมูล

คำสั่ง INSERT ใช้ในการเพิ่มข้อมูลแถวใหม่ลงในตาราง โดยอาจจะระบุข้อมูลไว้ในคำสั่งเลขหรือดึงข้อมูลออกจากตารางอื่นก็ได้ โดยบังคับว่าข้อมูลที่จะเติมลงไปนี้จะต้องมีข้อมูลในแต่ละคอลัมน์ในตารางที่จะเพิ่มเติมลงได้

รูปแบบของคำสั่ง INSERT เป็นดังนี้

INSERT INTO Tablename VALUES (Value)

หรือ

INSERT INTO Tablename : SELECT Statement

ตัวอย่างประโยค

INSERT INTO SLSREP VALUES (14 , 'Somchai' , '123 Ladprow 15' , 0.05)

การแสดงผลของประโยคนี้ก็จะเพิ่มข้อมูลเกี่ยวกับพนักงานขายคนใหม่เข้าไปในตาราง SLSREP โดยกำหนดให้มีหมายเลข 14 ชื่อ Somchai อยู่บ้านเลขที่ 123 Ladprow 15 โดยเริ่มต้นให้มีอัตราคอมมิชชั่น 0.05 ของราคาสินค้าที่ขายได้

2.6.3 การแก้ไขข้อมูล

คำสั่ง UPDATE ใช้ในการแก้ไขเปลี่ยนแปลงค่าในคอลัมน์ที่ต้องการในแถวที่มีเงื่อนไขสอดคล้องกับที่ระบุไว้

รูปแบบของคำสั่ง UPDATE เป็นดังนี้

UPDATE Tablename SET Columnname = Value WHERE Condition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างประโยค

UPDATE CUSTOMER SET NAME = 'Sommai' WHERE CUSTOM = 256

จะเปลี่ยนชื่อลูกค้าหมายเลข 256 ในตาราง CUSTOMER ให้เป็น Sommai

2.6.4 การลบข้อมูล

คำสั่ง DELETE ใช้ในการลบแถวของข้อมูลที่สอดคล้องกับเงื่อนไขที่ระบุออกจากตาราง

รูปแบบของคำสั่ง DELETE เป็นดังนี้

DELETE Tablename WHERE Condition

ตัวอย่างประโยค

DELETE CUSTOMER WHERE NAME = 'Somsri'

จะลบข้อมูลของลูกค้าที่ชื่อ Somsri ออกจากระบบ ซึ่งถ้าผู้ใช้ไม่ได้ส่วนของ WHERE เอาไว้ก็จะมีผลทำให้ลบข้อมูลทุกแถวออกจากตารางที่ระบุ

2.6.5 คำสั่งสำหรับนิยามตาราง

การใช้คำสั่งนี้เราจะต้องใส่ชื่อตารางพร้อมทั้งชื่อคอลัมน์, ชนิด และขนาดของข้อมูลของแต่ละคอลัมน์ โดยเราไม่สามารถกำหนดข้อบังคับ ตัวอย่างเช่น

```
CREATE TABLE SLSREP
( SLSREP_NUMBER          DECIMAL ( 2 ),
  SLSREP_NAME            CHAR ( 15 ),
  SLSREP_ADDRESS        CHAR ( 25 ),
  TOTAL_COMMISSION      DECIMAL ( 7 , 2 ),
  COMISSION_RATE        DECIMAL ( 2 , 2 ) )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยคำสั่งนี้ จะทำให้มีการนิยามตารางของพนักงานขาย ซึ่งประกอบด้วยคอลัมน์ 5 คอลัมน์ ได้แก่ รหัสประจำตัว , ชื่อ , ที่อยู่ , ค่าคอมมิชชั่น และอัตราคอมมิชชั่น ส่วนคำสั่งเมนการลบตารางคือ

DROP TABLE Tablename

2.6.6 คำสั่งสำหรับนิยามวิวของผู้ใช้

การวิวนี้เราถือว่าเป็นเพียงตารางสมมติเท่านั้นก็คือ ผู้ที่มีสิทธิใช้วิวจะสามารถใช้คำสั่งแฮตคิวแอล เรียกดูข้อมูลจากวิวนี้ได้เต็มที่

รูปแบบของคำสั่งคือ

```
CREATE VIEW Viewname
[ ( Columnname1 , Columnname2 , ... ) ]
AS SELECT ( SELECT Statement )
```

ซึ่งจะเป็นการสร้างตารางขึ้นมาใหม่ตามชื่อที่ระบุ ซึ่งจะประกอบด้วย คอลัมน์และแถวของข้อมูล ตามที่ได้จากประโยค SELECT ตารางที่สร้างขึ้นใหม่นี้ ถ้าไม่ได้ระบุชื่อคอลัมน์เอาไว้ก็จะมีชื่อคอลัมน์ตรงตามตารางเดิมที่ระบุไว้ในส่วนของประโยค SELECT

ตัวอย่างประโยค

```
CREATE VIEW CUST12 AS
SELECT CUSTOMER_NUMBER , NAME , ADDRESS , CURRENT_BALANCE
FROM CUSTOMER
WHERE CUSTOMER_NUMBER = 12
```

เป็นการสร้างตารางใหม่ชื่อ CUST12 ขึ้นจากข้อมูลในตาราง CUSTOMER โดยเลือกเฉพาะลูกค้าที่ใช้บริการของพนักงานขายหมายเลข 12 ซึ่งในลักษณะตัวอย่างนี้ ผู้บริหารฐานข้อมูลอาจจะกำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พนักงานขายหมายเลข 12 มีสิทธิใช้เฉพาะตาราง CUST12 โดยไม่มีสิทธิใช้ตาราง CUSTOMER ส่วนรวมก็ได้ ส่วนคำสั่งที่ตรงข้ามกับ CREATE VIEW ก็คือ คำสั่ง DROP VIEW นั้นก็จะเป็นคำสั่งยกเลิกการใช้วิว

2.6.7 คำสั่งสำหรับสร้างตารางดัชนี

ในการเรียกใช้ข้อมูล การระบุดัชนีที่เหมาะสมจะช่วยให้การค้นหาเป็นไปอย่างรวดเร็วยิ่งขึ้น นอกเหนือจากประโยชน์ในการเพิ่มความเร็วแล้ว การสร้างดัชนียังสามารถใช้ในการควบคุมค่าของข้อมูลได้ในระดับหนึ่ง กล่าวคือ ถ้าเราต้องการสร้างหลักประกันว่าไม่ต้องการให้ข้อมูลในตารางมีค่าของคอลัมน์ใดซ้ำกันมากกว่า 1 แถว เช่น ในตารางสินค้าไม่ต้องการให้ชื่อสินค้าที่เหมือนกันปรากฏในข้อมูลมากกว่า 1 แถว ก็สามารถควบคุมได้โดยเพิ่มคำว่า UNIQUE เข้าไปในคำสั่งได้ดังนี้

```
CREATE UNIQUE INDEX PART_INDEX
ON PART ( PART_DESCRIPTION )
```

ซึ่งก็จะเป็นการเพิ่มความเร็ว ถ้าเราใช้ชื่อสินค้าในการค้นหาข้อมูลอีกด้วย และถ้าดัชนีตัวใดไม่เป็นที่ต้องการใช้อีกต่อไป ก็สามารถยกเลิกได้โดยการใช้คำสั่ง

```
DROP INDEX PART_INDEX
```

2.7 รูปแบบบรรทัดฐาน (Normal form)

การออกแบบริเลชันเพื่อที่จะสามารถเรียกใช้ข้อมูลได้อย่างสะดวกและมีประสิทธิภาพ โดยพยายามให้เกิดความซ้ำซ้อนของการเก็บข้อมูลน้อยที่สุด ศาสตร์แขนงหนึ่งที่ว่านี้เป็นพื้นฐานในการออกแบบข้อมูลได้แก่ การออกแบบให้อยู่ในรูปแบบบรรทัดฐาน (Normal form) โดยอาศัยความรู้เกี่ยวกับฟังก์ชันการขึ้นต่อกัน (Functional dependency) ของข้อมูล

2.7.1 ฟังก์ชันการขึ้นต่อกัน

นิยาม สมมติให้ B เป็นแอททริบิวต์ (Attribute) ตัวหนึ่ง และ A เป็นแอททริบิวต์อีกตัวหนึ่ง เรากล่าวว่า B มีฟังก์ชันการขึ้นต่อกับ A ก็ต่อเมื่อ ค่าของ A สามารถใช้ในการเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(DETERMINE) ค่าของ B ได้เพียง 1 ค่าเสมอ สัญลักษณ์ของการขึ้นต่อกันในลักษณะเช่นนี้ จะเขียนว่า $A \rightarrow B$ ซึ่งอ่านว่า B มีฟังก์ชันการขึ้นต่อกับ A หรือ A มีฟังก์ชันในการเลือก B

2.7.2 รูปแบบบรรทัดฐานระดับที่ 1

คือการปรับรีเลชันให้ไม่เป็นแบบบรรทัดฐาน ซึ่งได้รีเลชันที่มีข้อมูลในบางช่องมากกว่า 1 ค่า

นิยาม รีเลชันใดๆกล่าวได้ว่าอยู่ในรูปแบบบรรทัดฐานระดับที่ 1 ถ้ารีเลชันไม่มีกลุ่มที่ซ้ำกัน

2.7.3 รูปแบบบรรทัดฐานระดับที่ 2

แม้ว่าเราได้จัดรูปแบบในรูปของรีเลชันให้เป็นรูปแบบบรรทัดฐานระดับที่ 1 แล้ว แต่ในบางครั้งอาจมีปัญหาดังกล่าวเกิดขึ้นเช่น การแก้ไขข้อมูล , ความขัดแย้งของข้อมูล , การเพิ่มเติมของข้อมูล , การลบข้อมูล

นิยาม เราเรียกแอททริบิวต์ที่ไม่ใช่ส่วนใดส่วนหนึ่งของคีย์หลักว่าแอททริบิวต์นั้นคีย์ (Attribute NonKey)

นิยาม รีเลชันใดๆจะจัดอยู่ในรูปแบบบรรทัดฐานระดับที่ 2 ถ้ารีเลชันนั้นเป็นรูปแบบบรรทัดฐานระดับที่ 1 และไม่มีแอททริบิวต์นั้นคีย์ตัวใดขึ้นต่อกับส่วนใดส่วนหนึ่งของคีย์

2.7.4 รูปแบบบรรทัดฐานระดับที่ 3

ถึงแม้เราจะได้ทำให้เป็นบรรทัดฐาน (Normalize) ความสัมพันธ์ให้อยู่ในระดับที่ 2 แล้ว แต่ปัญหายังคงอยู่ในบางครั้งก็คือ การแก้ไขข้อมูล , ความขัดแย้งของข้อมูล , การเพิ่มเติมของข้อมูล , การลบข้อมูล

นิยาม เราเรียกแอททริบิวต์ใดๆก็ตามที่สามารถเลือก (Determine) แอททริบิวต์ตัวอื่นๆ ได้ว่า ตัวเลือก (Determinant) จากนิยามนี้เป็นที่ปรากฏชัดว่าคีย์หลักและคีย์คู่แข่งจะต้องเป็นตัวเลือกอย่างแน่นอน

นิยาม รีเลชันใดๆจะจัดอยู่ในรูปแบบบรรทัดฐานระดับที่ 3 ถ้ารีเลชันนั้นเป็นรูปแบบบรรทัดฐานระดับที่ 2 และตัวเลือกทุกตัวจะต้องเป็นคีย์คู่แข่ง

สำหรับการขจัดตัวเลือกที่ไม่ใช่คีย์คู่แข่งออกไปก็มีวิธีการดังนี้

1. ค้างเอทริวิตที่ขึ้นกับตัวเลือกที่ไม่ใช่คีย์คู่แข่งออกไปสู่รีเลชั่นใหม่
2. กำหนดให้ตัวเลือกที่เกี่ยวข้องเป็นคีย์ของรีเลชั่นใหม่

2.8 สถาปัตยกรรมฐานข้อมูลแบบไคลเอนต์เซิร์ฟเวอร์

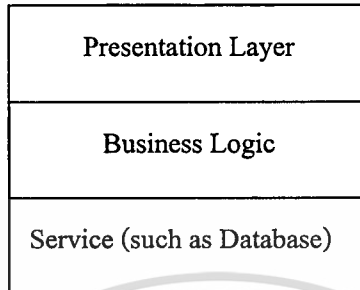
แอปพลิเคชันส่วนใหญ่จะมี 3 เลเยอร์ ส่วนบนสุดคือ 프리เซนต์ชันเลเยอร์ (Presentation Layer) ซึ่งช่วยในการติดต่อสื่อสารระหว่างคนกับเครื่องเรียกว่า ยูสเซอร์อินเทอร์เฟซ (User Interface) 프리เซนต์ชันเลเยอร์ทำหน้าที่จัดการอินพุตจากคีย์บอร์ด, เมาส์ หรืออุปกรณ์อื่นๆ รวมทั้งเอาพุตในรูปแบบของการแสดงผลบนหน้าจอ ส่วนตรงกลางคือ แอปพลิเคชันเลเยอร์ (Application Layer) หรือ บิซิเนส ลอจิก (Business Logic) ซึ่งเป็นส่วนของการทำงานที่บอกถึงลักษณะเฉพาะของแอปพลิเคชันเช่น แอปพลิเคชันลอจิก ทำให้เกิดความแตกต่างระหว่างระบบการตั้งสินค้าและระบบบริหารสินค้าคงคลัง เหตุผลที่เลเยอร์นี้ถูกเรียกว่า บิซิเนสลอจิก ก็เนื่องมาจากประกอบไปด้วยกฎข้อบังคับทางด้านธุรกิจ ซึ่งช่วยในการดำเนินกิจการ ส่วนล่างสุดคือ เซอร์วิส (Service) โดยจะให้บริการทั่วไปซึ่งร้องขอโดยเลเยอร์อื่นๆ รวมทั้งการให้บริการไฟล์, การให้บริการทางด้านการพิมพ์, การให้บริการด้านการติดต่อสื่อสาร และส่วนที่สำคัญที่สุดนั่นคือการให้บริการด้านฐานข้อมูล เพื่อให้ง่ายต่อการเข้าใจ จึงจำกัดเฉพาะเนื้อหาของโปรแกรมที่ต้องการการให้บริการด้านฐานข้อมูล ซึ่งประเด็นที่คำนึงถึงเป็นอันดับแรกเมื่อได้ยินคำว่า ไคลเอนต์เซิร์ฟเวอร์ ส่วนกฎข้อบังคับโดยทั่วไปยังใช้ได้กับการให้บริการทางอื่นๆ

จำนวนของเทียร์ (tier) ในไคลเอนต์เซิร์ฟเวอร์แอปพลิเคชันถูกกำหนดด้วยความสัมพันธ์ (แบบ tightly หรือ loosely) ของทั้ง 3 เลเยอร์ของโปรแกรมที่รวมอยู่ว่าเป็นแบบใด วันเทียร์ (One-tier) (ดังแสดงในรูปที่ 2.1) เป็นแอปพลิเคชันหนึ่งที่ 3 เลเยอร์ของโปรแกรมเชื่อมต่อกันอย่างเหนียวแน่น โดยเฉพาะอย่างยิ่งฟรีเซนต์ชันเลเยอร์ที่มีรายละเอียดเกี่ยวกับโครงสร้างฐานข้อมูล โดยมีแอปพลิเคชันเลเยอร์เป็นตัวเชื่อมยังฟรีเซนต์ชันเลเยอร์และเซอร์วิสเลเยอร์ ทั้ง 3 เลเยอร์รวมทั้งดาต้าเบสเอ็นจิน โดยทั่วไปแล้วจะทำงานอยู่บนเครื่องเดียวกัน

แอปพลิเคชันแบบวันเทียร์นั้นง่ายต่อการออกแบบและการเขียน โดยเฉพาะอย่างยิ่งเมื่อใช้ซอฟต์แวร์ทูลสมัยใหม่ที่มีอยู่ในปัจจุบัน คุณสามารถสร้างแอปพลิเคชันเลเยอร์แบบวันเทียร์ที่ใช้กับมัลติยูสเซอร์ได้ ด้วยการเรียกใช้พพลิเคชันบนเครื่องพีซีต่างๆที่ใช้ฐานข้อมูลร่วมกัน โดยฐานข้อมูลนั้นสามารถเก็บไว้บนพีซีเครื่องใดเครื่องหนึ่ง (กรณีเชื่อมต่อกันแบบเพียร์) หรือบนไฟล์เซิร์ฟเวอร์ สังเกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่าพีซีแต่ละตัวที่รันแอปพลิเคชันจะมีค่าเบสเอ็นจินเป็นของตัวเอง มีเพียงข้อมูลเท่านั้นที่ใช้ร่วมกัน และจะไม่มีค่าเบสลอจิกเป็นของตัวเอง

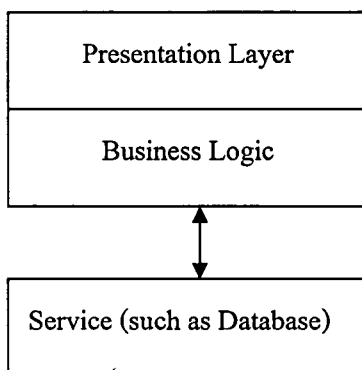


รูปที่ 2.1 แอปพลิเคชันแบบวันเทียร์

แอปพลิเคชันแบบวันเทียร์สามารถใช้งานได้กับมัลติยูสเซอร์จนถึงระดับหนึ่งเท่านั้น เมื่อจำนวนผู้ใช้เพิ่มมากขึ้น ปัญหาที่ตามมาคือ การทำงานของฐานข้อมูลทั้งหมดจะไปอยู่บนไคลเอนต์แต่ละตัว ตัวอย่างเช่น ถ้าโปรแกรมต้องการรายชื่อไคลเอนต์ทั้งหมดที่มีนามสกุลเป็น Smith ข้อมูลทั้งหมด (อินเดกซ์และเรคอร์ดของข้อมูล) ต้องส่งผ่านระบบเครือข่ายเพื่อไปยังฐานข้อมูล สำหรับการสอบถามข้อมูลที่มีความซับซ้อนนั้นจำเป็นต้องพิจารณาถึงจำนวนข้อมูลด้วย เพราะบางครั้งอาจหมายถึงการสอบถามข้อมูลทั้งฐานข้อมูล ในทางเทคนิคแล้วเป็นการยากที่จะจัดการกับค่าเบสเอ็นจินอิสระที่อยู่ตามเครื่องต่างๆให้ทำงานสอดคล้องกัน ในกรณี 2 ไคลเอนต์พยายามเข้าถึงหรืออัปเดตข้อมูลในเรคคอร์ดเดียวกัน ปัญหานี้เรียกว่า ปัญหาการถือครองเรคคอร์ด

วิธีเพิ่มประสิทธิภาพและแก้ไขปัญหาด้านเทคนิคที่เกี่ยวข้องกับ โปรแกรมวันเทียร์ที่ใช้กับมัลติยูสเซอร์คือ สถาปัตยกรรมไคลเอนต์เซิร์ฟเวอร์แบบทูเทียร์ (Two-tier) (ดังแสดงในรูปที่ 2.2) วิธีการเขียนแอปพลิเคชันชนิดนี้จะใกล้เคียงแบบวันเทียร์แต่มีข้อยกเว้นว่าค่าเบสเซอร์วิส (เอ็นจิน) นั้นไม่ได้ทำงานบนไคลเอนต์ แต่จะทำงานบนเซิร์ฟเวอร์แทน ซึ่งเซิร์ฟเวอร์จะมีฐานข้อมูลในตัวและยังมีวิธีที่ใช้ในการสื่อสารระหว่างแอปพลิเคชันเลเยอร์และค่าเบสเซอร์วิสด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



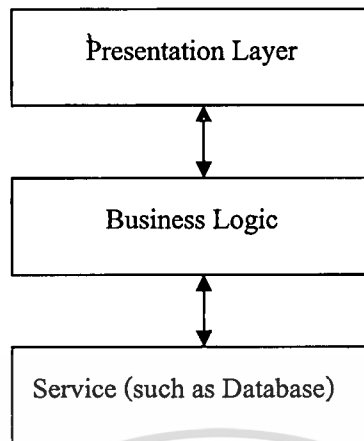
รูปที่ 2.2 แอปพลิเคชันแบบทิวเทียร์

วิธีที่นิยมใช้ในปัจจุบันคือ Structured Query Language หรือ เอสคิวแอล (SQL) ซึ่งเป็นคำสั่งที่สามารถรวมคำสั่งในการสอบถามข้อมูล (Query) ฐานข้อมูลที่มีความซับซ้อนให้อยู่ในรูปประโยคที่สั้นและกระชับคำสั่งเอสคิวแอล จะถูกส่งไปยังดาต้าเบสเซอร์ฟเวอร์เพื่อทำการประมวลผลและส่งข้อมูลที่ผู้ต้องการ (คำตอบ) กลับไปยังพีซีไคลเอนต์ สำหรับการออกแบบแอปพลิเคชันแบบทิวเทียร์นี้ จะมีเฉพาะส่วนของดาต้าเบสเซอร์วิสเท่านั้นที่แยกออกจากแอปพลิเคชัน ฟริเซนเทชั่นและบิซิเนสลोजิกเลเยอร์จะยังมีความสัมพันธ์กันอย่างเหนียวแน่น และจะมีรายละเอียดเกี่ยวกับฐานข้อมูลเช่นเดิม

แอปพลิเคชันแบบทิวเทียร์มีความซับซ้อนขึ้นเล็กน้อยในการเขียน แต่เครื่องมือในการพัฒนาส่วนใหญ่ให้การสนับสนุนอย่างมากในการพัฒนา เพื่อให้เกิดความสะดวกและรวดเร็วในการเขียนโปรแกรม เครื่องมือเหล่านี้มีความเหมาะสมเนื่องจากช่วยให้โปรแกรมแบบทิวเทียร์ สามารถพัฒนาได้เร็วเท่ากับแอปพลิเคชันแบบวันเทียร์ ข้อเสียที่มีคือ เรื่องของค่าใช้จ่าย เนื่องจากเครื่องมือส่วนใหญ่มีดาต้าเบสเอ็นจินที่สามารถจัดการกับการออกแบบในลักษณะวันเทียร์เท่านั้น (เช่น เจ็ตเอ็นจินที่ใช้ไมโครซอฟเอกเซสและวิซวลเบสิก) ถ้าแอปพลิเคชันเป็นแบบทิวเทียร์จะต้องใช้ดาต้าเบสตัวอื่น เช่น ออราเคิล , DB2 ของ ไอบีเอ็ม , ไซเบส หรือ เอสคิวแอลเซอร์ฟเวอร์ของไมโครซอฟ

ลักษณะของแอปพลิเคชันไคลเอนต์เซิร์ฟเวอร์แบบทรีเทียร์ (Three-tier) (ดังแสดงในรูปที่ 2.3) ก็เป็นไปตามที่คาดไว้คือ โปรแกรมเลเยอร์ทั้ง 3 แยกออกจากกันเป็นอิสระ การออกแบบแอปพลิเคชันแบบทรีเทียร์นั้น บิซิเนสลोजิกจะกลายเป็นอีกเซอร์วิสหนึ่ง และเซอร์วิสดังกล่าวดังกล่าวสามารถทำงานบนเครื่องคอมพิวเตอร์ของตัวเองได้ เมื่อบิซิเนสลोजิกกลายเป็นเซอร์วิสจึงนิยมเรียกว่า แอปพลิเคชันเซิร์ฟเวอร์ หรือเรียกง่าย ๆ ว่า แอปเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แอปพลิเคชันแบบทรีเทียร์

บ่อยครั้งที่แอปเซิร์ฟเวอร์ทำงานบนเครื่องเซิร์ฟเวอร์เดียวกันกับฐานข้อมูล บางครั้งการทำงานบนเครื่องเดียวกันให้ประโยชน์ในแง่ของประสิทธิภาพ ดังนั้นจึงเป็นทางเลือกแรกๆ ที่ผู้ใช้ส่วนใหญ่เลือก แต่ประโยชน์หลักของการมีแอปเซิร์ฟเวอร์คือ การจัดระบบให้เกิดความเหมาะสมเพื่อตอบสนองต่อความต้องการของผู้ใช้ได้มากที่สุด

ถ้าคุณทำธุรกิจที่มีการแบ่งเขตของการขาย ตัวอย่างเช่น บางเขตมีพนักงานขายจำนวนมากที่ใช้แอปเซิร์ฟเวอร์ร่วมกัน โดยที่แอปเซิร์ฟเวอร์นั้นทำงานบนคอมพิวเตอร์คนละเครื่องกัน เมื่อพนักงานขายคนหนึ่งซึ่งอยู่ในเขตที่ห่างไกลออกไป ต้องรับทั้งไคลเอนต์และแอปเซิร์ฟเวอร์บนเครื่องพีซีเครื่องเดียว ไม่ว่าจะไคลเอนต์และแอปเซิร์ฟเวอร์จะเป็นแบบใด แอปเซิร์ฟเวอร์ทั้งหมดควรได้รับการบริการด้านฐานข้อมูลจากคอมพิวเตอร์ส่วนกลางซึ่งอาจเป็นเมนเฟรม และไม่ว่าแอปเซิร์ฟเวอร์จะอยู่กระจายกันอย่างไร ผู้ใช้ทั้งหมดต้องสามารถใช้งานแอปพลิเคชันได้เหมือนกัน โดยไม่ขึ้นกับสถานที่ตั้งของแอปเซิร์ฟเวอร์ว่าเป็นเช่นไร

สำหรับแอปพลิเคชันไคลเอนต์เซิร์ฟเวอร์แบบทรีเทียร์ โดยทั่วไปพีริเซนต์เลเยอร์จะไม่มีรายละเอียดที่เกี่ยวกับฐานข้อมูล แต่พีริเซนต์เลเยอร์จะมีการติดต่อสื่อสารกับแอปพลิเคชันเซิร์ฟเวอร์โดยใช้เมสเสจที่กำหนดไว้แล้วแทน

วิธีที่ดีที่สุดในการทำความเข้าใจเรื่องนี้คือ ให้นึกถึงเรื่องเว็บไว้ ตัวอย่างเช่น บราวเซอร์จะไมู้ถึงโครงสร้างของฐานข้อมูลที่ thaiAmazon.com แต่ก็สามารถเข้าถึงฐานข้อมูลได้เมื่อต้องการสั่งซื้อหนังสือ ซึ่งเป็นผลมาจากอินเทอร์เน็ต โพรโตคอลที่ได้รับการออกแบบมาอย่างดีจึงช่วยให้ไคลเอนต์

(บราวเซอร์) ติดต่อกับแอปพลิเคชันเซิร์ฟเวอร์ (เว็บไซด์) ได้ นอกจากนี้ยังสามารถสั่งซื้อหนังสือจากเครื่องพีซีใดๆก็ได้ที่ไม่ใช่เครื่องของคุณ รวมทั้งสามารถเลือกใช้บริการเซิร์ฟเวอร์อื่นได้

แอปพลิเคชันแบบทรีเทียร์นั้นสร้างยากกว่าแอปพลิเคชันแบบทูเทียร์ อุปสรรคสำคัญคือสภาพแวดล้อมโดยรวมในการพัฒนาของซอฟต์แวร์ทั้งหลายนั้น ยังไม่เข้าใจถึงรูปแบบของทรีเทียร์เหมือนกับรูปแบบของทูเทียร์ ผลที่ตามมาคือ การเขียนแอปพลิเคชันแบบทรีเทียร์โดยส่วนใหญ่ต้องใช้การพัฒนาโปรแกรมด้วยมือ และแอปพลิเคชันแบบทรีเทียร์ยังยากต่อการออกแบบ เนื่องจากต้องใช้ในการเปรียบเทียบกับส่วนของทูเทียร์โดยตรง ผู้จัดจำหน่ายซอฟต์แวร์ทั้งหลายเริ่มวางจำหน่ายผลิตภัณฑ์ใหม่ๆที่สนับสนุนการพัฒนาทรีเทียร์หรือเอ็นเทียร์ (N-tier) ซึ่งเทคโนโลยีดังกล่าวยังสามารถพัฒนาต่อไปได้อีก

ประเด็นหนึ่งที่น่าสนใจคือ ความสัมพันธ์แบบทูเทียร์เกิดขึ้นระหว่างแอปเซิร์ฟเวอร์และดาต้าเบสเซิร์ฟเวอร์ แต่ไคลเอนต์ในแอปพลิเคชันแบบทรีเทียร์จะไม่มีรายละเอียดเกี่ยวกับฐานข้อมูลเหมือนกับที่แอปเซิร์ฟเวอร์มี

ปัจจุบันแนวความคิดของเอ็นเทียร์นั้นได้รับความนิยมพอสมควร แต่ข้อเท็จจริงคือทรีเทียร์เป็นสถาปัตยกรรมที่สูงสุดในขณะนี้ เอ็นเทียร์อ้างถึงข้อเท็จจริงที่ว่า แอปพลิเคชันเซิร์ฟเวอร์สามารถร้องขอบริการอื่นๆที่ต้องการ เพื่อให้การตอบสนองที่เหมาะสมกับไคลเอนต์ การส่งผ่านข้อมูลข่าวสารดังกล่าวนี้ อาจมีความซับซ้อนบ้าง

มิดเดิลแวร์ เป็นคำที่ใช้บ่อยในการอธิบายถึงแอปพลิเคชัน หรือบิซิเนสลोजิกที่อยู่ในแอปเซิร์ฟเวอร์ หรือบางครั้งอาจใช้อธิบายถึงบริการต่างๆทั่วไป ตัวอย่างเช่น ในการติดต่อสื่อสารทรีเทียร์อาจใช้ระบบของการส่งเมสเซจ ซึ่งเป็นความหมายของมิดเดิลแวร์ในแง่ของหน้าที่ระหว่างกลาง (หรือเป็นตัวกลาง) ของทรีเทียร์

สำหรับการออกแบบและสร้างแอปพลิเคชัน ไคลเอนต์เซิร์ฟเวอร์แบบทรีเทียร์ นักพัฒนาส่วนใหญ่ต้องตัดสินใจเกี่ยวกับจำนวนของบิซิเนสลोजิกที่ต้องใช้ในแอปเซิร์ฟเวอร์ที่เป็นอิสระจากกัน ตัวอย่างที่ดีในกรณีนี้คือ การตรวจสอบข้อมูลอินพุต ผู้ใช้ส่วนใหญ่ต้องการตอบสนองในทันทีเมื่อข้อมูลนั้นไม่ถูกต้อง ถ้าแอปเซิร์ฟเวอร์อยู่บนเครื่องคอมพิวเตอร์ต่างเครื่องกันก็อาจทำให้เกิดการล่าช้าในการตอบสนอง เป็นผลให้พีริเซมเทชั่นเลเยอร์ของไคลเอนต์ไม่สามารถรายงานได้ทันทีว่าข้อมูลที่รับเข้านั้นไม่ถูกต้อง ดังนั้นเพื่อให้มีประสิทธิภาพที่ดีขึ้น การตรวจสอบข้อมูลอินพุตจึงควรอยู่บนพีริเซมเทชั่นเลเยอร์ของไคลเอนต์แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าไม่มีบิซิเนสลोजิกหรือมีบิซิเนสลोजิกจำนวนไม่มากอยู่ในโคลเอนต์ (ในที่นี้ไม่รวมบนพีรีเซนเทชันเลเยอร์) โคลเอนต์ชนิดนี้จะเป็นแบบ ธิน โคลเอนต์ แต่ถ้ามีบิซิเนสลोजิกอยู่มากจะเรียกว่าแฟต โคลเอนต์ ตัวอย่างที่ดีที่สุดของธิน โคลเอนต์คือ เว็บบราวเซอร์ เนื่องจากมันสามารถเชื่อมต่อกับ แอปพลิเคชันที่แตกต่างกันได้ โดยไม่จำเป็นต้องมีรายละเอียดของแอปพลิเคชันเหล่านั้น และสามารถสร้างตัวติดต่อผู้ใช้ที่เหมาะสมได้ แนวความคิดทั้งหมดของเน็ตเวิร์กคอมพิวเตอร์มีประเด็นสำคัญอยู่ที่การสร้างค่าใช้จ่ายต่ำที่สุด และใช้อุปกรณ์ขนาดเล็กที่สุดในการทำงานกับเว็บเบราว์เซอร์



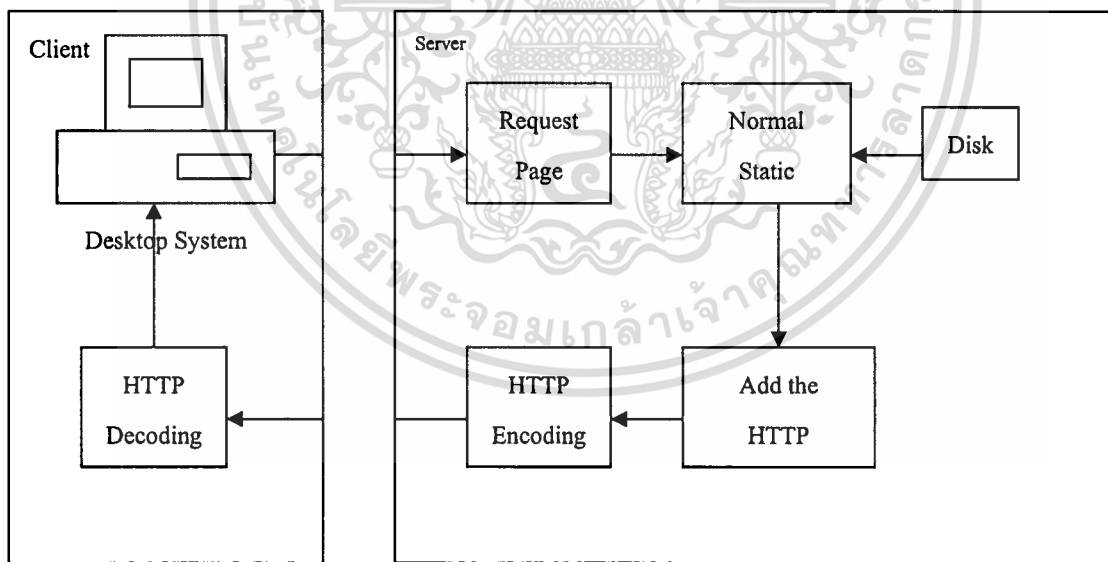
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ความรู้เบื้องต้นเกี่ยวกับเว็บเพจ*

3.1 ลักษณะของเว็บเพจแบบสถิต (Static Web Pages)

ลักษณะการทำงานของเว็บเพจแบบสถิต (ดังแสดงในรูปที่ 3.1) คือเมื่อผู้ใช้เรียกเว็บเพจผ่านทางบราวเซอร์หรือทางการเชื่อมโยงหลายมิติ (Hyperlink) คำสั่งที่ร้องขอหน้าเอกสารจะถูกส่งไปยังเซิร์ฟเวอร์ซึ่งเป็นที่เก็บไฟล์ของหน้าเอกสาร จากนั้นเว็บเซิร์ฟเวอร์จะโหลดหน้าเอกสารที่ร้องขอไปยังหน่วยความจำ ถ้าหน้าเอกสารนั้นเป็นเอกสารธรรมดา (Static HTML Page) เซิร์ฟเวอร์จะเพิ่มโพรโทคอลที่จำเป็นสำหรับการส่งข้อมูลให้กับเอกสารนั้นซึ่งเป็น โพรโทคอลที่ทำให้หน้าเอกสารสามารถส่งผ่านทางเอชทีทีพี (HTTP) ได้ จากนั้นเซิร์ฟเวอร์จะส่งเอกสารนั้นไปยังบราวเซอร์ ผู้ใช้จะได้เห็นรายละเอียดของหน้าเอกสารที่ได้ ซึ่งแปลมาจากเอกสารเอชทีเอ็มแอลที่ส่งมา แต่ทั้งหมดของเอกสารก็เป็นเพียงสำเนาของเอกสารที่อยู่ที่คิสบนเซิร์ฟเวอร์เท่านั้น



รูปที่ 3.1 ลักษณะการทำงานของเว็บเพจแบบสถิต

* อ้างอิงจาก ASP : Active Server Pages โดย Andrew M.Fedorchek และ David K.Resin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ลักษณะของเว็บเพจแบบพลวัต (Dynamic Web Pages)

เว็บเพจแบบพลวัตคือเอกสารที่ไม่ได้มาจากสำเนาที่อยู่ในคิสของเซิร์ฟเวอร์โดยตรงเท่านั้น แต่เป็นเอกสารที่มาจากการประมวลผลของเซิร์ฟเวอร์ด้วยเทคโนโลยีทางอินเทอร์เน็ตที่สามารถสร้างเว็บเพจแบบพลวัตได้เช่น

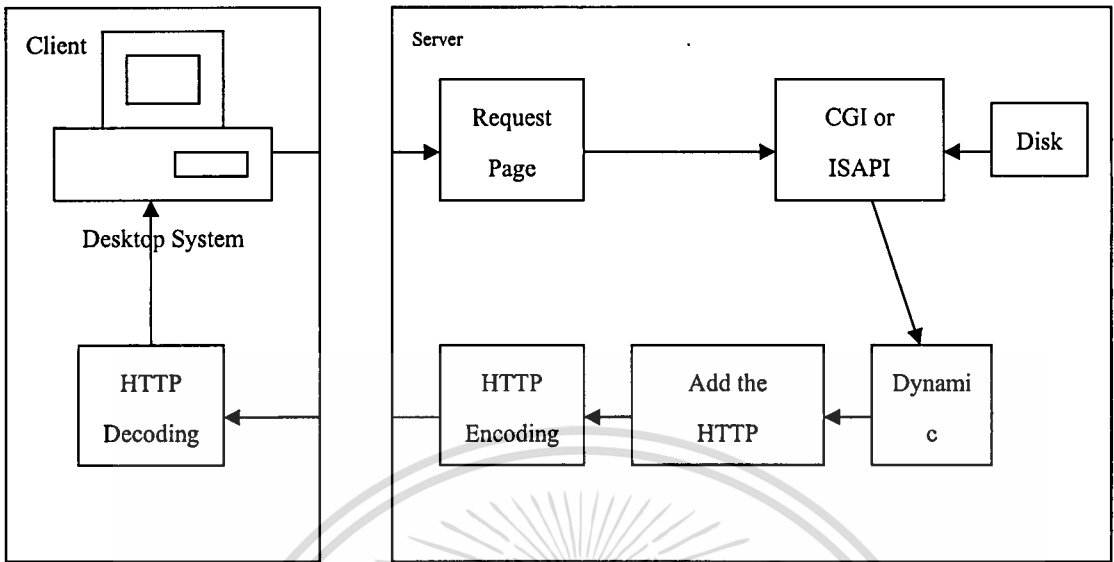
- ซีจีไอ (CGI) และไอเอสเอพีไอ (ISAPI)
- ซีจีไอสคริปต์และไอดีซี (Internet Database Connector)
- แอกทีฟเซิร์ฟเวอร์เพจส์ (Active Server Pages)

3.2.1 ซีจีไอและไอเอสเอพีไอ (CGI & ISAPI)

ซีจีไอและไอเอสเอพีไอมีลักษณะการทำงาน (ดังแสดงในรูปที่ 3.2) ดังนี้

ซีจีไอและไอเอสเอพีไอเป็นรูปแบบการสร้างเว็บเพจแบบพลวัตแบบหนึ่งที่ยอมรับโดยทั่วไป ซึ่งตัวเซิร์ฟเวอร์จะทำงานมากกว่าการบรรจุเอกสาร ซึ่งจะถูกร้องขอและส่งไปให้บราวเซอร์ ถ้าสิ่งถูกร้องขอจากบราวเซอร์เป็นซีจีไอและไอเอสเอพีไอ เซิร์ฟเวอร์จะโหลดแอปพลิเคชันนั้นขึ้นมาประมวลผล จากนั้นในตัวของแอปพลิเคชันนั้นจะสร้างผลลัพธ์ในรูปแบบของสารอักษร (A Stream of Text) และเอชทีเอ็มแอลโค้ด ซึ่งผลลัพธ์ที่ได้จะถูกเก็บไว้ในเอกสารชั่วคราว (Temporary page) ในเซิร์ฟเวอร์ จากนั้นผลลัพธ์จะถูกบรรจุ (Packaged up) สำหรับส่งผ่านทางเอชทีทีพีเพื่อส่งเอกสารไปยังบราวเซอร์ ทางผู้ใช้งานจะเห็นเป็นเพียงเว็บเพจแบบสถิตเท่านั้น เพราะสิ่งที่ส่งไปนั้นก็เพียงเอชทีเอ็มแอลโค้ด แต่อย่างไรก็ตามหน้าเอกสารจริงๆ ไม่ได้มาจากสำเนาเอกสารที่อยู่บนคิสของเซิร์ฟเวอร์ แต่เป็นผลมาจากการประมวลผลซึ่งทำให้หน้าเอกสารมีผลลัพธ์ที่แตกต่างกันในแต่ละครั้งที่แอปพลิเคชันถูกเรียกใช้ (ตามเงื่อนไขที่ถูกกำหนดไว้ในแอปพลิเคชัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ลักษณะการทำงานของซีจีไอและไอเอสเอพีไอ

3.2.2 ซีจีไอสคริปต์และไอดีซี (Internet Database Connector)

เทคโนโลยีซีซี (Internet Database Connector) จะสร้างเว็บเพจแบบพลวัตในลักษณะคล้ายกับแบบที่แล้ว อย่างไรก็ตามก็มีขั้นตอนการทำงาน (ดังแสดงในรูปที่ 3.3) บางอย่างที่แตกต่างกัน ดังนี้

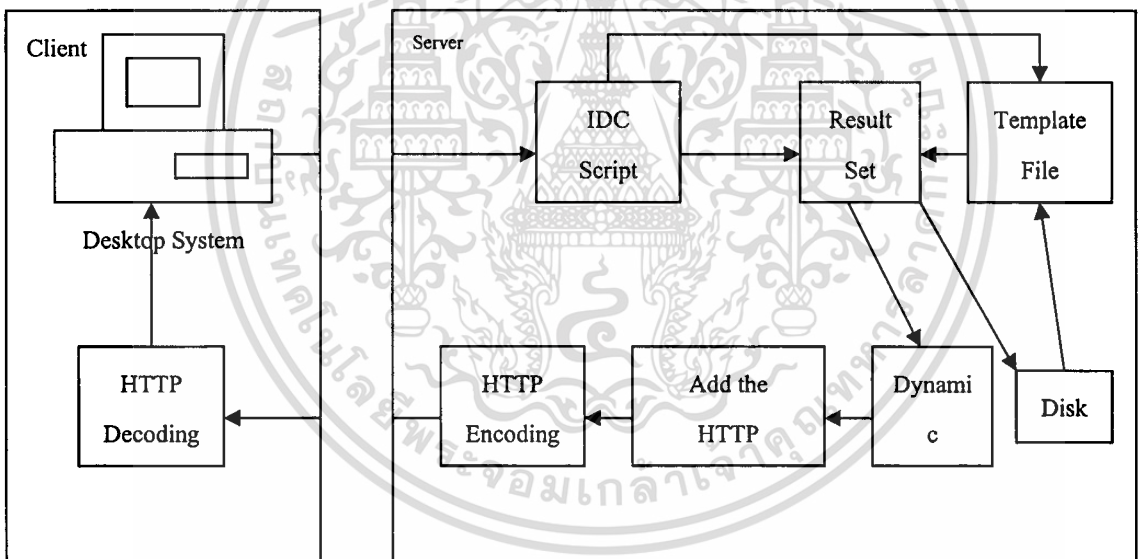
ในส่วนของสคริปต์ไฟล์, คำสั่ง และประโยคจะถูกเก็บในรูปของข้อความ (Text) บนดิสก์ของเซิร์ฟเวอร์ เพื่อรอการถูกโหลดไปประมวลผลซึ่งถ้าเป็นในกรณีของสคริปต์เช่น Perl หรือ Awk เหล่านี้จะถูกส่งผ่านไปให้ตัวแปลภาษา (Interpreter) ที่เหมาะสมที่อยู่บนเซิร์ฟเวอร์ ตัวแปลภาษานี้จะทำงานตามคำสั่งที่มีอยู่ในสคริปต์ ซึ่งจะทำได้ผลลัพธ์เป็นสารอักขระและเอชทีเอ็มแอลโค้ด (รูปแบบเดียวกันกับซีจีไอและไอเอสเอพีไอ) แต่จะมีขั้นตอนเพิ่มอีก 1 ขั้นตอนเพราะเทคโนโลยีตัวนี้จะใช้แผ่นแบบ (Template) ซึ่งทำหน้าที่เก็บรูปแบบของผลลัพธ์ที่จะส่งไปให้ผู้ใช้ (ผู้ร้องขอเอกสาร) อย่างไรก็ตามตัวแผ่นแบบนี้จะไม่เหมือนเว็บเพจแบบสถิต เพราะตัวแผ่นแบบจะไม่ถูกส่งไปให้ผู้ใช้โดยตรง แต่ตัวแผ่นแบบนี้จะเป็นเพียงตัวนำ (guide) ว่าหน้าเอกสารที่จะนำไปแสดงให้ผู้ใช้ดูนั้นมีลักษณะอย่างไร ในการทำงานเมื่อมีการร้องขอหน้าเอกสารที่ใช้เทคโนโลยีแบบนี้เช่น หน้าเอกสารที่มีนามสกุลเป็นไอดีซี (.idc) ตัวแปลภาษาจะโหลดและประมวลผลสคริปต์ไฟล์ที่ถูกอ้างอิงถึง (หน้าเอกสารที่ถูกร้องขอ) ในสคริปต์ไฟล์นั้นจะมีคำสั่งต่างๆที่ตัวแปลภาษาจะนำไปใช้สร้างค่าผลลัพธ์ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกต่างกันซึ่งบ่อยครั้งค่าที่ได้จะมาจากฐานข้อมูล โดยในสคริปต์จะประกอบด้วยชื่อของแผ่นแบบไฟล์ที่จะใช้ ซึ่งตัวแปรภาษาจะอ่านแผ่นแบบไฟล์จากคิส จากนั้นจะแทนที่เครื่องหมายพิเศษที่อยู่ในแผ่นแบบด้วยค่าที่ตัวแปรภาษาได้สร้างขึ้นตามคำสั่งที่มีอยู่ในสคริปต์ ต่อจากนั้นหน้าเอกสารที่เป็นผลลัพธ์ของแผ่นแบบจะถูกนำมาบรรจุ (Packaged up) และส่งไปให้บราวเซอร์

การใช้แผ่นแบบแยกออกมาอย่างนี้ให้ข้อได้เปรียบมากมายโดยเฉพาะอย่างยิ่งเมื่อต้องการแก้ไขรูปแบบการแสดงผลของหน้าเอกสารแทนที่เราจะเขียนส่วนของแอปพลิเคชันหรือซีจีไอสคริปต์ที่สร้างส่วนของการแสดงผลใหม่อีกครั้ง เราเพียงแค่แก้ไขรูปแบบได้โดยตรงที่ไฟล์แผ่นแบบนั้น

ในหลักการของแผ่นแบบแล้ว เมื่อบราวเซอร์อ้างถึงสคริปต์ไฟล์ ตัวสคริปต์ไฟล์จะอ้างถึงตัวแผ่นแบบไฟล์อีกครั้งหนึ่ง แต่ขอให้จำไว้ว่าทั้งตัวสคริปต์ไฟล์ และ / หรือ แผ่นแบบไฟล์ไม่ได้เป็นตัวที่ถูกส่งไปให้บราวเซอร์ แต่ทั้ง 2 ไฟล์จะทำงานร่วมกันเพื่อสร้างผลลัพธ์ที่เป็นลักษณะของเว็บเพจแบบพลวัตที่จะถูกส่งไปให้ผู้ใช้งาน



รูปที่ 3.3 ลักษณะการทำงานของซีจีไอสคริปต์และไอดีซี

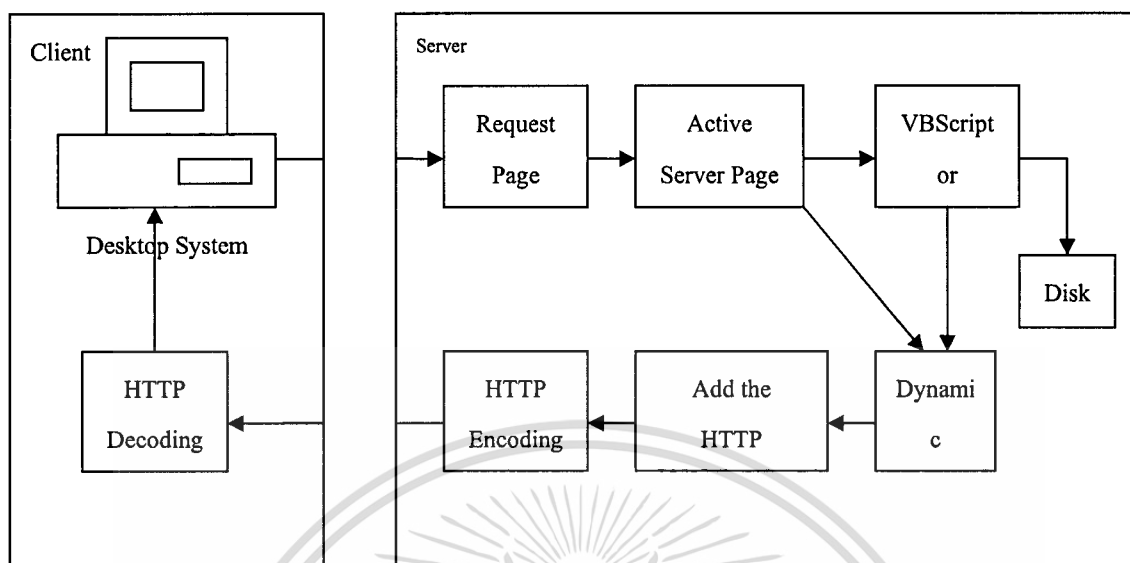
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 แอ็กทีฟเซิร์ฟเวอร์เพจส์ (Active Server Pages หรือ ASP)

ไอดีซีและเทคโนโลยีอื่นๆที่ใช้หลักการของแผ่นแบบเป็นรูปแบบในการทำหน้าเอกสารแสดงผลที่ให้ประโยชน์อย่างมากเช่น ถ้าต้องการปรับปรุงหน้าเอกสารแสดงผลก็สามารถกระทำได้โดยเพียงปรับปรุงที่ตัวแผ่นแบบ ซึ่งไม่เหมือนกับไอเอสเอพีไอที่ต้องปรับปรุงเปลี่ยนแปลงที่ตัวโค้ดโดยตรง และยังต้องคอมไพล์ (Compile) ใหม่อีกครั้งจึงจะใช้ได้ อย่างไรก็ตามตามหลักการของแผ่นแบบก็ยังมีจุดด้อยอีกประการคือ ตัวสคริปต์จะสร้างกลุ่มผลลัพธ์ (Result Set) แต่มีเพียงแผ่นแบบเท่านั้นที่จะถูกนำไปแปลงเป็นหน้าเอกสารได้ โดยส่วนมากตัวแผ่นแบบเองจะมีลักษณะเป็นโครงสร้างที่ตายตัว (A Fixed Structure) ทำให้มีข้อจำกัดในการจัดเก็บข้อมูลของกลุ่มผลลัพธ์ โดยทั่วไปแผ่นแบบจะมีความสามารถเพียงจัดเรียงรายการของค่าต่างๆที่ได้มาและประมวลผลเช่น เลือกทางเลือกแบบง่ายๆว่าจะอะไรควรอยู่ในหน้าเอกสารบ้าง โดยให้ขึ้นกับเงื่อนไขที่มี

แอ็กทีฟเซิร์ฟเวอร์เพจส์ (Active Server Pages หรือ ASP) ไม่ใช่หลักการของการทำงานร่วมกันของสคริปต์หลายๆตัวกับแผ่นแบบ แต่จะมีแบบการทำงาน (ดังแสดงในรูปที่ 3.4) ดังนี้คือ เมื่อตัวบราวเซอร์อ้างถึงหน้าเอกสารที่ต้องการ หน้าเอกสารนั้นก็จะถูกนำเข้าหน่วยความจำ (Memory) เหมือนกับเว็บเพจแบบสถิต อย่างไรก็ตามก่อนที่หน้าเอกสารจะถูกส่งไปให้บราวเซอร์ ตัวเซิร์ฟเวอร์จะประมวลผลหน้าเอกสารสำหรับสคริปต์ที่ตัวเซิร์ฟเวอร์สามารถจัดการได้ ซึ่งตัวสคริปต์นี้สามารถคำนวณและแทรก (Insert) หรือสร้างตัวอักขระพิเศษรวมทั้งเอชทีเอ็มแอลโค้ดที่ต้องการได้ในไฟล์เดียวกัน และเพราะว่ามันเป็นการทำงานโดยตรงกับหน้าเอกสาร ในขณะที่หน้าเอกสารนั้นถูกโหลดแทนที่จะเป็นการสร้างกลุ่มของผลลัพธ์ก่อน ดังนั้นมันจึงสามารถควบคุมการแสดงผลให้กับหน้าเอกสารที่จะนำไปแสดงได้อย่างดี เมื่อเซิร์ฟเวอร์เสร็จสิ้นกระบวนการในการโหลดหน้าเอกสาร(พร้อมทั้งขั้นตอนการประมวลผล)แล้ว หน้าเอกสารจะถูกบรรจุไว้ภายใต้เอชทีทีพีเว็บเปเปอร์ (HTTP wrapper) และส่งไปยังบราวเซอร์ เมื่อหน้าเอกสารมาถึงที่บราวเซอร์ มันก็จะเป็นเพียงตัวอักขระและเอชทีเอ็มแอลโค้ดธรรมดาเหมือนกับเว็บเพจแบบสถิต ดังนั้นก็กล่าวได้ว่าบราวเซอร์ได้รับหน้าเอกสารที่มันอ้างถึงซึ่งเป็นหน้าเอกสารที่ได้ปรับปรุง (Modify) แล้ว โดยค่าที่ได้จากการประมวลผลสคริปต์ที่อยู่บนหน้าเอกสารนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ลักษณะการทำงานของแอคทีฟเซิร์ฟเวอร์เพจส์

3.3 แอคทีฟเซิร์ฟเวอร์เพจส์ คืออะไร?

แอคทีฟเซิร์ฟเวอร์เพจส์เป็นเอกสารเอชทีเอ็มแอลที่ได้เพิ่มความสามารถบางอย่างเข้าไป ดังนั้นแอคทีฟเซิร์ฟเวอร์เพจส์จึงสามารถทำงานหลายๆสิ่งๆที่เอชทีเอ็มแอลสามารถทำได้เช่นกันเช่น สามารถบรรจุเอชทีเอ็มแอลแท็ก (HYML Tags) ซึ่งถูกแปล (Interpret) และแสดงผลโดยบราวเซอร์ และอีกหลายๆสิ่งๆที่เราใช้บรรจุให้กับหน้าเอกสารเอชทีเอ็มแอลธรรมดาเช่น Java applets , ตัวอักษรกระพิบ , สคริปต์ที่ใช้ประมวลผลทางบราวเซอร์ , ActiveX controls เหล่านี้สามารถนำมาไว้ที่แอคทีฟเซิร์ฟเวอร์เพจส์ได้ แต่อย่างไรก็ตามแอคทีฟเซิร์ฟเวอร์เพจส์ก็ยังมีคุณลักษณะ (Features) ที่สำคัญ 4 อย่างที่ทำให้โดดเด่นนั้นคือ

3.3.1 แอคทีฟเซิร์ฟเวอร์เพจส์สามารถบรรจุสคริปต์ที่ใช้ประมวลผลทางเซิร์ฟเวอร์ได้

การทำเช่นนี้ได้ทำให้เกิดประโยชน์มากเพราะจะทำให้เราสามารถสร้างหน้าเอกสารที่เป็นพลวัต (Dynamic) ได้ ตัวอย่างง่ายๆที่แสดงถึงประโยชน์ของคุณลักษณะนี้คือ เราสามารถที่จะทำเอกสารที่แสดงข้อความทักทายที่แตกต่างกันไปในแต่ละเวลาของวันได้

3.3.2 แอคทีฟเซิร์ฟเวอร์เพจส์ได้เตรียม Build-in object ไว้อย่างมากมาย

การที่มี Build-in object ในแอคทีฟเซิร์ฟเวอร์เพจส์นั้น ช่วยให้สคริปต์ของเรามีประสิทธิภาพในการทำงานมากขึ้น เพราะ Object ต่างๆเหล่านี้จะทำให้เราสามารถรับส่งข้อมูลระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์กับ ไคลเอนต์ (บราวเซอร์) ได้ ตัวอย่างเช่น การใช้ Object “ Request ” เราสามารถรับส่งข้อมูลจากผู้ใช้ที่ส่งมาทางฟอร์ม (Form) ของเอชทีเอ็มแอลและส่งข้อมูลนั้นต่อไปให้กับส่วนของสคริปต์ที่ต้องการได้อย่างง่ายดาย

3.3.3 แอ็กทีฟเซิร์ฟเวอร์เพจส์สามารถเพิ่มคอมโพเนนต์ (Component) ที่ต้องการได้

ไม่เฉพาะคอมโพเนนต์มาตรฐานที่แอ็กทีฟเซิร์ฟเวอร์เพจส์ได้เตรียมไว้ตอนที่ติดตั้งเท่านั้น แอ็กทีฟเซิร์ฟเวอร์เพจส์ยังสามารถเพิ่มคอมโพเนนต์ที่ผู้ใช้ต้องการเข้าไปได้อีก

3.3.4 แอ็กทีฟเซิร์ฟเวอร์เพจส์สามารถติดต่อกับฐานข้อมูลได้อย่างง่ายดาย

โดยการใช้ชุดของ Object พิเศษที่เรียกว่า ActiveX Data Object (ADO) (Object เหล่านี้เป็นมาตรฐานอยู่แล้วในแอ็กทีฟเซิร์ฟเวอร์เพจส์) คุณลักษณะในข้อนี้ทำให้แอ็กทีฟเซิร์ฟเวอร์เพจส์มีประสิทธิภาพมากที่สุดในการที่จะนำไปใช้งาน

ดังนั้นด้วยคุณลักษณะทั้ง 4 ข้อที่กล่าวมา ทำให้กล่าวได้ว่าแอ็กทีฟเซิร์ฟเวอร์เพจส์นั้นคือหน้าเอกสารเอชทีเอ็มแอลธรรมดาที่ได้เพิ่มการทำงานของสคริปต์ที่ประมวลผลทางเซิร์ฟเวอร์ โดยมี Object และคอมโพเนนต์เพิ่มเข้ามาช่วยในการทำงาน ทำให้สามารถสร้างเว็บเพจแบบพลวัตได้

3.4 แอ็กทีฟเซิร์ฟเวอร์เพจส์ ทำงานอย่างไร?

แนวทางที่ดีที่สุดที่จะเข้าใจได้ว่าแอ็กทีฟเซิร์ฟเวอร์เพจส์ทำงานได้อย่างไรนั้นคือ การพิจารณาเว็บเซิร์ฟเวอร์ที่สนับสนุนและไม่สนับสนุนแอ็กทีฟเซิร์ฟเวอร์เพจส์เปรียบเทียบกัน ดังนั้นในที่นี้จึงจะใช้ Internet Information Server (IIS) เป็นเว็บเซิร์ฟเวอร์ (ผู้อ่านสามารถเลือกใช้เว็บเซิร์ฟเวอร์ตัวอื่นๆในการใช้งานแอ็กทีฟเซิร์ฟเวอร์เพจส์ก็ได้)

การอธิบายจะเริ่มจากการทำงานโดยทั่วไปของ Internet Information Server ที่ทำงานกับเว็บเพจแบบสถิต ไปจนถึงการทำงานของ Internet Information Server กับเว็บเพจแบบพลวัตดังนี้

โดยทั่วไปของ Inter Information Server ก่อนที่จะมีแอ็กทีฟเซิร์ฟเวอร์เพจส์จะเป็นการให้บริการกับหน้าเอกสารแบบสถิตเท่านั้น ดังนั้นเมื่อผู้ใช้ร้องขอหน้าเอกสารเหล่านี้ Internet Information Server จะดึงข้อมูลหน้าเอกสารที่ต้องการจากดิสก์หรือหน่วยความจำ (ซึ่งกล่าวได้ว่าเป็นลักษณะของเว็บเพจแบบสถิต) แล้วส่งไปให้บราวเซอร์ของผู้ใช้ (ผู้ร้องขอหน้าเอกสารนั้น) ดังนั้น Internet Information Server จึงมีหน้าที่หลักเพียงเป็นตัวกลางระหว่างบราวเซอร์และไฟล์ที่อยู่ในดิสบนเซิร์ฟเวอร์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปการทำงานของ Internet Information Server กับเว็บเพจแบบสถิตได้ดังนี้

1. ผู้ใช้ใส่ที่อยู่ทางอินเทอร์เน็ตของไฟล์เอชทีเอ็มแอลผ่านทางเว็บเบราว์เซอร์ เพื่อร้องขอหน้าเอกสารที่ต้องการ ตัวอย่างเช่น [HTTP://www.kmitl.ac.th/hello.html](http://www.kmitl.ac.th/hello.html)
2. เบราวเซอร์จะส่งคำร้องขอสำหรับหน้าเอกสารนั้นแก่เว็บเซิร์ฟเวอร์
3. เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอหน้าเอกสารและจดจำได้ว่าหน้าเอกสารที่ร้องขอเป็นไฟล์เอชทีเอ็มแอลเพราะว่าไฟล์ที่ร้องขอมามีส่วนขยาย (นามสกุลไฟล์) เป็น .htm หรือ .html
4. เว็บเซิร์ฟเวอร์ดึงหน้าเอกสารที่ต้องการจากดิสก์หรือหน่วยความจำแล้วส่งกลับคืนไปยังเบราว์เซอร์ที่ร้องขอมา
5. ตัวไฟล์เอชทีเอ็มแอลที่ส่งมาจะถูกแปลความ โดยเบราว์เซอร์ของผู้ใช้ที่ร้องขอหน้าเอกสารนั้นแล้วแสดงผลบนหน้าจอให้เห็น

ต่างจากเอกทึฟเซิร์ฟเวอร์เพจส์คือ ในขณะที่ Internet Information Server สามารถทำงานบริการเว็บเพจแบบสถิตนั้น เอกทึฟเซิร์ฟเวอร์เพจส์จะยอมให้ Internet Information Server ให้บริการหน้าเอกสารที่เป็นแบบพลวัตได้ด้วย โดยตัวเว็บเซิร์ฟเวอร์เองจะกลายเป็นตัวสร้างหน้าเอกสารไปด้วยเพื่อความเข้าใจที่ชัดเจนขึ้น เราสามารถแยกขั้นตอนได้ดังนี้

สรุปการทำงานของ Internet Information Server กับเว็บเพจแบบพลวัตได้ดังนี้

1. ผู้ใช้ใส่ที่อยู่ทางอินเทอร์เน็ตของไฟล์เอกทึฟเซิร์ฟเวอร์เพจส์ผ่านทางเว็บเบราว์เซอร์ เพื่อร้องขอหน้าเอกสารเอกทึฟเซิร์ฟเวอร์เพจส์ ตัวอย่างเช่น
[HTTP://www.kmitl.ac.th/hello.asp](http://www.kmitl.ac.th/hello.asp)
2. เบราวเซอร์จะส่งคำร้องขอหน้าเอกสารเอกทึฟเซิร์ฟเวอร์เพจส์ไปยัง Internet Information Server
3. เว็บเซิร์ฟเวอร์จะรับคำร้องขอและจดจำได้ว่าเป็นการร้องขอไฟล์เอกทึฟเซิร์ฟเวอร์เพจส์ เพราะว่าเป็นการร้องขอไฟล์ที่มีส่วนขยาย (นามสกุลไฟล์) เป็น .asp
4. เว็บเซิร์ฟเวอร์จะรับไฟล์เอกทึฟเซิร์ฟเวอร์เพจส์ที่ถูกร้องขอจากดิสก์หรือหน่วยความจำ
5. เว็บเซิร์ฟเวอร์จะรับส่งไฟล์นั้น ไปยังโปรแกรมพิเศษที่ชื่อว่า ASP.dll

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ไฟล์แอกทีฟเซิร์ฟเวอร์เพจจะถูกประมวลผลจากบนลงล่าง (Top to Bottom) และทุกๆ คำสั่งที่มีในไฟล์จะประมวลผล แล้วเก็บผลลัพธ์ของการประมวลผลไว้ในรูปไฟล์เอชทีเอ็มแอลธรรมดา
7. ไฟล์เอชทีเอ็มแอลที่ได้จะถูกส่งกลับ ไปยังบราวเซอร์ที่ร้องขอ
8. ไฟล์เอชทีเอ็มแอลจะถูกแปลความโดยบราวเซอร์ของผู้ร้องขอหน้าเอกสารนั้น และแสดงผลกลับไปหน้าจอ

ในมุมมองของเว็บเซิร์ฟเวอร์แล้ว แอกทีฟเซิร์ฟเวอร์เพจจะมีความแตกต่างกับหน้าเอกสารธรรมดาอย่างมากเพราะหน้าเอกสารธรรมดาคจะถูกส่ง ไปให้บราวเซอร์ โดยไม่มีการประมวลผลใดๆเลย แต่สำหรับแอกทีฟเซิร์ฟเวอร์เพจแล้ว คำสั่งที่มีอยู่ในแอกทีฟเซิร์ฟเวอร์เพจจะต้องถูกประมวลผล (Execute) เพื่อที่จะสร้างผลลัพธ์ที่เป็นหน้าเอกสารเอชทีเอ็มแอล ซึ่งเป็นสิ่งที่ทำให้แอกทีฟเซิร์ฟเวอร์เพจบรรลุไปด้วยส่วนที่เป็นแบบพลวัต (Dynamic)

ในมุมมองของบราวเซอร์กล่าวได้ว่า แอกทีฟเซิร์ฟเวอร์เพจนั้นจะเหมือนกับหน้าเอกสารธรรมดาอย่างมาก สิ่งเดียวที่แตกต่างก็คือ แอกทีฟเซิร์ฟเวอร์เพจมีส่วนขยายเป็น .asp แทนที่จะเป็น .htm หรือ .html ที่เป็นแบบนี้เพราะว่าเมื่อมีการร้องขอหน้าเอกสารที่เป็นแอกทีฟเซิร์ฟเวอร์เพจ สิ่งที่จะส่งกลับคืนมานั้นก็คือหน้าเอกสารที่เป็นเอชทีเอ็มแอลธรรมดา ประโยชน์ที่ได้รับจากลักษณะเช่นนี้คือ แอกทีฟเซิร์ฟเวอร์เพจสามารถนำไปแสดงได้กับทุกบราวเซอร์ที่มีอยู่

3.5 ภาษาสคริปต์กับแอกทีฟเซิร์ฟเวอร์เพจ

แอกทีฟเซิร์ฟเวอร์เพจจะใช้ภาษาสคริปต์ในการใช้งานอย่างมาก ซึ่งภาษาที่ใช้อาจเป็นภาษาที่รู้จักกันดีเช่น VBScript , JavaScript (Jscript) หรือจะเป็นภาษาอื่น ๆ ที่มีคุณลักษณะที่เข้ากันได้กับมาตรฐานสคริปต์ นอกจากนั้น ActiveX ก็สามารถใช้ในแอกทีฟเซิร์ฟเวอร์เพจได้เช่นเดียวกัน

ในการเติมสคริปต์เข้าไปในไฟล์แอกทีฟเซิร์ฟเวอร์เพจ ดูเหมือนจะเป็นการยากที่จะเข้าใจได้ว่าสคริปต์จะถูกประมวลผลที่ไหน ที่ไคลเอนต์ (บราวเซอร์) หรือที่เซิร์ฟเวอร์? ตัวของเซิร์ฟเวอร์เองจะสามารถรู้ได้อย่างไรว่า สคริปต์ส่วนใดจะต้องถูกประมวลผลในขณะที่สร้างหน้าเอกสารแบบพลวัต หรือ สคริปต์ส่วนใดที่จะถูกส่งไปให้บราวเซอร์เพื่อแปลความและประมวลผลสคริปต์ส่วนนั้นอย่างที่มีมันควรจะเป็น? ในความเป็นจริงเป็นเรื่องไม่ยากที่ตัวเซิร์ฟเวอร์จะสามารถรู้ได้ว่าสคริปต์ส่วนใดควรจะถูกประมวลผลที่ใด เพราะโดยทั่วไปแอกทีฟเซิร์ฟเวอร์เพจจะใช้แท็ก (Tag) “ <% ... %> ” ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอกถึงส่วนของสคริปต์ที่จะถูกประมวลผลที่เซิร์ฟเวอร์ ในขณะที่ตัวบราวเซอร์เองจะใช้แท็ก (Tag) ของเอชทีเอ็มแอล “ < ... > ” ในการบอกถึงส่วนของสคริปต์ที่จะถูกประมวลผลที่บราวเซอร์ ปัญหาที่เกิดขึ้นเมื่อมีการใช้คุณลักษณะของเซิร์ฟเวอร์ไซด์สคริปต์ (Server-Side Script) ของแอททิฟเซิร์ฟเวอร์เพจส์คือ คำสั่งที่อยู่ในแท็ก “ <% ... %> ” จะถูกประมวลผลทันทีที่หน้าเอกสารนั้นถูกแปลความบนเซิร์ฟเวอร์ซึ่งเป็นหลักการเดียวกับคำสั่งที่อยู่ในแท็ก “ <Script> ... </Script> ” ที่บราวเซอร์จะประมวลผลทันทีที่หน้าเอกสารนั้นไปถึง ตัวอย่างเช่น ถ้าเราต้องการเขียนคำสั่งที่จะให้ไปประมวลผลที่บราวเซอร์ เราจะเขียนคำสั่งนั้นลงในรูปซับรูทีน (Subroutine) หรือฟังก์ชัน (Function) ดังนี้

```
<Script LANGUAGE = VBScript>                                `for execute in Browser
    Sub cmdShow_onClick()                                     `executed when a button is clicked
        Call ShowTheMeaning                                `calls the Show TheMeaning routine
    End Sub

    Sub ShowTheMeaning()                                     `execute only when called
        MsgBox gstrMyValue, 64, gintMyvalue                `displays the values
    End Sub

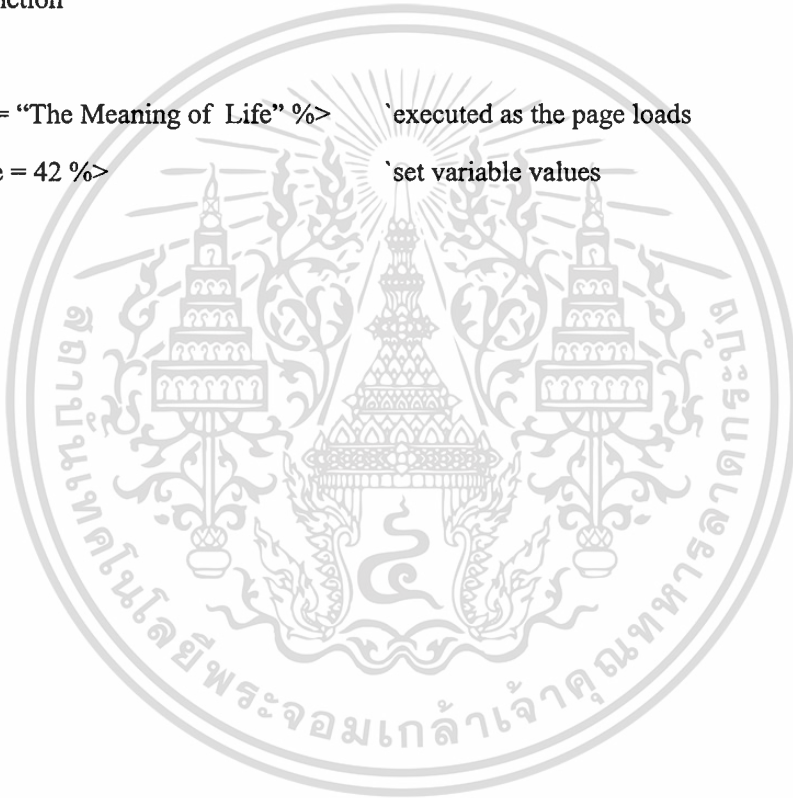
    gstrMyValue = “The Meaning of Life”                    `executed as the page loads
    gintMyValue = 42                                       `sets the variables values
</Script>
```

ปัญหาที่เกิดขึ้นก็คือถ้าเราต้องการใช้ซับรูทีนหรือฟังก์ชันในไฟล์แอททิฟเซิร์ฟเวอร์เพจส์ ตัวแอททิฟเซิร์ฟเวอร์เพจส์จะไม่ยอมให้มีการใช้ซับรูทีนหรือฟังก์ชันภายในแท็ก “ <% ... %> ” ด้วยเหตุผลที่ว่าแอททิฟเซิร์ฟเวอร์เพจส์จะยอมให้เฉพาะคำสั่งที่จะถูกประมวลผลทันทีที่หน้าเอกสารนั้นถูกแปลความ (Interpreted) เท่านั้น เนื่องจากในการใช้ซับรูทีนหรือฟังก์ชัน เราจำเป็นต้องเขียนไว้ในแท็ก “ <Script> ... </Script> ” ธรรมดาเหมือนกับคำสั่งที่เขียนสำหรับประมวลผลที่บราวเซอร์ ดังนั้นในการแก้ปัญหาเรื่องความสับสนที่จะเกิดขึ้นในการเขียนสคริปต์ (เพื่อจะประมวลผลที่เซิร์ฟเวอร์หรือที่ไคลเอนต์) และป้องกันความผิดพลาดที่จะส่งสคริปต์ที่ต้องการให้ประมวลผลที่เซิร์ฟเวอร์ไปที่บราวเซอร์ แอททิฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์เพจสได้เสนอคุณลักษณะ “ RUNAT ” ขึ้นมา ซึ่งถ้าต้องการให้เซิร์ฟเวอร์หรือฟังก์ชันประมวลผลที่ด้านเซิร์ฟเวอร์ เราสามารถทำได้ดังนี้

```
<Script LANGUAGE = VBScript RUNAT = Server>    `for execute in the Server
    Function GetTheMeaning()                    `return a new string executed only when called
                                                `From elsewhere in code
        GetTheMeaning = gstrMyText & “ is ” & gintMyValue
    End Function
</Script>
<% gstrMyText = “The Meaning of Life” %>      `executed as the page loads
<% gintMyValue = 42 %>                        `set variable values
```



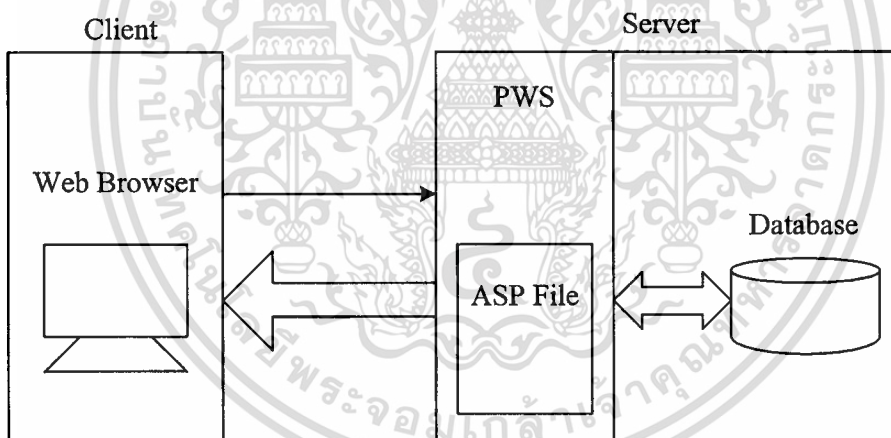
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบโปรแกรม

4.1 แนวความคิดในการออกแบบ

แนวความคิดในการออกแบบโปรแกรมนี้ (ดังแสดงในรูปที่ 4.1) จะใช้เว็บเซิร์ฟเวอร์เป็น Personal Web Server (PWS) เป็นตัวติดตั้งทั้งไคลเอนต์และฐานข้อมูล โดนที่แอกทีฟเซิร์ฟเวอร์เพจส์เป็นสคริปต์ในการติดต่อและใช้สถาปัตยกรรมแบบทูเทียร์ ซึ่งหมายความว่า จะมีอยู่ 2 ฝ่าย คือ ฝ่ายไคลเอนต์ซึ่งจะทำหน้าที่ร้องขอข้อมูลจากเซิร์ฟเวอร์และรอรับข้อมูลกลับมาเท่านั้น ในส่วนของฝั่งเซิร์ฟเวอร์เมื่อได้รับการร้องขอมาจากไคลเอนต์แล้วก็จะติดต่อและจัดการฐานข้อมูลผ่านทาง โอดีบีซี (ODBC) เมื่อเสร็จสิ้นกระบวนการที่ไคลเอนต์ร้องขอแล้ว ก็จะส่งผลลัพธ์หรือข้อมูลที่ไคลเอนต์ต้องการกลับมาให้ไคลเอนต์

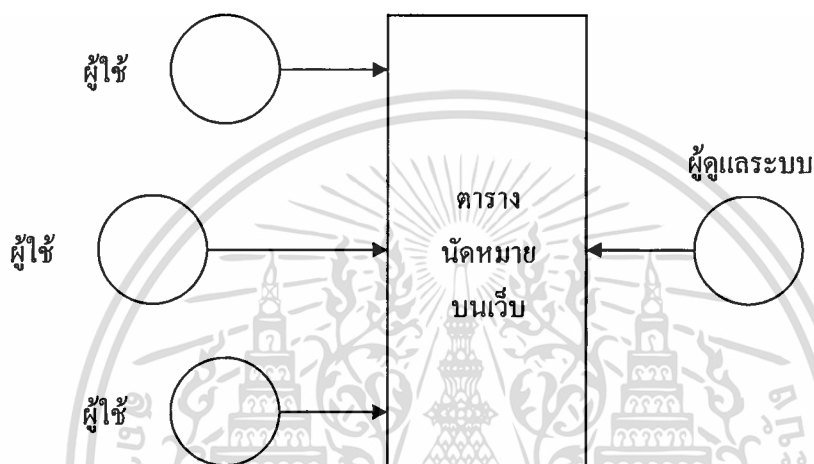


รูปที่ 4.1 การออกแบบโปรแกรม

ในโครงการนี้ใช้แอกทีฟเซิร์ฟเวอร์เพจส์ซึ่งเขียนด้วย VBScript การทำงานของมันจะฝั่งตัวอยู่ที่ฝั่งเซิร์ฟเวอร์ ซึ่งการใช้วิธีนี้มีข้อดีคือ โปรแกรมจะไม่ขึ้นกับไคลเอนต์ ทางฝั่งไคลเอนต์จะเป็นอย่างไรก็ได้ เพียงแค่ส่งข้อมูลไปและรอการตอบกลับจากเซิร์ฟเวอร์เท่านั้น แตกต่างกับการใช้วิธีอื่นเช่น JavaApplet เป็นต้น การใช้ JavaApplet จะรันที่ไคลเอนต์ทำให้สามารถแบ่งภาระจากทางฝั่งเซิร์ฟเวอร์มาได้ แต่การใช้ JavaApplet นั้นจะทำให้ระบบต้องขึ้นกับไคลเอนต์มากขึ้นซึ่งอาจจะมี

ปัญหาได้เช่น เว็บเบราว์เซอร์ไม่สามารถรัน JavaApplet ได้เป็นต้น อย่างไรก็ตาม การออกแบบในโครงการนี้จะทำให้เซิร์ฟเวอร์ต้องรับภาระทั้งหมด ซึ่งทำให้ประสิทธิภาพโดยรวมต่ำลง ดังนั้นในการพัฒนาต่อไปในอนาคตควรที่จะแบ่งภาระให้กับไคลเอนต์บ้าง จะทำให้ประสิทธิภาพของระบบโดยรวมดีขึ้น

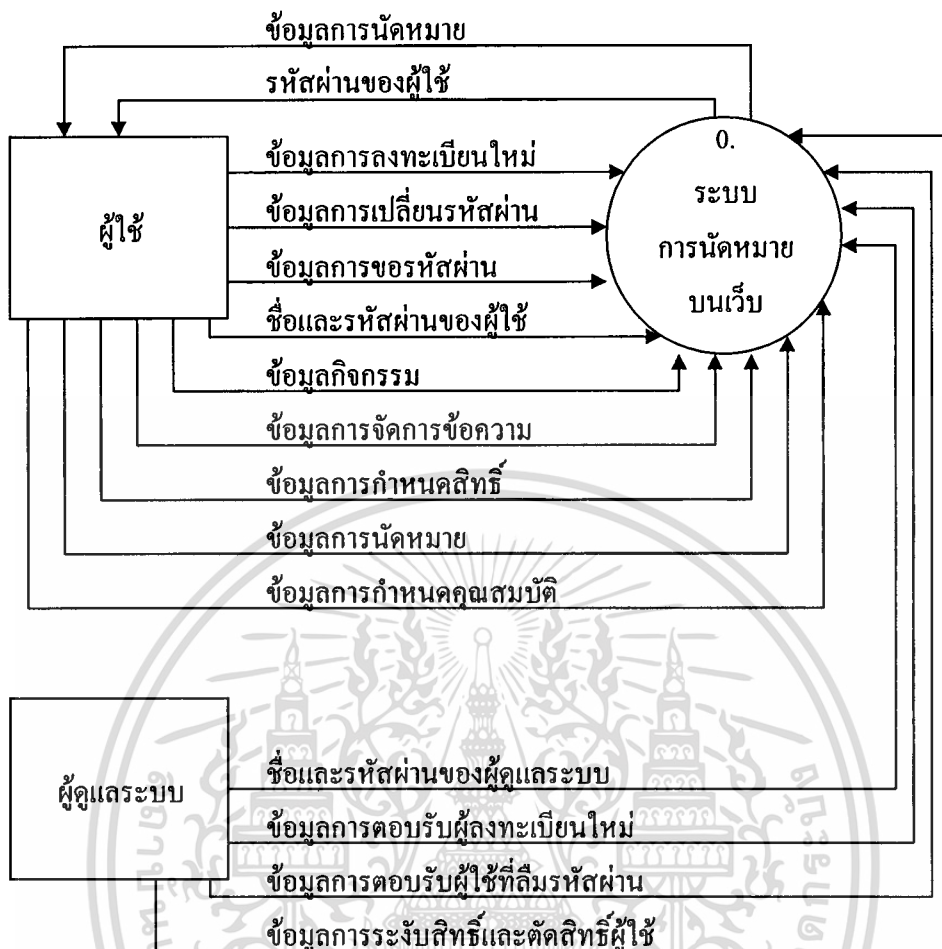
การออกแบบโปรแกรมนี้จะมีทั้งผู้ใช้ระบบและผู้ดูแลระบบซึ่งคอยควบคุมและดูแลระบบ (ดังแสดงในรูปที่ 4.2) เช่น อนุญาตผู้ที่เข้ามาใช้งานระบบ เป็นต้น



รูปที่ 4.2 ผู้ใช้และผู้ดูแลระบบในโปรแกรม

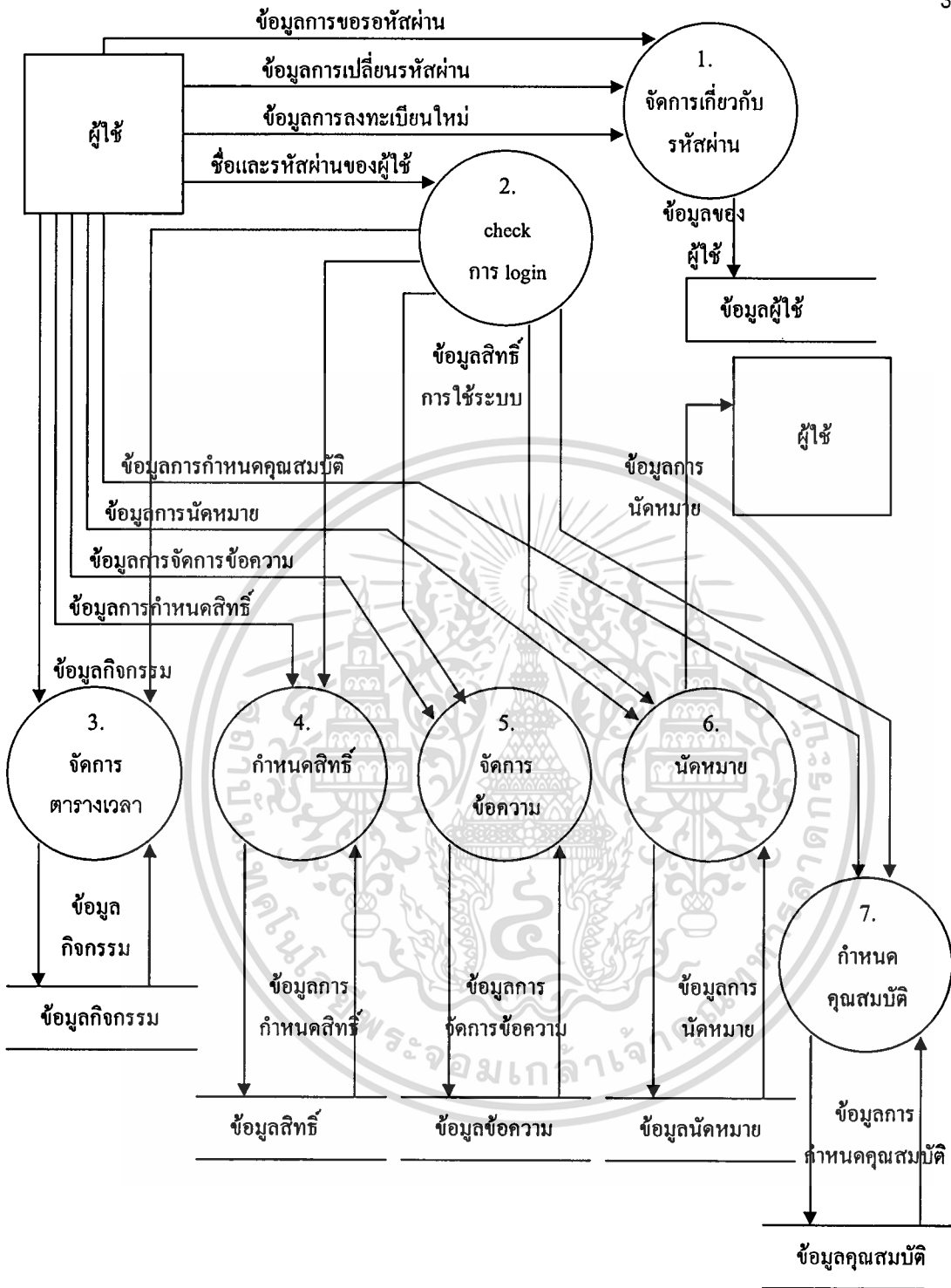
4.2 การออกแบบขั้นตอนการทำงานของระบบ

ซึ่งจะประกอบด้วย Context Diagram และ Data Flow Diagram (ดังแสดงในรูปที่ 4.3 และ 4.4) โดย Context Diagram จะแสดงถึงภาพรวมของระบบ ส่วน Data Flow Diagram นั้นจะแสดงถึงรายละเอียดต่างๆภายในระบบ



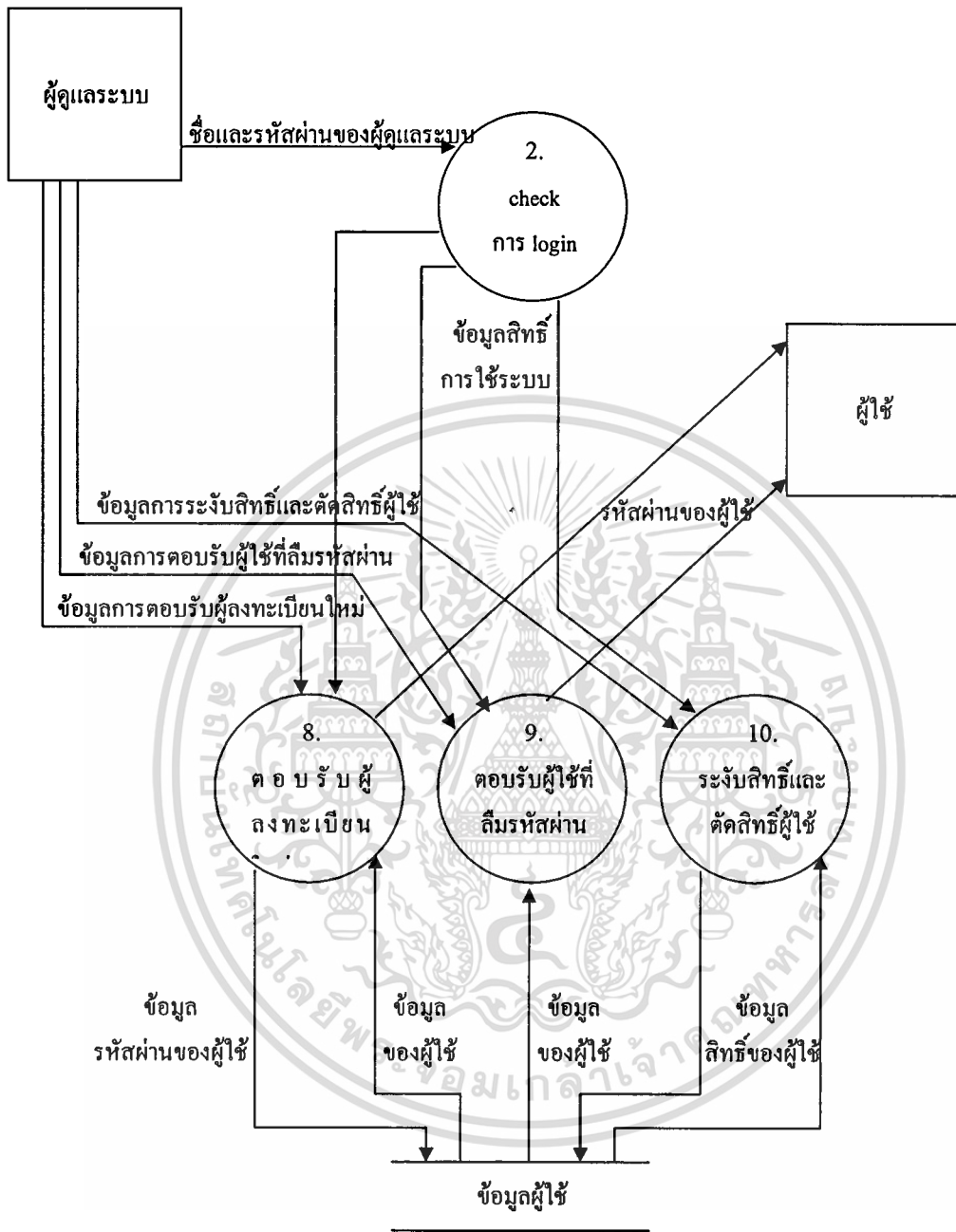
รูปที่ 4.3 Context Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4(ก) Data Flow Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

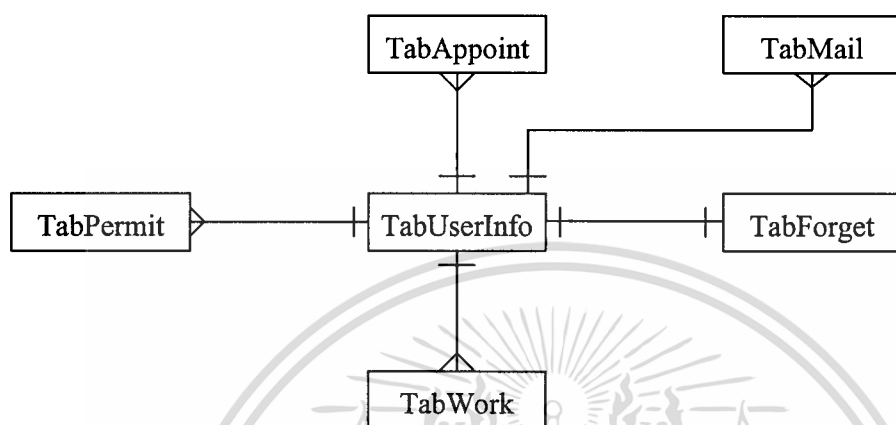


รูปที่ 4.4(ข) Data Flow Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลในโครงการนี้จะใช้ไมโครซอฟเอกเซล ซึ่งประกอบไปด้วยตารางทั้งหมด 6 ตารางโดยแต่ละตารางมีความสัมพันธ์กันดังนี้



รูปที่ 4.5 ความสัมพันธ์ของตาราง

โดยจะมี TabUserInfo เป็นตารางหลัก เนื่องจากการออกแบบฐานข้อมูลนี้ เราจะไม่เก็บชื่อหรือนามสกุลในทุกๆตาราง แต่จะเก็บเป็นหมายเลข ID แทน แต่การแสดงผลออกทางหน้าจอของไคลเอนต์ต้องแสดงชื่อและนามสกุล ดังนั้นทุกครั้งที่มีการสอบถามข้อมูล (Query) จึงต้องมีการสอบถามข้อมูลชื่อและนามสกุลจาก TabUserInfo นี้ทุกครั้ง

รายละเอียดของตารางมีดังต่อไปนี้

4.3.1 TabAppoint

เป็นตารางที่เก็บกิจกรรมต่างๆทั้งหมด มีรายละเอียดดังนี้

ตารางที่ 4.1 รายละเอียดของตาราง TabAppoint

ชื่อคอตัมน์	ชนิดข้อมูล	ขนาด	Null	ความหมาย
JobNo (PK)	Int	4	Not Null	Counter
UserID (FK)	Int	4	Null	ID ของผู้ใช้
Detail	Char	50	Null	รายละเอียดการนัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Start	Date/Time	-	Null	เวลาเริ่มนัด
Finish	Date/Time	-	Null	เวลาเสร็จนัด
AppointerID	Int	4	Null	ผู้นัด
AppointeeID	Int	4	Null	ผู้ถูกนัด
YesNo1	Int	1	Null	คำตอบการนัด
AppointStatusID	Int	4	Null	สถานะการนัด
PrivatePublic	Int	1	Null	1 = ข้อมูลส่วนตัว 2 = ผู้อื่นดูได้
DayNum	Int	4	Null	จำนวนวันที่ต้อง นัดหมายก่อน

AppointStatusID	ความหมาย
1	ใส่เอง
2	นัดแล้วรอคำตอบ
3	รอให้ผู้นัดตอบการนัด
4	การนัดหมายสำเร็จ

- อ้างอิงจากตาราง TabUserInfo

4.3.2 TabForget

เป็นตารางที่เก็บรายละเอียดของผู้ที่ใช้กรอกแบบฟอร์มการลืมรหัสผ่าน มีรายละเอียดดังนี้

ตารางที่ 4.2 รายละเอียดของตาราง TabForget

ชื่อคอลัมน์	ชนิดข้อมูล	ขนาด	Null	ความหมาย
Number (PK)	Int	4	Not Null	Counter
UserID (FK)	Int	4	Null	ID ของผู้ใช้
Name	Char	20	Null	ชื่อ
Surname	Char	30	Null	นามสกุล
Telephone	Char	20	Null	หมายเลขโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Char	100	Null	ที่อยู่
Username	Char	20	Null	ชื่อผู้ใช้
Email	Char	40	Null	Email ของผู้ใช้
RequestTime	Date/Time	-	Null	เวลาที่ร้องขอ
Status	Int	1	Null	สถานะ 0 = New Request 1 = ให้รหัสผ่านแล้ว 2 = ปฏิเสธการขอ รหัสผ่าน

- อ้างอิงจากตาราง TabUserInfo

4.3.3 TabMail

เป็นตารางที่เก็บข้อความที่ผู้ใช้ส่งถึงกันทั้งหมด มีรายละเอียดดังนี้

ตารางที่ 4.3 รายละเอียดของตาราง TabMail

ชื่อคอลัมน์	ชนิดข้อมูล	ขนาด	Null	ความหมาย
MailNo (PK)	Int	4	Not Null	Counter
UserID (FK)	Int	4	Null	ID ของผู้ใช้
Title	Char	50	Null	ชื่อ
Detail	Char	50	Null	ข้อความ
DateTimeSend	Date/Time	8	Null	เวลาส่ง
ReceiverID	Int	4	Null	ผู้รับ
MailStatusID	Int	4	Null	สถานะข้อความ

MailStatusID

ความหมาย

1

ยังไม่อ่าน

2

อ่านแล้ว

- อ้างอิงจากตาราง TabUserInfo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 TabPermit

เป็นตารางที่เก็บค่าที่ผู้ใช้ให้สิทธิ์แก่กัน มีรายละเอียดดังนี้

ตารางที่ 4.4 รายละเอียดของตาราง TabPermit

ชื่อคอลัมน์	ชนิดข้อมูล	ขนาด	Null	ความหมาย
UserID(PK,FK)	Int	4	Null	ID ของผู้ใช้
PrioritorID	Int	4	Null	ID ของผู้ถูกกำหนดสิทธิ์
PriorityID	Int	4	Null	ค่าของการกำหนดสิทธิ์

PriorityID

ความหมาย

2

ยอมให้มีกรนัดหมาย ไม่เห็นรายละเอียด นัดแล้วต้องรอคำตอบ

3

ยอมให้มีกรนัดหมาย เห็นรายละเอียด นัดแล้วต้องรอคำตอบ

4

ยอมให้มีกรนัดหมาย เห็นรายละเอียด นัดแล้วไม่ต้องรอคำตอบ

- อ้างอิงจากตาราง TabUserInfo

4.3.5 TabUserInfo

เป็นตารางที่เก็บรายละเอียดทุกอย่างของทุกคนเช่น ชื่อ นามสกุล ฯลฯ มีรายละเอียด

ดังนี้

ตารางที่ 4.5 รายละเอียดของตาราง TabUserInfo

ชื่อคอลัมน์	ชนิดข้อมูล	ขนาด	Null	ความหมาย
UserID (PK)	Int	4	Not Null	ID ของผู้ใช้
Name	Char	20	Null	ชื่อ
Surname	Char	30	Null	นามสกุล
Telephone	Char	20	Null	หมายเลขโทรศัพท์
Address	Char	100	Null	ที่อยู่
Username	Char	20	Null	ชื่อผู้ใช้
Password	Char	20	Null	รหัสผ่าน
Email	Char	40	Null	Email ของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LastLogin	Date/Time	-	Null	เวลา Login ล่าสุด
Status	Int	1	Null	สถานะ 0 = ลงทะเบียนใหม่ 1 = ผู้ใช้ปกติ 2 = ถูกระงับสิทธิ์ 3 = ถูกตัดสิทธิ์

4.3.6 TabWork

เป็นตารางที่เก็บค่าว่าผู้ใช้แต่ละคน ทำงานวันไหนและเวลาอะไรบ้าง มีรายละเอียดดังนี้

ตารางที่ 4.6 รายละเอียดของตาราง TabWork

ชื่อคอลัมน์	ชนิดข้อมูล	ขนาด	Null	ความหมาย
JobNo (PK)	Int	4	Not Null	Counter
UserID (FK)	Int	4	Null	ID ของผู้ใช้
Workday	Int	4	Null	วันทำงาน
Start	Date/Time	-	Null	เวลาเริ่มทำงาน
Finish	Date/Time	-	Null	เวลาเลิกงาน

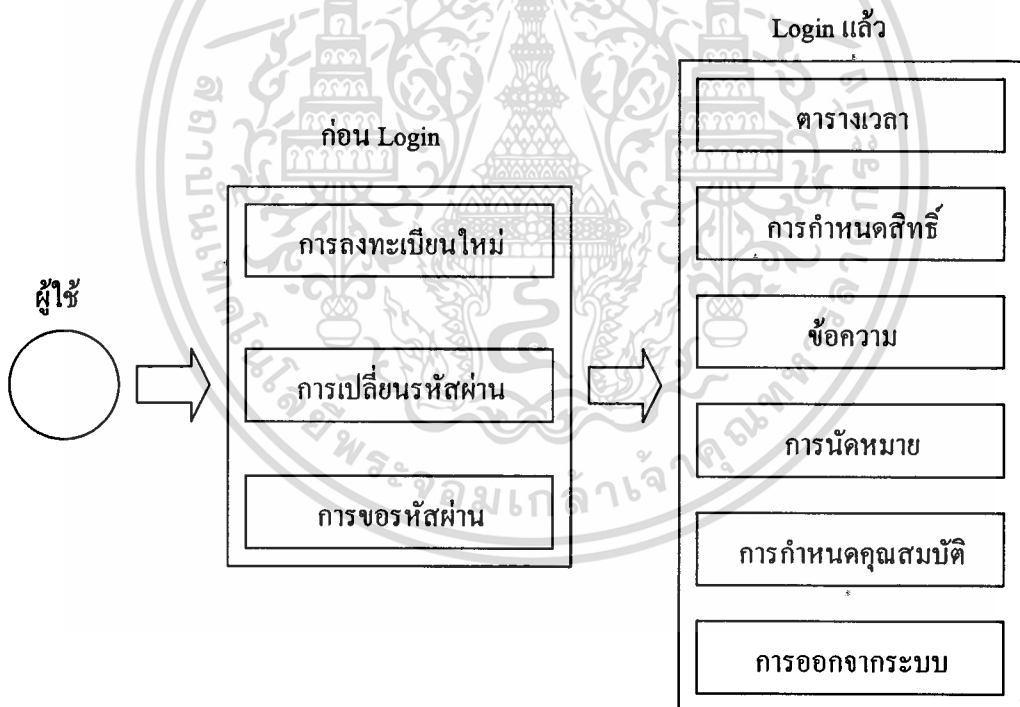
- อ้างอิงจากตาราง TabUserInfo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การออกแบบโปรแกรมฝั่งผู้ใช้

การออกแบบโปรแกรมฝั่งผู้ใช้ในโปรแกรมนี้จะใช้เอกทีพีเจิร์ฟเวอร์เพจ โดยมี VBScript และ HTML เป็นภาษาในการพัฒนา ซึ่งมีรายละเอียดดังนี้

- การลงทะเบียนใหม่
- การเปลี่ยนรหัสผ่าน
- การขอรหัสผ่านในกรณีลืมรหัสผ่าน
- ตารางเวลา
- การกำหนดคสิทธิ์
- ข้อความ
- การนัดหมาย
- การกำหนดคุณสมบัติ
- การออกจากระบบ



รูปที่ 4.6 การใช้งานของผู้ใช้

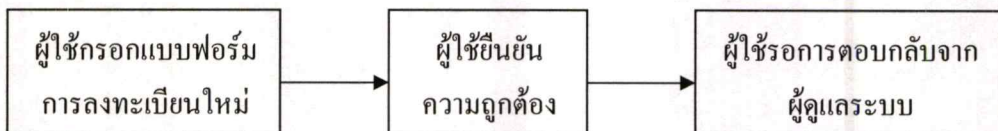
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 หน้าจอหลัก

เมื่อผู้ใช้ใส่ชื่อและรหัสผ่านแล้วคลิกปุ่ม Login โปรแกรมจะนำค่าชื่อและรหัสผ่านไปเปรียบเทียบกับค่าในตาราง TabUserInfo ถ้าชื่อและรหัสถูกต้อง ก็จะยินยอมให้เข้าสู่ระบบได้

4.4.1 การลงทะเบียนใหม่

ก่อนที่ผู้ใช้ทุกคนจะเข้ามาใช้โปรแกรมนี้จะต้องมีชื่อผู้ใช้และรหัสผ่านก่อน โดยจะต้องกรอกแบบฟอร์มให้ครบถ้วนตามที่กำหนดไว้ แล้วรอการติดต่อกลับจากผู้ดูแลระบบ โดยผู้ดูแลระบบจะทำการกำหนด Password ให้กับผู้ใช้



รูปที่ 4.8 ขั้นตอนการลงทะเบียนใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบฟอร์มการลงทะเบียน

ชื่อ :

นามสกุล :

โทรศัพท์ :

ที่อยู่ :

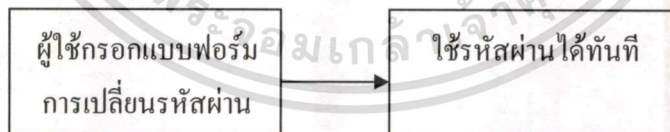
username :

email :

รูปที่ 4.9 หน้าจอแบบฟอร์มการลงทะเบียนใหม่

4.4.2 การเปลี่ยนรหัสผ่าน

ผู้ใช้งานสามารถเปลี่ยนรหัสผ่านได้ทุกครั้งตามต้องการในหัวข้อเปลี่ยนรหัสผ่าน และรหัสผ่านที่เปลี่ยนใหม่แล้วจะมีผลทันที



รูปที่ 4.10 ขั้นตอนการเปลี่ยนรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

http://localhost/enter.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail Stop

Address http://localhost/enter.asp Go Links DAP

แบบฟอร์มการเปลี่ยน Password

Username :

Password :

Confirm Password :

New Password :

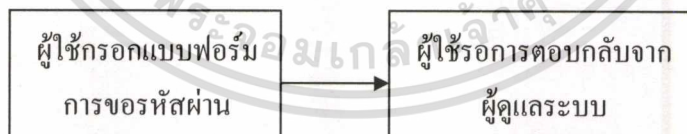
Confirm New Password :

Done Local intranet

รูปที่ 4.11 หน้าจอแบบฟอร์มการเปลี่ยนรหัสผ่าน

4.4.3 การขอรหัสผ่านในกรณีลืมรหัสผ่าน

ถ้าผู้ลืมรหัสผ่าน สามารถขอรหัสผ่านได้โดยกรอกแบบฟอร์มการขอรหัสผ่านให้ครบถ้วน จากนั้นก็รอการติดต่อกลับจากผู้ดูแลระบบ คล้ายๆกับการลงทะเบียนใหม่



รูปที่ 4.12 ขั้นตอนการขอรหัสผ่านในกรณีลืมรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

http://localhost/enter.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Refresh

Address http://localhost/enter.asp Go Links DAP

แบบฟอร์มการขอ Password

ชื่อ :

นามสกุล :

โทรศัพท์ :

ที่อยู่ :

username :

email :

Done Local intranet

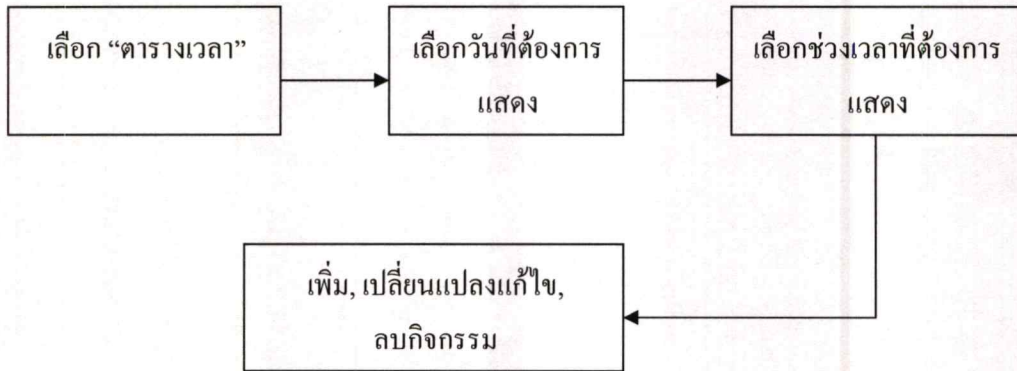
รูปที่ 4.13 หน้าจอแบบฟอร์มการขอรหัสผ่านในกรณีลืมรหัสผ่าน

4.4.4 ตารางเวลา

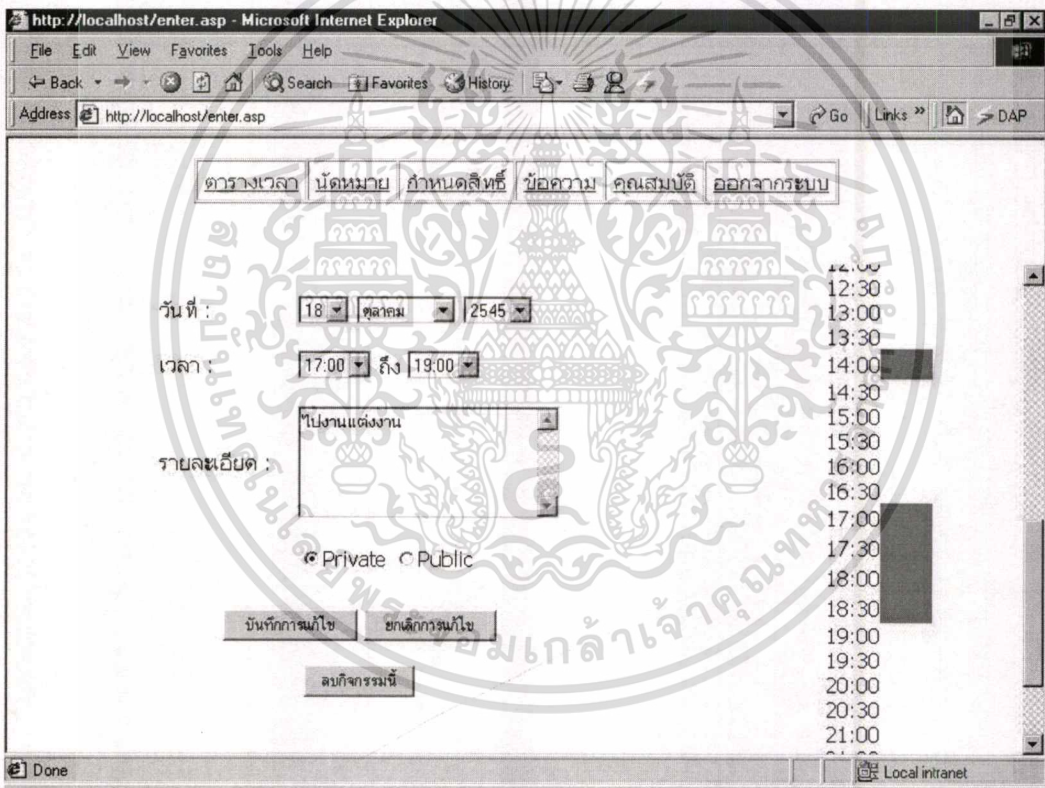
ขั้นตอนนี้เป็น การแก้ไขตารางเวลาของผู้ใช้เอง คือการป้อนข้อมูลเกี่ยวกับตัวผู้ใช้อเองเข้าไปในฐานข้อมูล เพื่อที่จะเป็นข้อมูลให้ผู้อื่นที่จะนัดหมายได้ เห็นรายละเอียดว่า ผู้ใช้มีเวลาว่างหรือไม่ ว่างวันไหนบ้าง ซึ่งจะทำให้ผู้อื่นนั้นนัดหมายตามเวลาที่ผู้ใช่ว่างได้อย่างถูกต้อง โดยการแก้ไขตารางเวลานี้สามารถกำหนดสิทธิ์ได้ 2 แบบคือ

1. Private ข้อมูลใดที่มีการกำหนดสิทธิ์แบบนี้ จะเป็นข้อมูลที่เจ้าของข้อมูลนี้เท่านั้นที่มีสิทธิ์ดูได้ คนอื่นจะมองเห็นเพียงไม่ว่างเท่านั้น แต่จะไม่เห็นรายละเอียดของข้อมูลนี้
2. Public ข้อมูลใดที่มีการกำหนดสิทธิ์แบบนี้ จะเป็นข้อมูล queผู้อื่นสามารถเห็นรายละเอียดของข้อมูลนี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 ขั้นตอนของตารางเวลา



รูปที่ 4.15 หน้าจอตารางเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.5 การกำหนดสิทธิ์

การนัดหมายในชีวิตประจำวันนั้น บุคคลต่างๆที่จะนัดหมายย่อมมีความสำคัญแตกต่างกันไปเช่น บุคคลที่ไม่รู้จัก, บุคคลที่รู้จักแต่ไม่สนิท หรือบุคคลที่สนิทคุ้นเคยกันมาก เป็นต้น ดังนั้นจึงต้องมีการกำหนดสิทธิ์ในการนัดหมายให้แก่บุคคลต่างๆ โปรแกรมตารางนัดหมายบนเว็บได้จัดการด้านนี้ไว้แล้ว โดยได้แบ่งการกำหนดสิทธิ์ออกเป็น 4 ระดับด้วยกันคือ

1. ระดับที่ 1 ไม่ยอมให้มีการนัดหมาย

ถ้าผู้ใช้ให้สิทธิ์นี้แก่ใครแล้ว ผู้นั้นจะไม่สามารถนัดหมายผู้ใช้ได้เลย ซึ่งการสิทธิ์แบบนี้จะเป็นค่าเริ่มต้น (Default) ของทุกคนที่เพิ่งเข้าสู่ระบบครั้งแรก

2. ระดับที่ 2 ยอมให้มีการนัดหมายได้ แต่ไม่เห็นรายละเอียด ต้องรอคำตอบ

การให้สิทธิ์แบบนี้ ผู้นัดหมายจะสามารถนัดหมายผู้ใช้ได้ แต่จะไม่เห็นรายละเอียดของข้อมูลนั้นๆ โดยจะรู้เพียงว่าผู้ถูกนัดไม่ว่างในเวลาใดเท่านั้น

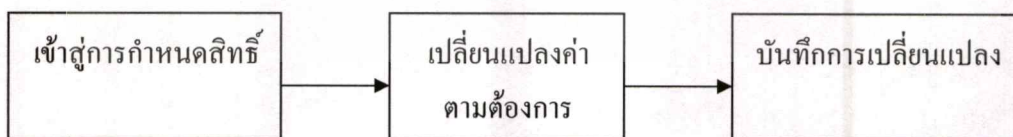
3. ระดับที่ 3 ยอมให้มีการนัดหมายได้ และเห็นรายละเอียด ต้องรอคำตอบ

การให้สิทธิ์แบบนี้ ผู้นัดหมายจะสามารถนัดหมายผู้ใช้ได้ และจะเห็นรายละเอียดของข้อมูลนั้นๆเช่น ไปดูหนัง เป็นต้น

4. ระดับที่ 4 ยอมให้มีการนัดหมายได้ และไม่ต้องรอคำตอบ

สิทธิ์แบบนี้จะเป็นแบบที่ให้สิทธิ์มากที่สุด เพราะไม่ต้องรอคำตอบ ระบบจะถือว่าผู้ใช้ตอบตกลงไปโดยอัตโนมัติ

อนึ่ง ถ้าข้อมูลนั้น เจ้าของได้ใส่ข้อมูลว่าเป็นแบบ Private ถึงแม้ว่าจะให้สิทธิ์แบบที่ 3 หรือ 4 ก็ตาม ก็จะมองไม่เห็นรายละเอียดของข้อมูลนั้นๆอยู่ดี แต่จะมองเห็นเพียงว่า ข้อมูลส่วนตัวเท่านั้น



รูปที่ 4.16 ขั้นตอนการกำหนดสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

http://localhost/enter.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites History

Address http://localhost/enter.asp Go Links DAP

ตารางเวลา นัดหมาย กำหนดสิทธิ์ ข้อความ คุณสมบัติ ออกจากระบบ

รายชื่อบุคคลในระบบ

หมายเลข	ชื่อ	นามสกุล	แบบที่ 1	แบบที่ 2	แบบที่ 3	แบบที่ 4
1	เรืองเดช	กุลมงคล	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	วิษิต	นันทรัตนพงศ์	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
3	วิเศษฐ	แหวนวิเศษ	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	พรพล	สารยศ	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Update Reset

หมายเหตุ
 แบบที่ 1. ไม่ยอมให้ทำการนัดหมายและไม่เห็นรายลขเอียด (Default)
 แบบที่ 2. ยอมให้มีการนัดหมาย ไม่เห็นรายลขเอียด นัดแล้วต้องรอคำตอบ
 แบบที่ 3. ยอมให้มีการนัดหมาย เห็นรายลขเอียด นัดแล้วต้องรอคำตอบ
 แบบที่ 4. ยอมให้มีการนัดหมาย เห็นรายลขเอียด นัดแล้วไม่ต้องรอคำตอบ

Done Local intranet

รูปที่ 4.17 หน้าจอการกำหนดสิทธิ์

4.4.6 ข้อความ

เมื่อมีการนัดหมาย โปรแกรมก็จะส่งข้อความไปถึงผู้รับเพื่อให้ผู้รับ ตอบตกลง ปฏิเสธ หรือยกเลิกการนัดหมาย

ส่วนเรื่องการลบข้อความนั้น ก็จะไม่ลบจริงๆ แต่จะมี field หนึ่งถูก set ค่า เช่นเป็น 0 หมายถึงลบแล้วเป็นต้น ข้อความเหล่านี้ก็จะมีลักษณะคล้ายกับ file ใน recycle bin ผู้ใช้สามารถเข้ามาอ่านข้อความย้อนหลังที่ลบไปแล้วได้ รวมทั้งยังสามารถ empty recycle bin ได้ด้วย ก็จะเป็นการลบจริงๆเลย เพราะผู้ใช้เป็นคนลบเอง แต่หากผู้ใช้ไม่ได้ทำการลบข้อความเลย ย้อนหลังไปนานมาก ก็จะมี store procedure ซึ่งจะคอยรันอยู่เสมอ อาจจะทุกวัน ตอนเที่ยงคืน คอยลบเมลที่เก่าเกินไปทิ้งไป ในส่วนนี้ก็แล้วแต่กำหนด เช่น เก่าเกิน 1 ปี เป็นต้น แต่ก็ต้องมีการประกาศไว้ในส่วนของหัว recycle bin ให้ผู้ใช้ได้ทราบด้วยว่า ระบบจะทำการลบเมลที่เก่าเกินกว่า 1 ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางเวลา นัดหมาย กำหนดสิทธิ์ ข้อความ คุณสมบัติ ออกจากระบบ

ข้อความ

Delete	No.	วันเวลาที่ได้รับ	ผู้ส่ง	ชื่อเรื่อง
<input type="checkbox"/>	1	16/10/2545 11:38:13	คุณ สมทบ กุลมงคล	พบปะสังสรรค์

ยืนยัน ปฏิเสธ
ยกเลิก

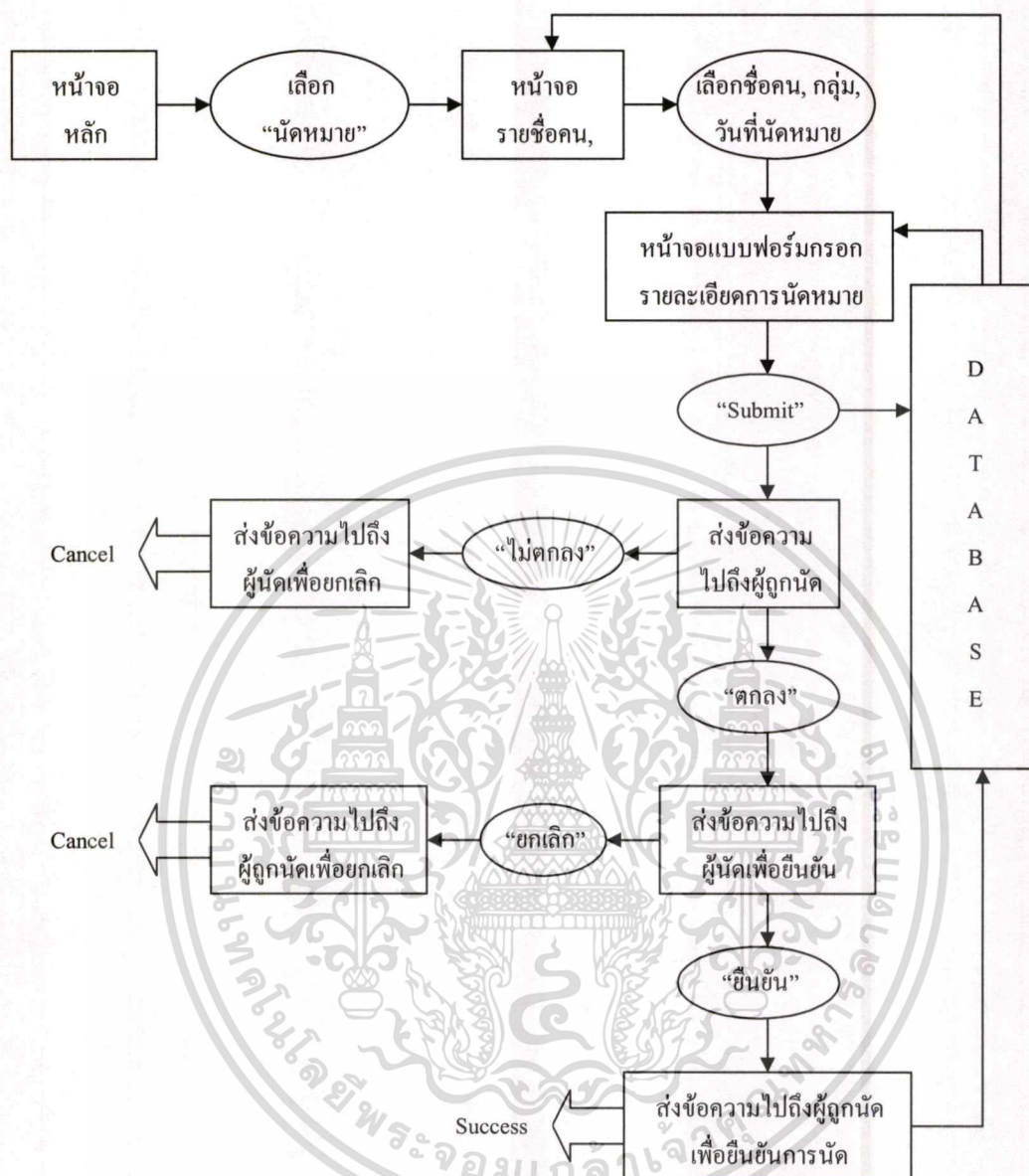
Done Local intranet

รูปที่ 4.18 หน้าจอข้อความ

4.4.7 การนัดหมาย

การนัดหมายมีขั้นตอนตามรูปที่แสดงไว้ด้านล่าง โดยสามารถนัดหมายคนเดียวหรือพร้อมกันครั้งละหลายคนก็ได้ ผู้นัดหมายสามารถยืนยันหรือยกเลิกการนัดหมายได้ด้วย โปรแกรมจะนัดหมายกับผู้ถูกนัดหมายที่ตอบตกลงเท่านั้น โดยจะถือว่าผู้ที่ไม่ตอบการนัดหมายคือการปฏิเสธ โดยสามารถนัดหมายได้ก่อนเป็นเวลา 6 เดือนนับจากปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

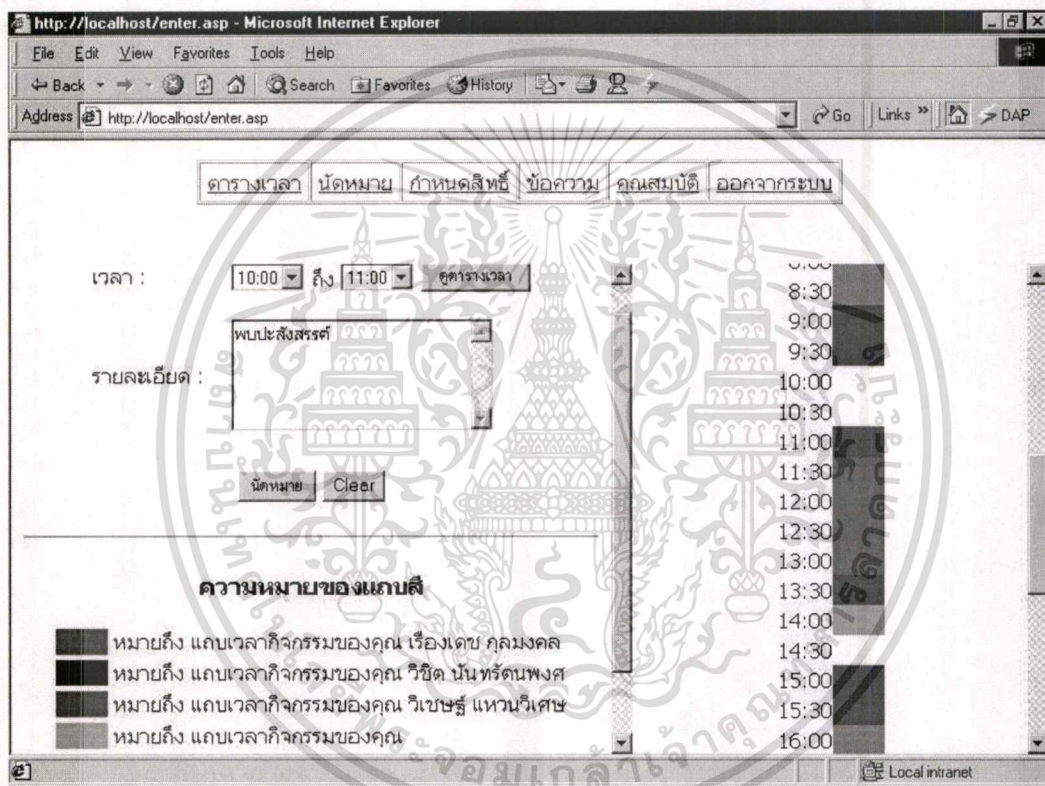


รูปที่ 4.19 ขั้นตอนการนัดหมาย

ขั้นตอนการนัดหมายจะอยู่ที่การกำหนดสิทธิ์ด้วยเพราะถ้าเป็นสิทธิ์แบบนัดได้เลย ก็ไม่ต้องมีการตอบกลับจากผู้ถูกนัดเพราะนัดได้เลยอยู่แล้ว แต่ถ้าเป็นการกำหนดสิทธิ์แบบอื่น ก็ต้องมีการตอบกลับจากผู้ถูกนัดเช่น นัดไป 5 คนพร้อมๆกัน มีการตอบกลับมา 3 คน ผู้บังคับก็จะได้รับข้อความกลับมา 3 ข้อความ ตอนนี้นัดหมายยังถือว่าไม่สมบูรณ์ ผู้บังคับก็จะเป็นผู้ตัดสินใจอีกทีว่า จะยังมีการนัดหมายนี้เกิดขึ้นอีกหรือไม่ สมมติว่ายังยืนยันการนัดอยู่ ก็จะส่งข้อความกลับไปบอกทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 คนว่ายืนยัน แต่ถ้าต้องการยกเลิก ก็จะส่งข้อความไปให้ทั้ง 5 คนเลยว่ายกเลิก เมื่อมีการยืนยันการนัดหมายเรียบร้อยแล้ว หากผู้นัดต้องการยกเลิกการนัดหมายนั้นๆก็สามารถทำได้โดยการคลิก “ยกเลิก” พร้อมทั้งกรอกเหตุผล ก็จะมีข้อความส่งไปบอกทั้ง 3 คนที่ยืนยันการนัดหมายกันไปแล้ว ก็จะถือว่ายกเลิกนัดเลย หรือ 1 ใน 3 คนที่เป็นผู้ถูกนัดติดธุระฉุกเฉินก็สามารถคลิก “ยกเลิก” พร้อมทั้งกรอกเหตุผล ก็จะมีข้อความส่งไปบอกผู้นัด แต่จุดนี้ยังไม่ถือว่ายกเลิกนัด ผู้นัดจะเข้ามาอ่านเมลล์แล้วจะเป็นผู้ตัดสินใจอีกทีว่าจะยกเลิกหรือไม่ หากยกเลิกก็จะส่งข้อความไปบอกทั้ง 3 คนว่ายกเลิกการนัดหมาย พร้อมทั้งเหตุผล



รูปที่ 4.20 หน้าจอการนัดหมาย

4.4.8 การกำหนดคุณสมบัติ

ผู้ใช้สามารถกำหนด วันและเวลาการทำงานของตัวเองได้ในหัวข้อนี้ ซึ่งจะมีผลให้ผู้อื่นไม่สามารถนัดหมายท่านได้นอกเหนือวันและเวลาทำงานของท่าน รวมทั้งการกำหนดการนัดหมายล่วงหน้าในกรณีที่ท่านให้สิทธิ์แบบที่ 4 เพื่อป้องกันการนัดหมายที่กระทบกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

http://localhost/enter.asp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Refresh

Address http://localhost/enter.asp Go Links DAP

ตารางเวลา นัดหมาย กำหนดสิทธิ์ ข้อความ คุณสมบัติ ออกจากระบบ

กรุณาเลือกวันทำงานของท่าน :

วันอาทิตย์ วันจันทร์ วันอังคาร วันพุธ วันพฤหัสบดี วันศุกร์ วันเสาร์

กรุณาเลือกช่วงเวลาทำงานของท่าน :

เวลาทำงาน : 07:00 ถึง 17:00

ผู้นัดหมายจะต้องนัดหมายท่านก่อนล่วงหน้าเป็นเวลา 5 วัน

บันทึกการเปลี่ยนแปลง ยกเลิกการเปลี่ยนแปลง

หมายเหตุ
ผู้อื่นจะไม่สามารถนัดหมายท่านได้ออกเหนือวันและเวลาทำงานของท่าน

Done Local intranet

รูปที่ 4.21 หน้าจอการกำหนดคุณสมบัติ

4.4.9 การออกจากระบบ

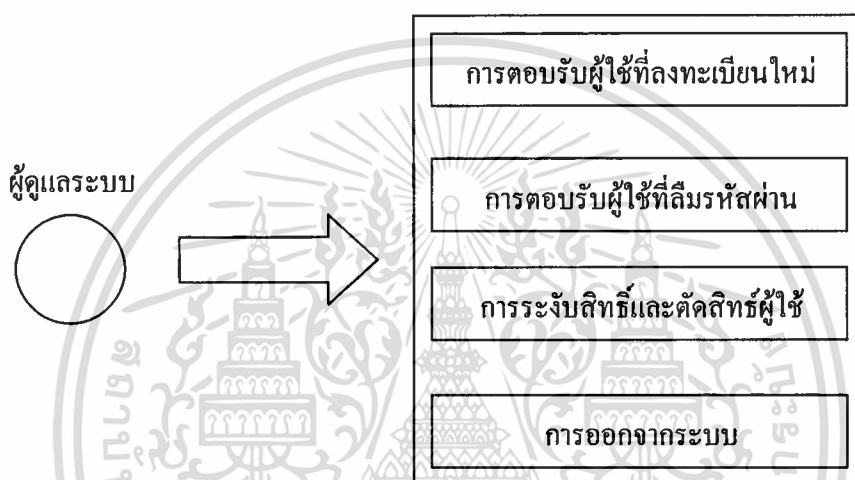
การออกจากระบบทำได้โดยการกดปุ่ม ออกจากระบบ โปรแกรมจะ Logout และกลับไปยังหน้าจอหลัก (หน้าจอ Login) อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การออกแบบโปรแกรมฝั่งผู้ดูแลระบบ

ผู้ดูแลระบบมีหน้าที่ตรวจสอบและดูแลระบบ เช่นการอนุญาตให้ผู้ใช้ใหม่ได้ลงทะเบียน เป็นต้น ซึ่งสามารถแบ่งเป็นหัวข้อย่อยๆ ได้ดังนี้

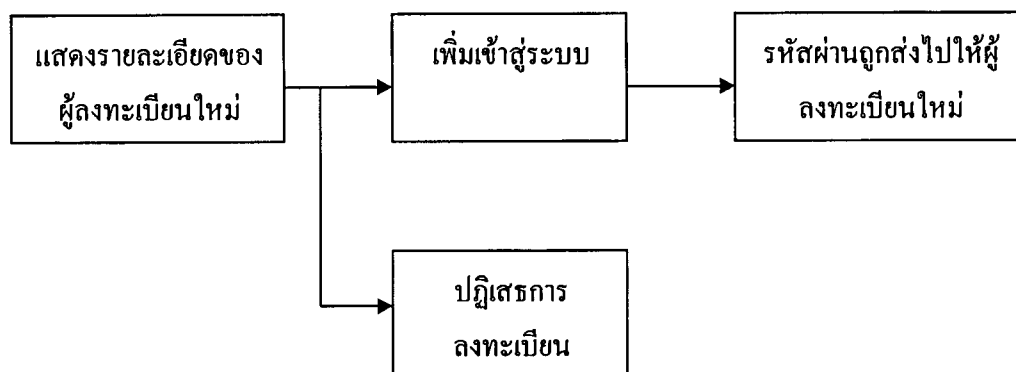
- การตอบรับผู้ใช้ที่ลงทะเบียนใหม่
- การตอบรับผู้ใช้ที่ลืมรหัสผ่าน
- การระงับสิทธิ์และตัดสิทธิ์ผู้ใช้
- การออกจากระบบ



รูปที่ 4.22 การใช้งานของผู้ดูแลระบบ

4.5.1 การตอบรับผู้ใช้ที่ลงทะเบียนใหม่

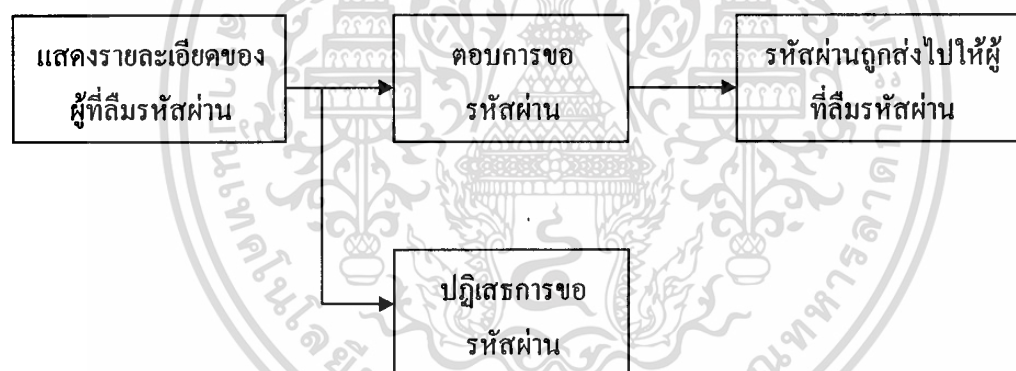
เมื่อผู้ใช้ได้มีการลงทะเบียนใหม่นั้น ข้อมูลที่ผู้ใช้ได้กรอกแบบฟอร์มมานั้นจะถูกส่งมาถึงผู้ดูแลระบบ ซึ่งจะเป็นผู้ตรวจสอบและอนุญาต



รูปที่ 4.23 ขั้นตอนการตอบรับผู้ใช้ที่ลงทะเบียนใหม่

4.5.2 การตอบรับผู้ใช้ที่ลืมรหัสผ่าน

เมื่อผู้ใช้กรอกแบบฟอร์มการลืมรหัสผ่านแล้ว ผู้ดูแลระบบจะเป็นผู้ตรวจสอบและสร้างรหัสผ่านให้ผู้ใช้ใหม่ และตอบกลับไปยังผู้ใช้นั้น

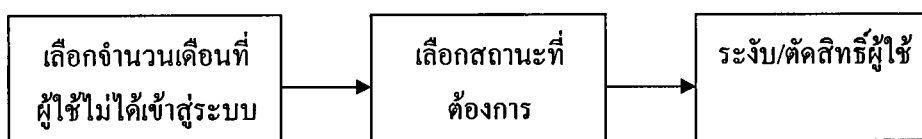


รูปที่ 4.24 ขั้นตอนการตอบรับผู้ใช้ที่ลืมรหัสผ่าน

4.5.3 การระงับสิทธิ์และตัดสิทธิ์ผู้ใช้

ผู้ดูแลระบบสามารถระงับสิทธิ์หรือตัดสิทธิ์ผู้ใช้ที่ไม่ได้เข้าสู่ระบบนานกว่า 3, 6 หรือ 12 เดือนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 ขั้นตอนการระงับสิทธิ์และตัดสิทธิ์ผู้ใช้

4.5.4 การออกจากระบบ

ใหม่ ใช้เพื่อออกจากระบบและกลับสู่หน้าจอ Login อีกครั้งหนึ่งเพื่อรอการ Login ครั้ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป

5.1 สรุป

โครงการนี้เป็นการสร้างโปรแกรมที่ทำหน้าที่เป็นตารางเวลาส่วนตัว สามารถจัดการเกี่ยวกับการนัดหมายได้ ไม่ว่าจะเป็นการนัดหมายคนคนเดียวหรือครั้งละหลายคนพร้อมกัน รวมทั้งยังคำนึงถึงการกำหนดสิทธิ์ของบุคคลต่างๆอีกด้วย

ประโยชน์ที่ได้จากโปรแกรมนี้อคือ ช่วยเตือนความจำในการจัดการด้านเวลาเช่น มีนัดกับใคร, มีกิจกรรมทำอะไร เวลาไหนบ้าง เป็นต้น จึงทำให้การจัดการด้านเวลาของผู้ใช้เป็นไปอย่างสะดวก รวดเร็ว และถูกต้อง หลีกเลี่ยงการนัดหมายที่ซ้ำซ้อนที่อาจเกิดขึ้นได้

5.2 ปัญหาของโครงการและข้อเสนอแนะ

- การออกแบบ Interface ยังไม่สวยงาม ควรออกแบบให้สวยงามกว่านี้
- การออกแบบฐานข้อมูลยังไม่ดีพอ มีการซ้ำซ้อนของข้อมูลมากทำให้มีตารางเกิดขึ้นเป็นจำนวนมาก ควรทำการ Normalization ให้ดี
- การส่งข้อมูลที่สำคัญยังไม่ปลอดภัยเพียงพอ ควรมีการเข้ารหัสเพื่อความปลอดภัย

บรรณานุกรม

บัณฑิต จามรภูติ. 2541. การใช้งานฐานข้อมูลเชิงสัมพันธ์ Microsoft SQL Server. กรุงเทพฯ:
ซีเอ็ดยูเคชั่น.

Andrew M.Fedorchek, David K.Resin. ASP : Active Server Pages. Foster City, CA:
IDG Books Worldwide.

Will Fastie. นฤตม์ วรรณเลิศลักษณ์. 2542. “เรียนรู้การทำงานของไคลเอ็นต์เซิร์ฟเวอร์.”
PC Magazine. 137-139



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

- ชื่อ นาย สมทบ กุลมงคล
- เกิดวันที่ 5 กุมภาพันธ์ พ.ศ.2522
- สถานที่เกิด อ.หาดใหญ่ จ.สงขลา
- ศึกษาในระดับประถมศึกษาที่ รร.เพชรรัตน์
ศึกษาในระดับมัธยมศึกษาตอนต้นที่ รร.ราชบพิตร
ศึกษาในระดับมัธยมศึกษาตอนปลายที่ รร.โยธินบูรณะ
ศึกษาในระดับอุดมศึกษาที่ ม.เกษตรศาสตร์ คณะ วิศวกรรมศาสตร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้