



การระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์

Accounting Management System for Engineering Faculty



นางสาวนุชจิรา ใจตรงค์ รหัส 45380063
นางสาวแคทรียา พิลาปิงคำ รหัส 45380187

5078874 e.2

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 13 พ.ย. 2549
เลขทะเบียน..... 4900133
เลขเรียกหนังสือ..... 2/5,
มหาวิทยาลัยนเรศวร 77240
2548.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2548



ใบรับรองโครงการนิสิต

หัวข้อโครงการ	ระบบการจัดการงานบัญชีของคณะวิทยาศาสตร์
ผู้ดำเนินโครงการ	นางสาว นุชจิรา ใจตุรงค์ รหัสนิสิต 45380063 นางสาว แคทรียา พิสีปิงคำ รหัสนิสิต 45380187
อาจารย์ที่ปรึกษา	ดร. สุรเชษฐ์ กานต์ประชา
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2548

คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะกรรมการสอบโครงการนิสิต


.....ประธานกรรมการ
(ดร. สุรเชษฐ์ กานต์ประชา)


.....กรรมการ
(ดร. สมยศ เกียรติวนิชวิไล)


.....กรรมการ
(อาจารย์ พันัส นัตถุทธิ์)

หัวข้อโครงการ	ระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์
ผู้ดำเนินโครงการ	นางสาว นุชจิรา ใจตุรงค์ รหัสบัณฑิต 45380063 นางสาว แคทรียา พิศปิงคำ รหัสบัณฑิต 45380187
อาจารย์ที่ปรึกษา	ดร. สุรเชษฐ์ กานต์ประชา
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2548

บทคัดย่อ

โครงการนี้เป็นการพัฒนาระบบฐานข้อมูล และการจัดทำเว็บไซต์ระบบฐานข้อมูลการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์ ให้มีประสิทธิภาพมากยิ่งขึ้น โดยมีวัตถุประสงค์ เพื่อสร้างโปรแกรมที่สามารถติดต่อฐานข้อมูล เพื่อดูแลข้อมูลเกี่ยวกับระบบบัญชีของคณะวิศวกรรมศาสตร์

จากผลการทดสอบประสิทธิภาพ ของโปรแกรมระบบการจัดการงานบัญชี ของคณะวิศวกรรมศาสตร์ พบว่า การทำงานของโปรแกรมสามารถทำงานได้เป็นอย่างดี แต่อย่างไรก็ตาม ทางผู้จัดทำเห็นว่าโปรแกรมยังมีข้อจำกัด โดยที่ในอนาคตสามารถที่จะมีการนำมาพัฒนาต่อไป เพื่อช่วยตัดสินใจในการอนุมัติงบประมาณประจำปีของผู้บริหารและสามารถวางแผนระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์ได้

Project Title Accounting Management System for Engineering Faculty
Name Miss. Nutchira Jaiturung ID. 45380063
Miss. Khatthareeya Pitpingkham ID. 45380187
Project Advisor Surachet Kanprachar ,Ph.D
Major Computer Engineering
Department Electrical and Computer Engineering
Academic Year 2004

ABSTRACT

In this project, the database system and website relate to accounting management of Engineering Faculty are developed. The main purpose of this work is to develop a computer program to communicate to the database and to take care about accounting in Engineering Faculty.

From the testing of the developed system, it is found that the database and the website of this project work appropriately and be able to serve as expected. However, there are some limitations needed to be developed further in order to help increasing the efficiency of the system.

กิตติกรรมประกาศ

โครงการวิศวกรรมคอมพิวเตอร์สำเร็จได้ด้วยดี ก็เนื่องด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงการ คือ ท่านอาจารย์ ดร. สุรเชษฐ์ กานต์ประชา ผู้ซึ่งกรุณาให้ความรู้ คำแนะนำและเอาใจใส่เป็นอย่างดี ระหว่างการดำเนินโครงการ อีกทั้งยังตรวจสอบ แก้ไขข้อบกพร่องต่าง ๆ จนโครงการนี้เสร็จสมบูรณ์ ทางคณะผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบคุณท่านอาจารย์ ดร.สมยศ เกียรติวนิชวิไล และ ท่านอาจารย์ พนัส นัถฤทธิ์ ที่ได้กรุณารับเป็น กรรมการตรวจสอบโครงการ และให้คำแนะนำ ตรวจสอบแก้ไขโครงการทำให้โครงการนี้สมบูรณ์ยิ่ง

ในโอกาสนี้ทางคณะผู้จัดทำโครงการ จึงขอขอบคุณทุก ๆ ท่านที่มีส่วนช่วยให้โครงการนี้ประสบความสำเร็จได้ด้วยดี



นางสาวนุชจิรา ใจตรงค์
นางสาวแคทรียา พิศปังคำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนวิธีการดำเนินการ	3
1.5 ผลที่คาดว่าจะได้รับ	4
1.6 งบประมาณที่ต้องใช้ในการดำเนินการ	4
บทที่ 2 หลักการและทฤษฎี	
2.1 ระบบฐานข้อมูล [1]	5
2.2 สถาปัตยกรรมของระบบฐานข้อมูล [1]	6
2.3 แบบจำลองของฐานข้อมูลเชิงสัมพันธ์ (Relation Database Model) [1]	7
2.4 โครงสร้างของฐานข้อมูลแบบ Relational [1]	8
2.5 การออกแบบฐานข้อมูล [1]	13
2.6 Data Flow Diagram (DFD) [3]	17
2.7 Entity-Relationship Model [1,4]	20
2.8 การทำงาน Normalization [1, 2]	38
2.9 ภาษา PHP [4]	46
2.10 Macromedia Dreamweaver [5]	50
2.11 MySQL [6]	51
2.12 Mambo [9]	63

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบโปรแกรมฐานข้อมูลการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์	
3.1 แผนภาพกระแสข้อมูล (Data Flow Diagram)	65
3.2 การออกแบบฐานข้อมูล	71
บทที่ 4 การพัฒนาโปรแกรม	
4.1 แผนภาพการทำงานของโปรแกรม	77
4.2 รูปแบบของโปรแกรมระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์	82
4.3 ทดสอบโปรแกรม Web Application	88
4.4 ทดสอบโปรแกรม PHP ติดต่อกับฐานข้อมูล	92
บทที่ 5 สรุปผลดำเนินการและข้อเสนอแนะ	
5.1 สรุปผลการดำเนินโครงการ	100
5.2 ข้อเสนอแนะ	100
เอกสารอ้างอิง	102
ภาคผนวก	
ก การออกแบบฐานข้อมูล	103
ข การติดตั้ง Web Server	109
ค รูปแบบการติดตั้งในส่วนของ Web Application	121
ประวัติผู้เขียนโครงการ	122

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่าง Relation [1].....	8
2.2 ตัวอย่าง Relation [1].....	8
2.3 ตัวอย่าง Relation [1].....	9
2.4 ตัวอย่าง Relation [1].....	9
2.5 ตัวอย่าง Relation [1].....	10
2.6 ตัวอย่าง Relation [1].....	11
2.7 ตัวอย่าง Relation [1].....	11
2.8 ตัวอย่าง Primary Key [1].....	12
2.9 ตัวอย่าง Primary Key [1].....	13
2.10 ตัวอย่าง Candidate Key [1].....	13
2.11 ตัวอย่าง Entity [1,4].....	21
2.12 ตัวอย่าง Property [1,4].....	22
2.13 ตัวอย่าง Identity [1,4].....	22
2.14 ตัวอย่าง Relationship [1,4].....	23
2.15 ตัวอย่าง Regular Entity [1,4].....	24
2.16 ตัวอย่าง Weak Entity [1,4].....	25
2.17 ตัวอย่าง Single-value Property [1,4].....	27
2.18 ตัวอย่าง Multi – valued Property [1,4].....	28
2.19 ตัวอย่าง Relationship [1,4].....	29
2.19 (ต่อ) ตัวอย่าง Relationship [1,4].....	29
2.20 ตัวอย่าง Relationship แบบ One – to – one [1,4].....	30
2.21 ตัวอย่าง Relationship แบบ One – to- many [1,4].....	31
2.21 (ต่อ) ตัวอย่าง Relationship แบบ One – to- many [1,4].....	32
2.22 ตัวอย่าง Relationship แบบ Many – to – many [1,4].....	33
2.22 (ต่อ) ตัวอย่าง Relationship แบบ Many – to – many [1,4].....	34
2.23 Relationship แบบ N – ary Relationship [1,4].....	36
2.24 Relationship แบบ Recursive Relationship [1,4].....	37
2.25 ตัวอย่าง First Normal Form [1, 2].....	39

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
2.25 (ต่อ) ตัวอย่าง First Normal Form [1, 2].....	40
2.26 ตัวอย่าง Second Normal Form [1, 2].....	42
2.27 ตัวอย่าง Third Normal Form [1, 2].....	44
2.28 ตัวอย่าง Fourth Normal Form [1, 2].....	45
2.28 (ต่อ) ตัวอย่าง Fourth Normal Form [1, 2].....	45
2.29 Relational Operator [6]	56
2.30 Bulletin Operator [6]	56
2.31 ตารางตรรก [6].....	56
3.1 ตารางพจนานุกรมข้อมูลของตาราง TbDepartment.....	72
3.2 ตารางพจนานุกรมข้อมูลของตาราง TbPlanMain.....	72
3.3 ตารางพจนานุกรมข้อมูลของตาราง TbPlanThird.....	75
3.4 ตารางพจนานุกรมข้อมูลของตาราง TbDocDetail.....	76
ก-1 ตารางพจนานุกรมข้อมูลของตาราง TbPlanFirst.....	103
ก-2 ตารางพจนานุกรมข้อมูลของตาราง TbPlanSecond.....	104
ก-3 ตารางพจนานุกรมข้อมูลของตาราง TbGroupDoc.....	105
ก-4 ตารางพจนานุกรมข้อมูลของตาราง TbProjectDoc.....	105
ก-5 ตารางพจนานุกรมข้อมูลของตาราง TbDepartBudget.....	106
ก-6 ตารางพจนานุกรมข้อมูลของตาราง TbTransferMain.....	106
ก-7 ตารางพจนานุกรมข้อมูลของตาราง TbTransferFrom.....	107
ก-8 ตารางพจนานุกรมข้อมูลของตาราง TbTransferTo.....	107

สารบัญรูป

รูปที่	หน้า
2.1 วงจรชีวิตการพัฒนาระบบสารสนเทศ [1].....	15
2.2 ขั้นตอนการออกแบบฐานข้อมูล [1].....	16
2.3 การแทนกระแสข้อมูลเป็นลูกศร [3].....	18
2.4 การแทนกระแสเป็นลูกศร [3].....	18
2.5 การแทนนามที่อยู่นอกระบบ [3].....	18
2.6 การแทนแหล่งเก็บข้อมูล [3].....	18
2.7 การแทนสัญลักษณ์เพิ่มเติม [3].....	19
2.8 การปฏิบัติงานของบริษัท หนังสือไทย [3].....	20
2.9 Regular Entity [1,4]	24
2.10 Weak Entity [1,4]	25
2.11 Weak Entity [1,4]	26
2.12 Composite Property [1,4]	26
2.13 Composite Property [1,4]	27
2.14 Multi-valued Property [1,4]	28
2.15 Relationship [1,4]	30
2.16 Relationship แบบ One – to – one [1,4].....	31
2.17 Relationship แบบ One – to – many [1,4].....	33
2.18 Relationship แบบ Many – to – many [1,4].....	34
2.19 Relationship แบบ Binary Relationship [1,4].....	35
2.20 Relationship แบบ Binary Relationship [1,4].....	35
2.21 Relationship แบบ Recursive Relationship [1,4].....	37
2.22 แผนผังการทำงานของ PHP [4].....	50
2.23 แผนผังการทำงานของ Mambo [9].....	63
3.1 DFD Level 0.....	66
3.2 แผนภาพ DFD ของกระบวนการ 0 ระดับ 1.....	67
3.3 แผนภาพ DFD ของกระบวนการที่ 1 ระดับที่ 1.....	68
3.4 แผนภาพ DFD ของกระบวนการที่ 2 ระดับที่ 1.....	69
3.5 แผนภาพ DFD ของกระบวนการที่ 3 ระดับที่ 1.....	70

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.6 แผนภาพ DFD ของกระบวนการที่ 4 ระดับที่ 1.....	71
3.7 แผนภาพ DFD ของกระบวนการที่ 5 ระดับที่ 1.....	71
4.1 เพิ่มข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ.....	77
4.2 ค้นหาข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ.....	78
4.3 แก้ไขข้อมูลทะเบียนรายละเอียดการเบิกจ่ายงบประมาณ.....	79
4.4 ลบข้อมูลทะเบียนรายละเอียดการเบิกจ่ายงบประมาณ.....	80
4.5 ฟังก์ชันของ โปรแกรมในส่วนแสดงข้อมูลทั่วไปเพื่อพิมพ์รายงาน.....	81
4.6 แสดงหน้าแรกเข้าโปรแกรม.....	82
4.7 แสดงเงินงบประมาณในแต่ละหน่วยงาน.....	83
4.8 แสดงการสรุปงบประมาณแต่ละหน่วยงาน.....	84
4.9 กราฟแสดงการสรุปงบประมาณแต่ละหน่วยงาน.....	85
4.10 ระบบการกรอกรายละเอียดการใช้จ่ายหมวดต่างๆ.....	86
4.11 กราฟแสดงการสรุปการเบิกจ่ายงบประมาณ.....	87
4.12 แสดงการพิมพ์รายละเอียดการขอเบิกเงินงบประมาณ.....	87
4.13 แสดงหน้าแรกของเว็บเพจ.....	88
4.14 แสดง LINKS ในส่วนของ MAIN MENU.....	89
4.15 แสดง Contact Us ในส่วนของ MAIN MENU.....	90
4.16 แสดง Administrator ในส่วนของ MAIN MENU.....	91
4.17 แสดงหน้าเข้าระบบบัญชีคณะวิศวกรรมศาสตร์.....	92
4.18 แสดงหน้า Login ระบบบัญชีคณะฯ สำหรับผู้บริหาร.....	93
4.19 แสดงหน้าแรกของยอดสรุปเงินงบประมาณของคณะวิศวกรรมศาสตร์.....	93
4.20 แสดงหน้าสรุปเงินงบประมาณในแต่ละกองทุน.....	94
4.21 แสดงหน้ารายละเอียดต่าง ๆ ของกองทุน และหมวดต่างๆ.....	94
4.22 แสดงหน้า Login ระบบบัญชีคณะฯ สำหรับหัวหน้าภาควิชา.....	95
4.23 แสดงหน้าแรกของยอดสรุปเงินงบประมาณของคณะวิศวกรรมศาสตร์.....	96
4.24 แสดงหน้าสรุปเงินงบประมาณในแต่ละกองทุน.....	96
4.25 แสดงหน้า Login ระบบบัญชีคณะฯ สำหรับ Admin.....	97
4.26 แสดงหน้าแรกของยอดสรุปเงินงบประมาณของคณะวิศวกรรมศาสตร์.....	98

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.27 แสดงหน้าสรุปรูปเงินงบประมาณในแต่ละกองทุน.....	98
4.28 แสดงหน้ารายละเอียดต่าง ๆ ของกองทุน และหมวดต่างๆ.....	99
ข-1 ตัว Setup โปรแกรม wmserver.....	109
ข-2 หน้า setup โปรแกรม (1).....	110
ข-3 หน้า setup โปรแกรม (2).....	110
ข-4 หน้า setup โปรแกรม (3).....	111
ข-5 หน้า setup โปรแกรม (4).....	111
ข-6 หน้า setup โปรแกรม (5).....	112
ข-7 หน้าตาโปรแกรมเมื่อทำการ setup เสร็จ.....	112
ข-8 หน้าของโปรแกรม Web Server ที่ทำงานแล้ว.....	113
ข-9 ตัวไอคอน wmserver รั้นอยู่ที่ Task bar.....	113
ข-10 เมนูของโปรแกรม Wmserver.....	114
ข-11 รูปไอคอน Wmserver Tools 3.2.....	114
ข-12 Wmserver Online.....	114
ข-13 เมนูการตั้งค่าโปรแกรม.....	115
ข-14 เมนูการตั้งค่าให้เป็นภาษาไทย (1).....	115
ข-15 เมนูการตั้งค่าให้เป็นภาษาไทย (2).....	116
ข-16 แสดงเมนูเป็นภาษาไทย.....	116
ข-17 การกำหนดค่าใน Text Editor (1).....	117
ข-18 การกำหนดค่าใน Text Editor (2).....	117
ข-19 แสดงการ stop service ของ apache.....	117
ข-20 MySQL Online และ Apache Offline.....	118
ข-21 แสดงหน้าของเว็บเบราว์เซอร์ ที่ Apache ไม่ได้ทำงาน.....	118
ข-22 Start Service ของ Apache.....	118
ข-23 MySQL Online และ Apache Online.....	118
ข-24 แสดงหน้าของเว็บเบราว์เซอร์ ที่ Apache ทำงาน.....	119
ข-25 ทำการ Copy File ชื่อ ControlBudget48 และ acc.....	119
ข-26 แสดงการกำหนดค่าใน ODBC.....	120

สารบัญรูป (ต่อ)

รูปที่

หน้า

ก-1 ทำการ Copy Folder acc และ web.....121



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบันไม่ว่าเราจะดำเนินงานใดๆ เราจะต้องเกี่ยวข้องกับข้อมูล อย่างใดอย่างหนึ่งอยู่เสมอ โดยเฉพาะอย่างยิ่งเมื่อเทคโนโลยีของโลกได้พัฒนาขึ้น ปัจจุบันมีการใช้คอมพิวเตอร์กันอย่างกว้างขวาง ข้อมูลต่างๆ ที่เคยจัดเก็บอยู่บนกระดาษได้ถูกนำมาจัดเก็บไว้บนคอมพิวเตอร์แทน ในตอนเริ่มต้นจะเก็บข้อมูลต่างๆ อยู่ในแต่ละแฟ้มข้อมูล แต่เมื่อปริมาณข้อมูลมีจำนวนเพิ่มมากขึ้น ประกอบกับความต้องการใช้ข้อมูลเพิ่มขึ้น การจัดเก็บข้อมูลจึงได้เปลี่ยนไปอยู่ในรูปของฐานข้อมูลแทน

เนื่องจาก ระบบการจัดการการบัญชี ของคณะวิศวกรรมศาสตร์ ยังไม่สามารถสนองความต้องการของผู้ใช้ได้เพียงพอ เช่น สรุบบประมาณของแต่ละปีการศึกษาในแต่ละภาควิชา ซึ่งในการรวบรวมข้อมูลนี้ อาจทำให้ใช้เวลานานในการเก็บรวบรวมข้อมูล ด้วยเหตุนี้ จึงได้มีการใช้ระบบฐานข้อมูลเข้ามาช่วย เพื่อความสะดวกและรวดเร็วต่อผู้ใช้งาน

1.2 วัตถุประสงค์ของโครงการ

1. สามารถเก็บรวบรวมข้อมูล เกี่ยวกับระบบบัญชี ของคณะวิศวกรรมศาสตร์ เพื่อนำไปออกแบบฐานข้อมูลได้
2. สามารถออกแบบฐานข้อมูล เพื่อเป็นประโยชน์ต่อการใช้งานในด้านระบบบัญชีได้
3. สามารถศึกษา และ เลือกเครื่องมือที่เหมาะสม เพื่อใช้ในการออกแบบโปรแกรมการจัดการฐานข้อมูลได้
4. สามารถออกแบบโปรแกรมการจัดการระบบฐานข้อมูล และฐานข้อมูลที่เกี่ยวข้องกับระบบบัญชีของคณะวิศวกรรมศาสตร์ได้
5. เพื่อความสะดวกรวดเร็วใน การจัดการระบบบัญชี ของแต่ละภาควิชาในคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

1.3 ขอบเขตของโครงการ

1. ศึกษาวิธีการออกแบบฐานข้อมูล

- การเก็บรวบรวมข้อมูล
- การเขียน Data Flow Diagram
- การเขียนแผนภาพ ER และการทำ Normalization

2. รวบรวมข้อมูลที่เกี่ยวข้องกับระบบบัญชี ของคณะวิศวกรรมศาสตร์ จากเจ้าหน้าที่ที่รับผิดชอบ

ขอบฝ่ายการเงิน

3. ออกแบบในส่วนของโปรแกรม (Application Design) และออกแบบในส่วน of ฐานข้อมูล (Database Design)

4. พัฒนาด้านแบบ (Prototype) ของระบบงาน ปรับปรุงระบบงาน และแก้ไขข้อผิดพลาด
5. นำระบบงานสารสนเทศที่พัฒนาเสร็จเรียบร้อยแล้วไปทดลองใช้งาน
6. วิเคราะห์ผลการทดสอบ การใช้งาน และสรุปผล



1.5 ผลที่คาดว่าจะได้รับ

1. เข้าใจวิธีการออกแบบระบบฐานข้อมูล และสามารถนำไปใช้งานได้จริง
2. เข้าใจการทำงานของโปรแกรมจัดการระบบฐานข้อมูลการจัดการระบบบัญชี
3. เป็นแนวทางในการนำระบบสารสนเทศไปประยุกต์ใช้ในงานด้านอื่น ๆ ได้ต่อไป
4. สามารถสร้าง และพัฒนาโปรแกรมการจัดการระบบบัญชี เพื่อเป็นประโยชน์กับคณะวิศวกรรมศาสตร์
5. ได้โปรแกรมจัดการฐานข้อมูลเกี่ยวกับระบบบัญชีของคณะวิศวกรรมศาสตร์
6. สะดวกและรวดเร็วต่อผู้ใช้งานเกี่ยวกับระบบบัญชีของคณะวิศวกรรมศาสตร์

1.6 งบประมาณที่ต้องใช้ในการดำเนินการ

1. ค่าปริญ์งาน	500	บาท
2. ค่าถ่ายเอกสารประกอบการทำโครงการ	500	บาท
3. ค่ากระดาษพิมพ์รายงาน	300	บาท
4. ค่าหมึกพิมพ์	400	บาท
5. อื่นๆ	300	บาท
รวมเป็นเงินทั้งสิ้น		2,000 บาท

(ถ้วนเฉลี่ยทุกรายการ)

บทที่ 2

หลักการและทฤษฎี

การพัฒนากระบวนการจัดการฐานข้อมูลในปัจจุบันเราจำเป็นต้องรู้จักการทำงานของระบบฐานข้อมูล องค์ประกอบ และคำศัพท์ต่างๆที่มีใช้ในการออกแบบฐานข้อมูล เพื่อจะได้ใช้เป็นความรู้ในการออกแบบ และพัฒนาระบบฐานข้อมูลได้ ทั้งยังสามารถมองถึงปัญหาของระบบและแก้ไข เพื่อให้มีประสิทธิภาพสูงสุดได้ ซึ่งจำแนกเนื้อหาได้ดังนี้

2.1 ระบบฐานข้อมูล [1]

2.1.1 ระบบเพิ่มข้อมูล

ในอดีตระบบการจัดเก็บข้อมูลยังมีการจัดเก็บเป็นแฟ้มเอกสาร เพื่อเป็นหมวดหมู่ ซึ่งอาจจะเป็นข้อมูลที่มีความสำคัญ หรือไม่มีความสำคัญ ประปนกันไป เช่น การจัดหมวดหมู่ของการเบิกเงิน มักจะจัดตามลักษณะการใช้จ่าย และเมื่อหน่วยงาน หรือองค์กรต่างๆ มีขนาดใหญ่ขึ้น เอกสารและข้อมูลทางฝ่ายบัญชี คณะวิศวกรรมศาสตร์จึงมีมากขึ้น เมื่อมีความจำเป็นต้องใช้หรือตรวจเช็คข้อมูล จึงยากและลำบากที่จะค้นหา จึงได้มีการนำคอมพิวเตอร์เข้ามาช่วยในการทำงานของหน่วยงานหรือองค์กรนั้นๆ ซึ่งรูปแบบของข้อมูลจะอยู่ใน รูปแบบแฟ้มข้อมูล เมื่อเพิ่มข้อมูลมารวมกันอยู่หลายๆ หรืออาจจะมีความสัมพันธ์กัน เราจึงนำมาไว้ด้วยกัน เรียกว่า “ระบบเพิ่มข้อมูล”

ซึ่งการที่เราจะนำข้อมูลที่อยู่ในระบบเพิ่มข้อมูลมานั้น เราจะต้องอาศัยผู้ที่มีความรู้เข้ามาช่วย ซึ่งเราเรียกว่า “โปรแกรมเมอร์” เพื่ออ่านข้อมูลต่างๆ เข้ามาช่วยในการโปรเซสของข้อมูล ซึ่งใช้ภาษาต่างๆ เช่น ฟอ์แทน, เบสิก เป็นต้น ซึ่งมีผลตามมาก็คือ ข้อจำกัดของการเรียกใช้ข้อมูล และความซับซ้อนในการพัฒนาโปรแกรม เช่นเมื่อต้องการเปลี่ยนแปลงเพิ่มข้อมูล โปรแกรมเมอร์ต้องเขียนโปรแกรมขึ้นมาใหม่

2.1.2 ปัญหาของระบบเพิ่มข้อมูล

เมื่อมีการจัดเก็บข้อมูลอยู่อย่างกระจัดกระจาย ของแต่ละหน่วยงาน หรือแต่ละองค์กร ซึ่งอาจทำให้เกิดปัญหาของ ข้อมูลซ้ำซ้อน เกิดการขัดแย้งของข้อมูลทำให้ผู้ใช้เข้าใจผิดเมื่อต้องการใช้ข้อมูลได้

2.1.3 ระบบฐานข้อมูล

จากการที่เกิดปัญหาจากหัวข้อที่ผ่านมา เราจึงต้องมีวิธีในการจัดเก็บข้อมูล ซึ่งเรานำมาเก็บในรูปแบบ “ฐานข้อมูล” นั้นเอง ซึ่งมีหลักสำคัญคือ การนำข้อมูลที่มีความสัมพันธ์กันมาไว้ในฐานข้อมูลเดียวกัน เพราะต้องอาศัยข้อมูลของกันและกันในฐานข้อมูลนั้นๆ

2.1.4 องค์ประกอบของระบบฐานข้อมูล

1. ข้อมูล
2. ฮาร์ดแวร์
3. ซอฟต์แวร์
4. ผู้ใช้หรือออกแบบฐานข้อมูล

2.2 สถาปัตยกรรมของระบบฐานข้อมูล [1]

เป็นการอธิบายถึง รูปแบบและโครงสร้างของข้อมูล โดยทั่วไปตามแนวคิด ไม่ขึ้นกับโครงสร้างของฐานข้อมูลนั้น สถาปัตยกรรมที่นิยมใช้กันคือ

2.2.1 สถาปัตยกรรม ANSI/SPARC แบ่งเป็น 3 ระดับคือ

1. ระดับ Internal ซึ่งจะเกี่ยวข้องกับ ระดับกายภาพ ในการจัดเก็บข้อมูลในฐาน ข้อมูลมากที่สุด ซึ่งจะเกี่ยวกับการจัดเก็บข้อมูล
2. ระดับ External ซึ่งจะเกี่ยวกับผู้ใช้งานที่สุด จะพูดถึงมุมมอง และความคิดของผู้ใช้ ในแต่ละคน
3. ระดับ Conceptual จะกล่าวถึง โครงสร้างข้อมูลในระดับแนวคิดแสดงออกมาในรูปแบบของภาพของ โครงสร้างแทนโครงสร้างทางกายภาพ

2.2.2 Mapping

ในสถาปัตยกรรม ANSI/SPARC นี้ มุมมองที่มีต่อข้อมูลในสถาปัตยกรรม ในระดับที่สูงกว่า สามารถที่จะถ่ายทอดมุมมองนั้นไปยังสถาปัตยกรรมที่ต่ำกว่าได้ เช่น การถ่ายทอดมุมมองที่มีต่อข้อมูลจากสถาปัตยกรรมในระดับ Conceptual ไปยังสถาปัตยกรรมในระดับ Internal เพื่อนำโครงสร้างของข้อมูลในระดับ Conceptual Schema ไปแปลงเป็นโครงสร้างของข้อมูลในระดับกายภาพ เพื่อใช้ในการจัดเก็บข้อมูล เป็นต้น สำหรับการถ่ายทอดมุมมองจากสถาปัตยกรรมในระดับที่สูงกว่าไปยังระดับที่ต่ำกว่านี้ จะเรียกว่า “การทำ Mapping” ในการทำ Mapping ตามสถาปัตยกรรม ANSI/SPARC สามารถแบ่งออกได้เป็น 2 ลักษณะดังนี้

2.2.3 สถาปัตยกรรมของระบบฐานข้อมูลกับ Database Administrator

ดังได้กล่าวมาแล้วในบทที่ 1 ว่า Database Administrator (DBA) เป็นบุคคลที่มีหน้าที่ในการกำหนดกลยุทธ์ และกลยุทธ์ในการใช้ข้อมูลในฐานข้อมูล รวมทั้งจัดหาเทคนิคที่จำเป็นต่อกลยุทธ์และกลยุทธ์ที่กำหนดขึ้น ดังนั้นผู้ใช้ทำหน้าที่เป็น DBA จึงมีความเกี่ยวข้องกับสถาปัตยกรรมของระบบฐานข้อมูลดังนี้

1. เป็นผู้กำหนดโครงสร้างของข้อมูลในสถาปัตยกรรม ในระดับ Conceptual เช่น การกำหนดว่า ข้อมูลใดบ้างที่ควรจะมีไว้ในฐานข้อมูล และมีโครงสร้างเป็นอย่างไร
2. เป็นผู้กำหนดโครงสร้าง ของข้อมูลในสถาปัตยกรรม ในระดับ Internal ซึ่งได้แก่ โครงสร้างทางกายภาพของข้อมูลในการจัดเก็บ เช่น ขนาดของแต่ละ Field ประเภทของข้อมูล ฯลฯ เป็นต้น
3. เป็นผู้ที่ทำหน้าที่ ตรวจสอบโครงสร้างข้อมูลที่กำหนดขึ้นว่า สามารถรับมือมุมมองหรือความต้องการใช้ข้อมูลของผู้ใช้ในสถาปัตยกรรมในระดับ External หรือไม่
4. เป็นผู้กำหนดการรักษาความปลอดภัย และกฎที่ใช้ ในการควบคุมความถูกต้อง ให้กับข้อมูล ซึ่งเป็นส่วนหนึ่งของการกำหนด โครงสร้างของข้อมูล ในสถาปัตยกรรมในระดับ Conceptual
5. เป็นผู้กำหนดวิธีในการสำรองข้อมูล (Data Backup) และการกู้ข้อมูลที่เสียหายกลับ มาใช้งาน (Data Recovery)
6. เป็นผู้ควบคุม ให้ระบบฐานข้อมูลมีประสิทธิภาพ และ ทันสมัยตามความต้องการ ที่เปลี่ยนแปลงไป

2.2.4 สถาปัตยกรรมของระบบฐานข้อมูล Database Management System

Database Management System (DBMS) เป็นส่วนที่มีความสัมพันธ์กับสถาปัตยกรรมของระบบฐานข้อมูล เนื่องจาก DBMS เป็นโปรแกรมที่ทำหน้าที่ รับคำสั่งต่างๆ ทั้งในกลุ่มคำสั่ง DML และกลุ่มคำสั่ง DDL ที่ผู้ใช้กำหนดในสถาปัตยกรรมในระดับ External ไปกระทำกับข้อมูลในฐานข้อมูล ที่มีโครงสร้างข้อมูลอยู่ในสถาปัตยกรรม ในระดับ Internal มาแปลงให้อยู่ในรูปแบบของโครงสร้างข้อมูลในสถาปัตยกรรมในระดับ Conceptual และ External ตามลำดับ เพื่อนำมาแสดงผลต่อผู้ใช้

2.3 แบบจำลองของฐานข้อมูลเชิงสัมพันธ์ (Relation Database Model) [1]

เป็นโครงสร้างในแบบ Relational พัฒนามาจากแบบจำลองที่กล่าวถึงความสัมพันธ์ระหว่างข้อมูลที่เรียกว่า Relative Model ข้อมูลที่จัดเก็บในแบบนี้จะเก็บเป็นส่วนย่อย ๆ ที่เรียกว่า Table อยู่ในตารางที่มีความสัมพันธ์ของสมการและแถว โดยข้อมูลแต่ละตารางจะเก็บเป็นเอกเทศ แต่สามารถเอามาสัมพันธ์กันได้ตามแนวคิด

2.4 โครงสร้างของฐานข้อมูลแบบ Relational [1]

ใน Relation Model ใช้นิยามต่างๆ คือ Relation, Attribute, Domain เพื่ออธิบายถึงข้อมูลใน Relation Model นี้

2.4.1 Relation

เป็นการจัดเก็บข้อมูลแบบตารางมีแถวและสดมภ์ ชื่อของแต่ละแถวเรียกว่า "Tuple" ส่วนชื่อของแต่ละสดมภ์เรียกว่า "Attribute"

ตารางที่ 2.1 ตัวอย่าง Relation [1]

EmpID	Name	Lastname	Sex	Salary	DepID
001	เกียรติศักดิ์	สิงห์ศิริเจริญกุล	M	10,000	02
002	พรเทพ	เกิดจันทิก	M	10,200	03

2.4.1.1 คุณสมบัติของ Relation

คุณสมบัติของ Tuple และ Attribute ของแต่ละ Relation จะประกอบด้วย

1. เนื่องจาก Relation ใน Relational Model อยู่ในรูปแบบของ เซตทางคณิตศาสตร์ ที่ภายในจะต้องประกอบด้วยสมาชิกที่มีค่าไม่ซ้ำกัน ดังนั้น ภายใน Relation ใดๆ จึงต้องมี Attribute ใด Attribute หนึ่ง ที่ทำให้แต่ละ Tuple ใน Relation มีข้อมูลที่ไม่ซ้ำกัน เช่น Relation "POPPULAR" ที่ใช้เก็บข้อมูลของประชากรในประเทศไทย ซึ่งถึงแม้ว่า จะมีบุคคลที่มีชื่อและนามสกุลที่ซ้ำกัน เช่น ประชากรชื่อ "นายสมบุรณ์ สุขมาก" แต่ข้อมูลแต่ละ Tuple ของ Relation "POPPULAR" ก็จะไม่ปรากฏข้อมูลที่ซ้ำกัน เนื่องจากค่าของข้อมูลใน Attribute "ID" ซึ่งใช้เก็บหมายเลขบัตรประจำตัวประชาชน มีข้อมูลที่ไม่ซ้ำกันดังตัวอย่างข้อมูลนี้

ตารางที่ 2.2 ตัวอย่าง Relation [1]

ID	Name	Surname	Sex
1230000100	สมบุรณ์	สุขมาก	M
1230000101	สมเกียรติ	เจริญพร	M
1230000102	สมบุรณ์	สุขมาก	M
1230000103	น้ำฝน	ม่วงทอง	F

2. ด้วยเหตุผลเช่นเดียวกับข้อที่ 1 ลำดับที่ของสมาชิกในเซตใดๆ จะไม่มีผลต่อเซตนั้น ดังนั้น ภายใน Relation จึงไม่มีการกำหนดลำดับที่ให้กับแต่ละ Tuple ใน Relation กล่าวคือ จะไม่มี

การกล่าวถึงคำว่า Tuple แรก หรือ Tuple สุดท้าย หรือ Tuple ลำดับที่ 5 หรือ Tuple ที่ผ่านมาใน Relation

3. ภายใน Relation จะไม่มีการกำหนดลำดับให้กับแต่ละ Attribute เนื่องจากในการอ้างอิง Attribute ใน Relation จะใช้ชื่อของ Attribute นั้นในการอ้างอิง ดังนั้น จึง ไม่มีการกล่าวถึงคำว่า Attribute แรก Attribute สุดท้ายหรือ Attribute ลำดับที่ 5 หรือ ที่ผ่านมา หรือ Attribute ถัดไปใน Relation เช่นเดียวกับ Tuple

4. ค่าในทุก Attribute ของ Relation จะต้องมีคุณสมบัติ Atomicity ซึ่งเป็นคุณสมบัติที่กำหนดให้ ค่าของข้อมูลในแต่ละ Attribute ของ Relation จะต้องมีความหมายใดความหมายหนึ่งเพียงความหมายเดียว ไม่ใช่กลุ่มของสิ่งใดสิ่งหนึ่ง หรือ กล่าวอีกในหนึ่ง ข้อมูลในแต่ละ Attribute ของ Relation จะต้องไม่ใช่ข้อมูลในลักษณะ Repeating Group เช่น กรณีพนักงานสามารถสังกัดฝ่ายได้มากกว่า 1 ฝ่าย ข้อมูลใน Attribute “DepID” ของแต่ละ Tuple ของ Relation “EMPLOYEE” ซึ่งใช้เก็บรหัสของฝ่ายที่ฝ่ายพนักงานแต่ละคนสังกัด จะไม่สามารถจัดเก็บรหัสของฝ่ายที่พนักงานคนนั้นสังกัดใน Tuple เดียวดังตัวอย่างข้อมูลต่อไปนี้ได้

ตารางที่ 2.3 ตัวอย่าง Relation [1]

EMPLOYEE

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01,03
00002	สมเกียรติ	เจริญพร	M	8,000	02
00003	จันจิรา	แจ้งเกิด	F	12,000	03

จากรูปจะสังเกตเห็นว่า ข้อมูลที่เก็บอยู่ใน Attribute “DepyID” พนักงานที่ชื่อ “สมบูรณ์” มีค่ามากกว่า 1 ค่า ซึ่งจากรูปคือ รหัสฝ่าย “01” และ “03” ดังนั้น relation นี้จึงต้องมีการแยกข้อมูลในลักษณะ Repeating Group นั้นออกมาเป็น Tuple ใหม่ดังตัวอย่างข้อมูลต่อไปนี้

ตารางที่ 2.4 ตัวอย่าง Relation [1]

EMPLOYEE

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
00001	สมบูรณ์	สุขมาก	M	10,000	03
00002	สมเกียรติ	เจริญพร	M	8,000	02

สำหรับคุณสมบัติของ Relation ในข้อนี้ ได้บังคับให้ทุก Relation มีรูปแบบการจัดเก็บข้อมูลที่เป็นไปตาม รูปแบบแรก (First Normal Form) ตามที่กำหนดไว้ในการทำ Normalization ให้กับ Relation ต่างๆ สำหรับรายละเอียดของการทำ Normalization ให้กับ Relation ต่างๆ สำหรับรายละเอียดของการทำ Normalization ให้กับ Relation จะกล่าวถึงในลำดับต่อไป

5. ชื่อของแต่ละ Attribute ใน Relation เดียวกัน จะต้องไม่ซ้ำกัน

6. ค่าที่ปรากฏในแต่ละ Attribute ใน Relation เดียวกัน จะต้องให้แทนข้อมูลที่ความหมายเดียวกัน เช่น Attribute “EmpID” ของ Relation “EMPLOYEE” จะต้องใช้เก็บข้อมูลรหัสพนักงานของแต่ละ Tuple เท่านั้น จะไม่สามารถใช้เก็บข้อมูลอื่นๆ เช่น เพศ ได้ดังยกตัวอย่างข้อมูลต่อไปนี้

ตารางที่ 2.5 ตัวอย่าง Relation [1]

EMPLOYEE

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
M	สมเกียรติ	เจริญพร	00002	8,000	02
00003	จันจิรา	แจ้งเกิด	F	12,000	03
00004	น้ำฝน	ม่วงทอง	F	9,500	01

2.4.1.2 ประเภทของ Relation

Relation สามารถแบ่งออกเป็นประเภทต่างๆ ได้ดังนี้

1. Named Relation

เป็น Relation ที่สร้างขึ้นด้วยคำสั่ง SQL ซึ่งอาจเป็น Relation จริงในฐานข้อมูลหรือเป็นเพียง Relation ที่สร้างขึ้นด้วยคำสั่งของ Query Language

2. Base Relation

เป็น Named Relation ในส่วนที่เป็น Relation จริงในฐานข้อมูล ซึ่งให้เก็บข้อมูลในหน่วยความจำสำรองดังนั้นจึงเป็น Relation จริงที่เกิดขึ้นจากการออกแบบฐานข้อมูล

3. Derived Relation

เป็น Named Relation ในส่วนของ Relation ซึ่งได้มาจากการใช้เงื่อนไขประกอบกับคำสั่งของ Query Language กับ Base Relation เช่น การเอา Relation “EMPLOYEE” ซึ่งเป็น Base Relation และมีข้อมูลดังตัวอย่างข้อมูลต่อไปนี้

ตารางที่ 2.6 ตัวอย่าง Relation [1]

EMPLOYEE

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
00002	สมเกียรติ	เจริญพร	M	8,000	02
00003	จันจิรา	แจ้งเกิด	F	12,000	03

มาใช้เงื่อนไข “เฉพาะพนักงานชาย” ผลที่ได้ ได้แก่ Derived Relation ที่มีชื่อว่า “MALE” ที่มีข้อมูลเฉพาะพนักงานที่มีเพศชายดังตัวอย่างข้อมูลดังต่อไปนี้

ตารางที่ 2.7 ตัวอย่าง Relation [1]

EMPLOYEE

EmpID	Name	Surname	Sex	Salary	DeptID
00001	สมบูรณ์	สุขมาก	M	10,000	01
00002	สมเกียรติ	เจริญพร	M	8,000	02

4. Expressible Relation

เป็น Relation ที่ได้มาจากการกระทำกับ Named Relation ด้วยเงื่อนไขทางด้านความสัมพันธ์ของข้อมูลระหว่าง Named Relation ที่ต้องการ ดังนั้นเมื่อนำเอาทุก Expressible Relation มาประกอบกัน ผลที่ได้ จึงได้แก่ทุกๆ Base Relation และ Derived Relation ที่ Expressible Relation นั้นที่ใช้สร้างขึ้น

5. View

เป็น Derived Relation ประเภทหนึ่ง แต่จะเป็น Relation เสมือน (Virtual Relation) ที่ถูกสร้างขึ้นไว้ในฐานข้อมูล

6. Snapshot

เป็น Derived Relation ประเภทหนึ่งเช่นเดียวกับ View แต่จะแตกต่างกันที่ Snapshot เป็น Relation ในฐานข้อมูลที่สามารถอ่านข้อมูลได้เพียงข้อมูลเดียว และสามารถกำหนดเวลาในการปรับปรุงค่าของข้อมูลใน Snapshot ได้ เช่น ทุกวัน ทุกสัปดาห์ เป็นต้น

7. Query Result

เป็น Relation ชั่วคราว ซึ่งเกิดจากการใช้ประโยคคำสั่งของ Query Language กับ Relation ในฐานข้อมูลใน Relation ประเภทนี้ จะเกิดขึ้นก็ต่อเมื่อมีการเรียกใช้เท่านั้นและจะหายไปเมื่อเลิกใช้งาน

8. Immediate Result

เป็น Relation ชั่วคราวที่เกิดขึ้น ในขณะที่ทำการประมวลผล ประโยคคำสั่งของ Query Language ที่มีความซับซ้อน เช่น คำสั่งต่อไปนี้

$((S \text{ JOIN } SP) \text{ WHERE } P\# = 'P2') [S\#, \text{CITY}]$

ในการประมวลผลจะเริ่มจากคำสั่งในวงเล็บก่อน ซึ่งได้แก่คำสั่ง “S” และ “SP” ก่อนที่จะนำไปใช้ในขั้นตอนต่อไปซึ่ง Relation ที่เกิดขึ้นชั่วคราวนี้ จะถูกเรียกว่า “Intermediate Result”

9. Stored Relation

เป็น Expressible Relation ที่สามารถจัดเก็บค่าของข้อมูลได้ ดังนั้น ในบางครั้งจึงอาจกล่าวได้ว่า Relation ประเภทนี้เป็น Base Relation

2.4.2 Key

Key คือ Attribute หรือชุดของ Attribute ที่ทำให้ข้อมูลในแต่ละ Tuple ใน Relation ไม่ซ้ำกัน เช่น Attribute EmpID ในตารางในข้อ 4.1 Key แบ่งเป็นดังนี้

1. Primary Key เป็น Key ที่ใช้ตรวจสอบการซ้ำกัน ของข้อมูลระหว่างข้อมูลที่ป้อนหาข้อมูลใหม่ให้กับ Relation เช่น Attribute EmpID

ตารางที่ 2.8 ตัวอย่าง Primary Key [1]

EmpID	Name	Lastname	Sex	Salary	DepID
001	เกียรติศักดิ์	สิงห์สุริยเจริญกุล	M	10,000	02
002	พรเทพ	เกิดจันทิก	M	10,200	03

2. Foreign Key เป็น Attribute ใด Attribute หนึ่งใน Relation ที่ใช้อ้างอิงไปยัง Attribute ที่ทำหน้าที่เป็น Primary Key ของอีกตารางหนึ่งใน “DepID” จะเป็น Foreign Key

ตารางที่ 2.9 ตัวอย่าง Foreign Key [1]

EmpID	Name	Lastname	Sex	Salary	DepID
001	เกียรติศักดิ์	สิงห์สุริจรินกุล	M	10,000	02
002	พรเทพ	เกิดจันทิก	M	10,200	03

DepID	Depname
02	การเงิน
03	พัสดุ

2.4.3 NULL

ข้อมูลที่จัดเก็บอยู่ในบานข้อมูล บ่อยครั้งที่ปรากฏข้อมูลที่ถูกจัดเก็บไม่ครบถ้วน ซึ่งอาจเกิดจากการกรอกข้อมูลไม่ครบถ้วน หรือการจัดเก็บข้อมูลที่ไม่ดีพอ เช่น ข้อมูลวันเดือนปีเกิด ของประชากร ซึ่งบางคนไม่สามารถระบุได้เพียงปี เป็นต้น แต่เนื่องจากทุก Attribute ในฐานข้อมูลแบบ Relation จะต้องมียค่า ดังนั้นเพื่อแก้ปัญหาดังกล่าว จึงได้มีการกำหนดค่าสำหรับข้อมูลที่ไม่สามารถระบุค่าได้ขึ้นมาซึ่งเรียกว่า Null ดังรูป

ตารางที่ 2.10 ตัวอย่าง Candidate Key [1]

Name	Sex	Birth_Date
สมบูรณ์ สุดขยัน	ชาย	1/8/2507
หฤทัย แสนประเสริฐ	หญิง	11/5/2511
แดง คำดี	ชาย	Null
ปานทิพย์ สุดซึ้ง	หญิง	2499

Null จะมีค่าที่แตกต่างจากช่องว่างหรือจำนวนศูนย์ เนื่องจาก Null จะไม่ใช่ค่าที่ปรากฏอยู่จริงในโลกของความเป็นจริง รวมทั้งเป็นค่าที่แทนข้อมูล ที่ไม่สามารถระบุค่าได้ ส่วนช่องว่าง หรือจำนวนศูนย์นั้น เป็นค่าที่ปรากฏอยู่จริง จึงถือว่าเป็นค่าของข้อมูลที่ระบุค่าได้

2.5 การออกแบบฐานข้อมูล [1]

ฐานข้อมูลเป็นส่วนที่สำคัญ สำหรับการออกแบบระบบสารสนเทศ ที่ใช้คอมพิวเตอร์ในการประมวลผล เนื่องจากเป็นส่วนที่ใช้ในการจัดเก็บข้อมูลต่างๆ ที่ใช้อินพุทของระบบสารสนเทศ จึงให้ความสำคัญกับการออกแบบฐานข้อมูลเช่นเดียวกับการออกแบบส่วนของการประมวลผล

2.5.1 วงจรชีวิตของการพัฒนาระบบงานสารสนเทศ (System Development Lift Cycle)

ในการพัฒนาระบบงานสารสนเทศ โดยทั่วไป จะดำเนินตามขั้นตอนต่างๆที่กำหนดไว้ใน System Development Lift Cycle (SDLC) แต่เนื่องจาก SDLC มีอยู่ด้วยกันหลายวิธี (Methodology) ดังนั้นจำนวนและรายละเอียดของขั้นตอนต่างๆจึงแตกต่างกันไปตาม Methodology ของ SDLC ที่นักพัฒนาระบบงานสารสนเทศเลือกใช้แต่อย่างไรก็ตาม ขั้นตอนต่างๆของแต่ละ Methodology ก็ไม่ได้แตกต่างกันอย่างสิ้นเชิง เนื่องจาก Methodology ของ SDLC ส่วนใหญ่ จะยึดแนวทางการแก้ไขปัญหาของ Federick Taylor ที่เรียกว่า Scientific Management เป็นหลักแต่เนื่องจากในหนังสือเล่มนี้ จะกล่าวถึงการออกแบบฐานข้อมูล ดังนั้น ขั้นตอนต่างๆในการพัฒนาระบบงานสารสนเทศของ SDLC ที่กล่าวถึงในที่นี้จึงเป็นขั้นตอนหลักๆ ที่พบอยู่ใน Methodology ต่างๆ ของ SDLC ซึ่งประกอบด้วยขั้นตอนต่างๆดังนี้

2.5.1.1 Feasibility Study เป็นขั้นตอนที่เกี่ยวกับการประเมินต้นทุนของทางเลือกต่างๆของการพัฒนาระบบงานสารสนเทศ เพื่อพิจารณาเลือกทางเลือกในการพัฒนาระบบงานสารสนเทศที่มีความคุ้มค่ามากที่สุด

2.5.1.2 Requirement Collection and Analysis เป็นขั้นตอนในการจกัรวบรวมความต้องการต่างๆ จากผู้ใช้ (User's Requirement) มาวิเคราะห์เพื่อจำแนกถึงปัญหา และความต้องการออกเป็นกลุ่ม เพื่อใช้กำหนดขอบเขต ให้กับระบบงานสารสนเทศที่จะพัฒนาขึ้น

2.5.1.3 Design เป็นขั้นตอนที่นำเอาปัญหา และความต้องการทางด้านต่าง ๆ ที่จำแนกไว้ในขั้นตอนที่ 2 มาใช้ในการออกแบบระบบงานสารสนเทศ

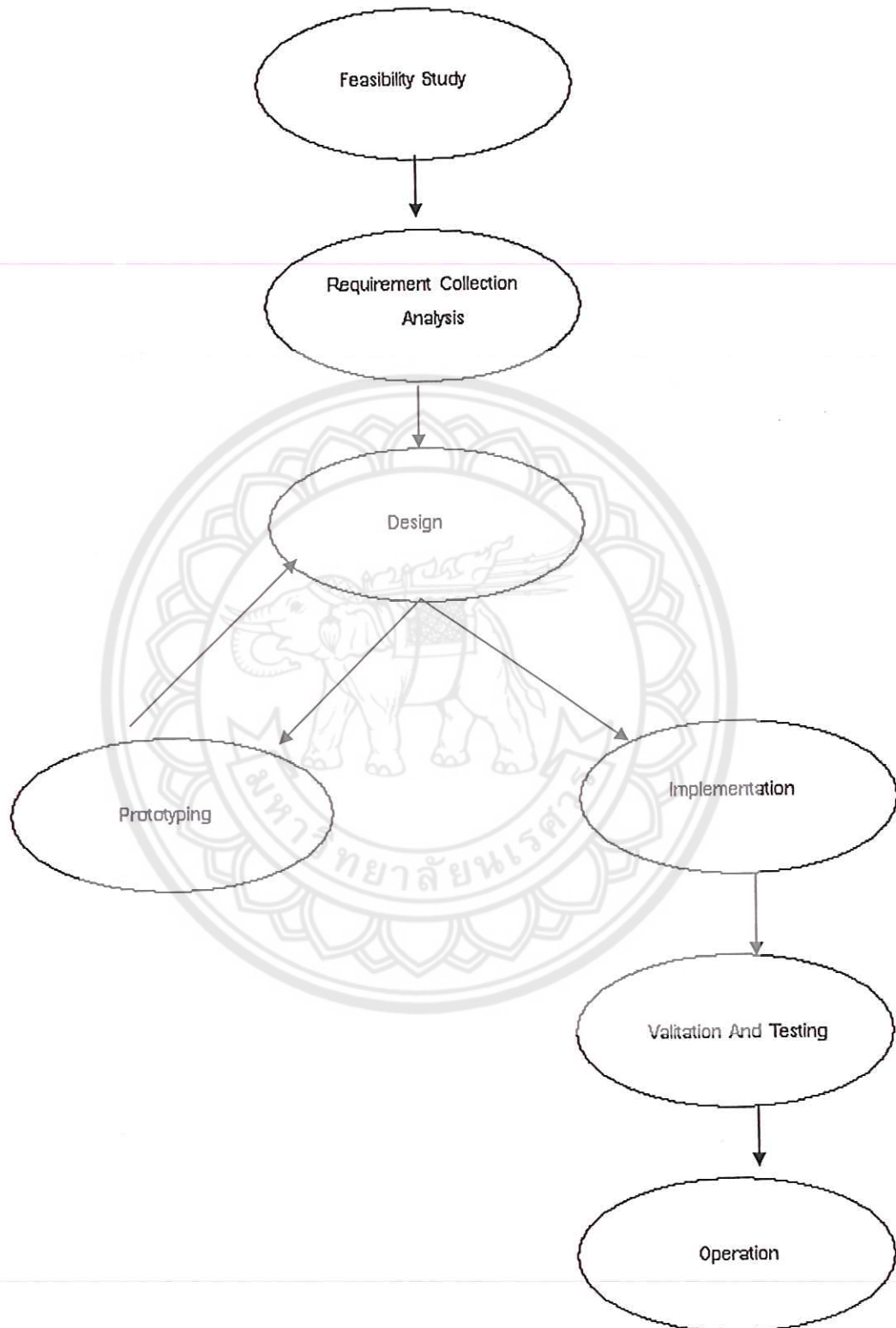
2.5.1.4 Prototyping เป็นขั้นตอนที่นำเอาส่วนต่างๆ ที่ได้ออกไว้ในขั้นตอนที่ 3 มาพัฒนาต้นแบบของระบบงาน (Prototype) เพื่อนำไปทดลองใช้ หาข้อผิดพลาดของระบบงาน ก่อนจะนำไปใช้จริง ในกรณีที่มีข้อผิดพลาดเกิดขึ้นรายละเอียดของข้อผิดพลาดต่างๆ จะถูกนำไปเป็นข้อมูลสำหรับขั้นตอนที่ 2 ได้ใหม่

2.5.1.5 Implementation เป็นขั้นตอนที่นำเอาระบบงานสารสนเทศที่พัฒนาเสร็จเรียบร้อยแล้วไปทดลองใช้งาน

2.5.1.6 Validation and Testing เป็นขั้นตอน ของการตรวจสอบความถูกต้อง ของระบบงานสารสนเทศที่พัฒนาขึ้น

2.5.1.7 Operation เป็นขั้นตอนสุดท้าย ซึ่งแน่ใจแล้วว่า ระบบงานสารสนเทศที่พัฒนาขึ้นสามารถทำงานได้อย่างถูกต้องจึงเริ่มนำข้อมูลต่าง ๆ มาใช้งานจริง

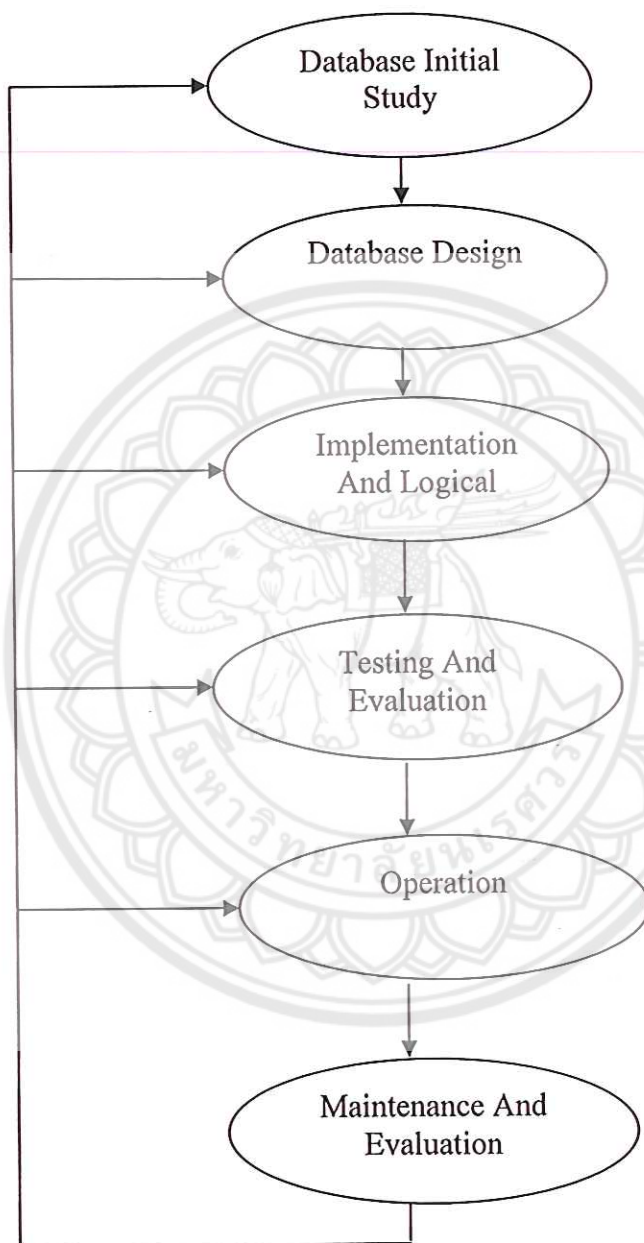
สำหรับทั้ง 7 ขั้นตอนนี้สามารถแสดงด้วยแผนภาพดังนี้



รูปที่ 2.1 วงจรชีวิตการพัฒนาาระบบสารสนเทศ [1]

2.5.2 วงจรชีวิตของการพัฒนาระบบฐานข้อมูล (Database Life Cycle)

วงจรชีวิตการพัฒนาระบบฐานข้อมูลเป็นขั้นที่กำหนดขึ้นเพื่อใช้เป็นแนวทางในการพัฒนาระบบฐานข้อมูลขึ้นใช้งาน ประกอบด้วยขั้นตอนต่างๆ ดังนี้



รูปที่ 2.2 ขั้นตอนการออกแบบฐานข้อมูล [1]

รายละเอียดที่ได้จากแต่ละขั้นตอนของการพัฒนาระบบฐานข้อมูลจะสามารถสะท้อนกลับไปขั้นตอนการทำงานก่อนหน้า ซึ่งจะช่วยปรับปรุงและแก้ไขข้อผิดพลาด ในการออกแบบฐานข้อมูล ในแต่ละขั้นตอนที่ผ่านมาได้เป็นอย่างดี

2.5.3 ขั้นตอนการออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลแบ่งออกได้เป็น 3 แบบดังนี้

2.5.3.1 การออกแบบฐานข้อมูลในระดับ Conceptual การออกแบบฐานข้อมูลในระดับ นี้ จะเป็นการกำหนดโครงสร้างเริ่มต้น ที่มีจุดมุ่งหมายเพื่ออธิบายโครงสร้างหลักๆ ของข้อมูลภายในระบบฐานข้อมูล โดยไม่คำนึงถึงฐานข้อมูลที่จะนำมาใช้การออกแบบ ในระดับนี้มีความสำคัญมาก เนื่องจากโครงสร้างที่ได้จากการออกแบบในขั้นตอนนี้จะถูกนำไปใช้ในขั้นตอนต่อไปนี้ โครงสร้างหรือที่เรียกว่า Schema ที่ได้จากการออกแบบในขั้นตอนนี้เรียกว่า Conceptual Schema

2.5.3.2 การออกแบบในระดับ Logical การออกแบบในระดับนี้จะป็นระดับที่ต่อเนื่องจากระดับ Conceptual กล่าวคือ การออกแบบในระดับนี้ จะอาศัยโครงสร้างที่ได้จากการออกแบบในระดับ Conceptual มาปรับปรุงให้มีโครงสร้างที่เป็นไปตามโครงสร้างข้อมูลที่จะนำมาใช้งาน โดยยังไม่คำนึงถึงผลิตภัณฑ์ทางด้านฐานข้อมูลที่จะนำมาใช้งาน การออกแบบในขั้นตอนนี้ต้องปรับปรุงโครงสร้างบางอย่างใน Conceptual Schema ให้สอดคล้องกับฐานข้อมูล ที่จะนำมาใช้งานการออกแบบในขั้นตอนนี้ จึงต้องมีการตรวจสอบความถูกต้องของโครงสร้างที่ออกแบบ ขึ้นกับส่วนประมวลผลต่างๆ ที่ไว้รวมทั้งจะต้องแปลงโครงสร้างต่างๆ ให้อยู่ในรูป Relation

2.5.3.3 การออกแบบฐานข้อมูลในระดับ Physical การออกแบบในระดับนี้ จะเป็นขั้นตอนสุดท้ายของการออกแบบฐานข้อมูลในขั้นตอนนี้ จะเป็นการปรับปรุงโครงสร้างของโครงสร้างที่ออกแบบเช่นเดียวกัน แต่การปรับปรุงโครงสร้างของการออกแบบฐานข้อมูล ในขั้นตอนนี้ จะเป็นการนำเอาโครงสร้างที่ได้จากการออกแบบในระดับ Logical มาปรับปรุงโครงสร้างให้เป็นไปตามโครงสร้างของผลิตภัณฑ์ ทางด้านฐานข้อมูลที่จะนำมาใช้งาน ผลิตภัณฑ์ที่ได้จากการออกแบบในระดับนี้ คือ โครงสร้างของระบบฐานข้อมูล ที่สามารถนำไปใช้ในการสร้างตัวฐานข้อมูลจริง

2.6 Data Flow Diagram (DFD) [3]

กรรมวิธีการวิเคราะห์ระบบอย่างมีโครงสร้างนั้น วิธีหนึ่งนิยมในทางปฏิบัติ คือ การมองภาพรวมในรูปแบบการไหลของข้อมูล (Data Flow) โดยที่วิธีชนิดนี้จะช่วยให้นักวิเคราะห์สามารถแบบเป็นระบบย่อยๆ ได้ง่ายขึ้นและสามารถตรวจสอบได้สะดวก

การนำเสนอระบบแบบการไหลของข้อมูลนั้นจะใช้สัญลักษณ์แทนการบรรยายการทำงานของระบบ ซึ่งลักษณะที่จะใช้เป็นรูปร่างกลม สีเหลี่ยมจัตุรัส สีเหลี่ยมผืนผ้าปลายเปิด เส้นโค้ง ลูกศร โดยนำลักษณะเหล่านี้เชื่อมต่อการแสดงต่อเนื่องของข้อมูลและการประมวลผล

2.6.1 สัญลักษณ์ Data Flow Diagram (DFD) [3]

ในแผนภาพของ DFD จะประกอบด้วยสัญลักษณ์ ดังนี้

1. ลูกศร ใช้แทนการไหลของข้อมูลพร้อมกับชื่อของข้อมูลนั้นๆ จะต้องกำกับไว้ด้วย

ชื่อข้อมูล



รูปที่ 2.3 การแทนกระแสข้อมูลเป็นลูกศร [3]

2. รูปวงกลม ใช้แทนการกระทำต่อข้อมูล ที่ไหลเข้ามาโดยไม่คำนึงถึงว่า จะเป็นการกระทำโดยคนหรือคอมพิวเตอร์ก็ตาม จะได้มาซึ่งผลลัพธ์ที่จะไหลออก จากวงกลมภายในวงกลมจะระบุคำสั่ง ๆ ที่จะใช้แทนการกระทำต่อข้อมูล



A = ข้อมูลเข้า

B = ข้อมูลออก

รูปที่ 2.4 การแทนกระแสเป็นลูกศร [3]

3. รูปสี่เหลี่ยม ใช้แทนนามที่อยู่ภายนอกระบบ ซึ่งเป็นการกำเนิดของข้อมูล หรือสิ้นสุดข้อมูลโดยมีชื่ออยู่ในสี่เหลี่ยม

ชื่อ

รูปที่ 2.5 การแทนนามที่อยู่นอกระบบ [3]

4. รูปสี่เหลี่ยมผืนผ้าปลายเปิด เป็นตัวแทนของแหล่งเก็บข้อมูล หรือ เพิ่มข้อมูล เสมือนเป็นตัวพักหรือช่วงขาดของการไหลของข้อมูล เพื่อนำไปเก็บเท่านั้น การกำหนดชื่อของแหล่งเก็บข้อมูลต้องอยู่ในสี่เหลี่ยม

ชื่อ

รูปที่ 2.6 การแทนแหล่งเก็บข้อมูล [3]

5. สัญลักษณ์เพิ่มเติม จะใช้เพิ่มเติมลงในสัญลักษณ์ ที่กล่าวมาข้างต้น เพื่อแสดงความเป็น
สิ่งเดียวกัน แต่จะถูกกล่าวหลาย ๆ ครั้งในแผนภาพ



รูปที่ 2.7 การแทน สัญลักษณ์เพิ่มเติม [3]

2.6.2 ลำดับชั้นใน Data Flow Diagram [3]

ในการเขียน DFD นักวิเคราะห์ระบบจะต้องมองระบบจากภาพรวมก่อน จากนั้นมองลึก
เข้าสู่รายละเอียดของระบบย่อยได้มากขึ้นเท่านั้น

DFD ระดับที่ 0:

ให้ถือว่าระบบทั้งระบบเป็น PROCESS หรือวงกลมหนึ่งวง มีลูกสรแทน INPUT และ
OUTPUT ตามที่จำเป็น

DFD ระดับที่ 1:

ให้แตกวงกลมระดับ 0 ออกเป็นวงกลมย่อย 2-5 วง ตามความเหมาะสม

DFD ระดับที่ 2:

ให้แตกวงกลมที่ลำดับ 1 ออกเป็นวงกลมย่อยลงไปอีกเท่าที่จะทำได้

DFD ระดับที่ 3:

ถ้าจำเป็นต้องตรวจสอบว่า วงกลมใดในภาพลำดับที่ 2 ยังมีความซับซ้อนที่จำเป็นต้องแตกย่อย
ก็ต้องสร้างแผนภาพประกอบด้วย วงกลมย่อยแทนวงกลมนั้น ให้รายละเอียดสุดท้าย

2.6.3 ทำไมจึงต้องใช้ Data Flow Diagram [3]

เหตุผลที่ต้องใช้ DFD เป็นผลภาพของระบบก็เพราะ

2.6.3.1 DFD ช่วยให้นักวิเคราะห์ระบบสามารถ

- สรุปข้อมูลเกี่ยวกับระบบ
- เข้าใจถึงปัญหาสำคัญของระบบและระบุส่วนของการทำงานที่ซ้ำซ้อน
- เข้าใจถึงความสัมพันธ์ ระหว่างส่วนต่างๆ ของระบบ และการประกอบกัน เป็น

ระบบ

- พัฒนาระบบได้อย่างมีประสิทธิภาพ

2.6.3.2 DFD เป็นเอกสารร่วมที่ช่วยให้นักวิเคราะห์ระบบและผู้ใช้สามารถเข้าใจระบบและ
ตรวจสอบความถูกต้องได้ทั้งสองฝ่าย

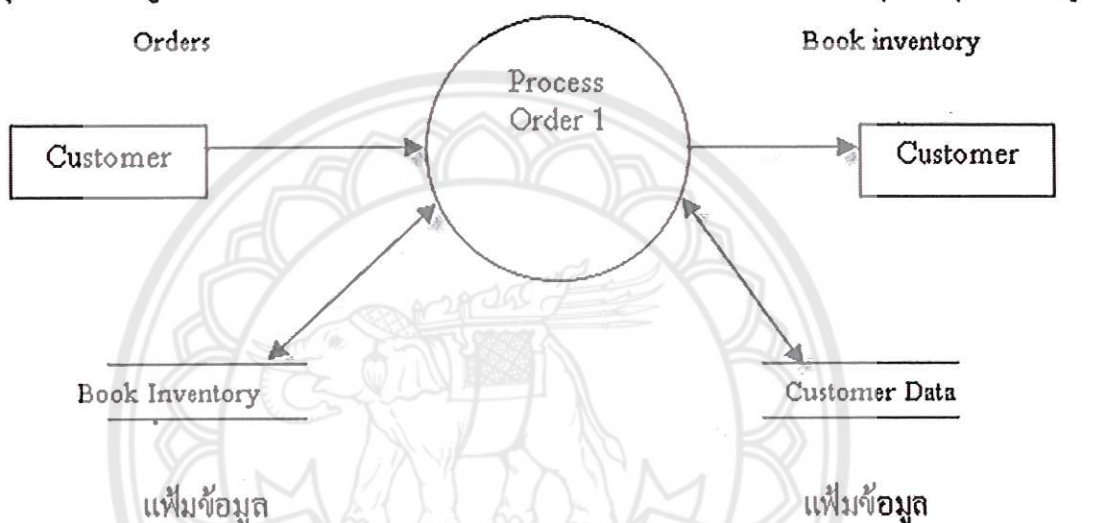
2.6.3.3 ในการตรวจสอบ เรื่องเวลาที่ใช้ใน แต่ละขบวนการนั้น นักวิเคราะห์สามารถใช้ DFD เป็นเครื่องมือที่ช่วยให้ทราบถึงของเขตการพัฒนาแบบ ของระบบว่ามีทางที่จะเป็นไปได้ อย่างไร

2.6.4 ตัวอย่างการใช้ Data Flow Diagram [3]

การเปลี่ยนแปลงข้อมูล

จุดเริ่มต้นข้อมูล

จุดสิ้นสุดของข้อมูล



รูปที่ 2.8 การปฏิบัติงานของบริษัท หนังสือไทย [3]

เป็นแผนภาพแสดง การสั่งซื้อหนังสือจากร้านหนังสือไทย ลูกค้าในช่องสี่เหลี่ยมซ้ายมือสั่งสินค้ามายังวงกลม สิ่งที่จะต้องทำก็คือ ตัด สต็อกของหนังสือที่ถูกส่งออกไปพร้อมกับใบเก็บเงินไป ยังลูกค้าด้านขวามือ โดยรูปแบบของสี่เหลี่ยมจัตุรัสของลูกค้านั้น ทั้งด้านซ้าย และขวามีลักษณะเหมือนกัน จะหมายถึงสิ่งเดียวกัน

2.7 Entity-Relationship Model [1,4]

การออกแบบฐานข้อมูลใช้งาน ในระบบสารสนเทศใดๆ ต้องอาศัยแบบจำลองข้อมูล เพื่อนำเสนอรายละเอียดต่างๆ ที่เกี่ยวข้องกับฐานข้อมูลที่ออกแบบ แบบจำลองข้อมูลที่นิยมใช้ ได้แก่ Entity-Relationship Model

2.7.1 Semantic Model

แบบจำลองของข้อมูลในยุคแรกๆ มีข้อจำกัด ในการนำเสนอรายละเอียดที่เกี่ยวข้องกับข้อมูลในฐานข้อมูล กล่าวคือ จะมีการนำเสนอเฉพาะรายละเอียดทางด้านโครงสร้างของฐานข้อมูล เช่น คุณสมบัตินี้ Atomicity ของข้อมูล กฎระเบียบต่างๆที่ใช้สำหรับควบคุมความถูกต้องของข้อมูล

ๆ เป็นต้น แต่ยังไม่มีการนำเสนอทางด้านความหมาย (Semantic) ของข้อมูลภายในฐานข้อมูลนั้นๆ เช่น จำนวน และน้ำหนักของสินค้าว่ามีความสัมพันธ์กันอย่างไร หรือ Domain ข้อมูลต่างๆ สามารถมีค่าเป็นอะไรได้บ้าง หรือข้อมูลที่ทำหน้าที่เป็น Candidate Key หรือ Foreign Key ใดๆ เป็นต้น ดังนั้น จึงมีการคิดค้นแบบจำลองข้อมูลแบบใหม่ที่เรียกว่า “Semantic Model”

2.7.2 คำศัพท์ที่ใช้ใน Semantic Model

Semantic Model ได้มีการนิยามคำขึ้นแทนข้อมูลในความหมายต่างๆเรียกว่า Concept ดังนี้

2.7.2.1 Entity

Entity เปรียบเหมือนกับคำนามที่สามารถระบุได้ในความเป็นจริง อาจเป็นสิ่งที่จับต้องได้ นายไมเคิล หนังสือชื่อ “ระบบฐานข้อมูล” หรือเป็นสิ่งที่อยู่ในรูปนามธรรมที่ไม่สามารถจับต้องได้ เช่น จำนวนวันลาพักร้อนของพนักงาน ซึ่งเมื่อแต่ละ Entity มารวมกันภายใต้คุณลักษณะหนึ่งๆ เหมือนกันแล้ว Entity เหล่านั้น จะถูกเรียกว่า “Entity Set” ดังตัวอย่างดังนี้

ตารางที่ 2.11 ตัวอย่าง Entity [1,4]

Entity ของนาย ก.

นาย ก.	ไทย
--------	-----

Entity ของนาย ข.

นาย ข.	ไทย
--------	-----

Entity Set “ประชากรสัญชาติไทย”

นาย ก.	ไทย
นาย ข.	ไทย

2.7.2.2 Property

Property หรือ Attribute คือ ข้อมูลที่แสดงลักษณะและคุณสมบัติของ Entity เช่น Property ของ Entity set ที่ชื่อ “รถยนต์” ที่ประกอบด้วย หมายเลขทะเบียนรถยนต์ หมายถึง เลขตัวถัง หมายเลขเครื่องยนต์ ยี่ห้อ รุ่น สี ดังนี้

ตารางที่ 2.12 ตัวอย่าง Property [1,4]

Property

E N T I T Y	ทะเบียน	หมายเลข ตัวถัง	หมายเลข เครื่องยนต์	ยี่ห้อ	รุ่น	สี
	4ข2100-	2AZ0012523	EE859222	TOYOTA	CORONA	ฟ้า
	1ศ1700-	1WE9944560	AS985112	HONDA	CIVIC	ม่วง
	8ช2545-	6AA4456455	EZ123444	NISSAN	SUNNY	แดง

2.7.2.3 Identity

แต่ละ Entity ภายใต้ Entity Set เดียวกัน ถึงแม้ว่าจะต้องมี Property ที่เหมือนกันแต่อย่างไรก็ตาม จะต้องต้องมี Property ใน property หนึ่งซึ่งเป็นเอกลักษณ์เฉพาะของ Entity นั้นเช่นกัน Property “หมายเลขประชาชน” ของแต่ละ Entity ใน Entity Set “ประชาชน” ซึ่งจะไม่มีหมายเลขใดที่ซ้ำกัน ดังตารางข้างล่าง

ตารางที่ 2.13 ตัวอย่าง Identity [1,4]

Identity

หมายเลขบัตรประชาชน	ชื่อ	นามสกุล	เพศ	สัญชาติ	วันเดือนปีเกิด
123456789	แดง	สด	ชาย	ไทย	12/12/22
987654321	ดำ	ดี	หญิง	ไทย	11/11/22

สำหรับ Property ที่สามารถนำมากำหนดเป็นเอกลักษณ์เฉพาะให้กับแต่ละ Entity นี้จะเรียกว่า “Identity”

2.7.2.4 Relationship

ได้แก่ Entity Set ที่สร้างขึ้นจาก 2 Entity Set เดิมหรือมากกว่า เพื่อใช้แสดงความสัมพันธ์ระหว่างแต่ละ Entity ใน Entity Set เดิมเหล่านั้นมาเชื่อมโยงข้อมูลกันภายใต้ค่าของ Property ที่เหมือนกัน ซึ่งการสร้างความสัมพันธ์ในลักษณะนี้ property ของ Relationship จะเกิดจากการนำเอา

Property ของแต่ละ Entity Set มารวมกันเช่น Relationship ที่ชื่อ “สังกัดคณะ” ซึ่งเกิดจากการที่ Entity Set “นักศึกษา” และ “คณะ” มี Property “รหัสคณะ” ที่เหมือนกันดังรูป

ตารางที่ 2.14 ตัวอย่าง RelationShip [1,4]

Entity Set “นักศึกษา”

รหัสนักศึกษา	ชื่อ-สกุล	เพศ	รหัสคณะ
380012	เอก โท	ชาย	02
381202	ตรี จัตวา	หญิง	01
380052	เอก ตรี	ชาย	03

Entity Set “คณะ”

รหัสคณะ	คณะ
01	วิศวกรรมศาสตร์
02	บริหารธุรกิจ

Relationship “สังกัดคณะ”

รหัสนักศึกษา	ชื่อ-สกุล	เพศ	รหัสคณะ	คณะ
380012	เอก โท	ชาย	02	บริหารธุรกิจ
381202	ตรี จัตวา	หญิง	01	วิศวกรรมศาสตร์
380052	เอก ตรี	ชาย	03	วิทยาศาสตร์

2.7.3 Entity-Relationship Model

Semantic Model ที่นิยมใช้มากที่สุด ได้แก่ Entity-Relationship Model หรือที่เรียกกันสั้นๆ ว่า “E-R Model” E-R Model นับเป็นแบบจำลองที่ครอบคลุมนิยามต่างๆ ที่กำหนดไว้ใน Semantic Model เนื่องจากมีรูปแบบที่ใช้กับทุกๆ แนวความ คือที่กำหนดไว้ Semantic Model ซึ่งได้แก่ Entity, Property, Relation และ Subtype สำหรับแผนภาพที่สร้างขึ้นโดยรูปภาพต่างๆ ภายใน E-R Model เพื่อแสดงความเป็นจริงต่างๆ ของข้อมูล จะเรียกว่า Entity-Relationship Diagram หรือที่นิยมเรียกกันสั้น ๆ ว่าแผนภาพ E-R (E-R Diagram)

2.7.3.1 Entity

ได้แก่ Entity Set ต่างๆ ที่นิยามไว้ใน Semantic Model แต่ใน E-R Model จะเรียก Entity Set ว่า Entity แทนสำหรับ Entity ใน E-R Model จะแบ่งออกเป็น 3 ประเภทดังนี้

1. **Regular Entity** หรือบางครั้งเรียกว่า Strong Entity ได้แก่ Entity ส่วนใหญ่ที่ปรากฏอยู่ในรูปฐานข้อมูล ซึ่งสมาชิกภายใน Entity ประเภทนี้ สามารถมีคุณสมบัติ Identity ได้ด้วยตัวมันเอง เช่น Entity “EMPLOYEE” ซึ่งสมาชิกภายใน Entity นี้ ได้แก่ ข้อมูลของพนักงานแต่ละคนในบริษัท ซึ่งประกอบด้วย Property ต่าง ๆ ดังนี้

ตารางที่ 2.15 ตัวอย่าง Regular Entity [1,4]

Entity “EMPLOYEE”

EmpID	NAME	SEX	SALARY
0001	สมชาย นิลกัตต์	M	8500
0002	สมถวิล กลั่นเจริญ	F	9000
0003	เจริญ ก้าวหน้า	M	12000

จากตัวอย่างข้อมูลของ Entity “EMPLOYEE” จะสังเกตเห็นว่า สมาชิก ซึ่งได้แก่ ข้อมูลในแต่ละแถวของตาราง สามารถมีคุณสมบัติ Identity ได้โดยอาศัยค่าของ Property “EmpID” เนื่องจากพนักงานแต่ละคน จะมีหมายเลขประจำตัวพนักงานที่ไม่ซ้ำกัน สำหรับรูปภาพที่ใช้แทน Entity ประเภทนี้ ได้แก่ รูปสี่เหลี่ยมผืนผ้า โดยมีชื่อของ Entity นั้นอยู่ภายใน ดังรูป

EMPLOYEE

รูปที่ 2.9 Regular Entity [1,4]

2. **Weak Entity** เป็น Entity ที่มีลักษณะตรงข้ามกับ Regular Entity กล่าวคือ สมาชิกของ Entity ประเภทนี้จะสามารถมีสมบัติ Identity ได้ จะต้องอาศัย Property ใด Property หนึ่งของ Regular Entity มาประกอบกับ Property ของตัวมันเอง เช่น Entity “TIME_IN_OUT” ซึ่งสมาชิกของ Entity นี้ได้แก่ เวลาเข้าออกของพนักงานแต่ละคนในแต่ละวัน ซึ่งประกอบไปด้วย Property ต่างๆ ดังนี้

i 50 ๗8๐74

ตารางที่ 2.16 ตัวอย่าง Weak Entity [1,4]

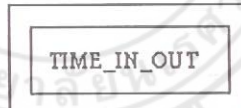
4900133

Entity "TIME_IN_OUT"

EmpID	Date	Time_In	Time_Out
0001	15/9/41	7.30	17.13
0002	15/9/41	8.00	18.00
0003	15/9/41	7.45	17.49
0001	16/9/41	8.00	18.00
0002	16/9/41	7.45	17.49
0003	16/9/41	7.30	17.13

s/s.
๗7247.
2548.
C.2

จากตัวอย่างข้อมูลของ Entity "TIME_IN_OUT" จะสังเกตเห็นว่า แต่ละสมาชิกของ Entity "TIME_IN_OUT" จะสามารถมีคุณสมบัติ Identity ได้ จะต้องอาศัยค่าของ Property "EmpID" ซึ่งเป็น Property ของ Entity "TIME_IN_OUT" มาประกอบกับ Property "Date" ซึ่งเป็น Property ของตนเอง จึงจะทำให้สมาชิกของแต่ละ Entity "TIME_IN_OUT" มีคุณสมบัติ Identity สำหรับรูปภาพที่ใช้แทน Entity ประเภทนี้ ได้แก่ รูปสี่เหลี่ยมผืนผ้าสองรูปซ้อนกัน โดยมีชื่อของ Entity นั้นอยู่ ดังรูป



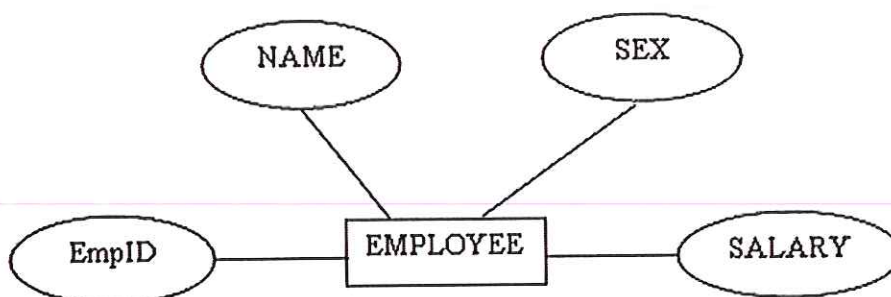
รูปที่ 2.10 Weak Entity [1,4]

2.7.3.2 Property

ได้แก่ Property ต่างๆ ของ Entity หรือ Relationship ที่นิยามไว้ใน Semantic Model เช่น Property "EmpID", "NAME", "SEX" และ "SALARY" ของ Entity "EMPLOYEE" เป็นต้น สำหรับ Property ใน E-R Model จะสามารถแบ่งย่อยได้ดังนี้

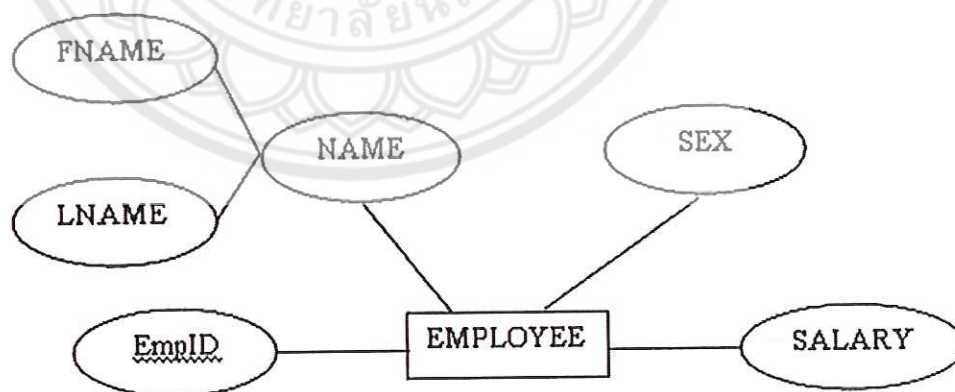
1) **Simple Property** ได้แก่ Property ที่ค่าภายใน Property นั้นไม่สามารถแบ่งย่อยได้อีก ซึ่งได้แก่ Property ของ Entity โดยทั่วไป เช่น Property "SEX" ของ Entity "EMPLOYEE" ที่ใช้เก็บเพศของพนักงานแต่ละคน หรือ Property "SALARY" ของ Entity "EMPLOYEE" ที่ใช้เก็บเงินเดือนของพนักงานแต่ละคน ซึ่งค่าทั้ง 2 Property นี้ไม่สามารถแบ่งออกเป็นค่าย่อยได้อีกสำหรับรูปภาพที่ใช้แทน Property ประเภทนี้ ได้แก่ วงรีที่มีเส้นเชื่อมต่อไปยัง Entity ที่เป็นเจ้าของ Property

นั้น โดยมีชื่อของ Property นั้นอยู่ภายใน เช่น Property ต่างๆ ของ Entity “EMPLOYEE” ดังรูปที่ 2.11



รูปที่ 2.11 Weak Entity [1,4]

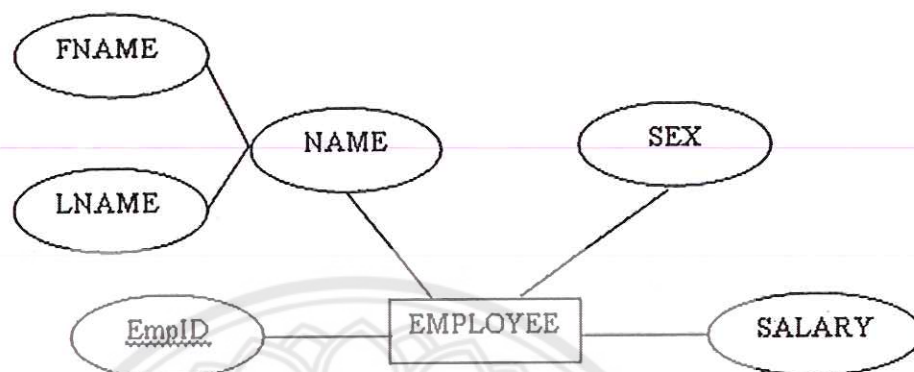
2) Composite Property เป็น Property ที่มีลักษณะตรงข้ามกับ Simple Property กล่าวคือ จะเป็น Property ที่ค่าภายใน Property นั้น ยังสามารถแยกเป็น Property ย่อยได้อีก เช่น Property “NAME” ของ Entity “EMPLOYEE” ที่สามารถแบ่งย่อยออกได้เป็น Property “FNAME” ที่ใช้เก็บชื่อและ Property “SNAME” ที่ใช้เก็บนามสกุล เป็นต้น สำหรับรูปภาพที่ใช้แทน Property ประเภทนี้จะใช้วงรีเดียวกับ Simple Property แต่จะเป็นวงรีที่เชื่อมกับวงรีของ Simple Property ที่เป็นเจ้าของ Composite Property นั้น เช่น Property “FNAME” และ “SNAME” ของ Entity “EMPLOYEE” ดังรูปที่ 2.12



รูปที่ 2.12 Composite Property [1,4]

3) Key เป็น Property หรือกลุ่มของ Property ที่มีค่าในแต่ละสมาชิกของ Entity ไม่ซ้ำกัน ซึ่งถูกนำมาใช้กำหนดคุณสมบัติ Identity ให้กับ Entity เช่น Property “EmpID” ของ Entity “EMPLOYEE” ซึ่งใช้แทนรหัสประจำตัวพนักงาน สำหรับรูปภาพที่ใช้แทน Key ของ Entity จะใช้รูปวงรีเช่นเดียวกับ Property แต่จะมีเส้นขีดอยู่ให้ Property ที่เป็น Key ดังรูป

4) **Single-valued Property** เป็น Property ที่มีค่าของข้อมูลภายใต้ค่าของ Property ใด Property เพียงค่าเดียว เช่น Property “SALARY” ที่ใช้เก็บเงินเดือนของพนักงานซึ่งพนักงานแต่ละคนสามารถมีเงินเดือน ได้เพียงค่าเดียว ดังตัวอย่างข้อมูลต่อไปนี้



รูปที่ 2.13 Composite Property [1,4]

ตารางที่ 2.17 ตัวอย่าง Single-value Property [1,4]

Entity “EMPLOYEE”

EmpID	NAME	SEX	SALARY
0001	สมชาย นิลกถัด	M	8500
0002	สมถวิล กลั่นเจริญ	F	9000
0003	เจริญ ก้าวหน้า	M	12000
0004	ชุตินา สกุลดี	F	10000
0005	นิวัติ เหล่าสุวรรณ	M	25000

สำหรับรูปภาพที่ใช้แทน Property ประเภทนี้ จะใช้รูปภาพเช่นเดียวกับ Simple Property

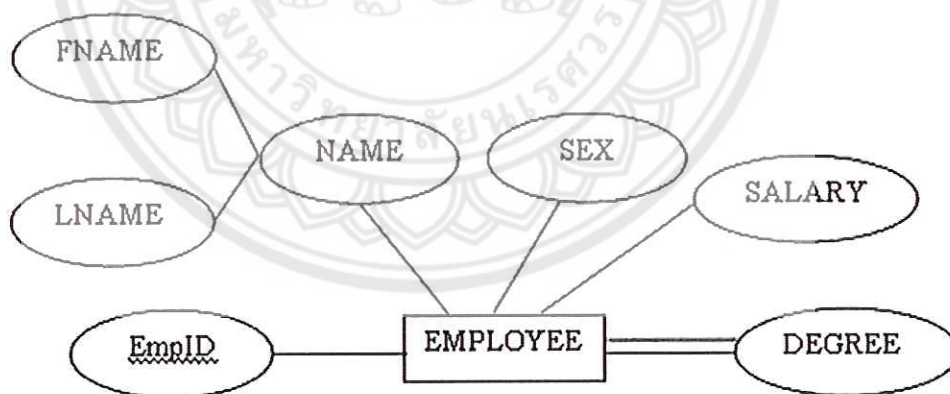
5) **Multi-valued Property** เป็น Property ที่มีลักษณะตรงข้ามกับ Property แบบ Single-valued กล่าวคือ เป็น Property ที่มีค่าของข้อมูลได้หลายค่าภายใต้ค่าของ Property ใด Property หนึ่งเช่น Property “DEGREE” ที่ใช้ระบุระดับการศึกษาของพนักงานแต่ละคน ซึ่งพนักงานแต่ละคน จะมีระดับการศึกษาได้หลายระดับ ดังตัวอย่างข้อมูลต่อไปนี้

ตารางที่ 2.18 ตัวอย่าง Multi – valued Property [1,4]

Entity “EMPLOYEE”

EmpID	NAME	SEX	SALARY	DEGREE	SCHOOL/UNIVERSITY
0001	สมชาย นิลกลัด	M	8500	0	ร.ร ขอนแก่นวิทยา
0002	สมถวิล กลั่นเจริญ	F	9000	0	ร.ร อ่างทองวิทยา
0003	เจริญ ก้าวหน้า	M	12000	0	ร.ร วัดสุทธิวราราม
				1	ม .กรุงเทพ
0004	ชุติมา สกุศลดี	F	10000	0	ร.ร นรีเวชวิทยา

จากตัวอย่างข้อมูลจะสังเกตเห็นว่า Property “DEGREE” ภายใต้พนักงานแต่ละคนสามารถมีค่าข้อมูลได้มากกว่า 1 ค่า เช่น Property “DEGREE” ของพนักงานที่ชื่อ “เจริญ ก้าวหน้า” จะมีอยู่ด้วยกัน 2 ค่า คือ 0 และ 1 เนื่องจากพนักงานคนนี้ จบการศึกษาระดับปริญญาตรี จึงมีข้อมูลของทั้งระดับมัธยมศึกษา และปริญญาตรี หรือ Property “DEGREE” ของพนักงานที่ชื่อ “นิวัติ เหล่าสุพรรณ” จะมีอยู่ด้วยกัน 3 ค่า คือ 1 ,0 และ 2 เนื่องจากพนักงานคนนี้จบการศึกษาระดับปริญญาโท เป็นต้น Entity หรือ Relation จะใช้เส้น 2 เส้นแทน ดังรูปที่ 2.14



รูปที่ 2.14 Multi-valued Property [1,4]

2.7.3.3 Relationship

ได้แก่ Relationship ที่นิยามไว้ใน Semantic Model สำหรับรูปภาพที่ใช้แทน Relationship ใน E – R Model จะได้แก่ รูปสี่เหลี่ยมข้าวหลามตัด ที่มีชื่อของ Relationship นั้นอยู่ภายใน สำหรับรูปภาพของ Relationship นี้ไม่สามารถปรากฏอยู่เดี่ยวๆ ได้ แต่จะต้องปรากฏอยู่คู่กับ Entity เสมอ Relationship โดยทั่วไป จะกำหนดขึ้นจาก Entity ที่มี Property ร่วมกันเช่น Entity “EMPLOYEE” และ “Department” ซึ่งสามารถแสดงด้วยตัวอย่างข้อมูลได้ดังนี้

ตารางที่ 2.19 ตัวอย่าง Relationship [1,4]

Entity "EMPLOYEE"

EmpID	NAME	SEX	SALARY	DEP_ID
0001	สมชาย นิลกลัด	M	8500	01
0002	สมถวิล กลั่นเจริญ	F	9000	03
0003	เจริญ ก้าวหน้า	M	12000	01
0004	ชุตินา สกุลดี	F	10000	02

Entity "Department"

DEP_ID	DEP_NAME
01	ธุรการ
02	บุคคล
03	การเงิน

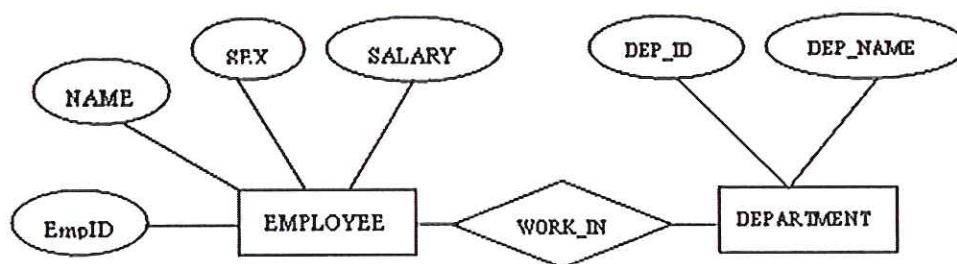
จากตัวอย่างข้อมูล จะสังเกตเห็นว่า ทั้ง 2 Entity มี Property ร่วมกันคือ Property "DEP_ID" นี้ได้ ซึ่งในที่นี้ได้แก่ Relationship เพื่อแสดงความสัมพันธ์ ซึ่งแสดงถึงฝ่ายที่พนักงานแต่ละคนสังกัด ดังนี้

ตาราง 2.19 (ต่อ) ตัวอย่าง Relationship [1,4]

Relation "WORK_IN"

EmpID	NAME	SEX	SALARY	DEP_ID	DEP_NAME
0001	สมชาย นิลกลัด	M	8500	01	ธุรการ
0002	สมถวิล กลั่นเจริญ	F	9000	03	การเงิน
0003	เจริญ ก้าวหน้า	M	12000	01	ธุรการ
0004	ชุตินา สกุลดี	F	10000	02	บุคคล
0005	นิวัติ เหล่าสุวรรณ	M	25000	02	บุคคล

และสามารถแสดงด้วยรูปภาพได้ดังนี้



รูปที่ 2.15 Relationship [1,4]

2.7.3.4 Cardinality Ratio

สมาชิกใน Entity ที่เกี่ยวข้องกับ Relation จะถูกเรียกว่า Participant ซึ่งจำนวนของ Participant นี้จะถูกเรียกว่า Degree ของ Relationship นั้น และจะถูกนำไปใช้กำหนดประเภทของ Relationship ที่เรียกว่า “Cardinality Ratio”

1. One – to – one Relationship เป็น Relationship ที่แต่ละ Participant ของ Entity หนึ่งจะมีความสัมพันธ์กับอีก participant ของอีก Entity หนึ่งเพียง Participant เดียว ดังตัวอย่างข้อมูลต่อไปนี้

ตารางที่ 2.20 ตัวอย่าง Relationship แบบ One – to – one [1,4]

Entity “CUSTOMER”

NAME	ADDRESS	ACCT_NO
แพง พลเมืองดี	111 บางพลัด กทม.	111111111
จิราพร สมตน	222 บางซื่อ กทม.	222222222
สุภาพร อุดมศิลป์	333 ปทุมวัน กทม.	333333333
กิตติ มั่นคง	444 บางบอน กทม.	444444444

Entity “ACCOUNT”

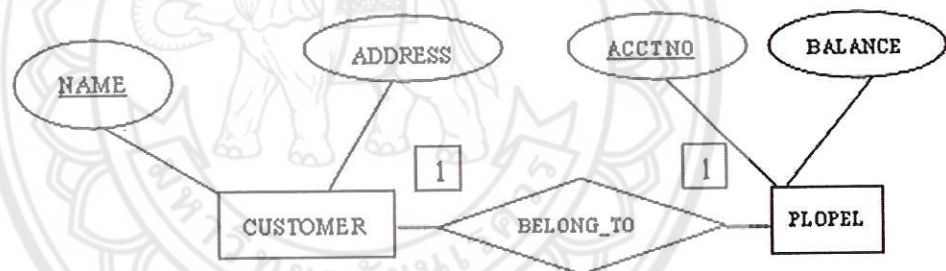
ACCT_NO	BALANCE
111111111	54000
222222222	12000
333333333	14000
444444444	100000

Relationship “BELONG_TO”

NAME	ADDRESS	ACCT_NO	ACCT_NO	BALANCE
แพง พลเมืองดี	111 บางพลัด กทม.	111111111	111111111	54000
จิราพร สมตน	222บางซื่อ กทม.	222222222	222222222	12000
สุภาพร อุดมศิลป์	333ปทุมวัน กทม.	333333333	333333333	14000
กิตติ มั่นคง	444บางบอน กทม.	444444444	444444444	100000

จากตัวอย่างข้อมูลของ Relationship “BELONG_TO” จะสังเกตเห็นว่าลูกค้าซึ่งได้แก่แต่ละรายการใน Entity “CUSTOMER” จะมีความสัมพันธ์กับรายการใน Entity “ACCOUNT” ได้เพียงรายการเดียว และในมูมกลับกัน แต่ละรายการใน Entity “ACCOUNT”

สำหรับรูปภาพที่ใช้แทนความสัมพันธ์นี้ จะใช้รูปภาพรายการเดียวกับ Relationship โดยทั่วไป ดังนั้น รูปภาพของความสัมพันธ์นี้ จึงมีลักษณะดังรูปที่ 2.16



รูปที่ 2.16 Relationship แบบ One – to – one [1,4]

2. One – to – many Relationship เป็น Relationship ที่แต่ละ Participant ของ Entity หนึ่ง มีความสัมพันธ์กับอีก Participant ของอีก Entity หนึ่งมากกว่า 1 Participant เช่น กรณีลูกค้าสามารถมีเงินฝากได้มากกว่า 1 บัญชีและแต่ละบัญชีเงินฝากจะต้องมีเจ้าของบัญชีเพียงคนเดียว ดังตัวอย่างข้อมูลต่อไปนี้

ตารางที่ 2.21 ตัวอย่าง Relationship แบบ One – to – many [1,4]

Entity “CUSTOMER”

NAME	ADDRESS	ACCT_NO
แพง พลเมืองดี	111 บางพลัด กทม.	111111111
แพง พลเมืองดี	111 บางพลัด กทม.	111111112
จิราพร สมตน	222บางซื่อ กทม.	222222222

Entity "ACCOUNT"

ACCT_NO	BALANCE
111111111	54000
222222222	12000
333333333	14000
444444444	100000
111111112	4000

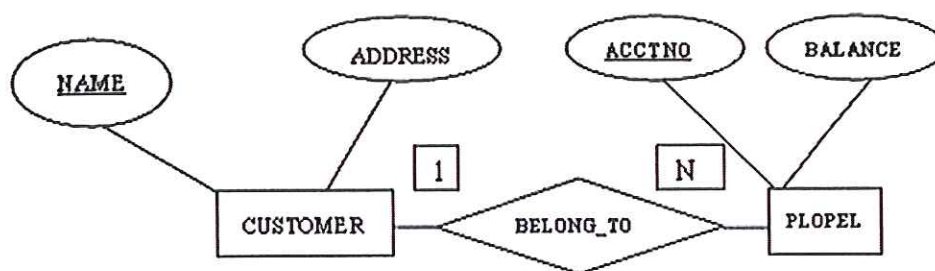
ตารางที่ 2.2 1 (ต่อ) ตัวอย่าง Relationship แบบ One – to – many [1,4]

Relationship "BELONG_TO"

NAME	ADDRESS	ACCT_NO	BALANCE
แพง พลเมืองดี	111 บางพลัด กทม.	111111111	54000
		111111112	4000
จิราพร สมตน	222บางซื่อ กทม.	222222222	12000
สุภาพร อุดมศิลป์	333ปทุมวัน กทม.	333333333	14000
กิตติ มั่นคง	444บางบอน กทม.	444444444	100000

จากตัวอย่างข้อมูลของ Relationship "BELONG_TO" จะสังเกตเห็นว่า ลูกค้าที่ชื่อ "แพง พลเมืองดี" เป็นเจ้าของบัญชีเงินฝาก 2 บัญชี คือ บัญชีเงินฝากเลขที่ "111111111" และ "111111112" แต่ในมุมมองกลับกันแต่ละบัญชีจะมีเจ้าของได้เพียงคนเดียว

สำหรับสัญลักษณ์ที่ใช้กับ Relationship ประเภทนี้ ได้แก่ ตัวเลข 1 และอักษร M โดยตัวเลข 1 จะถูกกำหนดไว้ทางด้านของ Entity ที่มีจำนวน Participant ที่เกี่ยวข้องกับ Relationship เพียง Participant เดียว ส่วนอักษร M จะถูกกำหนดไว้ทางด้านของ Entity ที่มีจำนวน Participant ที่เกี่ยวข้องกับ Relationship มากกว่า 1 Participant ซึ่งจากรูป ด้าน Entity "CUSTOMER" เป็น Entity ที่มีจำนวน Participant ที่เกี่ยวข้องกับ Relationship เพียง Participant เดียว ส่วนด้าน Entity "ACCOUNT" จะเป็น Entity ที่มีจำนวน Participant ที่เกี่ยวข้องกับ Relationship มากกว่า 1 Participant ดังนั้นจึงปรากฏตัวเลข 1 ไว้ทางด้าน Entity "CUSTOMER" และตัวอักษร M ทางด้าน Entity "ACCOUNT" ดังรูปที่ 2.17



รูปที่ 2.17 Relationship แบบ One – to – many [1,4]

3. Many – to – many Relationship เป็น Relationship ที่ Participant มากกว่า 1 Participant ของ Entity หนึ่ง มีความสัมพันธ์กับอีก Participant ของอีก Entity หนึ่งมากกว่า 1 Participant เช่น กรณีลูกค้าสามารถมีเงินฝากมากกว่า 1 บัญชี และแต่ละบัญชีเงินฝากสามารถมีเจ้าของบัญชีได้มากกว่า 1 คน ดังตัวอย่างข้อมูลต่อไปนี้

ตารางที่ 2.22 ตัวอย่าง Relationship แบบ Many – to – many [1,4]

Entity “CUSTOMER”

NAME	ADDRESS	ACCT_NO
แพง พลเมืองดี	111 บางพลัด กทม.	111111111
แพง พลเมืองดี	111 บางพลัด กทม.	111111112
จิราพร สมตน	222 บางซื่อ กทม.	222222222
สุภาพร อุดมศิลป์	333 ปทุมวัน กทม.	333333333
เอกสิทธิ์ ทับทอง	444 บางบอน กทม.	111111112

Entity “ACCOUNT”

ACCT_NO	BALANCE
111111111	54000
222222222	12000
333333333	14000
444444444	100000

Relationship "BELONG_TO"

NAME	ADDRESS	ACCT_NO	BALANCE
แพง พลเมืองดี	111 บางพลัด กทม.	111111111	54000
		111111112	4000
จิราพร สมตน	222บางซื่อ กทม.	222222222	12000
สุภาพร อุดมศิลป์	333ปทุมวัน กทม.	333333333	14000
เอกสิทธิ์ ทับทอง	444บางบอน กทม.	111111112	4000

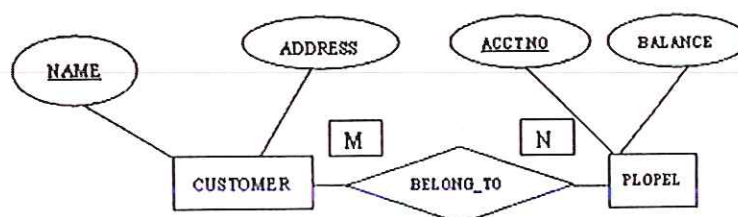
และในมุมมองกลับกัน เห็นว่า บัญชีเลขที่ "11111112" มีเจ้าของบัญชี 2 คน คือ "แพง พลเมืองดี" และ "เอกสิทธิ์ ทับทอง" ซึ่งแสดงด้วย Relationship "BELONG_TO" ได้เช่นเดียวกัน ดังนี้

ตารางที่ 2.22 (ต่อ) ตัวอย่าง Relationship แบบ Many – to –many [1,4]

Relationship "BELONG_TO"

ACCT_NO	NAME	ADDRESS	BALANCE
111111112	แพง พลเมืองดี	111 บางพลัด กทม.	4000
	เอกสิทธิ์ ทับทอง	444บางบอน กทม.	4000
111111111	แพง พลเมืองดี	111 บางพลัด กทม.	54000
222222222	จิราพร สมตน	222บางซื่อ กทม.	12000
333333333	สุภาพร อุดมศิลป์	333ปทุมวัน กทม.	14000
444444444	กิตติ มั่นคง	444บางบอน กทม.	100000

ความสัมพันธ์ระหว่างลูกค้าและบัญชีเงินฝาก จึงเป็น Many-to-many Relationship สำหรับสัญลักษณ์ที่ใช้กับ Relationship ประเภทนี้ได้แก่ ตัวอักษร M กำหนดไว้ทั้ง 2 ด้านของ Entity ดังรูปที่ 2.18

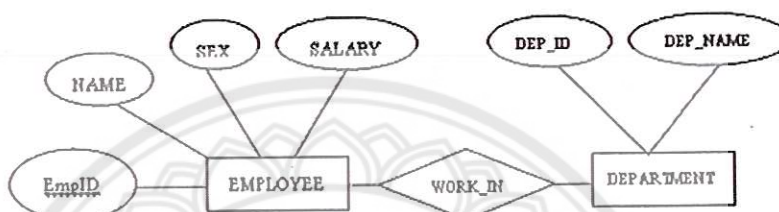


รูปที่ 2.18 Relationship แบบ Many – to –many [1,4]

2.7.3.5 ประเภทของ Relationship

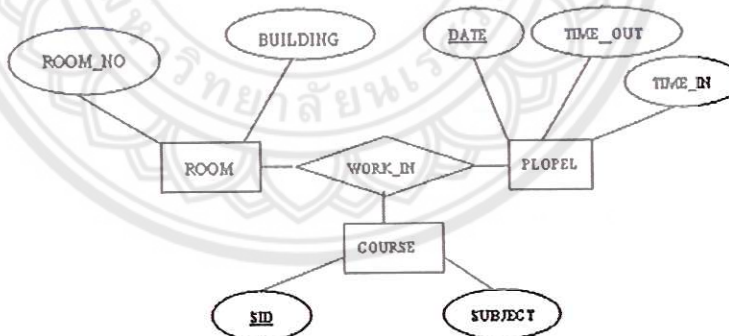
นอกเหนือจากการใช้จำนวนของ Participant ในการจัดประเภทของ Relationship แล้วยังสามารถใช้จำนวนของ Entity ที่มีความสัมพันธ์กับแต่ละ Relationship มากำหนดประเภทของ Relationship ได้ดังนี้

1. **Binary Relationship** เป็น Relationship ที่พบมากที่สุดในแผนภาพ E_R โดยเป็น Relationship ที่เกิดขึ้นระหว่าง 2 Entity ใดๆ เช่น Relationship “WORK_IN” ดังรูปที่ 2.19



รูปที่ 2.19 Relationship แบบ Binary Relationship [1,4]

2. **N-ary Relationship** เป็น Relationship เกิดขึ้นระหว่าง Entity มากกว่า 2 Entity ขึ้นไป เช่น Relationship “SCHEDULE” ซึ่งใช้แสดงตารางเรียนวิชาต่างๆ ดังรูปที่ 2.20



รูปที่ 2.20 Relationship แบบ Binary Relationship [1,4]

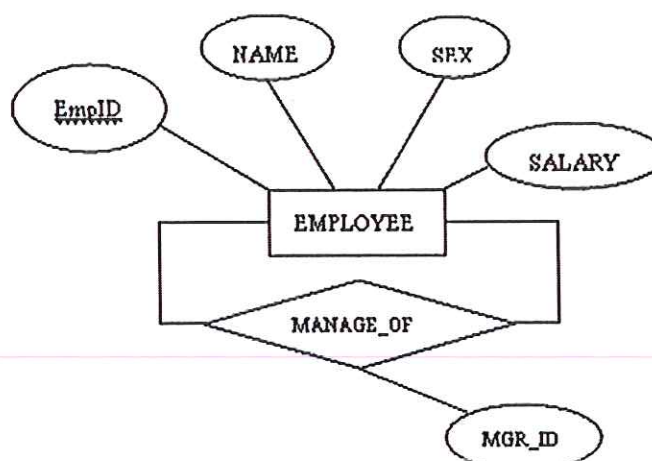
ตารางที่ 2.23 Relationship แบบ N-ary Relationship [1,4]

Entity "ROOM"		Entity "TIME"			Entity "COURSE"	
ROOM_NO	BUILDING	TID	DATE	TIME	SID	SUBJECT
KMB101	KMB	T1	จ.อ	8.00-11.00	EN101	อังกฤษ 1
KLB201	KLB	T2	จ.อ	13.00-16.00	MA111	คณิตศาสตร์ 1
LTB201	LTB	T3	พ.พฤ	8.00-11.00	PH111	ฟิสิกส์ 1
		T4	พ.พฤ	13.00-16.00	CH111	เคมี 1
		T5	ศ.ศ	8.00-11.00	CH112	เคมี 2
		T6	ศ.ศ	13.00-16.00	EN102	อังกฤษ 2

Entity "SCHEDULE"		
SID	TID	ROOM_NO
EN101	T1	KMB101
MA111	T2	KMB101
PH111	T3	LTB201
CH111	T2	KLB201
CH112	T3	KLB201
EN102	T1	LTB201

จากตัวอย่างข้อมูลจะสังเกตเห็นว่า Relationship "SCHEDULE" กำหนดขึ้นจาก Property "ROOM_NO" ของ Entity "ROOM" และ Property "DATE" ของ Entity "TIME" และ Property "SID" ของ Entity "COURSE" มารวมกันดังนั้น Relationship "SCHEDULE" นี้จึงมีความสัมพันธ์กับ 3 Entity ดังกล่าว

3. Recursive Relationship เป็น Relationship ที่เกิดขึ้นกับ Entity เดียว ในกรณีนี้ที่ Property ของ Entity นั้นสามารถสร้างความสัมพันธ์กับอีก Property หนึ่งภายใน Entity เดียวกัน เช่น Relationship "MANAGE_OF" ซึ่งใช้แสดงชื่อหัวหน้างานแต่ละคน ดังรูปที่ 2.21



รูปที่ 2.21 Relationship แบบ Recursive Relationship [1,4]

ซึ่งสามารถแสดงด้วยตัวอย่างข้อมูลได้ดังนี้

ตารางที่ 2.24 Relationship แบบ Recursive Relationship [1,4]

Entity "EMPLOYEE"

EmpID	NAME	SEX	SALARY	MGR_ID
0001	สมชาย นิลกลัด	M	8500	00003
0002	สมถวิล กลั่นเจริญ	F	9000	00003
0003	เจริญ ก้าวหน้า	M	12000	-
0004	ชุตินา สกุลดี	F	10000	00003
0005	นิวัติ เหล่าสุวรรณ	M	25000	00003

Relationship "MANAGE_OF"

EmpID	NAME	SEX	SALARY	MGR_ID	MGR_NAME
0001	สมชาย นิลกลัด	M	8500	00003	เจริญ ก้าวหน้า

2.7.3.6 คุณสมบัติของแผนภาพ E-R ที่ดี

เนื่องจากแผนภาพ E-R ถูกใช้เป็นที่ระบอเครื่องมือในการนำเสนอความเป็นจริงเกี่ยวกับข้อมูล ดังนั้นจึงควรที่จะต้องมีคุณสมบัติดังนี้

1. Expressiveness แผนภาพ E-R ที่ดี จะต้องสามารถอธิบายโครงสร้างของข้อมูลได้เป็นอย่างดีและครบถ้วน
2. Simplicity แผนภาพ E-R ที่ดี จะต้องมึรูปแบบที่ง่ายต่อความเข้าใจ

3. Minimality แผนภาพ E-R ที่ดีจะต้องมีความชัดเจน และไม่สามารถตีความเป็นอื่น
4. Formality แผนภาพ E-R ที่ดี จะต้องไม่ซ้ำซ้อน และมีรูปแบบที่เป็นมาตรฐาน

2.7.3.7 การค้นหา Entity

สิ่งที่จำเป็นสำหรับการวาดแผนภาพ E - R ได้แก่ การกำหนด Entity, Property, Relationship และ Cardinality ที่ใช้ข้อมูลต่างๆ ภายในฐานข้อมูลที่ ต้องการออกแบบนั้น ซึ่งมีขั้นตอนที่ค่อนข้างซับซ้อนแต่อย่างไรก็ตามผู้ออกแบบสามารถที่จะกำหนด Entity ขึ้นภายในระบบได้อย่างคร่าวๆ โดยวิธีการค้นหา Entity ด้วยวิธีของ Data Perspective ซึ่งกำหนดไว้ว่า Entity จะถูกกำหนดขึ้นจากสิ่งต่างๆ ที่ประกฏอยู่ในความต้องการของผู้ใช้ดังต่อไปนี้

1. ขั้นตอนการทำงาน เอกสาร และรายงานต่างๆ ที่ปรากฏอยู่ในระบบ
2. อุปกรณ์ต่างๆ ที่ระบบใช้การติดต่อ
3. ระบบงานที่เกี่ยวข้องในการส่งผ่านข้อมูลระหว่างระบบกัน
4. เหตุการณ์ที่เกิดขึ้นในระบบและระบบจะต้องมีการบันทึกไว้ เป็นหลักฐาน
5. บุคคลหรือสิ่งที่เกี่ยวข้องกับระบบงาน
6. สถานที่ที่เกี่ยวข้องกับระบบงาน
7. หน่วยงานต่างๆ ในองค์กรที่เกี่ยวข้องกับระบบงาน

2.8 การทำงาน Normalization [1, 2]

การออกแบบฐานข้อมูลด้วย E - R Model มีจุดมุ่งหมายเพื่อนำเสนอข้อเท็จจริงต่างๆ ที่เกี่ยวข้องกับข้อมูล โดยไม่ได้คำนึงถึงว่าฐานข้อมูลที่ออกมาจะมีปัญหาทางด้านความซับซ้อนของข้อมูลความถูกต้องของข้อมูล ความผิดพลาดในการเพิ่ม ลบ และแก้ไข หรือไม่ ดังนั้น จึงต้องมีวิธีการตรวจสอบ และแก้ไขปัญหาต่างๆ เหล่านี้ วิธีดังกล่าว “การทำ Normalization”

2.8.1 Normalization [1, 2]

เป็นวิธีการที่ใช้ในการตรวจสอบ และแก้ไขปัญหาทางด้านความซับซ้อนของข้อมูล โดยดำเนินการให้ข้อมูลในแต่ละ Relation อยู่ในรูปที่เป็นหน่วยเล็กที่สุด ไม่สามารถแตกแยกออกเป็นหน่วยย่อยๆ ได้อีก โดยยังคงความสัมพันธ์ระหว่างข้อมูลใน Relation ต่างๆ ไว้ตามหลักการที่กำหนดไว้ใน Relation Model

การทำ Normalization นี้เป็นการดำเนินงานอย่างเป็นระดับที่กำหนดไว้ด้วยกันเป็นขั้นตอนตามปัญหาที่เกิดขึ้นในขั้นตอน ซึ่งแต่ละขั้นตอนจะมีชื่อตาม โครงสร้างข้อมูลที่กำหนดไว้ดังนี้

1. ขั้นตอนการทำ First Normal Form (1NF)
2. ขั้นตอนการทำ Second Normal Form (2NF)
3. ขั้นตอนการทำ Thrid Normal Form (3NF)
4. ขั้นตอนการทำ Boyce - codd Normal Form (BCNF)

5. ขั้นตอนการทำ Fourth Normal Form (4NF)

ในแต่ละขั้นตอนของการทำ Normalization จะมีการระบุรูปแบบของโครงสร้างของข้อมูล ที่ควรจะเป็นเรียกว่า Normal Form ไว้ ซึ่งโครงสร้างเหล่านี้สามารถแก้ไขปัญหาที่เกิดขึ้น ในโครงสร้างที่ระบุนี้จะสามารถแก้ไขปัญหาก็จะเกิดขึ้น ในโครงสร้างข้อมูลของขั้นตอนก่อนหน้าได้ หรือกล่าวอีกนัยหนึ่ง การทำ Normalization ในขั้นตอน จะต้องอาศัยผลที่ได้ จากการทำ Normalization ในขั้นตอนก่อนหน้ามาปรับปรุง เพื่อให้มีโครงสร้างเป็นไปตามโครงสร้างที่กำหนดไว้ในขั้นตอนนี้ๆ แต่อย่างไรก็ตาม ในการทำ Normalization ไม่จำเป็นต้องเริ่มจากขั้นตอนการทำ First Normal Form เสมอไป กล่าวคือ การทำ First Normal Form และสิ้นสุดในขั้นตอนการทำ Fourth Normal Form เสมอไป

2.8.2 First Normal Form (1NF) [1, 2]

เป็นขั้นตอนสำหรับปรับโครงสร้างของข้อมูลของ Relation เพื่อให้ทุก Attribute ของ Relation มีคุณสมบัติ Atomicity กล่าวคือ โครงสร้างข้อมูลของ Relation ในแบบ 1 NF นี้จะต้องประกอบด้วย Attribute ที่ไม่อยู่ในรูป Repeating Group เช่น ตัวอย่างข้อมูลต่อไปนี้

ตารางที่ 2.25 ตัวอย่าง First Normal Form [1, 2]

ORDER					
CUST_NO	CUST_NAME	CITY	ZONE_SALE	ORDER_CONTENT	
					ORDER_QTY
C001	นารี เกิดสว่าง	อยุธยา	001	P001	24
				P003	30
				P004	50
C002	สลักจิต สว่างภพ	ศรีสะเกษ	002	P001	29
				P002	40
				P004	30

จากตารางจะสังเกตเห็นว่า (Attribute "CUST_NO") 1คนสามารถมีรายการสินค้าที่สั่งซื้อ (Attribute "CUST_NAME") ได้มากกว่า 1 รายการ ดังนั้น จึงกล่าวได้ว่า (Attribute "CUST_NO") นี้มีความสัมพันธ์กับ Attribute "ORDER_CONTENT" ในแบบ Repeating Group ส่งผลให้ Relation นี้มีโครงสร้างที่ไม่สอดคล้องกับ 1 NF ดังนั้นจึงต้องการ Normalization โดยการแปลง Attribute ที่อยู่ในรูป Repeating Group ให้มีคุณสมบัติ Atomicity พร้อมกับการกำหนดให้ Attribute ดังกล่าวเป็น Relation Key ของ Relation ดังนั้น จากตัวอย่างข้างต้นจึงถูกแปลงให้อยู่ในรูปดังนี้

ตารางที่ 2.25 (ต่อ) ตัวอย่าง First Normal Form [1, 2]

ORDER

CUST_NO	CUST_NAME	CITY	ZONE_SALE	PRODUCT_ID	ORDER_QTY
C001	นารี เกิดสว่าง	อยุธยา	001	P001	24
C001	นารี เกิดสว่าง	อยุธยา	001	P003	30
C001	นารี เกิดสว่าง	อยุธยา	004	P004	50
C002	สลักจิต สว่างภพ	ศรีสะเกษ	001	P001	29

จะสังเกตเห็นว่า แต่ละค่าที่เป็น Repeating Group ของ Attribute “PRODUCT_ID” และ “ORDER_QTY” จะถูกแยกออกมา Tuple ใหม่ พร้อมกำหนดให้ Attribute “PRODUCT_ID” ทำหน้าที่ Relation Key เป็นร่วมกับ Attribute “CUST_NO”

อย่างไรก็ตาม Relation ที่อยู่ในรูป 1 NF ถึงแม้จะทำให้ทุก Attribute มีคุณสมบัติ Atomicity แต่กลับเกิดปัญหาความซ้ำซ้อนกันของข้อมูล (Redundancy) ขึ้นใน Attribute “CUST_NO”, “CUST_NAME”, “CITY” และ “ZONE_SALE” และก่อให้เกิดปัญหาทางด้าน Anomaly ตามมา ดังนี้

1. Insert Anomaly

เมื่อพิจารณาจากตัวอย่าง จะสังเกตเห็นว่าการเพิ่มข้อมูลลูกค้า จะทำได้ก็ต่อเมื่อลูกค้านั้นมีรายการสั่งซื้อสินค้าแล้วเท่านั้น หมายถึงว่า Relation นี้จะไม่สามารถจัดเก็บข้อมูลของลูกค้าที่ยังไม่มีการสั่งซื้อสินค้าได้ เช่น เมื่อต้องการเพิ่มข้อมูลลูกค้ารหัส “C006” โดยที่ยังไม่มีการกำหนดการสั่งซื้อสินค้าใน Attribute “PRODUCT_ID” จะไม่สามารถกระทำได้ เนื่องจากการ Attribute “PRODUCT_ID” นี้ถูกกำหนดให้เป็น Relation Key จึงไม่สอดคล้องตามกฎของ Entity Integrity Rule ในส่วนที่ว่า Attribute หรือกลุ่ม Attribute ที่เป็น Relation Key จะมีค่าเป็น Null ไม่ได้

2. Delete Anomaly

เมื่อพิจารณาจากตัวอย่าง จะสังเกตเห็นว่า การลบข้อมูลรายการสั่งซื้อ (Attribute “PRODUCT_ID” และ “ORDER_QTY”) บาง Tuple ใน Relation นี้จะทำให้ข้อมูลลูกค้าบางคนสูญหายไป เช่น เมื่อลบข้อมูลรายการสั่งซื้อสินค้ารหัส “P005” ของลูกค้ารหัส “C003” จะทำให้ข้อมูลลูกค้ารหัส “C003” ถูกลบตามไปด้วย

3. Update Anomaly

เมื่อพิจารณาจากตัวอย่าง จะสังเกตเห็นว่า การปรับปรุงข้อมูลใน Tuple ที่มีค่าของข้อมูลซ้ำซ้อนกันไม่ครบถ้วน อาจก่อให้เกิดความขัดแย้งของข้อมูลที่ซ้ำซ้อนกันนั้นได้ เช่น การเปลี่ยนชื่อลูกค้าจาก “สลักจิต สว่างภพ” เป็น “รินลณี สว่างภพ” ของลูกค้ารหัส “C002” ซึ่งถ้าแก้ไขไม่ครบแล้ว จะทำให้ลูกค้ารหัส “C002” มีชื่อทั้ง “สลักจิต สว่างภพ” และ “รินลณี สว่างภพ”

2.8.3 Second Normal Form (2NF) [1, 2]

ในการทำ Normalization ในขั้นตอน Second Normal Form จำเป็นต้องรู้จักถึง Prime Attribute และ Nonprime Attribute เนื่องจาก Attribute ทั้ง 2 ประเภทนี้ จะมีความสำคัญต่อการทำ Normalization แบบ Second Normal Form

Prime Attribute ได้แก่ ทุก Attribute ที่ทำหน้าที่เป็น Relation Key ของ Relation ส่วน Nonprime Attribute ได้แก่ Attribute ที่ไม่ได้เป็นส่วนหนึ่งของ Relation Key

1. ต้องมีโครงสร้างเป็นไปตามโครงสร้าง 1 NF

2. ทุก Nonprime Attribute จะต้องไม่ขึ้นกับ Relation Key ที่อยู่ในรูป Subset

ตัวอย่างเช่น Relation "ORDER1" ในตัวอย่าง ที่ผ่านมา ซึ่งเป็น Attribute ที่ทำให้ข้อมูลในแต่ละ Tuple มีค่าไม่ซ้ำกัน ดังนั้น Attribute ทั้ง 2 จึงทำหน้าที่เป็น Relation Keys ซึ่งสามารถเขียนได้ด้วย Functional Dependency ได้ดังนี้

$$\text{FD : CUST_NO, PRODUCT_ID} \rightarrow \text{CUST_NAME, CITY, ZONE_SALE, ORDER_O}$$

เมื่อพิจารณาค่าของ Attribute "CUST_NO", "CUST_NAME", "CITY", "ZONE_SALE" จะสังเกตว่า Tuple ประกอบขึ้นมาจาก Attribute เหล่านี้ จะมีข้อมูลที่ซ้ำกันเป็นชุดๆ และมีเพียง Attribute "ORDER_QTY" เท่านั้น ที่มีค่าเปลี่ยนแปลงตามค่าของ Relation Key ดังนั้นจึงสามารถเขียนด้วย Functional Dependency ได้ดังนี้

$$\begin{aligned} \text{d1: CUST_NO, PRODUCT_ID} &\rightarrow \text{ORDER_QTY} \\ \text{d2: CUST_NO} &\rightarrow \text{CUST_NAME, CITY, ZONE_SALE} \end{aligned}$$

ใน d2 จะสังเกตเห็นว่า Attribute "CUST_NAME", "CITY" และ "ZONE_SALE" เป็น Nonprime Attribute ของ Relation ที่ไม่ได้ขึ้นอยู่กับเฉพาะ Relation "ORDER1" จึงไม่มีคุณสมบัติเป็นไปตามคุณสมบัติของ 2 NF จึงต้องแตก Relation "ORDER1" ออกเป็น 2 Relation ตาม d1 และ d2 ดังนี้

ตารางที่ 2.26 ตัวอย่าง Second Normal Form [1, 2]

CustOrder

CUST_NO	PRODUCT_ID	ORDER_QTY
C001	P001	24
C001	P003	30
C001	P004	50
C002	P001	29

Cust

CUST_NO	CUST_NAME	CITY	ZONE_SALE
C001	นารี เกิดสว่าง	อยุธยา	001
C002	สลักจิต สว่างภาพ	ศรีสะเกษ	002
C003	สุทิสรา แจ็กสกุล	เชียงใหม่	004
C004	ฟ้า เพิ่มพร	ศรีสะเกษ	002
C009	ต้นสาย ต้นเจริญ	เชียงใหม่	004

สำหรับโครงสร้างของ Relation “CustOrder” และ “Cust” นี้ จะสังเกตเห็นว่าสามารถ
แก้ปัญหา Anomaly ที่เกิดขึ้นใน Relation “ORDER1” ได้ดังนี้

- Relation “cust” สามารถเพิ่มข้อมูลลูกค้าได้ โดยไม่ต้องมีการสั่งซื้อสินค้า ที่เกิดขึ้น
เนื่องจากข้อมูลการสั่งซื้อสินค้า จะถูกแยกจัดเก็บใน Relation “CustOrder” จึงสามารถแก้ปัญหา
Insert Anomaly ที่เกิดขึ้นได้

- สามารถลบรายการสั่งซื้อสินค้ารหัส “P005” ของลูกค้ารหัส “C003” ใน Relation
“CustOrder” ได้โดยไม่ต้องส่งผลกระทบต่อข้อมูลลูกค้ารหัส “C003” เนื่องจากข้อมูลลูกค้าถูกแยกจัดเก็บใน
Relation “Cust” จึงสามารถแก้ปัญหา Delete Anomaly mujgdbf-7howfh

- เนื่องจากข้อมูลของลูกค้าที่ซ้ำซ้อน จะถูกแยกมาจัดเก็บใน Relation “Cust” ดังนั้น การ
เปลี่ยนแปลงรายละเอียดข้อมูลลูกค้า จึงกระทำกับ Relation “Cust” เพียง Relation เดียว จึง
ก่อให้เกิดปัญหา Yupdate Anomaly

2.8.4 Third Normal Form (3NF) [1, 2]

สำหรับ Relation ที่จะมีโครงสร้างในแบบ 3 NF จะต้องมีคุณสมบัติดังนี้

1. ต้องมีคุณสมบัติของ 2 NF

2. ต้องไม่มี Functional Dependency เกิดขึ้นระหว่าง Nonprime Attribute ด้วยกันเองที่เรียกว่า “Transitive Dependency” จาก Relation “Cust” ในหัวข้อที่ผ่านมา ถึงแม้ว่าจะมีโครงสร้างเป็นไปตามคุณสมบัติของ 2 NF แต่จะสังเกตเห็นว่า ค่าของ Attribute “CITY” และ “ZONE_SALE” ถ้าปรากฏข้อมูลที่มีค่าซ้ำซ้อนกันอยู่เป็นคู่ๆ หรือกล่าวอีกนัยหนึ่งทั้ง 2 Attribute สามารถที่จะระบุค่าระหว่างกันได้ กล่าวคือ เมื่อระบุค่าให้กับ Attribute “ZONE_SALE” จะสามารถทราบถึงชื่อเมืองใน Attribute “CITY” ได้ ซึ่งความสัมพันธ์ในลักษณะนี้ จะเรียกว่า Transitive Dependency ดังนั้น Relation นี้จึงขาดคุณสมบัติของ 3NF และยังก่อให้เกิดปัญหาความผิดพลาดทางด้าน Anomaly ดังนี้

1. Insert Anomaly

ใน Relation “Cust” เมื่อต้องการเพิ่มข้อมูลให้กับ Attribute “CITY” และ “ZONE_SALE” ซึ่งมีความสัมพันธ์ในแบบ Transitive Dependency จะไม่สามารถกระทำได้เนื่องจากข้อมูลใน 2 Attribute นี้ ไม่ใช่ Relation Key ดังนั้น จึงต้องเพิ่มข้อมูลนี้ของลูกค้านำใหม่ให้กับ Attribute “CUST_NO” และ “CUST_NAME” ตามไปด้วย

2. Update Anomaly

ใน Relation “Cust” จะสังเกตเห็นว่า เมื่อมีการแก้ไขข้อมูลใน Attribute “ZONE_SALE” จาก “004” ไปเป็น “005” จะต้องทำการแก้ไขข้อมูลในทุกๆ Tuple ที่มีค่าของ Attribute “ZONE_SALE” เท่ากับ “004” ให้ครบถ้วน เนื่องจากเมื่อทำการแก้ไขข้อมูลไม่ครบถ้วนจะก่อให้เกิดข้อมูลใน Relation ที่มีความขัดแย้งกันได้

3. Delete Anomaly

ใน Relation “Cust” จะสังเกตเห็นว่าถ้ามีการลบข้อมูลของ Tuple ที่จัดเก็บข้อมูลในกลุ่ม Transitive Dependency ที่ปรากฏอยู่เพียงชุดเดียวใน Relation จะส่งผลให้ข้อมูลในกลุ่ม Transitive Dependency นั้นสูญหายไปจาก Relation ได้เช่น เมื่อทำการลบข้อมูลใน Tuple ของลูกค้ารหัส “C001” นอกจากจะทำให้ข้อมูลของลูกค้ารหัส “C001” หายไปแล้ว ยังส่งผลให้ข้อมูลของเขตการขายที่อยู่สูญหายไปด้วย

จากปัญหา Anomaly ที่เกิดขึ้นจาก Transitive Dependency เหล่านี้ จึงต้องทำการแยก Nonprime Attribute ที่ก่อให้เกิด Transitive Dependency ของ Relation “Cust” ออกมาเป็น Relation ใหม่ ซึ่งจากตัวอย่าง ได้แก่ Attribute “CITY” และ “ZONE_SALE” ดังนี้

ตารางที่ 2.27 ตัวอย่าง Third Normal Form [1, 2]

Cust3		
CUST_NO	CUST_NAME	CITY
C001	นารี เกิดสว่าง	อยุธยา
C002	สลักจิต สว่างภพ	ศรีสะเกษ
CUST_NO	CUST_NAME	CITY
C003	สุพิศ แจกสกุล	เชียงใหม่
C004	ฟ้า เพิ่มพร	ศรีสะเกษ

CityZone	
CITY	ZONE_SALE
อยุธยา	001
ศรีสะเกษ	002
เชียงใหม่	004

ซึ่งสามารถเขียนด้วย Functional Dependency ได้ดังนี้

$d1: \text{CUST_NO} \rightarrow \text{CUST_NAME, CITY}$ $d2: \text{CITY} \rightarrow \text{ZONE_SALE}$
--

ข้อสังเกต ในการแยก Nonprime Attribute ที่ก่อให้เกิด Transitive Dependency ออกมาเป็น Relation ใหม่ มีหลักการอยู่ 2 ข้อดังนี้

1. แยก Nonprime Attribute ที่ก่อให้เกิด Transitive Dependency ออกมาเป็น Relation ใหม่
2. กำหนดให้ Nonprime Attribute ที่เป็นตัวระบุค่า ให้เป็น Relation Key ของ Relation ใหม่

2.8.5 Fourth Normal Form (4NF) [1, 2]

สำหรับ Relation ที่จะมีโครงสร้างแบบ 4NF จะต้องมีคุณสมบัติดังนี้

1. ต้องมีคุณสมบัติของ BCNF
2. ต้องไม่ปรากฏความสัมพันธ์ระหว่าง Attribute ในแบบ Multi-value Dependency เช่น ตัวอย่างข้อมูลของ Relation "EMPLOYEE_SKILL" ซึ่งใช้จัดเก็บข้อมูลเกี่ยวกับความสามารถของพนักงาน ด้านการใช้คอมพิวเตอร์ Attribute "COMPUTER_SKILL" แล ทางด้านภาษาต่างประเทศ Attribute "LANGUAGE_SKILL" ดังนี้

ตารางที่ 2.28 ตัวอย่าง Fourth Normal Form [1, 2]

EMPLOYEE_SKILL

EMPLOYEE#	COMPUTER_SKILL	LANGUAGE_SKILL
1267	Word Processing	ฝรั่งเศส
1267	Word Processing	เยอรมัน
1345	Spreadsheets	สเปน
1267	Spreadsheets	ฝรั่งเศส
1267	Spreadsheets	เยอรมัน
1345	COBEL	สเปน
1193	Word Processing	ฝรั่งเศส

จากตัวอย่างข้อมูลจะสังเกตเห็นว่า Relation นี้ มีคุณสมบัติเป็น BCNF แต่ยังไม่เป็น 4 NF เนื่องจากปรากฏโครงสร้างข้อมูลในแบบ Multi-value Dependency กล่าวคือ เมื่อระบุค่าของ Attribute "EMPLOYEE#" ซึ่งทำหน้าที่เป็น Determinant จะสามารถแสดงค่าของ Attribute "COMPUTER_SKILL" และ Attribute "LANGUAGE_SKILL" ที่ทำหน้าที่เป็น Dependency ได้มากกว่า 1 ค่า

ดังนั้น จึงต้องแบ่ง Relation นี้ ออกเป็น Relation ใหม่ตามโครงสร้างข้อมูลแบบ Multi-value Dependency ที่ปรากฏอยู่ใน Relation "COMPUTER_SKILL" และ "LANGUAGE_SKILL" ดังนี้

ตารางที่ 2.28 (ต่อ) ตัวอย่าง Fourth Normal Form [1, 2]

COMPUTER_SKILL

EMPLOYEE#	COMPUTER_SKILL
1267	Word Processing
1345	Spreadsheets
1267	Spreadsheets
1345	COBEL
1193	Word Processing

LANGUAGE_SKILL

EMPLOYEE#	LANGUAGE_SKILL
1267	ฝรั่งเศส
1267	เยอรมัน

2.8.6 สรุป [1, 2]

โครงสร้างของฐานข้อมูลที่ออกแบบขึ้น ควรที่จะนำปรับปรุง โดยใช้วิธีการทำ Normalization เพื่อปรับเปลี่ยนโครงสร้างของ Relation ต่างๆ ที่ได้ออกแบบไว้ให้มีโครงสร้างที่เป็นไปตามคุณสมบัติของ Relation ที่กำหนดไว้ใน Relation Model รวมทั้งมีโครงสร้างที่เหมาะสมต่อการนำไปใช้งานในการทำ Normalization จะมีอยู่ด้วยกัน 5 ขั้นตอน แต่ส่วนใหญ่ในทางปฏิบัติแล้วการทำ Normalization จะกระทำถึงขั้นตอน Boyce-Codd Normal Form (BCNF) เท่านั้นก็เพียงพอแล้ว

2.9 ภาษา PHP [4]

ช่วงแรก PHP คือ Professional Home Page แต่ในปัจจุบัน PHP หมายถึง PHP Hypertext Preprocessor ซึ่งเป็นภาษาสคริปต์แบบหนึ่งที่เรียกว่า Server Side Script ที่ประมวลผลฝั่งเซิร์ฟเวอร์แล้วส่งผลลัพธ์ไปฝั่งไคลเอนต์และเว็บเบราว์เซอร์เช่นเดียวกับ ASP (Active Server Page) ปัจจุบันได้รับความนิยมเป็นอย่างมากในการนำมาช่วยพัฒนางานบนเว็บที่เรียกว่า Web Development หรือ Web Programming เนื่องจากมีจุดเด่นหลายประการ รูปแบบของภาษา PHP มีเค้าโครงมาจากภาษา C และ Perl ที่นำมาปรับปรุงทำให้มีประสิทธิภาพสูงและทำงานได้เร็วขึ้น

2.9.1 ความเป็นมาของ PHP [4]

PHP เกิดขึ้นในปี 1994 โดยโปรแกรมเมอร์ชาวสหรัฐอเมริกาชื่อ Rasmus Lerdorf ได้พัฒนาเครื่องมือที่ใช้สำหรับการสร้างเว็บเพจส่วนตัวของเขา โดยตอนแรกใช้ภาษา Perl แต่เกิดอุปสรรคในเรื่องความเร็ว เขาจึงพัฒนาเครื่องมือใหม่นี้ขึ้นมาโดยใช้ไวยากรณ์ภาษา C และเรียกว่า Personal Home Page ในขณะเดียวกันก็พัฒนาส่วนที่ติดต่อกับฐานข้อมูลที่เรียกว่า Form Interpreter (FI) ทั้งสองส่วนรวมกันเป็น PHP/FI ตรงนี้เองที่เป็นจุดเริ่มต้นของ PHP เนื่องจากเมื่อมีผู้เข้าชมเว็บเพจต่างนิยมชมชอบ จึงติดต่อขอโค้ดเพื่อนำไปพัฒนาต่อ ในลักษณะที่เรียกว่า Open Source ด้วยเหตุนี้ในปี 1997 มีเว็บไซต์มากกว่า 50,000 แห่งที่ใช้ PHP/FI เพื่องานในด้านต่างๆ ทั้งการติดต่อฐานข้อมูล, การแสดงข้อมูลแบบไดนามิก และอื่นๆ อีกมากมาย

เมื่อมีผู้ใช้งานมากขึ้นก็มีคำร้องขอให้พัฒนาประสิทธิภาพของ PHP/FI ให้ยิ่งสูงขึ้น Zeev Suaski กับ Andi Gutmans ชาวอิสราเอล เข้ามาปรับปรุงโค้ดเดิมของ Lerdorf โดยใช้ C++ และมี

ทีมงานเพิ่มเติมอีก 3 คน คือ Stig Bakken, Shane Caraveo และ Jim Winstead โดยนาย Stig Bakken รับผิดชอบเกี่ยวกับความสามารถในการสนับสนุน Oracle, Shane Caraveo ดูแล PHP บน Window9x/NT และ Jim Winstead คอยตรวจสอบข้อบกพร่องต่างๆ อีกครั้ง และได้ชื่อเป็น Professional Home Page สำหรับ PHP3 ที่ออกสู่สายตาโปรแกรมเมอร์เมื่อ มิถุนายน 1998 คือการสนับสนุนหลายแพลตฟอร์มของระบบปฏิบัติการ ทั้ง (Window 95/98/ME/NT และ Linux) และเว็บเซิร์ฟเวอร์ (เช่น IIS, PWS, Apache, OmniHTTPd เป็นต้น) นอกจากนี้ยังสนับสนุน SNMP (Simple Network Management Protocol) และ IMAP (Internet Message Access Protocol)

ปัจจุบัน Zeev และ Andi Gutmans ได้ร่วมกันพัฒนาต่อเป็น PHP4 (พฤศจิกายน 2000 ออกเวอร์ชันล่าสุดคือ 4.02) โดยจะตั้งชื่อให้ ว่า Zend ซึ่งเป่าหมายนั้นก็คือ ประสิทธิภาพ ที่เหนือกว่า ASP โดยที่ Zend (www.zend.com) จะเป็น compile script ซึ่งต่างจากเดิมที่เป็น embed script interpreter ซึ่งทำให้ทำงานเร็วกว่าปัจจุบันซึ่งสามารถดาวน์โหลดได้แล้วที่ www.php.net ในขณะนี้ทีมงานประกอบด้วย

1. Rasmus Lerdorf ชาวสหรัฐอเมริกา
2. Zeev Suraski ชาวอิสราเอล
3. Andi Gutmans ชาวอิสราเอล
4. Shane Caraveo ชาวสหรัฐอเมริกา
5. Stig Bakken ชาวนอร์เวย์
6. Andrey Zmievski ชาวสหรัฐอเมริกา
7. Sascha Schumann ชาวเยอรมัน
8. Thies C.Amtzen ชาวเยอรมัน
9. Jim Winstead ชาวสหรัฐอเมริกา

ปัจจุบัน PHP จะหมายถึง PHP Hypertext Preprocessor ซึ่งบ่งบอกได้ว่าจะมีประสิทธิภาพในระดับโปรเฟสเซอร์เบื้องต้นสำหรับไฮเปอร์เท็กซ์

2.9.2 จุดเด่นของ PHP [4]

ถึงแม้จะรู้จักและนำมาใช้งานได้ไม่มากนัก แต่ PHP กลับได้รับความนิยมในการใช้เป็นเครื่องมือเพื่อพัฒนาเว็บเพจ เนื่องจาก PHP มีจุดเด่นดังนี้

1. Free เนื่องจากสิ่งที่ต้องการสูงสุดของโปรแกรมเมอร์ในการพัฒนาเว็บคือ ของฟรี PHP ได้ตอบสนองโปรแกรมเมอร์เป็นอย่างดี เพราะเครื่องมือที่ใช้เพื่อพัฒนาทุกอย่างสามารถหาได้ฟรีๆ (คุณสามารถเลือกใช้ได้จากแผ่นซีดี “พัฒนา Web Database ด้วย PHP”) ไม่ว่าจะเป็นระบบปฏิบัติการ (Windows, Linux) โปรแกรมเว็บเซิร์ฟเวอร์ (IIS, PWS, Apache, OmniHTTPd) โปรแกรมระบบฐานข้อมูล (MySQL, SQL) และ Server Site Script อย่าง PHP

2. **Speed** เนื่องจาก PHP นำข้อดีของภาษาสคริปต์ที่เคยมีในภาษา C, Perl และ Java รวมกับความเร็วของ CGI นำมาพัฒนาอยู่ใน PHP

3. **Open Source** เนื่องจากการพัฒนาของ PHP ไม่ได้ยึดติดกับบุคคล หรือกลุ่มคนเล็กๆ แต่เปิดโอกาสให้โปรแกรมเมอร์ทั่วไปได้เข้ามาช่วยพัฒนา ทำให้มีคนใช้งานจำนวนมาก และพัฒนาได้เร็วขึ้น

4. **Crossable Platform** เนื่องจาก PHP ใช้ได้กับหลายๆ ระบบปฏิบัติการไม่ว่าบน Windows, Unix, Linux หรืออื่นๆ โดยแทบจะไม่ต้องเปลี่ยนแปลงโค้ดคำสั่งเลย

5. **Database Access** เนื่องจาก PHP สามารถติดต่อกับ ฐานข้อมูล อย่าง dBASE, Access, SQL Server, Oracle, Sybase, Informix, PostgreSQL, MySQL, Empress, FilePro, PostgreSQL ได้ อย่างมีประสิทธิภาพ

6. **Protocol Support** เนื่องจาก PHP สามารถสนับสนุนโปรโตคอลหลายแบบ ทั้ง IMAP, SNMP, NNTP, POP3, HTTP

7. **Library** เนื่องจาก PHP มีไลบรารี สำหรับการติดต่อกับแอปพลิเคชันได้มากมาย

8. **Flexible** ด้วยเหตุที่ PHP มีความยืดหยุ่นตัวสูง ทำให้สามารถนำไปสร้างแอปพลิเคชันได้หลายประเภท

9. **Easy** เนื่องจาก PHP เป็นภาษาสคริปต์ภาษาหนึ่ง ทำให้สามารถแทรกตำแหน่งใดก็ได้ ในแท็กของ HTML

2.9.3 PHP ทำอะไรได้บ้าง [4]

PHP ทำทุกสิ่งที่คุณต้องการรวมทั้งการจัดการเกี่ยวกับกราฟฟิก และไดนามิก HTML ด้วย ตามคู่มือของ PHP ที่กล่าวว่า “The goal of the language is to allow Web developers to write dynamically generated page quickly” นั่นคือเป้าหมายหลักของ PHP โดยเฉพาะเรื่องไดนามิกที่สามารถเขียนได้อย่างรวดเร็ว ต่อไปนี้งานทั่วไปที่ PHP สามารถทำได้ทำตามฟังก์ชันของระบบ ได้แก่ การสร้าง การเปิด การอ่าน และการปิดไฟล์ในระบบ

1. เอ็กซิวคัตคำสั่งของระบบ ได้แก่ การสร้างโฟลเดอร์ และการปรับแต่งสิทธิการใช้งาน
2. จัดการข้อมูลจากฟอร์ม ได้แก่ การจัดเก็บข้อมูลลงไฟล์ ,การส่งข้อมูลผ่านทางอีเมลล์ , ส่งค่าข้อมูลจากการประมวลผลกลับไปยังผู้ใช้
3. การติดต่อฐานข้อมูล ได้แก่ การสร้างอินเทอร์เฟซแบบเว็บเพื่อเพิ่มข้อมูล ,ลบข้อมูล ,การแก้ไขและอัปเดตข้อมูลในฐานข้อมูล
4. เซตคุกกี้และแอ็กเซสตัวแปรคุกกี้
5. ใช้ PHP เพื่อรักษาความปลอดภัยของเว็บเพจ
6. เข้ารหัสข้อมูล

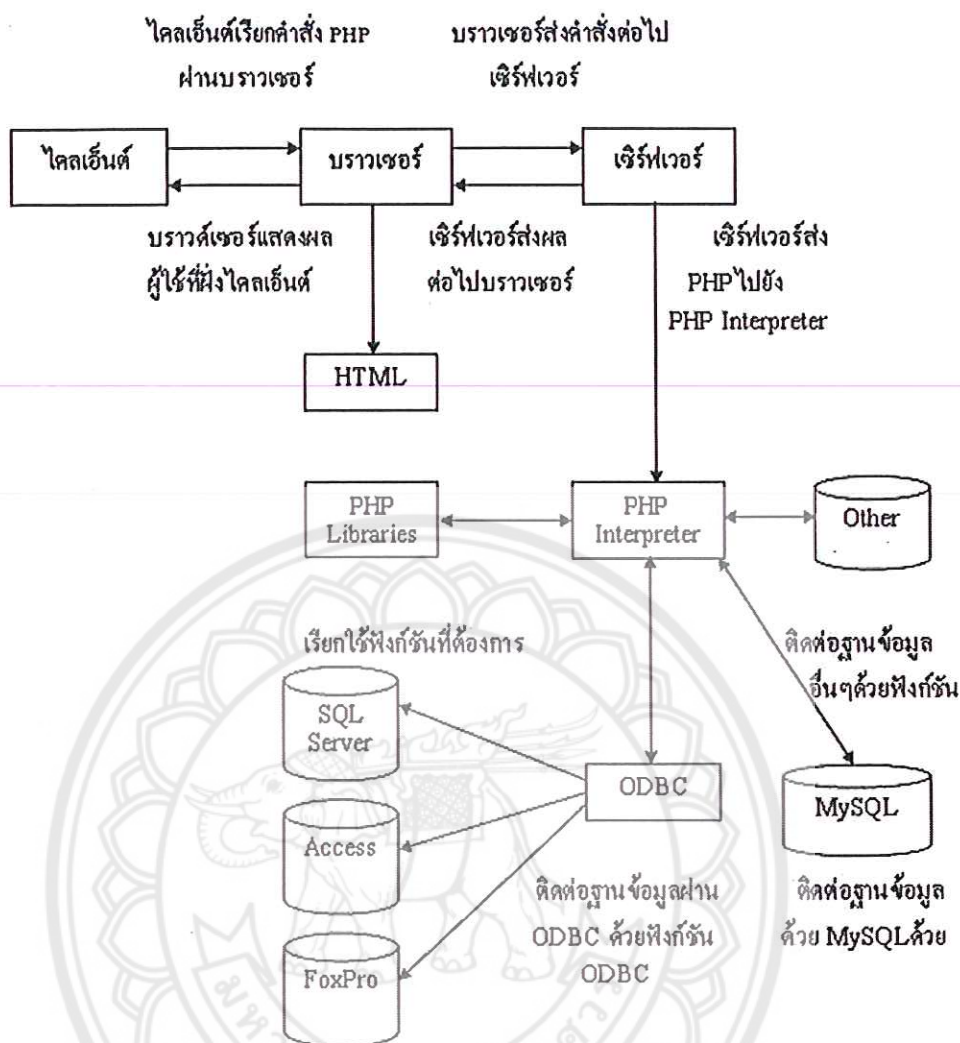
2.9.4 ขอบเขต Open Source [4]

ผู้ใช้งานคงเคยได้ยินคำว่า Open Source สำหรับซอฟต์แวร์หรือโปรแกรมใดก็ตามที่เป็น Open Source แสดงว่าโค้ดนั้นสามารถเปิดเผยต่อสาธารณชน เพื่อร่วมกันพัฒนาหรือปรับแต่ง สามารถศึกษารายละเอียดเพิ่มเติมได้จาก www.opensource.org เนื่องจาก PHP เป็น Open source จึงมีขอบเขตและลักษณะสำคัญดังนี้

1. แจกจ่ายฟรี
2. โปรแกรมนั้น จะต้องมีซอร์สโค้ด และต้องอนุญาตให้แจกจ่าย ซอร์สโค้ดนั้น ได้ เช่นเดียวกับรูปแบบที่คอมไพล์แล้ว
3. ใบอนุญาต (license) ต้องยอมให้แก้ไข ปรับแต่ง และแก้ปัญหาให้งานของคุณได้
4. ต้องมีซอร์สโค้ดของผู้เขียน
5. ผู้ใช้นำไปใช้งานโดยไม่ผิดกฎหมาย
6. ผู้ใช้นำไปปรับแต่งโดยไม่ผิดกฎหมาย
7. มีใบอนุญาตการแจกจ่าย
8. ใบอนุญาตไม่เฉพาะเจาะจงผลิตภัณฑ์
9. ใบอนุญาตต้องไม่ก่อความเดือดร้อนให้ซอฟต์แวร์อื่น

2.9.5 หลักการทำงานของ PHP [4]

เนื่องจาก PHP จะทำงานโดยมีตัวแปรและเอ็กซิควิต์ที่ฝั่งเซิร์ฟเวอร์ อาจจะเรียกการทำงานว่าเป็นเซิร์ฟเวอร์ไซด์ (Server Side) ส่วนการทำงานของบราวเซอร์ของผู้ใช้เรียกว่า ไคลเอนต์ไซด์ (Client Side) โดยการทำงานจะเริ่มต้นที่ผู้ใช้ส่งความต้องการผ่านเว็บเบราว์เซอร์ทาง HTTP (HTTP Request) ซึ่งอาจจะเป็นการกรอกแบบฟอร์มหรือใส่ข้อมูลที่ต้องการข้อมูลเหล่านั้น จะเป็นเอกสาร PHP (เอกสารนี้จะมีส่วนขยายเป็น php หรือ php3 แล้วแต่ผู้ใช้กำหนด เช่น search.php เป็นต้น) เมื่อเอกสาร PHP เข้ามาถึงเว็บเซิร์ฟเวอร์ก็จะถูกส่งไปให้ PHP เพื่อทำหน้าที่แปลคำสั่งแล้วเอ็กซิควิต์คำสั่งนั้น หลังจากนั้น PHP จะสร้างผลลัพธ์ในรูปแบบเอกสาร HTML ส่งกลับไปให้เว็บเซิร์ฟเวอร์เพื่อส่งต่อไปให้เบราว์เซอร์แสดงผลทางฝั่งผู้ใช้ต่อไป (HTTP Response) ซึ่งลักษณะการทำงานแบบนี้จะคล้ายกับการทำงานของ CGI (Common Gateway Interface) หรืออาจจะกล่าวได้ว่า PHP ก็คือโปรแกรม CGI ประเภทหนึ่งก็ได้ซึ่งจะทำงานคล้ายกับ ASP นั่นเอง ดังแสดงไว้ในรูปที่ 2.22



รูปที่ 2.22 แผนผังการทำงานของ PHP [4]

2.10 Macromedia Dreamweaver [5]

Dreamweaver เป็นเครื่องมือในการสร้างเว็บเพจที่มีประสิทธิภาพสูง เป็นผลิตภัณฑ์ของบริษัท Macromedia ซึ่งเป็นบริษัทที่พัฒนา และออกแบบโปรแกรมทางด้านกราฟฟิก รวมถึงเครื่องมือสร้างเว็บเพจต่างๆ ในโครงการนี้เราจะใช้ Dreamweaver เวอร์ชัน MX โดยความสามารถของ Dreamweaver จะคล้ายกับ โปรแกรมประเภท FrontPage, NetObject Fusion, HomeSite

2.10.1 ความสามารถโดยรวมของ Dreamweaver [5]

สนับสนุนการทำงานแบบ WYSIWYG (What You See Is What You Get) หมายความว่า "อะไรก็ตามที่เราทำบนหน้าจอ Dreamweaver ก็จะปรากฏเช่นนั้นบนหน้าจอเว็บเพจ" ซึ่งช่วยให้การปรับปรุงแก้ไขเว็บเพจนั้นทำได้ง่ายไม่จำเป็นต้องมีความรู้ในภาษา HTML มีความสามารถทำการติดต่อกับฐานข้อมูล เพื่อความสะดวกในการเขียนแอปพลิเคชันรองรับมัลติมีเดีย เช่น เสียง กราฟฟิก และอนิเมชัน ที่สร้างโดยโปรแกรม Flash, Shockwave มีเครื่องมือในการสร้างรูปแบบ

หน้าจอบริบท และมีความยืดหยุ่นในการใช้งานสูง สนับสนุนภาษาสคริปต์ต่างๆ ทั้งฝั่งไคลเอ็นต์ และเซิร์ฟเวอร์ เช่น Java, ASP, PHP มีเครื่องมือในการอัปโหลด (Upload) หน้าเว็บเพจไปที่เครื่องเซิร์ฟเวอร์เพื่อทำการเผยแพร่งานที่สร้างในอินเทอร์เน็ต โดยการส่งผ่าน FTP หรือใช้โปรแกรมภายนอก

2.10.2 เครื่องคอมพิวเตอร์ที่เหมาะสมกับการใช้งาน Dreamweaver [5]

แม้ว่า Dreamweaver จะเป็นโปรแกรมที่ไม่ต้องใช้เครื่องที่มีประสิทธิภาพสูงมากนัก แต่ในการใช้งานโดยปกติ เราอาจจำเป็นต้องใช้โปรแกรมอื่นๆ ร่วมในการทำงานด้วย เช่น โปรแกรมตกแต่งภาพ ดังนั้น สเปคเครื่องที่เรานำมาใช้ควรมีขนาดขั้นต่ำดังนี้

1. หน่วยประมวลผลกลาง (CPU) รุ่น Pentium II 300 MHz ขึ้นไป
2. ระบบปฏิบัติการ Window 98/ME หรือ Window NT Version 4.0, Windows 2000, Windows XP หรือ OS 8.1 หรือสูงกว่า สำหรับ Macintosh
3. หน่วยความจำ (RAM) 96 MB และมีเนื้อที่ฮาร์ดดิสก์ อย่างน้อย 275 MB สำหรับบรรจุโปรแกรม Dreamweaver
4. ความละเอียดของจอภาพแสดงผล 800 x 600 pixel
5. CD – ROM

2.11 MySQL [6]

MySQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) หรือที่เรียกง่ายๆ ก็คือ Database Server MySQL ถูกพัฒนามาจากโปรแกรม MySQL โปรแกรมสามารถสนับสนุนการทำงานบนระบบปฏิบัติการ Linux หรือ Windows นอกจากนี้ยังทำงานร่วมกับ Java, C, C++, PHP, ASP, หรือ Perl ได้อีกด้วย ก่อนที่จะใช้ PHP ร่วมกับ MySQL ได้ต้องทำการกำหนดสิทธิให้กับผู้ที่เข้ามาใช้งาน MySQL ก่อน และนอกจากนี้โปรแกรม MySQL ยังเป็นโปรแกรมประเภท Open Source ซึ่งมีลิขสิทธิ์ในแบบ GPL สามารถดาวน์โหลดเซอร์สโค้ดได้จากอินเทอร์เน็ตโดยไม่ต้องเสียค่าใช้จ่าย ทำให้เป็นโปรแกรมด้านฐานข้อมูลที่นิยมใช้ในปัจจุบัน

ระบบการจัดการฐานข้อมูล MySQL เหมาะกับธุรกิจขนาดเล็ก และขนาดกลาง และไม่จำเป็นต้องสิ้นเปลืองงบประมาณจำนวนมาก กับการซื้อระบบจัดการฐานข้อมูล แม้ว่า MySQL จะขาดคุณสมบัติบางอย่าง เช่น Stored Procedures และ Triggers (ปัจจุบันสนับสนุน Transactions) ก็ยังเพียงพอสำหรับระบบงานส่วนใหญ่

2.11.1 ข้อดีของ MySQL [6]

1. ฟรี สามารถใช้งานได้โดยไม่ต้องเสียค่าลิขสิทธิ์
2. เปิดเผยแพร่เซอร์สโค้ด
3. มีความเร็วในการทำงานสูง

4. มีเสถียรภาพสูง
5. ซึ่งจะทำงานได้กับระบบปฏิบัติการ ที่หลากหลาย ไม่ว่าจะเป็น UNIX, Linux, Windows 2000, Windows NT, Windows Me, Windows 9x, Solaris และอื่นๆ อีกมากมาย
6. มีผู้ใช้เป็นจำนวนมาก ทำให้มีการพัฒนาและออกเวอร์ชันใหม่ๆ อย่างสม่ำเสมอ
7. ติดตั้งและใช้งานง่าย มีคู่มือให้ดาวน์โหลดได้ฟรี

2.11.2 ความสามารถของ MySQL [6]

1. ระบบจัดการบัญชีผู้ใช้
2. สิทธิต่างๆ ในการเข้าใช้ฐานข้อมูล
3. ระบบสำรองข้อมูล (Backup)
4. ระบบคืนสภาพข้อมูล (Recovery)
5. ระบบโอนถ่ายข้อมูลไปยังโปรแกรมฐานข้อมูลตัวอื่นๆ
6. จัดเก็บข้อมูลได้หลายชนิดข้อมูล เช่น ข้อความ ตัวเลข รูปภาพ และอื่นๆ
7. ด้วยความสามารถ ของ MySQL ที่มีมากมาย ทำให้มีผู้นั้นใช้งานตัวโปรแกรม

MySQL มีมากขึ้น และในอนาคตคาดว่าจะอาจจะเป็นคู่แข่งของ Microsoft SQL Server หรือ Oracle

2.11.3 รูปแบบและคำสั่งของ MySQL [6]

การกำหนดผู้ใช้ (login)

เมื่อได้ติดตั้ง MySQL ตลอดจนทดสอบแล้วว่าโปรแกรมสามารถทำงานได้ เราก็จะเข้าสู่ MySQL การเข้าสู่ MySQL สามารถทำได้หลายทางทั้งทาง Client โดยการใช้ Telnet หรือ ทาง Consol โดย Login เป็น User

ก่อนอื่นจะต้องทำการกำหนด Mysqldadmin เพื่อความปลอดภัยของระบบ โดย Mysqldadmin เท่านั้นที่สามารถเข้าสู่ MySQL ได้

รูปแบบคำสั่งมีดังนี้

```
#/usr/local/bin/mysqladmin -u root password secret
```

-u ตามด้วยชื่อ user หมายถึงการกำหนดชื่อ user

password ตามด้วย password หมายถึงการกำหนด password ให้กับ user ดังกล่าวข้างต้น

รูปแบบคำสั่ง login เข้า MySQL มีดังนี้

```
#/usr/local/bin/mysql -u root password
```

Enter password: ***** (ให้ใส่ password ที่กำหนดลงไป)

เมื่อ login เข้ามาใน MySQL แล้วจะมีเครื่องหมายพร้อมดังนี้ mysql>

การออกจาก MySQL มีรูปแบบคำสั่งดังนี้ mysql> OUT

การติดต่อไฟล์ Database

MySQL มีคำสั่งให้แสดง file database คือคำสั่ง SHOW DATABASES

รูปแบบคำสั่งการติดต่อ file database

USE database

คำอธิบาย

USE database

คำอธิบาย

USE เป็นคำสั่งที่ใช้ติดต่อ file database

Database เป็นชื่อ file database ที่ต้องการติดต่อ

ตัวอย่าง

```
mysql>USE datatest ;
```

การสร้าง Table ก่อนที่จะสร้าง table จะสร้าง file database ก่อน จากตัวอย่างที่ผ่านมาได้สร้าง file database ชื่อ “datatest” ไว้แล้ว ให้นำ file “datatest” มาใช้อีกครั้งโดย ใช้คำสั่ง USE database

การสร้าง table จะต้อง มี Option เป็นส่วนประกอบ โครงสร้างดังตารางนี้

รูปแบบคำสั่งสร้าง table

```
CREATE TABLE data1 (filed1, filed2, filed3, filedn)
```

Create_definition :

```
col_name type [NOT NULL | NULL] {DEFAULT [default_value]
```

```
[AUTO_INCREMENT][PRIMARY KEY] [reference_definition]
```

```
or PRIMARY KEY (index_col_name,...)
```

```
or KEY [index_name] (index_col_name,...)
```

```
or INDEX [index_name] (index_col_name,...)
```

```
or UNIQUE [INDEX] [index_name] (index_col_name,...)
```

คำอธิบาย

CREATE TABLE เป็นคำสั่งสร้าง table

Data1 เป็นชื่อ table ที่ต้องการสร้าง

File1 เป็นชนิดของ column

ตัวอย่าง

```
mysql> CREATE TABLE phonebook(
```

```
-> name VARCHAR(25)
```

```
-> email VARCHAR(30),
```

```
-> phone INT,
```

```
-> ID INT NOT NULL AUTO_INCREMENT,
```

-> PRIMARY KEY(ID)) ;

ขอดูโครงสร้าง Table

Table ทุก Table จำเป็นต้องมีโครงสร้าง โครงสร้าง Table จะบ่งบอกถึงลักษณะของการ Design และยังช่วยให้การนำเข้าข้อมูลถูกต้องตามที่ Design ไว้ ถ้านำเข้าข้อมูล input ผิดก็ส่งผลกระทบต่อ database

รูปแบบคำสั่งการขอดูโครงสร้าง table

DESC tablename

คำอธิบาย

DESC tablename

DESC เป็นคำสั่งสร้างดูโครงสร้าง table

Tablename เป็นชื่อ table ที่ต้องการดู

Mysql> DESC phonebook ;

การขอดู Table ใน file Database

หลังจากสร้าง table แล้วก็จะดู table ก่อนอื่นจะต้องทำการติดต่อ file database ที่มี table

รูปแบบคำสั่งการขอดู table ใน file database

SHOW TABLES

คำอธิบาย

SHOW TABLES

SHOW เป็นคำสั่งของดู

TABLES เป็น table ที่ต้องการดูใน file database

ตัวอย่าง

mysql> SHOW TABLES;

การแก้ไข Table

เมื่อพบว่าต้องการแก้ table MySQL แต่การแก้ไขภายหลังจำเป็นต้องใช้ความระมัดระวังอย่างมาก เนื่องจากภายใน table นั้นได้มีการ add data ไว้เรียบร้อยแล้ว

รูปแบบคำสั่งการแก้ไข table

ALTER TABLE tableold RENAME tablenew ;

หรือ

ALTER TABLE table AD fiednew DATATYPE ;

คำอธิบาย

ALTER TABLE tableold RENAME tablenew ;

ALTER TABLE เป็นคำสั่งแก้ไข table

tableold เป็นชื่อของ table เดิม

RENAME ให้แก้ไขเปลี่ยนชื่อตาราง

Tablenew เป็นชื่อของ table ใหม่ที่เราต้องการแก้ไข

หรือ

ALTER TABLE table ADDfiednewDATATYPE ;

ALTER TABLE เป็นคำสั่งแก้ไข table

Table เป็นชื่อของ table ที่ต้องการแก้ไข

ADD ให้แก้ไขเปลี่ยนเพิ่ม field เข้าไปใหม่

Fiednew เป็นชื่อของ field ใหม่ที่เราต้องการแก้ไขเพิ่มเติม

DATATYPE เป็นลักษณะของ field ใหม่ที่เราต้องการให้เพิ่มไป

ตัวอย่าง

ALTER TABLE tableold RENAMEtablenew ;

การลบ Table

เมื่อสร้าง table ได้ ก็ต้องมีการลบได้ แต่ก่อนลบนั้น จะต้องมั่นใจก่อนว่า จะลบ table จริงๆ

จะนั้นแล้ว ข้อมูลของบริษัทหรือองค์กร จะหายไปในพื้นที่ ทางที่ดีควร backup ไว้

รูปแบบคำสั่งการลบ table ใน file database

DROP TABLE tablename

คำอธิบาย

DROP TABLE tablename

DROP TABLE เป็นคำสั่งลบ table

Tablename เป็นชื่อ table ที่ต้องการลบ

ตัวอย่าง

DROP TABLE phonebook ; รูปแบบคำสั่งการลบ table

Operator

ตัว Operator สามารถแยกได้ 2 ประเภทดังนี้

1. Relational Operator ดังแสดงไว้ในตารางที่ 2.29
2. Bulletin Operator ดังแสดงไว้ในตารางที่ 2.30

ตารางที่ 2.29 Relational Operator [6]

Operator	ความหมาย
=	เท่ากับ
>	มากกว่า
<	น้อยกว่า
>=	มากกว่า หรือเท่ากับ
<=	น้อยกว่า หรือเท่ากับ
<>	ไม่เท่ากับ

ตารางที่ 2.30 Bulletin Operator [6]

Operator	ความหมาย
AND	และ
OR	หรือ
NOT	เป็นเท็จ

เงื่อนไขการใช้ Bulletin operator

การใช้ Operator จัดการข้อมูลให้ได้ข้อมูลที่ถูกต้องและมีประสิทธิภาพ จำเป็นต้องทำความเข้าใจตรรก ดังแสดงไว้ในตารางที่ 2.31

ตารางที่ 2.31 ตารางตรรก [6]

Operator	เงื่อนไข	ตัวอย่าง	ผลลัพธ์
AND	T AND F	(1=1) AND (2=2)	T (จริง)
	F AND T	(1>1) AND (2=2)	F (เท็จ)
	T AND F	(1=1) AND (2<2)	F (เท็จ)
	F AND F	(1<>1) AND (2<>2)	F (เท็จ)

การเพิ่มข้อมูล

จะนำข้อมูลเข้าสู่ Table โดยมีวิธีการดังนี้

1. สร้าง file database 1 file ชื่อว่า datatest
2. สร้าง table ชื่อว่า phone book
3. เพิ่มข้อมูลตามลำดับ

รูปแบบคำสั่งการเพิ่มข้อมูลลงใน table

```
INSERT INTO tablename (filed1, filed2, filedn) VALUES ('filed1', 'filed2', filedn');
```

คำอธิบาย

```
INSERT INTO tablename ('filed1', 'filed2', filedn);
```

INSERT INTO เป็นคำสั่งเพิ่มข้อมูลเข้าสู่ table

tablename เป็นชื่อ table ที่ต้องการเพิ่มข้อมูล

(filed1, filed2, filedn) เป็นชื่อกลุ่ม field ที่ต้องการเพิ่มข้อมูล

VALUES กลุ่มของข้อมูลที่ต้องการเพิ่ม

('filed1', 'filed2', filedn') เป็นข้อมูลของแต่ละ field ที่ต้องการเพิ่ม

field 1 เป็น field ที่ 1 ของข้อมูลที่ต้องการเพิ่ม (จากที่ได้ออกแบบ Table ไว้)

field 2 เป็น field ที่ 2 ของข้อมูลที่ต้องการเพิ่ม

field n เป็น field ที่ n ของข้อมูลที่ต้องการเพิ่ม

ตัวอย่างเช่น

```
mysql > INSERT INTO phonebook (name, email, phone) VALUES
```

```
mysql > ('บ้านเย็น น้ำอุ่น',
```

```
mysql > 'ccpasskn@hotmail.com'
```

```
mysql > 223102, NULL);
```

การสืบค้นข้อมูล

MySQL มีระบบการสืบค้นข้อมูลสามารถสืบค้น โดยการสร้างเงื่อนไข ให้ตรงตามความต้องการ โดยมีวิธีการดังนี้

1. การสืบค้นข้อมูลโดยไม่มีเงื่อนไข

- ขอข้อมูลจากตารางทั้งหมด

- ขอข้อมูลบางส่วนในตารางทุก Record

รูปแบบคำสั่งสืบค้นข้อมูลโดยขอข้อมูลทั้งหมด table

```
SELECT * FROM tablename ;
```

รูปแบบคำสั่งสืบค้นข้อมูลโดยขอข้อมูลบางส่วน (บาง field) ทั้งหมด table

SELECT fieldname FROM tablename ;

คำอธิบาย

SELECT fieldname FROM tablename ;

SELECT คำสั่งเลือกข้อมูล

Fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table อะไร

Tablename table ที่ต้องการสืบค้น

ตัวอย่าง

mysql > SELECT * FROM phonebook ;

2. การสืบค้นข้อมูลโดยมีเงื่อนไข

การสืบค้นข้อมูลโดยมีเงื่อนไขจำเป็นต้องใช้ Operator เข้าช่วย การใช้ Operator สามารถใช้ได้ ในโอกาสที่แตกต่างกัน สามารถแยกการใช้ได้ดังนี้

การสืบค้นโดยใช้ Operator WHERE

การสืบค้นโดยใช้ Operator LIKE

การสืบค้นโดยใช้ Operator AND

การสืบค้นโดยใช้ Operator OR

การสืบค้นโดยใช้ Operator >

การสืบค้นโดยใช้ Operator >=

การสืบค้นโดยใช้ Operator <

การสืบค้นโดยใช้ Operator <=

การสืบค้นโดยใช้ Operator <>

การสืบค้นโดยใช้ Operator WHERE

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator WHERE

SELECT fieldname FROM tablename WHERE (fieldname = 'namerecord');

คำอธิบาย

SELECT fieldname FROM tablename WHERE (fieldname = 'namerecord');

SELECT คำสั่งเลือกข้อมูล

Fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table อะไร

WHERE ข้อมูลที่ต้องการสืบค้น

(fieldname=namerecord) field และ record ที่ต้องการ

ตัวอย่าง

```
mysql > SELECT * FROM phonebook WHERE name = 'passkorn';
```

การสืบค้นโดยใช้ Operator LIKE

Operator LIKE%

Operator %LIKE%

Operator %LIKE

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator LIKE%

```
SELECT fieldname FROM tablename WHERE (fieldname LIKE 'namerecord%');
```

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator%LIKE SELECT fieldname FROM tablename WHERE (fieldname LIKE% 'namerecord');

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator %LIKE

```
SELECT fieldname FROM tablename WHERE (fieldname LIKE '%namerecord');
```

คำอธิบาย

SELECT คำสั่งเลือกข้อมูล

Fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table อะไร

WHERE ที่ที่ต้องการสืบค้นข้อมูล

(fieldname1 = namerecord1) field และ record ที่ต้องการสืบค้น

OR เงื่อนไขหรือหมายความว่าต้องเป็นจริงเหตุการณ์นี้จึงจะแสดงผล

(fieldname2 = namerecord2) field และ record ที่ต้องการสืบค้น

ตัวอย่าง

```
mysql > SELECT * FROM phonebook WHERE
```

```
mysql > ( name = 'passkorn') OR (phone = '223102');
```

การสืบค้นโดยใช้ Operator >

การสืบค้นโดยใช้ Operator >=

การสืบค้นโดยใช้ Operator <

การสืบค้นโดยใช้ Operator <=

การสืบค้นโดยใช้ Operator <>

ในส่วนนี้ทำตัวอย่างพร้อมกันทีเดียว 5 Operator เนื่องจากใช้หลักการเดียวกันหมด รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator >

```
SELECT fieldname FROM tablename WHERE fieldname > data;
```

รูปแบบคำสั่งสืบค้นข้อมูลโดยใช้ Operator > =

SELECT fieldname FROM tablename WHERE fieldname > = data;

รูปแบบคำสั่งสืบค้นข้อมูล โดยใช้ Operator <

SELECT fieldname FROM tablename WHERE fieldname < data;

รูปแบบคำสั่งสืบค้นข้อมูล โดยใช้ Operator < =

SELECT fieldname FROM tablename WHERE fieldname < = data;

รูปแบบคำสั่งสืบค้นข้อมูล โดยใช้ Operator < >

SELECT fieldname FROM tablename WHERE fieldname < > data;

คำอธิบาย

SELECT fieldname FROM tablename WHERE fieldname > data;

SELECT คำสั่งเลือกข้อมูล

Fieldname field ที่ต้องการ ถ้าต้องการทุก field ให้ใช้เครื่องหมาย "*"

FROM จาก table อะไร

WHERE ที่ที่ต้องการสืบค้นข้อมูล

รูปแบบคำสั่ง (ต่อ)

Fieldname > data field ที่ต้องการให้แสดงผลว่า มากกว่า ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

Fieldname >= data field ที่ต้องการให้แสดงผลว่า มากกว่าหรือเท่ากับ ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

Fieldname < data field ที่ต้องการให้แสดงผลว่า น้อยกว่า ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

Fieldname <= data field ที่ต้องการให้แสดงผลน้อยกว่าหรือเท่ากับ ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

Fieldname <> data field ที่ต้องการให้แสดงผลไม่เท่ากับ ข้อมูลที่กำหนด data คือข้อมูลที่กำหนด

ตัวอย่าง

mysql > SELECT * FROM phonebook WHERE (salary > '20000');

mysql > SELECT * FROM phonebook WHERE (salary >= '20000');

mysql > SELECT * FROM phonebook WHERE (salary < '20000');

mysql > SELECT * FROM phonebook WHERE (salary <= '20000');

mysql > SELECT * FROM phonebook WHERE (salary <> '20000');

การเรียงลำดับ

การจัดการข้อมูลจำเป็นต้องมีการจัดเรียงลำดับข้อมูล เพื่อง่ายแก่การดูแลและตัดสินใจ เพื่อจะได้นำไปใช้ประโยชน์ ได้ตรงตามความต้องการ เช่น ฝ่ายบุคลากรต้องการทราบข้อมูลของพนักงานที่มีเงินเดือนเรียงลำดับจากมากไปน้อย เพื่อนำไปคำนวณหาฐานเงินเดือน การจัดลำดับข้อมูล Sort Data มีรูปแบบคำสั่งมีดังต่อไปนี้

รูปแบบคำสั่งจัดลำดับข้อมูล

```
SELECT fieldname From tablename ORDER BY fieldname ;
```

คำอธิบาย

```
SELECT fieldname From tablename ORDER BY fieldname ;
```

SELECT เป็นคำสั่งขอข้อมูล

Fieldname เป็นคำสั่งเลือกดูข้อมูลใน field (ถ้าต้องการเลือกทุก field ให้ใช้เครื่องหมาย *)

ORDER BY คำสั่งจัดเรียงลำดับข้อมูล

Fieldname เป็นชื่อ field ที่ต้องการเรียง

ตัวอย่าง

```
SELECT * FROM phonebook ORDER BY salary ;
```

การแก้ไขข้อมูล

รูปแบบคำสั่งการแก้ไขข้อมูล

```
UPDATE tablename SET fieldname = 'datanew' WHERE fieldname = 'dataold' ;
```

คำอธิบาย

```
UPDATE tablename SET fieldname = 'datanew' WHERE fieldname = 'dataold' ;
```

UPDATE เป็นคำสั่งแก้ไขข้อมูล

Tablename เป็นชื่อ table ที่ต้องการแก้ไขข้อมูลภายใน table

Fieldname = 'datanew' เป็นชื่อ field และข้อมูลใหม่ที่ต้องการแก้ไขข้อมูล

WHERE คำสั่งค้นหาข้อมูล

Fieldname = 'datanew' เป็นชื่อ field และข้อมูลใหม่ที่ต้องการแก้ไขข้อมูล

WHERE คำสั่งค้นหาข้อมูล

Fieldname = 'dataold' ; เป็นชื่อ field และข้อมูลเก่าที่ต้องถูกแก้ไข

ตัวอย่าง

```
mysql > UPDATE phonebook SET name = 'somjai' WHERE name = 'somsee' ;
```

การลบข้อมูล

จะต้องระวังเป็นพิเศษ เนื่องจาก MySQL จะไม่ทวนคำถามก่อนว่าจะยืนยันการลบหรือไม่ ข้อมูลที่ลบไปแล้วไม่สามารถ Undo กลับมาได้ ถ้าต้องการเรียกคืนจะต้องสร้างขึ้นใหม่

รูปแบบคำสั่งการลบข้อมูล

```
DELETE FROM tablename WHERE (fieldname = 'datadel');
```

คำอธิบาย

```
DELETE FROM tablename WHERE (fieldname = 'datadel');
```

DELETE FROM เป็นคำสั่งลบข้อมูล

Tablename เป็นชื่อ table ที่ต้องการลบข้อมูลภายใน table

WHERE คำสั่งค้นหาข้อมูล

fieldname = 'datadel' เป็นชื่อ field และข้อมูลที่ต้องการลบ

ตัวอย่าง

```
Mysql > DELETE FROM phonebook WHERE (name = 'somsee');
```

รูปแบบคำสั่ง

รูปแบบคำสั่งการสร้าง file database

```
CREATE DATABASE dtatbasename
```

รูปแบบคำสั่ง การติดต่อ file database

```
USE database
```

รูปแบบคำสั่งสร้าง table

```
CREATE TABLE data1 (field1, field2, field3, fieldn)
```

รูปแบบคำสั่งการเพิ่มข้อมูลลงใน table

```
INSERT INTO tablename (field1, field2, fieldn) VALUES ('field1', 'field2', 'field3');
```

รูปแบบคำสั่งสืบค้นข้อมูลโดยขอดูข้อมูลทั้งหมด table

```
SELECT * FROM tablename;
```

รูปแบบคำสั่งสืบค้นข้อมูลโดยขอดูข้อมูลบางส่วน (บาง field) ทั้งหมด table

```
SELECT fieldname FROM tablename;
```

รูปแบบคำสั่งการแก้ไขข้อมูล

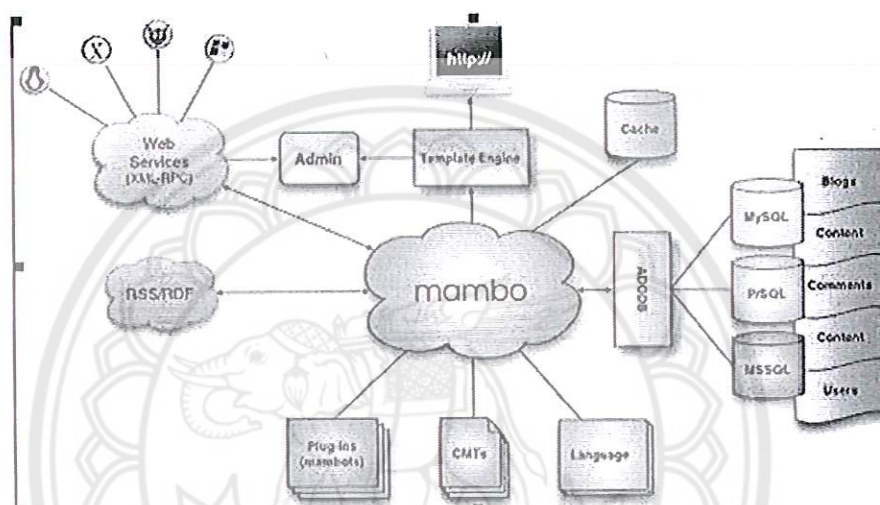
```
UPDATE tablename SET fieldname = 'datanew' WHERE fieldname = 'dataold';
```

รูปแบบคำสั่งการลบข้อมูล

```
DELETE FROM tablename WHERE (fieldname = 'datadel');
```

2.12 Mambo [9]

Mambo คือ ระบบจัดการเว็บไซต์หรือที่เรานิยมเรียกกันว่า Content Management System (CMS) ซึ่งก็คือ ระบบที่ทำงานอยู่เบื้องหลังของเว็บไซต์ที่จะทำให้การจัดการข้อมูล อาทิเช่น การสร้างบทความ, การจัดการบทความ และการเผยแพร่บทความนั้น กลายเป็นเรื่องง่ายดาย และมีประสิทธิภาพ คุณลักษณะที่ดีของระบบ CMS ซึ่งเป้าหมายของการทำ Mambo ก็เพื่อตอบสนองกับความต้องการของผู้ใช้ ตามหลักการของระบบ CMS ที่ดี



รูปที่ 2.23 แผนผังการทำงานของ Mambo [9]

โดยการที่จะให้ได้ระบบที่พร้อมนั้นการพัฒนาก็ต้องมีการพัฒนาแกนหลักของระบบให้มีความสมบูรณ์ที่สุด เพื่อที่จะให้ระบบสามารถที่จะทำงานร่วมกับส่วนประกอบย่อย ที่ถูกพัฒนาขึ้นจากผู้พัฒนาอิสระได้อย่างมีประสิทธิภาพ ซึ่งแนวคิดของการพัฒนาแบบนี้เองที่จะช่วยให้ Mambo นั้นกลายเป็นระบบที่มีประสิทธิภาพสูงและได้รับความนิยมอย่างแพร่หลาย และก็ทำให้การจัดการเนื้อหาบนเว็บไซต์นั้นเป็นไปได้โดยง่ายและสามารถที่จะตอบสนองกับความต้องการของผู้นำไปใช้ในทุกรูปแบบไม่ว่าจะเป็นในองค์กรขนาดใหญ่ ขนาดกลาง หรือขนาดเล็ก หรือแม้กระทั่งเว็บสำหรับครอบครัว

MAMBO คือสคริปต์ก้อนหนึ่งที่เขียนด้วยภาษา PHP โดยให้มีการทำงานร่วมกับฐานข้อมูล MySQL โดยมีจุดมุ่งหมายหลัก เพื่อให้คนที่ไม่มีความรู้เรื่องการเขียนโปรแกรม และฐานข้อมูลทำการสร้างเว็บแบบ Dynamic ได้ โดย MAMBO นั้นจะมีหน้าที่เอาไว้จัดการ Content มากมายหลายประเภทเพื่อให้รองรับการสร้างเว็บได้ทุกรูปแบบตามความต้องการของ Admin ไม่ว่าจะเป็น เว็บข่าว เว็บ portal เว็บขายของ หรือ เว็บอัลบั้มภาพ โดยเราจะเรียกรวมๆ เว็บที่มีระบบดังกล่าวเหล่านี้ว่า CMS หรือชื่อเต็มๆ ว่า Content Management System ซึ่งก็มีอยู่หลายตัวในท้องตลาด เช่น PHPNUKE, POSTNUKE หรือแม้แต่ MAMBO นั้นก็เป็น CMS ที่เพิ่งเกิดมาใหม่และลบข้อจำกัด

ของ CMS ตัวเดิมๆ ในเรื่องการ Design ออกไป โดย MAMBO นั้นเราสามารถที่จะออกแบบหน้าตาได้อย่างที่เราต้องการโดยไม่ยึดติดกับรูปแบบใดรูปแบบหนึ่ง โดยการเพิ่มโมดูลหรือคอมโพเนนต์ก็ทำได้ง่ายกว่า CMS ตัวอื่นๆ เช่นกัน แม้ mambo จะมีข้อดีมากมายนั้นแต่ MAMBO ก็ยังมีข้อด้อยบางประการที่ยังทำให้ไม่ได้รับความนิยมเท่าที่ควร โดยจุดหนึ่งนั้น MAMBO เป็น CMS ที่ออกมาใหม่ ไม่ได้มีการพัฒนามาจากตัวใดเลย จึงทำให้แกนของระบบมีความซับซ้อนและแตกต่างจาก CMS ตัวอื่น จึงทำให้มีผู้ที่พัฒนาโมดูลขึ้นมาใช้และเผยแพร่ให้น้อย และอีกจุดคือเนื่องจากระบบค่อนข้างซับซ้อนจึงทำให้ผู้ใช้อาจจะเกิดความสับสนได้

รายละเอียดที่กล่าวมาในบทที่ 2 จะอธิบายถึงหลักการทำงานและทฤษฎีที่เกี่ยวข้องในการทำงานของโปรแกรม ซึ่งจะนำไปสู่การออกแบบโปรแกรมฐานข้อมูลการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์ในบทที่ 3 ต่อไป



บทที่ 3

การออกแบบโปรแกรมฐานข้อมูล การจัดการงานบัญชีของคณะวิศวกรรมศาสตร์

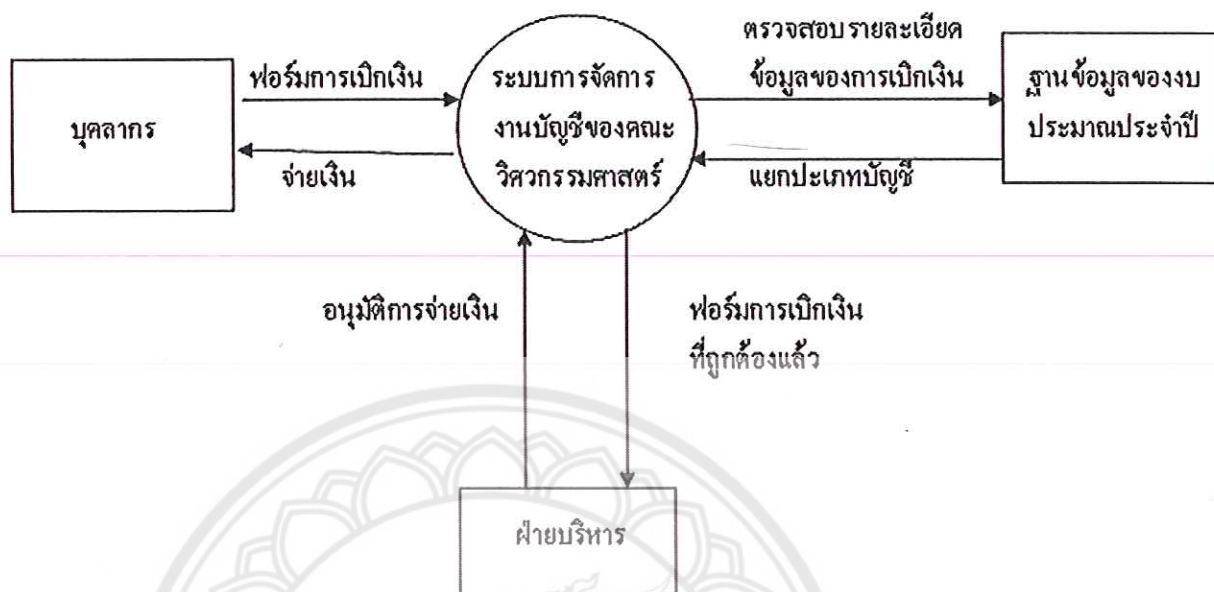
จากการที่เข้าไปรับทราบ และได้เก็บข้อมูลเบื้องต้น แล้วพอที่จะเห็นปัญหาที่เกิดขึ้น เนื่องจากการเก็บข้อมูลอยู่ในรูปแบบเอกสารอาจมีปัญหาการจัดเก็บเอกสารเกิดขึ้น หากเอกสารมีรายละเอียดการเบิกจ่ายเงินงบประมาณเกิดสูญหายจะทำให้มีปัญหาอื่นเกิดตามมาอีกมากมาย การเก็บข้อมูลในรูปแบบเอกสารจะทำให้ค้นหาข้อมูลได้ยาก ใช้ระยะเวลานานในการตรวจสอบข้อมูล และไม่สามารถรู้ยอดเงินคงเหลือในแต่ละหมวดได้อย่างชัดเจน ทำให้เราไม่สามารถรับทราบการเบิกจ่ายงบประมาณ ได้อย่างแน่นอน ณ เวลานั้น เป็นผลทำให้เราอาจมีการเบิกจ่ายเกินเงินงบประมาณนั้นๆ ได้ ซึ่งจำนวนเงินงบประมาณที่เหลือในแต่ละหมวดจะมีผลในการตัดสินใจในการอนุมัติการเบิกจ่ายได้ ดังนั้นปัญหาแรกๆ ที่ควรจัดการแก้ไขควรที่จะเป็นปัญหาการจัดการฐานข้อมูลระบบบัญชีคณะวิศวกรรมศาสตร์

ในบทนี้ จึงจะบอกกล่าวถึงแนวทางในการศึกษา และพัฒนาโครงการ โดยนำเอาขั้นตอนการวิเคราะห์และออกแบบระบบ มาใช้ในการจัดการระบบและฐานข้อมูลที่จะใช้ในโครงการมีรายละเอียดดังต่อไปนี้

3.1 แผนภาพกระแสข้อมูล (Data Flow Diagram)

เป็นแผนภาพกระแสข้อมูลที่ได้มีการวิเคราะห์ในแบบเชิงโครงสร้าง โดยแผนภาพกระแสข้อมูลนี้ใช้เป็นเครื่องมือในการพัฒนาระบบงาน เพื่อแสดงความสัมพันธ์ระหว่างโปรเซสกับข้อมูลที่เกี่ยวข้อง โดยข้อมูลในแผนภาพจะทำให้ทราบถึงที่มาของข้อมูลว่าข้อมูลดังกล่าวมาจากไหนและข้อมูลไปที่ไหน ข้อมูลเก็บที่ใดและเกิดเหตุการณ์ใดกับข้อมูลในระหว่างทาง

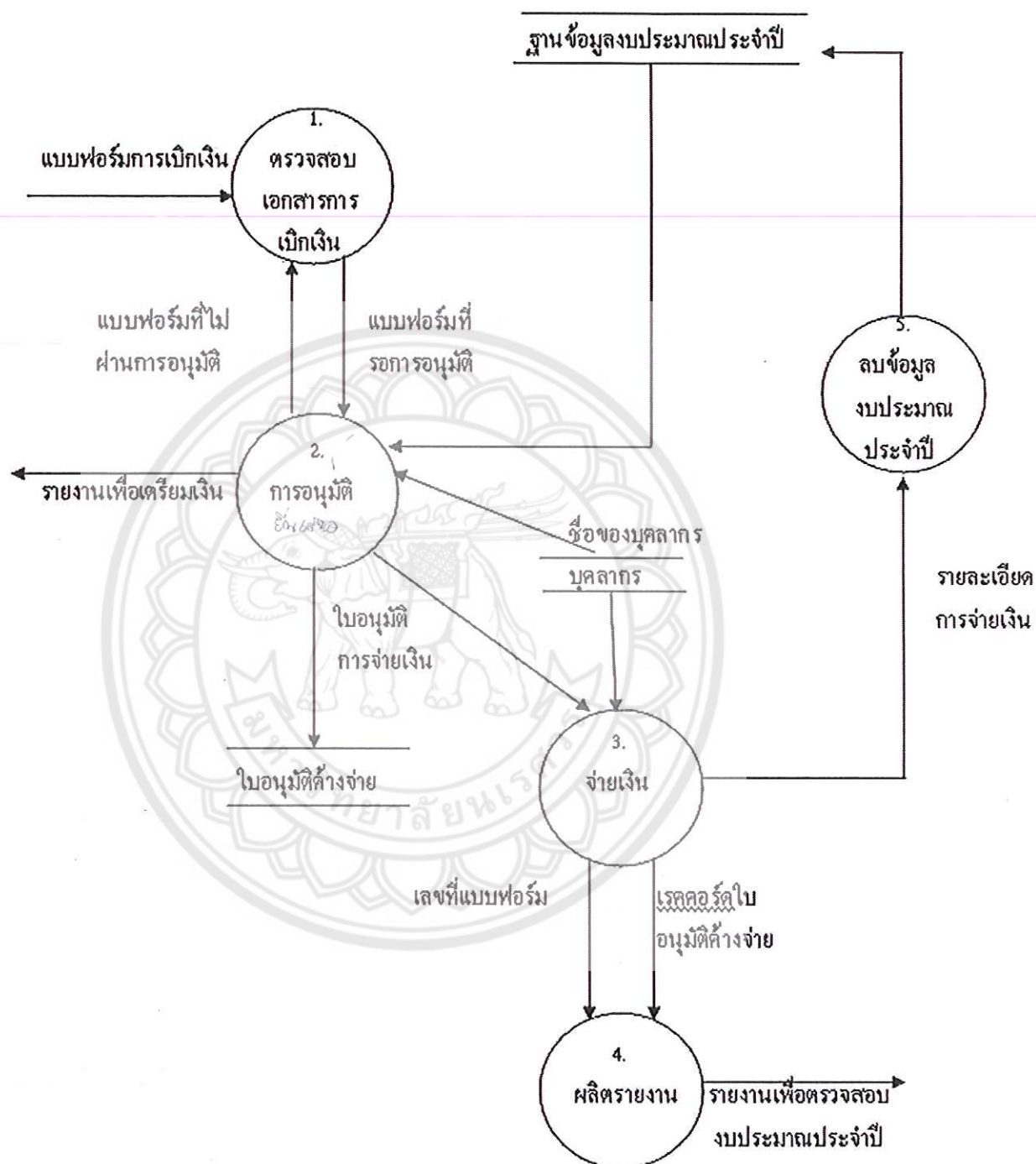
DFD Level 0



รูปที่ 3.1 DFD Level 0

จากรูปที่ 3.1 เป็น DFD ระดับสูงของระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์ ซึ่งมีชื่อเรียกอีกอย่างหนึ่งว่า “Context Diagram” ซึ่งระดับนี้จะบอกว่าระบบมีอินพุตเป็นอะไร ผลลัพธ์ที่ได้จากระบบ และข้อมูลที่เกี่ยวข้องกับระบบภายนอก

Process 0 Level 1



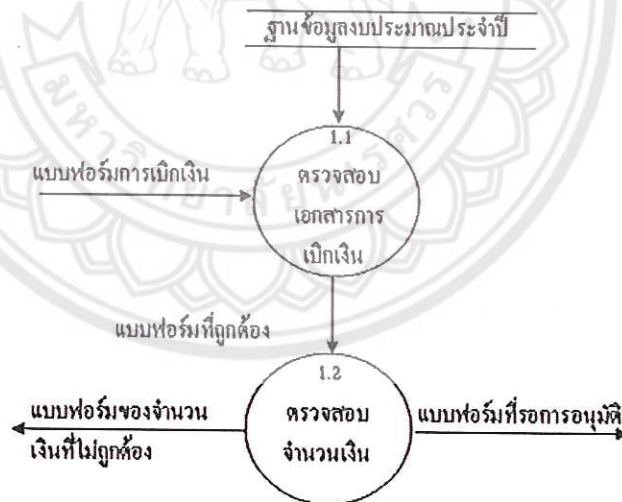
รูปที่ 3.2 แผนภาพ DFD ของกระบวนการ 0 ระดับ 1

จากแผนภาพการไหลของ Level1 นี้ เราจะแสดงรายละเอียดของกระบวนการที่ 0 ใน Level 0 ออกเป็น 5 ส่วนประกอบด้วย

- ✓1. ระบบตรวจสอบเอกสารการเบิกเงิน
- 2. ระบบการอนุมัติ
- ✓3. ระบบการจ่ายเงิน
- ✓4. ระบบการผลิตรายงาน
- ✓5. ระบบการลงข้อมูลงบประมาณประจำปี

จากรูปที่ 3.2 มีอินพุต คือ แบบฟอร์มการเบิกเงินเข้าสู่โปรแกรมตรวจสอบเอกสารการเบิกเงิน แบบฟอร์ม ที่ผ่านการตรวจสอบแล้ว จะเข้าสู่โปรแกรมอนุมัติ โดยจะใช้ข้อมูลจากฐานข้อมูลงบประมาณประจำปี และข้อมูลจากฐานข้อมูลบุคลากรมาใช้ในการพิจารณาอนุมัติการเบิกเงินงบประมาณ แบบฟอร์มที่ไม่ผ่านการอนุมัติจะถูกส่งกลับไปยังฝ่ายตรวจสอบเอกสาร สำหรับแบบฟอร์มที่ผ่านการอนุมัติจะถูกเก็บไว้ในฐานข้อมูลใบอนุมัติค้างจ่าย และเข้าสู่โปรแกรมจ่ายเงิน ซึ่งโปรแกรมจ่ายเงินจะแสดงรายละเอียดการจ่ายเงิน เพื่อลงข้อมูลงบประมาณประจำปีในระบบฐานข้อมูลงบประมาณประจำปี ส่งเลขที่ฟอร์ม และเรคคอร์ดใบอนุมัติค้างจ่าย เพื่อจัดทำรายงานตรวจสอบงบประมาณประจำปี

Process 1 Level 1



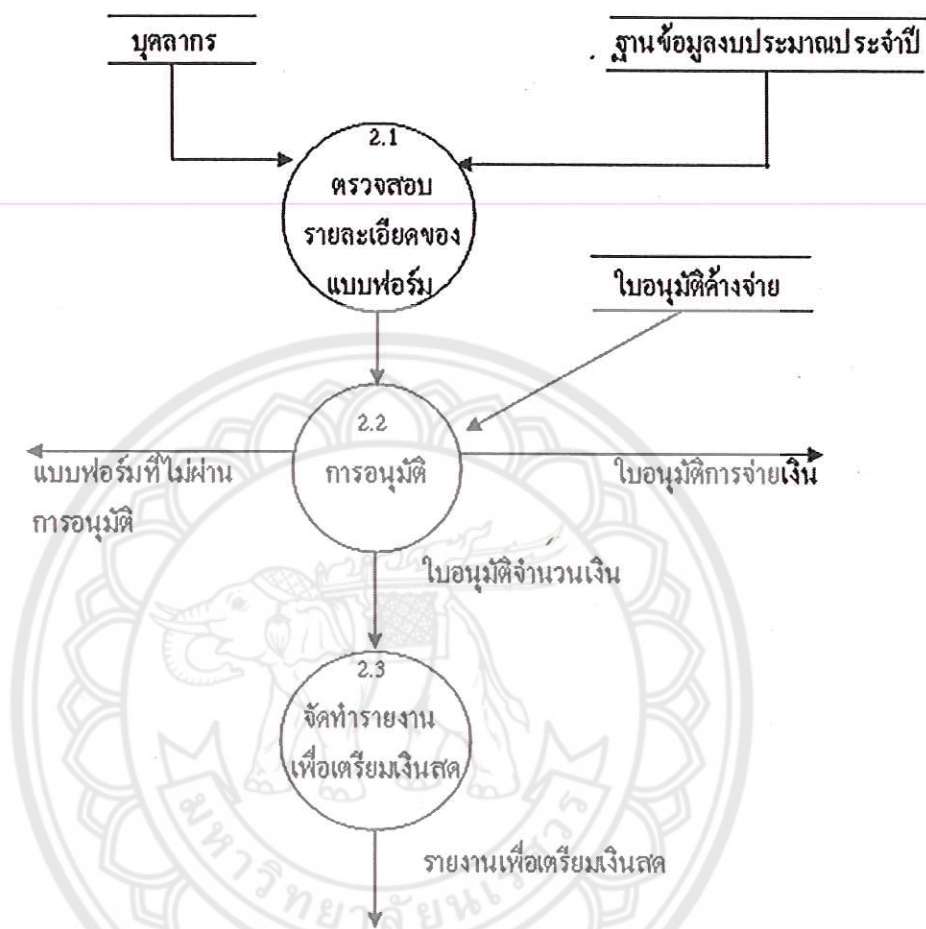
Process1.1 : การตรวจสอบเอกสารการเบิกเงิน

Process1.2 : การตรวจสอบจำนวนเงิน

รูปที่ 3.3 แผนภาพ DFD ของกระบวนการที่ 1 ระดับที่ 1

จากรูปที่ 3.3 ฐานข้อมูลงบประมาณประจำปี มาใช้ในการตรวจสอบข้อมูลในระบบฐานข้อมูลงบประมาณประจำปี กับแบบฟอร์มการเบิกเงินและจำนวนเงิน เมื่อทำการตรวจสอบแล้วแบบฟอร์มจะแบ่งออกเป็น 2 ประเภทคือ แบบฟอร์มของจำนวนเงินที่ไม่ถูกต้อง และแบบฟอร์มที่รอการอนุมัติ

Process 2 Level 1



Process 2.1 : การตรวจสอบรายละเอียดของแบบฟอร์ม

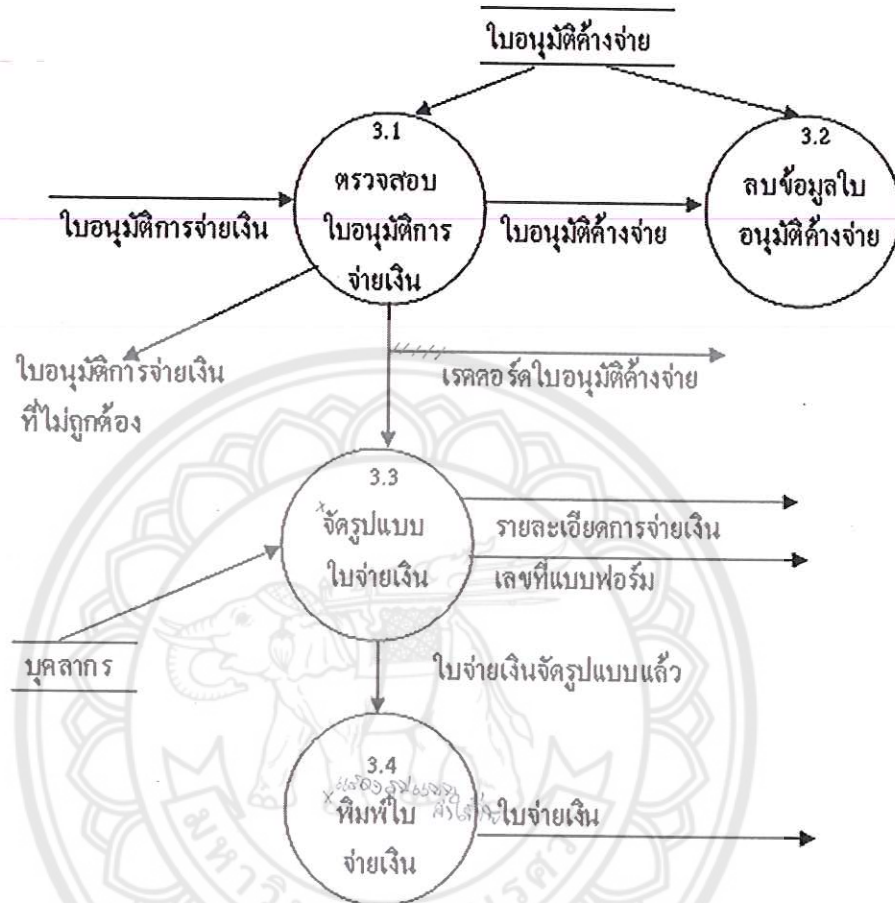
Process 2.2 : การอนุมัติ

Process 2.3 : การจัดทำรายงานเพื่อเตรียมเงินสด

รูปที่ 3.4 แผนภาพ DFD ของกระบวนการที่ 2 ระดับที่ 1

นำฐานข้อมูลของบุคลากร และฐานข้อมูลงบประมาณประจำปี มาช่วยในการตรวจสอบรายละเอียดของแบบฟอร์ม เพื่อช่วยในการอนุมัติงบประมาณประจำปี ข้อมูลที่ได้รับคือ ใบอนุมัติการจ่ายเงิน และใบอนุมัติจำนวนเงิน เพื่อใช้ในการจัดทำรายงานเพื่อเตรียมเงินสด

Process 3 Level 1



Process3.1 : การตรวจสอบใบอนุมัติการจ่ายเงิน

Process3.2 : การลบข้อมูลใบอนุมัติค้างจ่าย

Process3.3 : การจัดรูปแบบใบจ่ายเงิน

Process3.4 : การพิมพ์ใบจ่ายเงิน

รูปที่ 3.5 แผนภาพ DFD ของกระบวนการที่ 3 ระดับที่ 1

จากรูปที่ 3.5 ฐานข้อมูลใบอนุมัติค้างจ่าย ซึ่งถูกใช้ในการตรวจสอบการจ่ายเงินในโปรแกรม 3.1 เพื่อจะดึงเลขที่แบบฟอร์มเข้ามา ข้อมูลใบอนุมัติค้างจ่ายจะถูกลบออก เมื่อจ่ายเงินเรียบร้อยแล้ว ฐานข้อมูลบุคลากรถูกใช้ เพื่อดึงข้อมูลของบุคลากร นอกจากนั้นเลขที่บุคลากรจะอยู่ในเรคคอร์ดใบอนุมัติค้างจ่าย โปรแกรม 3.3 และ 3.4 ใช้สำหรับพิมพ์รายงานการจ่ายเงิน ในแผนภาพนี้มีข้อมูลที่เกิดผิดพลาดเพียงอย่างเดียวคือ ใบอนุมัติค้างจ่ายที่ไม่ถูกต้อง

Process 4 Level 1



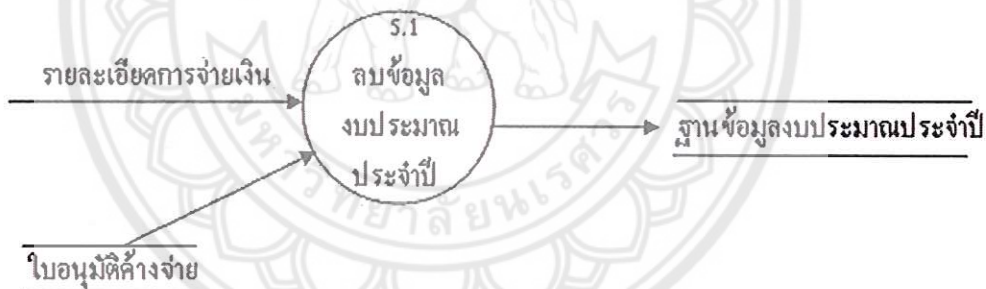
Process4.1 : การจัดรูปแบบรายงานการตรวจสอบงบประมาณประจำปี

Process4.2 : สรุปรายงาน

รูปที่ 3.6 แผนภาพ DFD ของกระบวนการที่ 4 ระดับที่ 1

จากรูปที่ 3.6 จะเป็นโพรเซสสำหรับจัดรูปแบบรายงานการตรวจสอบงบประมาณประจำปี และสรุปรายงาน เพื่อพิมพ์รายงานตรวจสอบงบประมาณประจำปี

Process 5 Level 1



Process5.1 : การลบข้อมูลงบประมาณประจำปี

รูปที่ 3.7 แผนภาพ DFD ของกระบวนการที่ 5 ระดับที่ 1

จากรูปที่ 3.7 ข้อมูลใบอนุมัติค้างจ่ายจะถูกลบข้อมูลงบประมาณประจำปี ออกเมื่อจ่ายเงินเรียบร้อยแล้ว และเก็บไว้ในฐานข้อมูลงบประมาณประจำปี

3.2 การออกแบบฐานข้อมูล

หลังจากทำการศึกษาในส่วนของฐานข้อมูลแล้วเมื่อทำการออกแบบ Database ให้ครอบคลุม Requirement และง่าย สะดวกที่สุด ซึ่ง Database จะแบ่งเป็นส่วนหลักๆ ดังนี้

3.2.1 TbDepartment เป็นตารางเกี่ยวกับหน่วยงาน ที่ใช้สำหรับเก็บรายละเอียด ของหน่วยงานคณะวิศวกรรมศาสตร์ ซึ่งมีรายชื่อหน่วยงาน ดังนี้

- สำนักงานเลขานุการคณะ (so)
- ภาควิชาวิศวกรรมเครื่องกล (me)
- ภาควิชาวิศวกรรมโยธา (ce)
- ภาควิชาวิศวกรรมอุตสาหกรรม (ie)
- ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ (ee)
- โครงการปริญญาตรีภาคพิเศษ (tw)
- หน่วยงานนอกคณะ

ตารางที่ 3.1 ตารางพจนานุกรมข้อมูลของตาราง TbDepartment

Field	Type	Length	Description	Remark
DepartID	Text	2	รหัสหน่วยงาน	Primary Key
DepartName	Text	40	ชื่อหน่วยงาน	

จากตารางที่ 3.1 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbDepartment

3.2.2 TbPlanMain เป็นตารางที่ใช้สำหรับเก็บรายละเอียดของ กองทุน/แผนงาน/งาน คณะวิศวกรรมศาสตร์ ซึ่งมีรายชื่อกองทุน/แผนงาน/งาน ดังนี้

- กองทุนเพื่อการศึกษา/แผนงานจัดการศึกษาอุดมศึกษา/งานสนับสนุนการจัดการศึกษา (01)
- กองทุนเพื่อการศึกษา/แผนงานจัดการศึกษาอุดมศึกษา/งานจัดการศึกษา (02)
- กองทุนสินทรัพย์ถาวร/แผนงานจัดการศึกษาอุดมศึกษา/งานสนับสนุนการจัดการศึกษา (03)
- กองทุนสินทรัพย์ถาวร/แผนงานจัดการศึกษาอุดมศึกษา/งานจัดการศึกษา (04)
- กองทุนกิจการนิสิต/แผนงานจัดการศึกษาอุดมศึกษา/งานสนับสนุนการจัดการศึกษา (05)
- กองทุนกิจการนิสิต/แผนงานจัดการศึกษาอุดมศึกษา/งานจัดการศึกษา (06)
- กองทุนวิจัย/แผนงานวิจัย/งานวิจัยพัฒนาและเทคโนโลยี (07)

ตารางที่ 3.2 ตารางพจนานุกรมข้อมูลของตาราง TbPlanMain

Field	Type	Length	Description	Remark
PIMainID	Text	2	รหัสกองทุน/แผนงาน/ งาน	Primary Key
PIMainName	Text	100	ชื่อกองทุน/แผนงาน/ งาน	

จากตารางที่ 3.2 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbPlanMain

3.2.5 TbPlanThird เป็นตารางเกี่ยวกับหมวด 3 ที่ใช้สำหรับเก็บรายละเอียด ของรหัสหมวด 3 คณะวิศวกรรมศาสตร์ ซึ่งมีรายชื่อรหัสหมวด 3 ดังนี้

- หมวดค่าจ้างชั่วคราว

- ค่าจ้างชั่วคราว

- ค่าจ้างชั่วคราวรายเดือน

- ค่าจ้างชั่วคราวรายวัน

◁ หมวดค่าตอบแทนใช้สอยและวัสดุ

× ค่าตอบแทน

◁ ค่าทำการนอกเวลา

- ค่าตอบแทนตำแหน่งบริหาร

◁ ค่าตอบแทนในการสอน

- ค่าตอบแทนนิสิตช่วยงาน

× สาขาขาดแคลน

- ค่าตอบแทนรายวิชานอกคณะ

- ค่าเบี้ยประชุม

× ค่าตรวจกระดาษคำตอบ

- ค่าสอนเกินภาระงาน

× ค่าตอบแทนอื่นๆ

× ค่าใช้สอย

× ค่าเดินทางไปราชการ

× ค่าประชุมและพิธีการ

- ค่าจ้างถ่ายเอกสารและเข้าเล่ม

× ค่าเบี้ยประกันสังคม

- ค่าซ่อมแซมครุภัณฑ์

× ค่าใช้สอยอื่นๆ

× ค่าวัสดุ

× ค่าวัสดุสำนักงาน

× ค่าวัสดุคอมพิวเตอร์

× ค่าวัสดุไฟฟ้าและวิทยุ

- ค่าวัสดุน้ำมันเชื้อเพลิงและหล่อลื่น
- ค่าวัสดุก่อสร้าง
- ค่าวัสดุงานบ้านงานครัว
- ค่างานวัสดุการเกษตร
- ค่าวัสดุหนังสือและวารสาร
- ค่าวัสดุการศึกษา
- ค่าวัสดุวิทยาศาสตร์และการแพทย์
- ค่าวัสดุโฆษณาและเผยแพร่
- ค่าวัสดุอื่นๆ

- หมวดค่าสาธารณูปโภค

- ค่าไฟฟ้า

- ค่าไฟฟ้า

- ค่าโทรศัพท์

- ค่าโทรศัพท์เคลื่อนที่

- ค่าโทรศัพท์ส่วนกลางคณะฯ * (ตั้งไว้เพื่อเงิน งบประมาณ ไม่พอโอนให้มหาวิทยาลัย)

- หมวดเงินอุดหนุน

- อุดหนุนทั่วไป

โครงการ *** พิจารณาอีกทีว่าจะเจาะลงลึกแค่ไหน

- หมวดค่าครุภัณฑ์ที่ดินและสิ่งก่อสร้าง

- ค่าครุภัณฑ์

ค่าครุภัณฑ์ *** พิจารณาอีกทีว่าจะเจาะลงลึกแค่ไหน

- ค่าสิ่งก่อสร้าง

- ค่าสิ่งก่อสร้าง

ตารางที่ 3.3 ตารางพจนานุกรมข้อมูลของตาราง TbPlanThird

Field	Type	Length	Description	Remark
PITrdD	Text	10	รหัสหมวด3	Primary Key
PITrdName	Text	50	ชื่อหมวด3	
PITrdBudget	Currency		จำนวนเงินในหมวด3	เฉพาะเงินโครงการ ในหมวดอุดหนุน และค่าครุภัณฑ์ *** พิจารณาอีกทีว่าจะ เจาะลงลึกแค่ไหน ***
PIStdID	Text		รหัสหมวด2	Foreign Key
DepartID	Text	2	รหัสหน่วยงาน	Foreign Key

จากตารางที่ 3.3 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbPlanThird

3.2.6 TbDocDetail เป็นตารางที่ใช้สำหรับ เก็บรายละเอียดของเอกสาร คณะวิศวกรรมศาสตร์ ซึ่งมีรายละเอียดเอกสาร ดังนี้

ตารางที่ 3.4 ตารางพจนานุกรมข้อมูลของตาราง TbDocDetail

Field	Type	Length	Description	Remark
DocID	Text		รหัสเอกสาร	Primary Key
DocDetail	Text	50	รายละเอียดเอกสาร	
PreDate	Date		วันที่เสนอขอจัดซื้อ/จ้าง.../	now
PrePay	Currency		จำนวนเงินประมาณการ	
PostDate	Date		วันที่จัดซื้อ-จ้างแล้ว	
Paid	Currency		จำนวนจ่ายจริง	
ReferNo	Text	20	เลขที่เอกสารอ้างอิง เช่น ใบส่งสินค้า	
PITrdID	Text	10	รหัสหมวด3	Foreign Key
Remark	Text	50	หมายเหตุ	
GroupID	Text	4	รหัสฎีกา	Foreign Key
Status	Text	1	สถานะเอกสาร มี 4 สถานะ 1.กำลังเสนอขออนุมัติ 2.อนุมัติให้จัดซื้อ/จ้าง.../ 3.จัดซื้อ-จ้างแล้ว (มีใบส่ง ของแต่เอกสารยังไม่ถึง การเงิน) 4.การเงินตั้งฎีกาเบิก 5.กัณเงิน*** 6 .ยกเลิก	

จากตารางที่ 3.4 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbDocDetail ในส่วนของการแสดงรายละเอียดตารางอื่น ๆ จะอยู่ในภาคผนวก ก บทที่ 3 นี้จะเป็นการนำเอาการวิเคราะห์และออกแบบระบบ มาใช้ในการจัดการระบบและฐานข้อมูลที่จะใช้ในโครงการ ซึ่งเป็นขั้นตอนเบื้องต้น ที่จะนำไปสู่การพัฒนาโปรแกรม ในบทที่ 4 ต่อไป

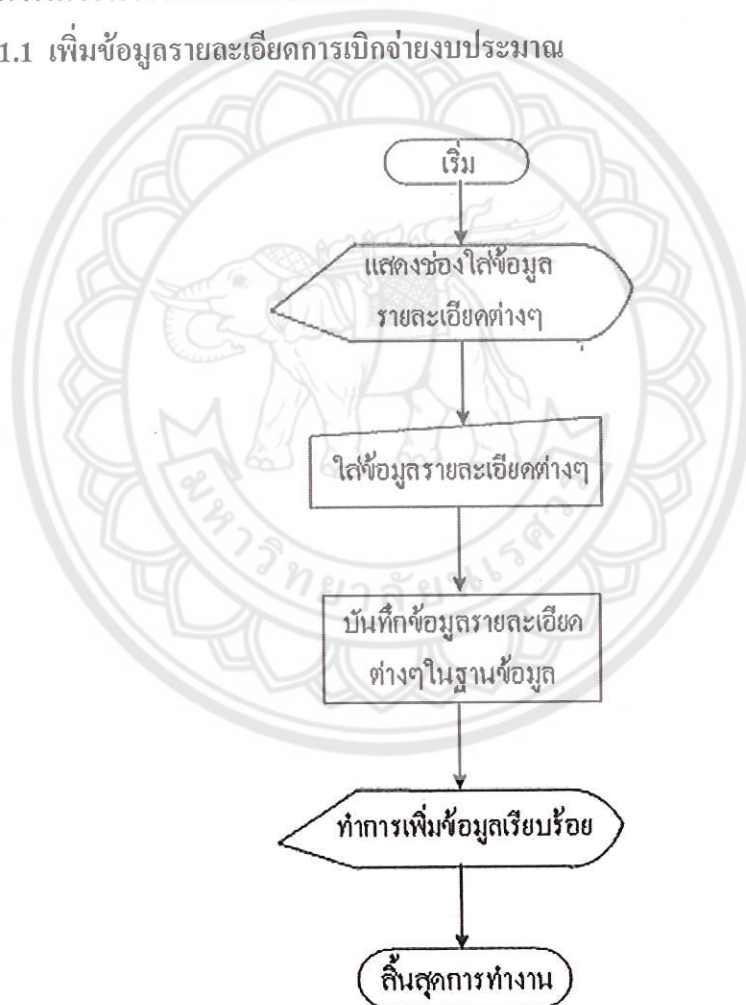
บทที่ 4

การพัฒนาโปรแกรม

ในบทนี้จะกล่าวถึง การทดสอบและวิเคราะห์การใช้งานจริงของโปรแกรม ซึ่งโปรแกรมที่ได้จัดทำขึ้นนั้นได้ใช้โปรแกรม Visual Basic 6.0 ทำการติดต่อกับฐานข้อมูล เพื่อจะได้ทราบว่าสามารถทำงานได้ถูกต้องตามที่ออกแบบไว้หรือไม่

4.1 แผนภาพการทำงานของโปรแกรม

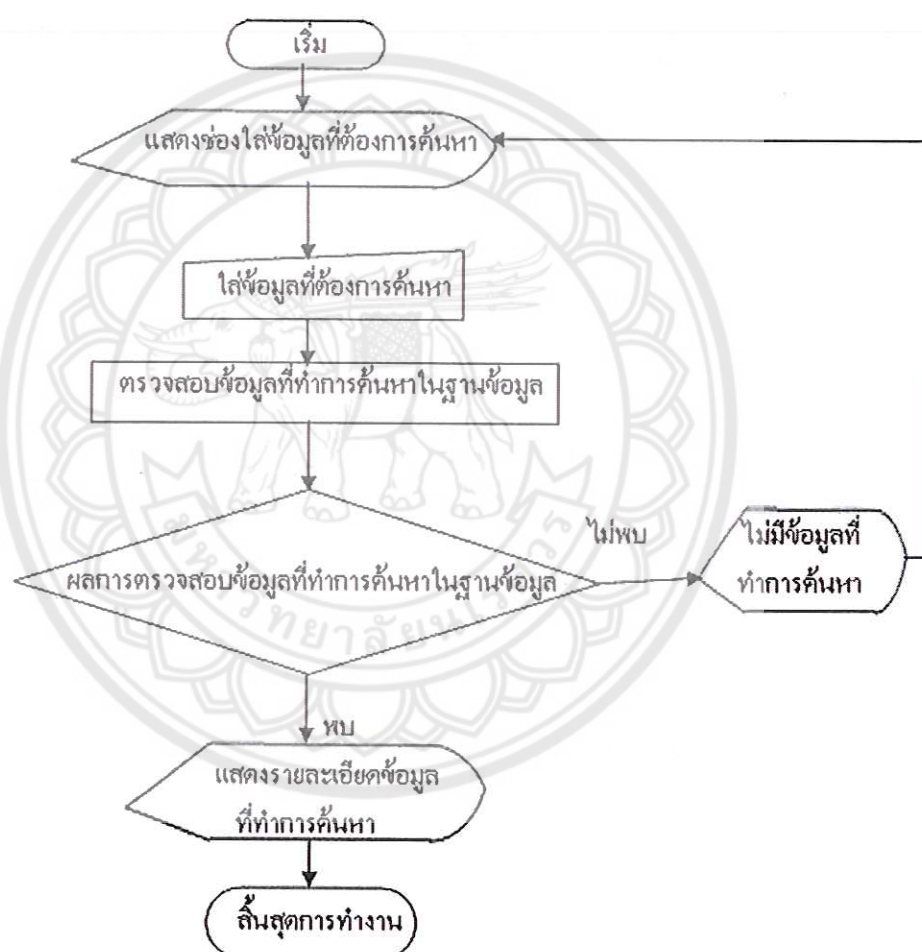
4.1.1 เพิ่มข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ



รูปที่ 4.1 เพิ่มข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ

เพิ่มรายละเอียดข้อมูลการเบิกจ่ายงบประมาณ เริ่มต้นหน้าโปรแกรมเพิ่มข้อมูล รายละเอียดการเบิกจ่ายงบประมาณ โดยโปรแกรมจะแสดงช่องใส่ข้อมูลรายละเอียดต่างๆ ได้แก่ รหัสเอกสาร, เลขฎีกา, สถานะเอกสาร, รายละเอียด, วันที่ส่ง, วันที่จ่ายจริง, งบประมาณการ, จ่ายจริง, หมวด3, เอกสารอ้างอิงและหมายเหตุระบบกรอกรายละเอียดข้อมูลต่างๆ ไปในฐานข้อมูลและจะแสดงหน้าต่างข้อความขึ้นมาว่าทำการเพิ่มข้อมูลเรียบร้อยแล้ว เป็นการเสร็จขั้นตอนการเพิ่มข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ ดังแสดงไว้ในรูปที่ 4.1

4.1.2 ค้นหาข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ

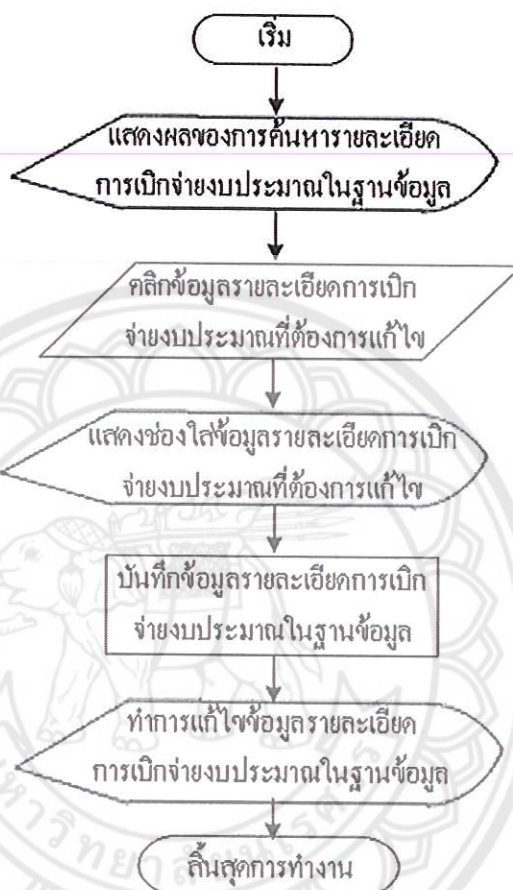


รูปที่ 4.2 ค้นหาข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ

ค้นหาข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ เริ่มต้นหน้าโปรแกรมค้นหาข้อมูลรายละเอียดการเบิกจ่ายงบประมาณ โดยโปรแกรมจะแสดงช่องใส่ข้อมูล รหัสเอกสารที่ต้องการค้นหา โปรแกรมจะทำการค้นหาข้อมูลรายละเอียดเอกสารในฐานข้อมูล เมื่อผลการค้นหาออกมาว่าไม่พบรหัสเอกสารในฐานข้อมูล โปรแกรมจะทำการขึ้นหน้าต่างเตือนว่าไม่มีข้อมูลที่ต้องการค้นหา และเมื่อผลการค้นหาออกมาว่า พบข้อมูลรายละเอียดเอกสารในฐานข้อมูล โปรแกรมจะแสดงข้อมูล

รายละเอียดเอกสารที่ทำการค้นหานั้นออกมา เป็นการเสร็จขั้นตอนการค้นหารายละเอียดการเบิกจ่ายงบประมาณในฐานข้อมูล ดังแสดงไว้ในรูปที่ 4.2

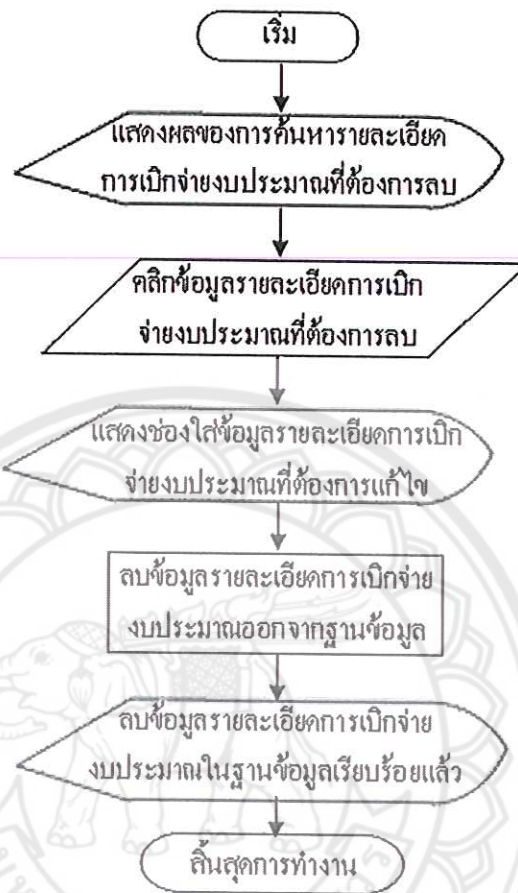
4.1.3 แก้ไขข้อมูลทะเบียนรายละเอียดการเบิกจ่ายงบประมาณ



รูปที่ 4.3 แก้ไขข้อมูลทะเบียนรายละเอียดการเบิกจ่ายงบประมาณ

แก้ไขข้อมูลรายละเอียดการเบิกจ่าย เริ่มค้นหาโปรแกรมแก้ไขข้อมูลรายละเอียด การเบิกจ่าย โดยจะเริ่มค้นจากหน้าผลการค้นหาข้อมูลรายละเอียดการเบิกจ่ายเมื่อคลิกข้อมูลรายละเอียด เอกสารที่จะทำการแก้ไข โปรแกรมจะแสดงช่องใส่ข้อมูลรายละเอียดเอกสารที่ต้องการแก้ไข ผู้ใช้ ระบบกรอกข้อมูลที่ต้องการแก้ไขเข้าไปที่โปรแกรม โปรแกรมจะทำการบันทึกข้อมูลรายละเอียด การเบิกจ่ายงบประมาณ เป็นการเสร็จขั้นตอนการแก้ไขข้อมูลรายละเอียดเอกสาร ดังแสดงไว้ในรูป ที่ 4.3

4.1.4 ลบข้อมูลทะเบียนรายละเอียดการเบิกจ่ายงบประมาณ

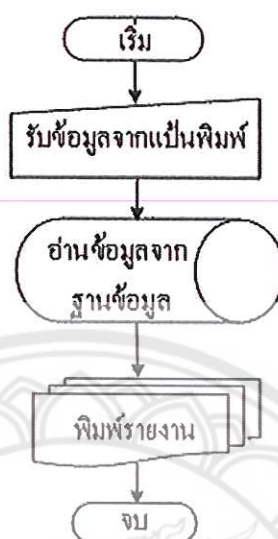


รูปที่ 4.4 ลบข้อมูลทะเบียนรายละเอียดการเบิกจ่ายงบประมาณ

ลบข้อมูลรายละเอียดการเบิกจ่ายงบประมาณเริ่มต้นหน้าตาโปรแกรมลบข้อมูลรายละเอียดการเบิกจ่ายจะเริ่มค้นจากหน้าผลของการค้นหาข้อมูลรายละเอียดการเบิกจ่ายและเมื่อคลิกข้อมูลรายละเอียดการเบิกจ่ายที่ต้องการลบไฟล์ โปรแกรมจะทำการลบข้อมูลการเบิกจ่ายงบประมาณออกจากฐานข้อมูล เป็นการเสร็จขั้นตอนการลบข้อมูลทะเบียนรายละเอียดการเบิกจ่ายงบประมาณ ดังแสดงไว้ในรูป 4.4

4.1.5 การสั่งพิมพ์รายงาน

ในกรณีการสั่งพิมพ์รายงานนั้นจะมีขั้นตอนในการทำงานของ โปรแกรมดังรูปที่ 4.5

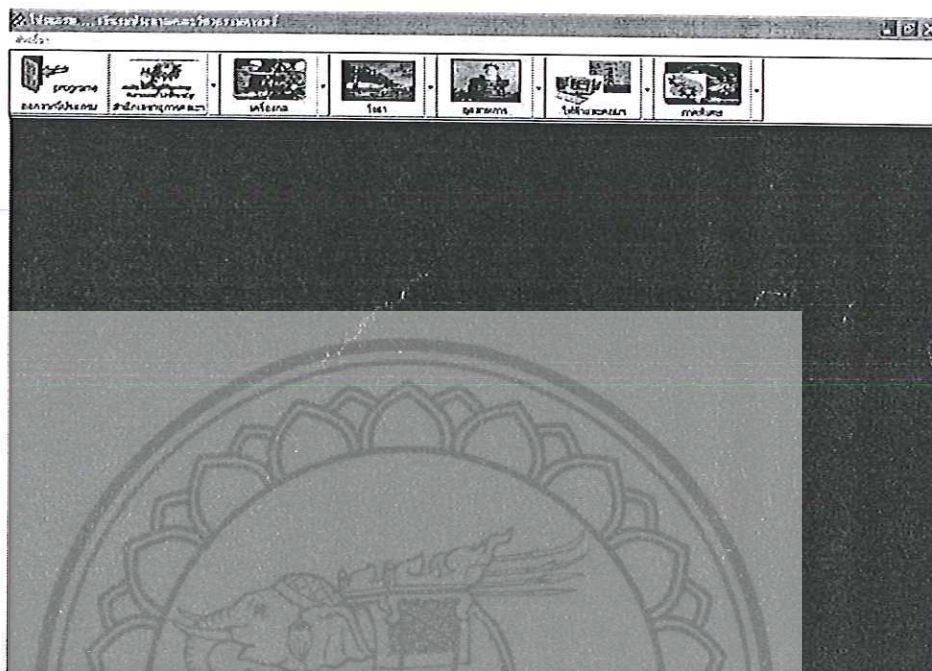


รูปที่ 4.5 ผังงานของ โปรแกรมในส่วนแสดงข้อมูลทั่วไปเพื่อพิมพ์รายงาน

จากรูปที่ 4.5 พิมพ์รายงานการเบิกจ่ายงบประมาณ เริ่มต้นที่หน้าโปรแกรม รายละเอียดการเบิกจ่ายงบประมาณ เมื่อกรอกข้อมูลครบแล้ว เราสามารถสั่งพิมพ์รายงาน โดยคลิกที่ปุ่มพิมพ์ ก็จะแสดงรายงานรายละเอียดการเบิกจ่ายงบประมาณออกมา

4.2 รูปแบบของโปรแกรมระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์

4.2.1 หน้าแรกของโปรแกรม



รูปที่ 4.6 แสดงหน้าแรกเข้าโปรแกรม

หน้านี้จะแสดงแต่ละหน่วยงานในภาควิชาวิศวกรรมศาสตร์ ให้ผู้ที่ต้องการกรอกข้อมูลได้เห็นเวลาผู้ใช้เข้าโปรแกรม ระบบบัญชีของคณะวิศวกรรมศาสตร์ ด้านบนของโปรแกรมจะเป็นส่วนที่ใช้แสดงงบประมาณของหน่วยงานต่างๆ ในคณะวิศวกรรมศาสตร์ แสดงวิธีการใช้งานโปรแกรม แสดงรายละเอียดการใช้จ่ายเงินในแต่ละหมวด และส่วนออกจากโปรแกรม ถัดลงมาจะเป็นส่วนให้เลือกกรอกรายละเอียดการใช้จ่ายของแต่ละหน่วยงานในคณะวิศวกรรมศาสตร์คือสำนักเลขานุการ คณะฯ ภาควิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมอุตสาหกรรม ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ และโครงการปริญญาตรีภาคพิเศษ และส่วนที่ออกจากโปรแกรม ดังแสดงในรูปที่ 4.6

4.2.2 เงินงบประมาณ

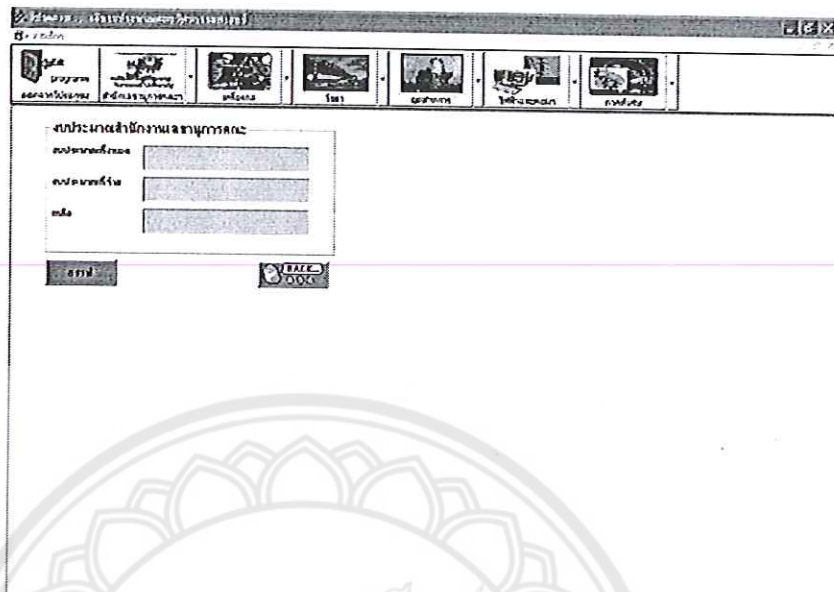
The screenshot shows a software application window with a menu bar and a toolbar. Below the toolbar is a data entry form with several text boxes and buttons. The form contains the following data:

รหัสหมวด3	ชื่อหมวด3	จำนวนเงินหมวด3	รหัสหมวด2	ชื่อหมวด2
00000100	ค่าจ้างบรรณกร	100000	00001	งบ
00000200	ค่าจ้างครู	100000	00001	งบ
00000300	ค่าตอบแทนพิเศษ	100000	00001	งบ
00000400	ค่าตอบแทนพิเศษ	100000	00001	งบ
00000500	ค่าตอบแทนพิเศษ	100000	00001	งบ
00000600	ค่าตอบแทนพิเศษ	100000	00001	งบ
00000700	ค่าตอบแทนพิเศษ	100000	00001	งบ
00000800	ค่าตอบแทนพิเศษ	100000	00001	งบ
00000900	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001000	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001100	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001200	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001300	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001400	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001500	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001600	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001700	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001800	ค่าตอบแทนพิเศษ	100000	00001	งบ
00001900	ค่าตอบแทนพิเศษ	100000	00001	งบ
00002000	ค่าตอบแทนพิเศษ	100000	00001	งบ

รูปที่ 4.7 แสดงเงินงบประมาณในแต่ละหน่วยงาน

ฐานข้อมูลเงินงบประมาณหน่วยงานต่างๆ ภายในภาควิชาวิศวกรรมศาสตร์ ซึ่งจะมีงบประมาณของสำนักเลขาธิการคณะฯ ภาควิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมอุตสาหกรรม ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ และ โครงการปริญญาตรี ภาคพิเศษ ฐานข้อมูลเงินงบประมาณจะเก็บข้อมูลต่างๆ คือ รหัสหมวด3 ชื่อหมวด3 จำนวนเงินในหมวด3 และรหัสหมวด2 การจัดเก็บข้อมูลลงในฐานข้อมูลเงินงบประมาณ ทำได้โดยให้ผู้ใช้กรอกข้อมูลลงใน TextBox ที่ระบุชื่อข้อมูลอยู่ข้างหน้าเมื่อต้องการเพิ่มข้อมูลลงไป ในฐานข้อมูลเงินงบประมาณ ให้ผู้ใช้คลิกปุ่ม “เพิ่มข้อมูล” แล้ว TextBox ต่างๆ จะว่างพร้อมที่จะพิมพ์ข้อมูลลงไปได้ ผู้ใช้ทำการพิมพ์ข้อมูลต่างๆ ของแต่ละหัวข้อได้เมื่อกรอกข้อมูลครบเรียบร้อยแล้วจึงทำการคลิกปุ่ม “บันทึก” จะเป็นการนำเอาข้อมูลที่ผู้ใช้กรอกไปเก็บลงในฐานข้อมูล ก่อนที่ผู้ใช้จะคลิกปุ่ม “บันทึก” ถ้าผู้ใช้ต้องการยกเลิกการเพิ่มข้อมูลก็ให้คลิกปุ่ม “ยกเลิก” ได้ จะปรากฏ MessageBox ถ้าผู้ใช้ว่าต้องการที่จะยกเลิกหรือไม่ เมื่อผู้ใช้ต้องการแก้ไขข้อมูลที่มีอยู่ก่อนแล้วให้ผู้ใช้คลิกปุ่ม “แก้ไข” TextBox ต่างๆ จะอยู่ในสถานะพร้อมที่จะแก้ไขข้อมูล เมื่อแก้ไขข้อมูลเรียบร้อยแล้วจึงทำการคลิกปุ่ม “บันทึก” ผู้ใช้ต้องการค้นหาข้อมูลให้ผู้ใช้คลิกปุ่ม “ค้นหา” แล้วจึงทำการพิมพ์รหัสหมวด3 ของสิ่งที่ต้องการค้นหาลงไป เมื่อผู้ใช้คลิกปุ่ม “บันทึก” โปรแกรมจะมีการตรวจสอบรูปแบบข้อมูล ถ้าไม่ถูกต้องจะมี MessageBox แสดงต่อผู้ใช้ว่าที่ใดที่ผู้ใช้กรอกข้อมูลผิดพลาด เพื่อความถูกต้องของข้อมูลในฐานข้อมูล ดังแสดงในรูปที่ 4.7

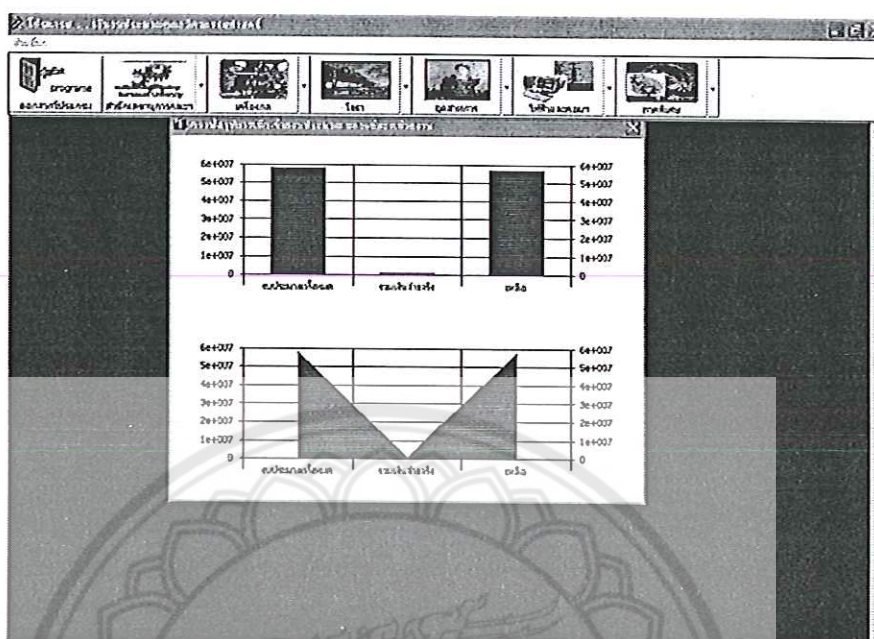
4.2.3 สรุปงบประมาณแต่ละหน่วยงาน



รูปที่ 4.8 แสดงการสรุปงบประมาณแต่ละหน่วยงาน

* สรุปการเบิกจ่ายงบประมาณของแต่ละหน่วยงานของภาควิชาวิศวกรรมศาสตร์ คือ สรุปการเบิกจ่ายงบประมาณสำนักเลขานุการคณะฯ สรุปการเบิกจ่ายงบประมาณภาควิชาวิศวกรรมเครื่องกล สรุปการเบิกจ่ายงบประมาณภาควิชาวิศวกรรมโยธา สรุปการเบิกจ่ายงบประมาณภาควิชาวิศวกรรมอุตสาหกรรม สรุปการเบิกจ่ายงบประมาณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ และสรุปการเบิกจ่ายงบประมาณโครงการปริญญาตรีภาคพิเศษ ซึ่งสิ่งที่น่าสนใจมีงบประมาณทั้งหมด งบประมาณที่จ่ายจริง และงบประมาณที่เหลือ ดังแสดงในรูปที่ 4.8

4.2.4 กราฟแสดงการสรุปงบประมาณแต่ละหน่วยงาน



รูปที่ 4.9 กราฟแสดงการสรุปงบประมาณแต่ละหน่วยงาน

เป็นส่วนที่นำจำนวนเงินที่ได้จากการสรุปงบประมาณ มาแสดงเป็นรูปแบบของกราฟ ทำให้เห็นภาพได้ชัดเจนว่า มีการใช้งบประมาณทั้งหมด งบประมาณที่จ่ายจริง และงบประมาณที่เหลือแตกต่างกันมากหรือน้อยเพียงใด แสดงสรุปการเบิกจ่ายงบประมาณของแต่ละหน่วยงาน ออกมาเป็นลักษณะกราฟแท่ง และกราฟพื้นที่ เพื่อแสดงให้เห็นความแตกต่างมากยิ่งขึ้น ดังแสดงในรูปที่ 4.9

4.2.5 ระบบการกรอกรายละเอียดการใช้จ่ายหมวดต่างๆ ในคณะวิศวกรรมศาสตร์

The screenshot shows a web application interface for recording expenses. At the top, there are navigation icons and a title bar. The main content area is titled 'ภาควิศวกรรมโยธา CO' and contains a form with several sections:

- กรอกรายละเอียด:** Fields for account type (ประเภทบัญชี), date (วันที่), and amount (จำนวนเงิน).
- ข้อมูล:** Fields for account number (บัญชีเลขที่), date (วันที่), and amount (จำนวนเงิน).
- สรุปยอด:** Summary fields for total amount (รวมยอดเงิน), date (วันที่), and amount (จำนวนเงิน).
- กรอกรายละเอียดการหัก:** A section for recording deductions with a 'ลบ' (minus) button.
- Table:** A table with columns for 'วันที่', 'รายการ', 'ยอดเงิน', 'วันที่', 'จำนวนเงิน', and 'วันที่'.

At the bottom, there are standard browser navigation buttons like 'Back', 'Forward', 'Home', and 'Print'.

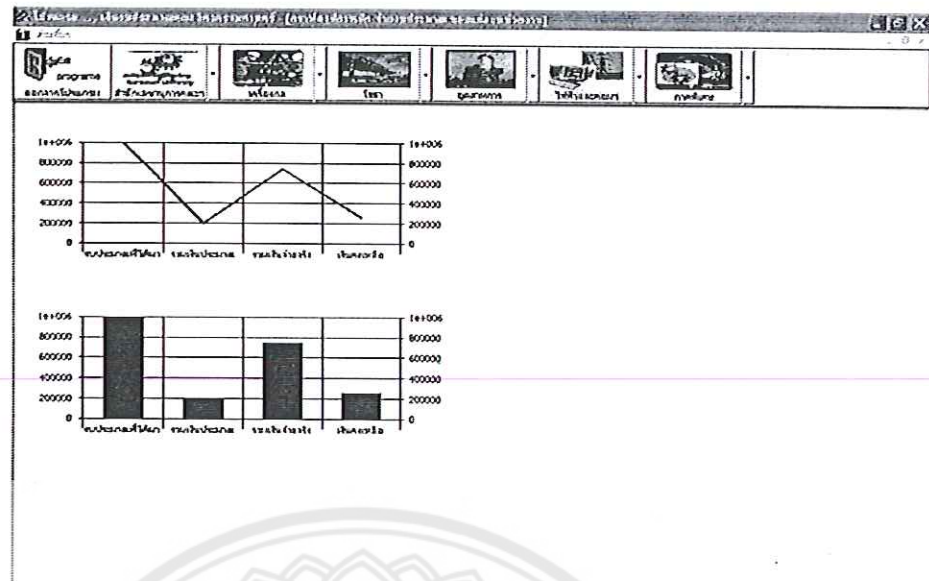
รูปที่ 4.10 ระบบการกรอกรายละเอียดการใช้จ่ายหมวดต่างๆ

การจัดเก็บข้อมูลในฟอร์มการกรอกรายละเอียด สิ่งที่จัดเก็บประกอบไปด้วยข้อมูลพื้นฐานของเอกสารต่างๆ จำนวนเงิน รหัสหมวด¹ และวันที่ เพื่อที่จะทำให้เราทราบได้ว่า การเบิกจ่ายเงินแต่ละครั้งมีรหัสเอกสารที่เท่าไร ทราบรายละเอียดการจ่ายเงิน วันที่จ่ายเงิน จำนวนเงินที่ขอเบิก และจำนวนเงินที่จ่ายไป รายละเอียดการทำงานของฟอร์มการกรอกรายละเอียด มีดังนี้

- การเพิ่มข้อมูล หลังจากกดปุ่มเพิ่มข้อมูล ช่องแสดงผลของข้อมูล (Text Box) จะกลายเป็นช่องว่าง เพื่อที่จะคอยรับค่าข้อมูลที่ป้อนเข้าไป และปุ่มคอนโทรลต่างๆ ก็จะถูกทำให้ไม่ได้ยกเว้นปุ่มบันทึกข้อมูลและปุ่มยกเลิกการแก้ไข หลังจากกรอกข้อมูลเรียบร้อยแล้ว ถ้าต้องการบันทึกข้อมูลให้กดปุ่มบันทึกข้อมูล หากไม่ต้องการบันทึกข้อมูลที่กรอกลงไปให้กดปุ่มยกเลิกการแก้ไข การแก้ไขข้อมูลปกติแล้วที่ช่องแสดงข้อความ จะไม่สามารถกรอกได้หรือแก้ไขข้อมูลได้ ดังนั้นเมื่อเราต้องการแก้ไขข้อมูลต้องทำการกดปุ่มแก้ไขข้อมูลจึงจะสามารถแก้ไขข้อมูลได้

*- การลบข้อมูล เมื่อต้องการลบข้อมูลต้องทำการเลื่อนพอยน์เตอร์ไปยังเรคคอร์ดที่ต้องการลบที่ (โดยการค้นหาจากช่องค้นหาข้อมูล) หลังจากนั้นก็ทำการกดปุ่มลบข้อมูล โปรแกรมก็จะทำการลบข้อมูลออกจากฐานข้อมูล¹

- สำหรับการค้นหาข้อมูล เราสามารถค้นหาข้อมูลได้จากรหัสเอกสาร ดังแสดงในรูปที่



รูปที่ 4.11 กราฟแสดงการสรุปการเบิกจ่ายงบประมาณ

- กราฟ เป็นส่วนที่ใช้แสดงการสรุปการเบิกจ่ายงบประมาณของแต่ละหน่วยงาน ซึ่งจะแสดงงบประมาณที่ได้มา รวมเงินประมาณ รวมเงินจ่ายจริง เงินคงเหลือ ถ้าต้องการแสดงกราฟขึ้นมาให้กดปุ่มกราฟ ดังแสดงในรูปที่ 4.11

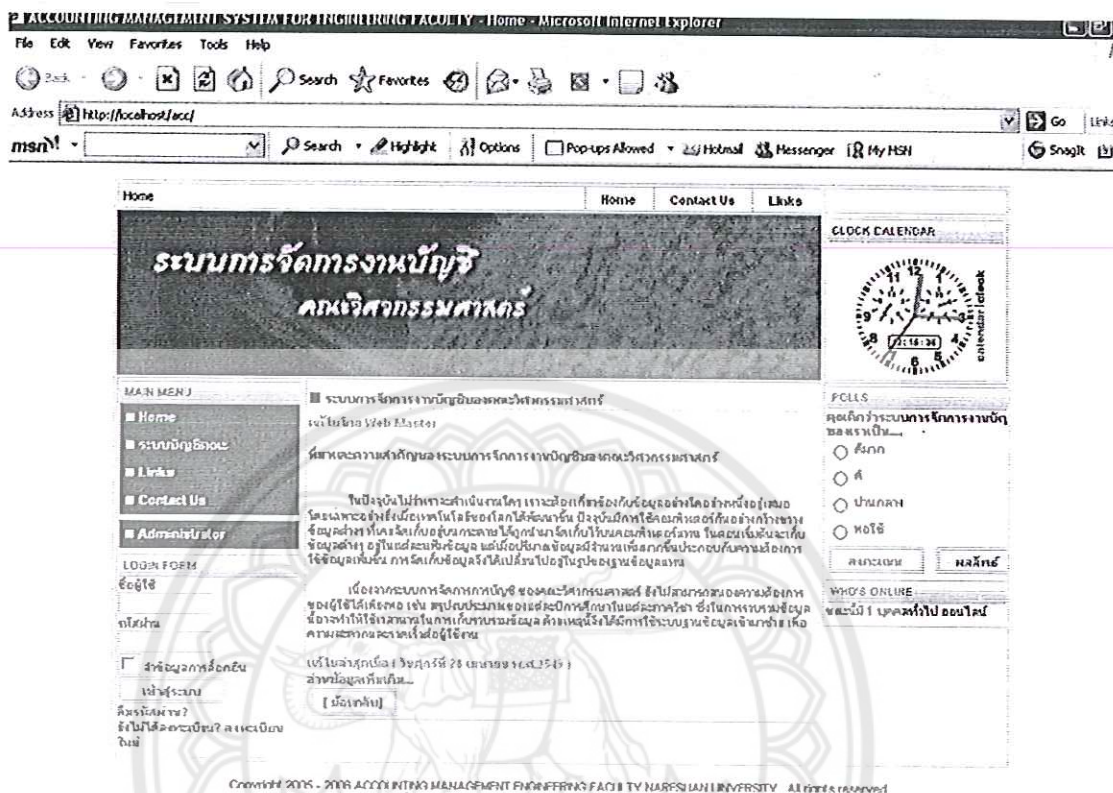
The figure shows a screenshot of a software application displaying a table of financial data. The table is titled 'รายละเอียดการใช้งบประมาณของคณะวิศวกรรมศาสตร์' and 'แสดงคงเหลือยอด 5'. The table has columns for 'ประเภทบัญชี', 'จำนวน', 'ปีงบประมาณ', 'ปีบัญชี', and 'จำนวน'. The data is organized into two columns.

ประเภทบัญชี	จำนวน	ปีงบประมาณ	ปีบัญชี	จำนวน
เงินสด	20110000	ปีงบประมาณ	ปีบัญชี	2000
เงินสด	12	ปีบัญชี	ปีบัญชี	2000
เงินสด		ปีบัญชี	ปีบัญชี	
เงินสด	1	ปีบัญชี	ปีบัญชี	ปีบัญชี

รูปที่ 4.12 แสดงการพิมพ์รายละเอียดการขอเบิกเงินงบประมาณ

- พิมพ์ แสดงรายละเอียดของการเบิกงบประมาณออกมาในรูปแบบเอกสารถ้าต้องการตั้งพิมพ์ให้กดปุ่มพิมพ์ ดังแสดงในรูปที่ 4.12

4.3 ทดสอบโปรแกรม Web Application



รูปที่ 4.13 แสดงหน้าแรกของเว็บเพจ

4.3.1 หน้าแรกเข้าโปรแกรม

จากรูปที่ 4.13 หน้านี้จะรวมข้อมูลที่ต้องแสดงให้ผู้ชมได้เห็นเวลาที่เข้าเยี่ยมชมเว็บไซต์ จะรวมข้อมูลหลักในเว็บเพจต่างๆ ดังนี้ ด้านบนของเว็บเพจจะเป็นส่วนที่เป็นข้อมูลหลักดังนี้ Home Contact Us Links ด้านซ้ายมือของเว็บเพจจะเป็นส่วนของ MAIN MENU แบ่งออกเป็น Home ระบบบัญชีคณะ Links Contact Us และ Administrator ถัดลงมาจะเป็นส่วนของ LOGIN FORM แบ่งออกเป็น ชื่อผู้ใช้ และรหัสผ่าน ตรงกลางของเว็บเพจจะเป็นส่วนของที่มา และความสำคัญของระบบการจัดการงานบัญชี ของคณะวิศวกรรมศาสตร์ วัตถุประสงค์ของระบบการจัดการงานบัญชี ของคณะวิศวกรรมศาสตร์ ด้านขวามือด้านบนของเว็บเพจจะเป็นในส่วนของนาฬิกา ปฏิทิน ถัดลงมาจะเป็น POLLS และด้านล่างจะเป็นส่วนของ WHO'S ONLINE

4.3.2 LINKS

เป็นเมนูหนึ่งที่น่ามาใช้สำหรับเลือก Category Web และเลือก URL เพื่อเข้าชมเว็บภาควิชาต่างๆ ของคณะวิศวกรรมศาสตร์ได้ง่ายขึ้น โดยไม่ต้องพิมพ์ URL ในส่วนหน้านี้มีดังนี้ สำนักงานเลขานุการคณะฯ ภาควิชาวิศวกรรมศาสตร์ ซึ่งก็จะมีแยกภาควิชาอีก 5 ภาควิชา มีดังนี้ โครงการ

ปริญญาตรีภาคพิเศษ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ ภาควิชาวิศวกรรมเครื่องกล
ภาควิชาวิศวกรรมอุตสาหกรรมและภาควิชาวิศวกรรมโยธา ดังแสดงไว้ในรูปที่4.14



รูปที่ 4.14 แสดง LINKS ในส่วนของ MAIN MENU

4.3.3 Contact Us

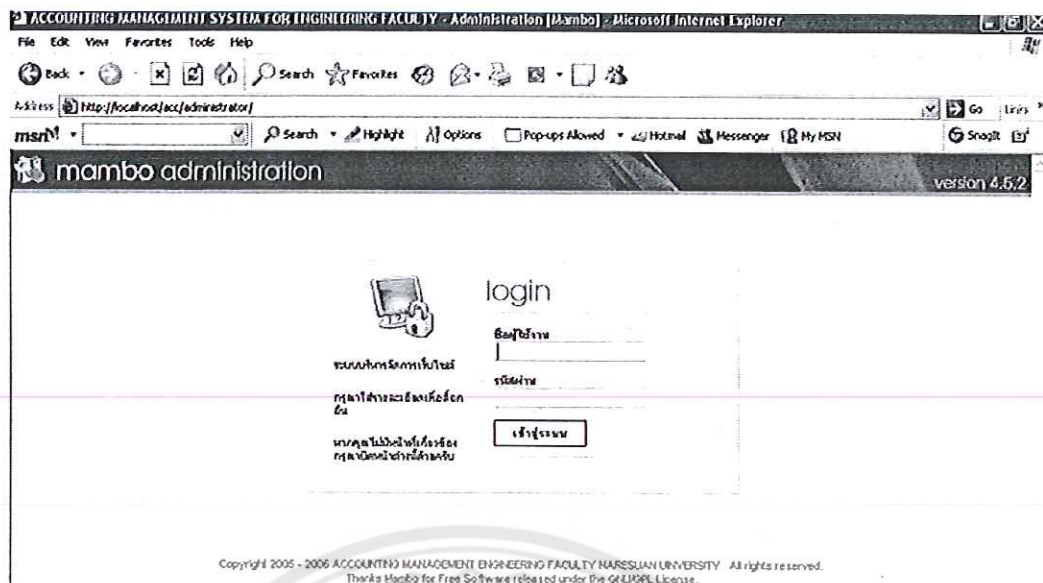
เป็นเมนูหนึ่งที่ใช้สำหรับผู้ที่ต้องการติดต่อเจ้าหน้าที่ ที่เกี่ยวกับระบบการจัดการงานบัญชี
ดังแสดงไว้ในรูปที่4.15



รูปที่ 4.15 แสดง Contact Us ในส่วนของ MAIN MENU

4.3.4 Administrator

เป็นเมนูหนึ่งที่มีไว้ให้ผู้ดูแลเว็บไซต์เข้าบริหารจัดการเว็บไซต์ข้อมูลต่างๆ โดยที่ผู้ดูแลเว็บไซต์นั้น ต้องกรอกข้อมูลในการ LOGIN เพื่อความปลอดภัยของระบบ ดังแสดงไว้ในรูปที่4.16



รูปที่ 4.16 แสดง Administrator ในส่วนของ MAIN MENU

4.3.5 LOGIN FORM

เป็นเมนูหนึ่งที่นำมาใช้ สำหรับให้ผู้ที่เข้ามาชมเว็บไซต์ ที่อยากจะสมัครเป็นสมาชิกของเว็บไซต์ ซึ่งมีไว้เพื่อให้ผู้ดูแลระบบได้รู้ว่า มีผู้สนใจเว็บไซต์ ระบบการจัดการงานบัญชีคณะวิศวกรรมศาสตร์มากน้อยเพียงใด ซึ่งเป็นผลให้ผู้ดูแลระบบ สามารถพัฒนาเว็บไซต์ให้มีประสิทธิภาพมากยิ่งขึ้น

4.3.6 POLL

เป็นเมนูหนึ่งที่นำมาใช้ เป็นแบบสำรวจความคิดเห็น ของผู้ที่เข้ามาชมเว็บไซต์ระบบการจัดการงานบัญชีคณะวิศวกรรมศาสตร์ รายละเอียดในส่วนนี้จะมีคำถาม และมีตัวเลือกให้คลิกว่ามีความเห็นอย่างไร

4.4 ทดสอบโปรแกรม PHP ติดต่อกับฐานข้อมูล

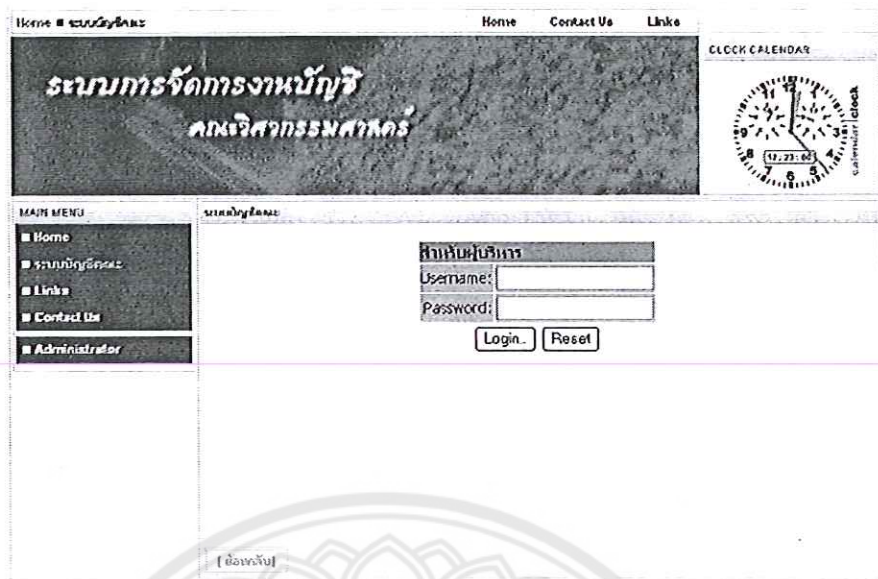
ในส่วนนี้จะเป็นโปรแกรมภาษา PHP ติดต่อกับฐานข้อมูล MYSQL ซึ่งจะเป็นฐานข้อมูลระบบการจัดการงานบัญชี ของคณะวิศวกรรมศาสตร์



รูปที่ 4.17 แสดงหน้าเข้าระบบบัญชีคณะ

4.4.1 ผู้บริหาร

ในส่วนนี้จะกำหนด Username และ Password ของผู้บริหารในฐานข้อมูลตั้งแต่แรกแล้ว เพื่อความสะดวกของผู้ใช้ในกรณีที่มีการเปลี่ยนตำแหน่งผู้บริหาร และผู้บริหารสามารถขอ Username และ Password ได้ที่ผู้ดูแลระบบ ซึ่งผู้บริหารสามารถเข้าระบบ โดยกรอก Username และ Password ที่ถูกต้อง แล้วกดปุ่ม Login ก็สามารถเข้าระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์ได้ ดังแสดงไว้ในรูปที่4.18



Copyright 2005 - 2006 ACCOUNTING MANAGEMENT ENGINEERING FACULTY NARESUAN UNIVERSITY . All rights reserved.
 Thanks Mantis for Free Software released under the GNU GPL License.

รูปที่ 4.18 แสดงหน้า Login ระบบบัญชีคณะฯ สำหรับผู้บริหาร

ซึ่งหลังจากที่ผู้บริหาร Login เข้าสู่ระบบแล้ว ในส่วนของผู้บริหารนี้ สามารถที่จะเข้ามาดูยอดสรุปเงินงบประมาณคณะวิศวกรรมศาสตร์ได้ โดยสามารถดูข้อมูลของ สำนักเลขานุการคณะฯ และดูข้อมูลได้ทั้ง 5 ภาควิชา ดังนี้ ภาควิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมอุตสาหกรรม ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ และ โครงการปริญญาตรีภาคพิเศษ โดยที่ในแต่ละภาควิชา นั้นจะมีกองทุนต่างๆ ดังแสดงไว้ในรูปที่ 4.19



Copyright 2005 - 2006 ACCOUNTING MANAGEMENT ENGINEERING FACULTY NARESUAN UNIVERSITY . All rights reserved.
 Thanks Mantis for Free Software released under the GNU GPL License.

รูปที่ 4.19 แสดงหน้าแรกของยอดสรุปเงินงบประมาณของคณะวิศวกรรมศาสตร์

ในกรณีของผู้บริหารนี้ สามารถดูได้ทั้งในส่วนของสรุปเงินงบประมาณของแต่ละกองทุน และสามารถดูรายละเอียดต่าง ๆ ในแต่ละกองทุนได้ ดังแสดงไว้ในรูปที่ 4.20 และ รูปที่ 4.21

กองทุนเพื่อการศึกษาสำนักวิทยบริการคณะ	
งบประมาณค่าใช้จ่าย แบ่งออกเป็นส่วนตัวต่าง ดังรายละเอียดต่อไปนี้	
หมวดค่าจ้างชั่วคราว	
เงินเดือน	6000 บาท
รายเดือน	8000 บาท
หมวดค่าตอบแทน	
ค่าตอบแทน	
ค่าทำการนอกเวลา	7000 บาท
ค่าเบี้ยประชุม	6000 บาท
ค่าตอบแทนตำแหน่งบริหาร	6000 บาท
ค่าตอบแทนผู้สอน	6000 บาท
สาขาขาดแคลน	6000 บาท
ค่าตอบแทนนิสิตช่วยงาน	6000 บาท
ค่าตรวจกระดาษคำตอบ	6000 บาท
ค่าสอนเกินภาระงาน	5000 บาท
ค่าตอบแทนรายวิชาออกคณะ07	6000 บาท
ค่าตอบแทนรายวิชาออกคณะ10	6000 บาท
ค่าใช้สอย	
ค่าเดินทางไปราชการ	6000 บาท

รูปที่ 4.20 แสดงหน้าสรุปเงินงบประมาณในแต่ละกองทุน

รหัสเอกสาร	เลขที่เบิก	รายละเอียด	วันที่สั่ง	รายละเอียด		จำนวน	สถานที่	รหัสเอกสาร	รหัสหมวด	เอกสารอ้างอิง	หมายเหตุ
				ประเภทการ	วันที่รับจริง						
905	2	jk j jk	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	1	01010102so	h ghj	gh gy	
906	2	jk h kh k	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	2	01010102so	klk	kl	
907	2	jk f j k j	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	3	01010102so	ty yu	ty uty	
908	2	kl gh gh h	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	4	01010102so	uy vu	h j hg	
8745645	1	d f gh df	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	5	01010102so	rey tre	ret	
965656	2	hg j Stu	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	6	01010102so	fgh th	fgh g	

First Previous Next Last

รูปที่ 4.21 แสดงหน้ารายละเอียดต่าง ๆ ของกองทุน และหมวดต่างๆ

4.4.2 หัวหน้าภาควิชา

ในส่วนนี้จะกำหนด Username และ Password ของหัวหน้าภาควิชาในฐานข้อมูลตั้งแต่แรกแล้วเพื่อความสะดวกของผู้ใช้ในกรณีที่มีการเปลี่ยนตำแหน่งหัวหน้าภาควิชา และหัวหน้าภาควิชาคณะสามารถขอ Username และ Password ได้ที่ผู้ดูแลระบบ ซึ่งหัวหน้าภาควิชาสามารถเข้าระบบโดยกรอก Username และ Password ที่ถูกต้อง แล้วคลิกปุ่ม Login ก็จะสามารถเข้าระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์ได้ ดังแสดงไว้ในรูปที่ 4.22

Home ■ ระบบบัญชีคณะ Home Contact Us Links

ระบบการจัดการงานบัญชี
คณะวิศวกรรมศาสตร์

CLOCK CALENDAR

MAIN MENU

- Home
- ระบบบัญชีคณะ
- Links
- Contact Us
- Administrator

ระบบบัญชีคณะ

สำหรับหน้าภาควิชา

Username:

Password:

[ย้อนกลับ]

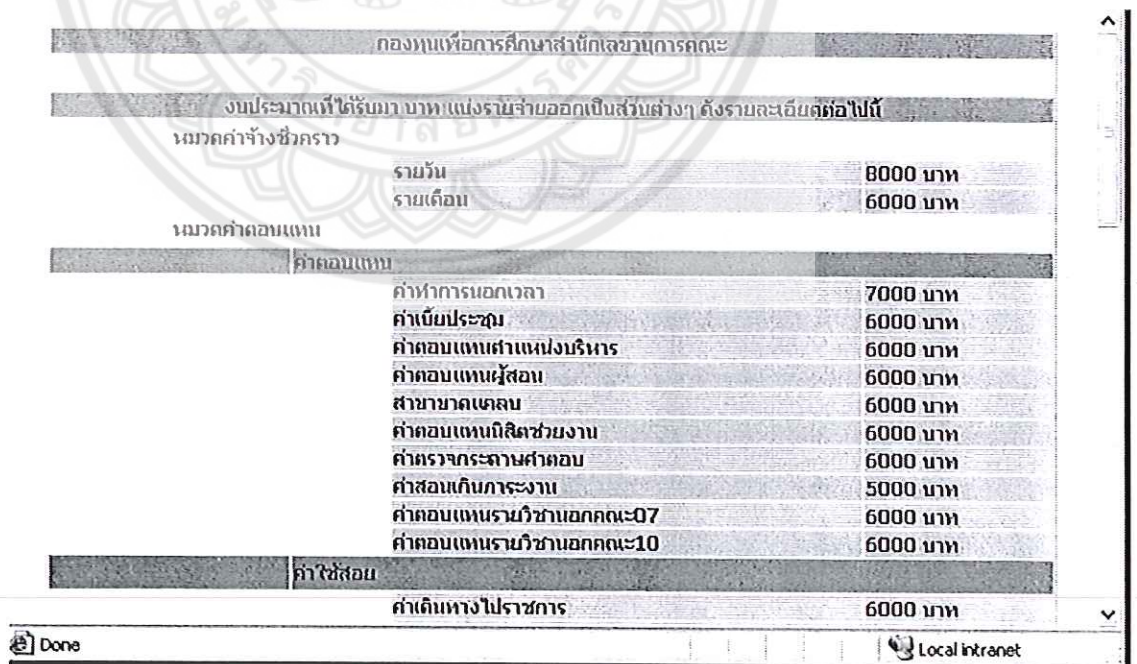
รูปที่ 4.22 แสดงหน้า Login ระบบบัญชีคณะฯ สำหรับหัวหน้าภาควิชา

ซึ่งหลังจากที่หัวหน้าภาควิชา Login เข้าสู่ระบบแล้ว ในส่วนของหัวหน้าภาควิชานี้ สามารถที่จะเข้ามาดูยอดสรุปรายงบประมาณคณะวิศวกรรมศาสตร์ได้ โดยสามารถดูข้อมูลของ สำนักเลขานุการคณะฯ และดูข้อมูลได้ทั้ง 5 ภาควิชา ดังนี้ ภาควิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ และโครงการปริญญาตรีภาคพิเศษ โดยที่ในแต่ละภาควิชา นั้นจะมีกองทุนต่างๆ ดังแสดงไว้ในรูปที่ 4.23



รูปที่ 4.23 แสดงหน้าแรกของยอดสรุปเงินงบประมาณของคณะวิศวกรรมศาสตร์

ในกรณีของหัวหน้าภาควิชาฯ สามารถดูได้เฉพาะในส่วนของสรุปเงินงบประมาณของแต่ละกองทุนเท่านั้น ไม่สามารถดูรายละเอียดต่าง ๆ ในแต่ละกองทุนได้ ดังแสดงไว้ในรูปที่ 4.24



รูปที่ 4.24 แสดงหน้าสรุปเงินงบประมาณในแต่ละกองทุน

4.4.3 ผู้ดูแลระบบ หรือ Admin

ในส่วนนี้จะกำหนด Username และ Password ของผู้ดูแลระบบในฐานข้อมูลตั้งแต่แรกผู้ดูแลระบบสามารถเข้าระบบโดยกรอก Username และ Password ที่ถูกต้อง แล้วกดปุ่ม Login ก็ สามารถเข้าระบบการจัดการงานบัญชีของคณะวิศวกรรมศาสตร์ได้ ดังแสดงไว้ในรูปที่ 4.25



รูปที่ 4.25 แสดงหน้า Login ระบบบัญชีคณะฯ สำหรับ Admin

ซึ่งหลังจากที่ Admin หรือผู้ดูแลระบบ Login เข้าระบบแล้ว ในส่วนของ Admin หรือผู้ดูแลระบบนี้ สามารถที่จะเข้ามาดูยอดสรุปรายเงินงบประมาณคณะวิศวกรรมศาสตร์ได้ โดยสามารถดูข้อมูลของ สำนักเลขานุการคณะฯ และดูข้อมูลได้ทั้ง 5 ภาควิชา ดังนี้ ภาควิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมอุตสาหกรรม ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ และ โครงการปริญญาตรีภาคพิเศษ โดยที่ในแต่ละภาควิชา นั้นจะมีกองทุนต่างๆ ดังแสดงไว้ในรูปที่ 4.26



รูปที่ 4.26 แสดงหน้าแรกของยอดสรุปเงินงบประมาณของคณะวิศวกรรมศาสตร์

ในกรณีของ Admin หรือผู้ดูแลระบบนี้ สามารถดูได้ทั้งในส่วนของสรุปเงินงบประมาณของแต่ละกองทุนและสามารถรายละเอียดต่างๆ ในแต่ละกองทุนได้ ดังแสดงไว้ในรูปที่ 4.27 และรูปที่ 4.28



รูปที่ 4.27 แสดงหน้าสรุปเงินงบประมาณในแต่ละกองทุน

ระบบข้อมูล										
รหัสเอกสาร	เลขที่ผู้ทำ	รายละเอียด	วันที่ส่ง	ประเภทการ	วันที่จ่ายจริง	จ่ายจริง	สถานะเอกสาร	รหัสหมวด3	เอกสารอ้างอิง	หมายเหตุ
905	2	jk jkljk	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	1	01010102so	h ghj	ghjgy
906	2	klh klhjk	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	2	01010102so	kk	kl
907	2	jk jkljk	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	3	01010102so	tytyu	tyuty
908	2	kl klhgh	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	4	01010102so	uygu	h hg
0745645	1	d ghdf	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	5	01010102so	reylre	ret
965656	2	hg u5tu	2005-04-03 00:00:00	1000	2005-04-03 00:00:00	1000	6	01010102so	fghth	fghfg

first Previous Next Last

รูปที่ 4.28 แสดงหน้ารายละเอียดต่างๆ ของกองทุน และหมวดต่างๆ

บทที่ 4 นี้จะกล่าวถึงการทำงานของโปรแกรม รูปแบบการทำงานของโปรแกรมในระบบการจัดการงานบัญชีคณะวิศวกรรมศาสตร์ และการทำงานในส่วนของ Web Application พร้อมทั้งทดสอบเพื่อดูผลของการทำงานเมื่อมีการดึงฐานข้อมูลมาใช้ร่วมกัน



บทที่ 5

สรุปผลการดำเนินการและข้อเสนอแนะ

ในบทนี้ จะขอกล่าวถึงการสรุปผลการดำเนินงานที่ได้จากการศึกษา และ ดำเนินการเพื่อทำการจัดการ ระบบบัญชีคณะวิศวกรรมศาสตร์จนได้โครงการที่ผู้พัฒนาคิดว่า มีประสิทธิภาพในการจัดเก็บข้อมูลที่เป็นระบบและมีประสิทธิภาพมากยิ่งขึ้น

5.1 สรุปผลการดำเนินโครงการ

จากการดำเนินงาน การจัดการระบบบัญชีคณะวิศวกรรมศาสตร์ มีวัตถุประสงค์เพื่อจัดทำระบบฐานข้อมูลระบบบัญชี เพื่อให้มีประสิทธิภาพมากยิ่งขึ้น ได้แก่ ตรวจสอบเกี่ยวกับข้อมูลงบประมาณและรายละเอียดการเบิกจ่ายงบประมาณ เพื่อเพิ่มประสิทธิภาพในการจัดเก็บข้อมูลของฝ่ายบัญชีคณะวิศวกรรมศาสตร์ และพัฒนาให้ฝ่ายบัญชีคณะวิศวกรรมศาสตร์มีการจัดการภายในระบบที่มีประสิทธิภาพสูงสุดซึ่งได้ผลการดำเนินการดังนี้

- สามารถตรวจสอบข้อมูลเกี่ยวกับงบประมาณ และรายละเอียดการเบิกจ่ายงบประมาณ ได้สะดวกขึ้น ทำให้การตรวจสอบและการค้นหาข้อมูลมีความรวดเร็วมากขึ้น อีกทั้งข้อมูลที่ได้นั้นยังมีความถูกต้อง

- สามารถนำข้อมูลที่ได้ ทำการจัดเก็บไว้ที่นั้นนำออกมาทำการประมวลผล เพื่อช่วยตัดสินใจทางด้านการอนุมัติงบประมาณได้

5.2 ข้อเสนอแนะ

ผู้ศึกษาโครงการนี้มีข้อเสนอแนะในการทำโครงการดังต่อไปนี้

- จากการที่ได้ศึกษาโครงการนี้ สามารถที่จะพัฒนาโปรแกรมเกี่ยวกับ การจัดการงานบัญชีคณะวิศวกรรมศาสตร์ เพื่อให้ใช้ได้กับบัญชีอื่นๆ ที่มีระบบการจัดการที่มีความคล้ายคลึงกัน โดยมีความแตกต่างไม่มากนัก อาจสามารถปรับเปลี่ยนบางส่วน เนื่องจากโปรแกรมที่ได้จัดทำขึ้นนี้ได้จากการ ศึกษาการทำงาน ของฝ่ายบัญชีของคณะวิศวกรรมศาสตร์ และ ข้อมูลต่างๆ ล้วนมาจากบัญชีคณะวิศวกรรมศาสตร์ โดยนำข้อมูลที่ได้มาประกอบการเขียนโปรแกรม เพื่อแก้ปัญหาที่เกิดขึ้นจริงและตรงตามความต้องการของผู้ใช้ระบบ

- เนื่องจากโปรแกรมที่จัดทำขึ้นนี้ มีความเหมาะสม และใช้ได้กับการทำงานการจัดการระบบบัญชีในปัจจุบันเท่านั้น ซึ่งในอนาคตหากการจัดการระบบบัญชี มีการเปลี่ยนแปลงกระบวนการทำงานซับซ้อนมากยิ่งขึ้น รวมทั้งการจัดการที่แตกต่างออกไปจากปัจจุบัน เพราะโปรแกรมที่ได้จัดทำขึ้นในครั้งนี้นั้น ทางผู้จัดทำไม่สามารถรองรับการทำงานของระบบได้ดีเท่าที่ควร ดังนั้น จึงควรที่มีการพัฒนา โปรแกรมการจัดการงานบัญชี ของคณะวิศวกรรมศาสตร์ เพื่อให้ทันสมัย สะดวก และใช้งานได้อย่างมีประสิทธิภาพอยู่ตลอดเวลา

- เนื่องจากการที่การจัดการระบบบัญชี ได้มีการนำระบบคอมพิวเตอร์เข้ามาใช้งานอย่างจริงจังเป็นครั้งแรกซึ่งถือเป็นของใหม่สำหรับบุคลากรที่ทำงานอยู่เดิมดังนั้นจึงควรที่จะมีการอบรมให้ความรู้เกี่ยวกับคอมพิวเตอร์ และ การใช้งานโปรแกรมให้กับบุคลากร เพื่อที่จะได้บุคลากรที่มีคุณภาพมากยิ่งขึ้นและจะส่งผลให้การทำงานและการใช้โปรแกรมที่จัดทำมีประสิทธิภาพมากยิ่งขึ้นตามไปด้วย

- โครงการวิศวกรรม “โปรแกรมระบบการจัดการบัญชีคณะวิศวกรรมศาสตร์” อยู่บนพื้นฐานของการออกแบบฐานข้อมูลฝ่ายบัญชีจริง โครงสร้างของฐานข้อมูลที่ออกแบบมานั้นอาจจะไม่สมบูรณ์ตามทฤษฎี โปรแกรมการจัดการระบบบัญชีคณะวิศวกรรมศาสตร์นั้นยังมีข้อจำกัด ซึ่งหากเป็นไปได้สามารถที่จะพัฒนาต่อไปให้มีประสิทธิภาพมากขึ้นได้ในอนาคต

เอกสารอ้างอิง

- [1] กิตติ ภัคดีวัฒนกุล . จำลอง ครูอุตสาหกรรม. **คัมภีร์ระบบฐานข้อมูล**. พิมพ์ครั้งที่ 6. กรุงเทพมหานคร : ไทยเจริญการพิมพ์. 2545.
- [2] ยุทธนา ลีลาศวัฒนกุล . อมรพันธุ์ คำอรธ. **สร้างระบบฐานข้อมูลด้วย PL/SQL Oracle Developer**. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร : อินโฟเพรส. 2545.
- [3] อำไพ พรประเสริฐกุล. **การวิเคราะห์และออกแบบระบบ**. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. 2543.
- [4] สมประสงค์ รัตนินิธิ. **PHP4**. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : หอสมุดแห่งชาติ. 2543.
- [5] สุรเชษฐ์ วงศ์ชัยพรพงษ์ . ทินกร วัฒนเกษมสกุล. **Web Programming DreamweaverMX 2004 และ PHP**. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร : บริษัท เคทีพี คอมพ์แอนด์ คอนซิลท์ จำกัด. 2547.
- [6] มณีโชติ สมานไทย. **การออกแบบฐานข้อมูล และภาษา SQL**. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : บริษัท Infopress Developer Book . 2546.
- [7] Alden C. Lorents and James N. Morgan. **Database System Concept Management and Applications**. The Dryden Press Harcourt Brace College Publishers : Inc. 1998.
- [8] Silberschatz,Korth. Sudarshan. **Database System Concept**. Singapore : Inc. 2002.
- [9] Project Development Team. **“What is Mambo?”** [Online]. Available : [Http://www.mamboserver.com/index.php?option=com_content&task=view&id=81&Itemid=86](http://www.mamboserver.com/index.php?option=com_content&task=view&id=81&Itemid=86)

ภาคผนวก ก

การออกแบบฐานข้อมูล

ทุกฐานข้อมูลจะต้องมีส่วนที่ใช้เก็บข้อมูลในลักษณะ Metadata ซึ่งเป็นข้อมูลที่บอกถึงลักษณะของตัวข้อมูลที่เกี่ยวข้องในฐานข้อมูล เช่น โครงสร้างของข้อมูล โครงสร้างของ Table โครงสร้างของ Index กฎที่ใช้ควบคุมความถูกต้องของข้อมูล (Integrity Rule) กฎที่ใช้ในการรักษาความปลอดภัย ให้กับข้อมูล (Security Rule) ฯลฯ ข้อมูลเหล่านี้จัดเป็นข้อมูลที่มีความจำเป็นต่อโปรแกรม DBSM ในการตัดสินใจที่จะดำเนินการใดๆ กับฐานข้อมูล เช่น ข้อมูลที่เกี่ยวข้องกับกฎที่ใช้ในการรักษาความปลอดภัยให้กับข้อมูล จะถูกนำมาใช้ในการพิจารณาให้สิทธิแก่ผู้ใช้ในการใช้งานฐานข้อมูล เป็นต้น สำหรับส่วนที่ใช้จัดเก็บข้อมูลในลักษณะของ Metadata นี้ ได้แก่ Data Dictionary หรือ Catalog

การออกแบบ Data Dictionary

หลังจากทำการศึกษาในส่วนของฐานข้อมูลแล้ว เมื่อทำการออกแบบ Database ให้ครอบคลุม Requirement และง่าย สะดวกที่สุด ซึ่ง Database จะแบ่งเป็นส่วนหลักๆ ดังนี้

1. TbPlanFirst เป็นตาราง ที่ใช้สำหรับเก็บรายละเอียด ของรหัสหมวด 1 คณะวิศวกรรมศาสตร์ซึ่งมีรายชื่อรหัสหมวด1 ดังนี้

- หมวดค่าจ้างชั่วคราว
- หมวดค่าตอบแทนใช้สอยและวัสดุ
- หมวดค่าสาธารณูปโภค
- หมวดเงินอุดหนุน
- หมวดค่าครุภัณฑ์ที่ดินและสิ่งก่อสร้าง

ตารางที่ ก-1 ตารางพจนานุกรมข้อมูลของตาราง TbPlanFirst

Field	Type	Length	Description	Remark
PIFstID	Text	4	รหัสหมวด1	Primary Key
PIFstName	Text	50	ชื่อหมวด1	
PIMainID	Text		รหัสกองทุน/แผนงาน/ งาน	Foreign Key

จากตารางที่ ก-1 แสดงรายละเอียด Field, Type, Length, Description, Remark การเก็บข้อมูลในตาราง TbPlanFirst

2. **TbPlanSecond** เป็นตารางเกี่ยวกับหมวด2 ที่ใช้สำหรับเก็บรายละเอียด ของรหัสหมวด 2 คณะวิศวกรรมศาสตร์ ซึ่งมีรายชื่อรหัสหมวด 2 ดังนี้

- หมวดค่าจ้างชั่วคราว
 - ค่าจ้างชั่วคราว
- หมวดค่าตอบแทนใช้สอยและวัสดุ
 - ค่าตอบแทน
 - ค่าใช้สอย
 - ค่าวัสดุ
- หมวดค่าสาธารณูปโภค
 - ค่าไฟฟ้า
 - ค่าโทรศัพท์
- หมวดเงินอุดหนุน
 - อุดหนุนทั่วไป
- หมวดค่าครุภัณฑ์ที่ดินและสิ่งก่อสร้าง
 - ค่าครุภัณฑ์
 - ค่าสิ่งก่อสร้าง

ตารางที่ ก-2 ตารางพจนานุกรมข้อมูลของตาราง TbPlanSecond

Field	Type	Length	Description	Remark
PlSndD	Text	6	รหัสหมวด2	Primary Key
PlSndName	Text	50	ชื่อหมวด2	
PlFstID	Text		รหัสหมวด1	Foreign Key

จากตารางที่ ก-2 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbPlanSecond

3. **TbGroupDoc** เป็นตาราง ที่เกี่ยวกับฎีกา ซึ่งจะใช้สำหรับ การที่จะเก็บรายละเอียดฎีกา ของคณะวิศวกรรมศาสตร์ ซึ่งฎีกามีรายละเอียด ดังนี้

- เลขที่ฎีกา

ตารางที่ ก-3 ตารางพจนานุกรมข้อมูลของตาราง TbGroupDoc

Field	Type	Length	Description	Remark
GroupID	Text	4	รหัสผู้ึก	Primary Key
GroupDate	Date		วันที่ตั้งผู้ึก	
PLSndD	Text	6	รหัสหมวด2	Foreign Key
DepartID	Text	2	รหัสหน่วยงาน	Foreign Key

จากตารางที่ ก-3 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbGroupDoc

4. TbProjectDoc เป็นตารางเกี่ยวกับรายละเอียด ที่ใช้สำหรับเก็บรายละเอียดเอกสาร ในแต่ละโครงการ (หมวดเงินอุดหนุน) ของคณะวิศวกรรมศาสตร์ ซึ่งมีเอกสารในแต่ละโครงการมีรายละเอียด ดังนี้

ตารางที่ ก-4 ตารางพจนานุกรมข้อมูลของตาราง TbProjectDoc

Field	Type	Length	Description	Remark
ProjectDocID	Text	4	รหัสเอกสาร โครงการ	Primary Key
ProjectDocDetail	Text	50	รายละเอียด	Primary Key
Budget	Currency		จำนวนเงิน	
DocID	Text		รหัสเอกสาร	Foreign Key

จากตารางที่ ก-4 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbGroupDoc

5. TbDepartBudget เกิดจาก TbDepartment + TbPlanFirst เป็นตารางที่ใช้สำหรับเก็บรายละเอียดของจำนวนเงินตามหมวดต่างๆ ของคณะวิศวกรรมศาสตร์ ซึ่งมีรายละเอียด ดังนี้

ตารางที่ ก-5 ตารางพจนานุกรมข้อมูลของตาราง TbDepartBudget

Field	Type	Length	Description	Remark
DepartID	Text	2	รหัสหน่วยงาน	Primary Key
PIFstID	Text	4	รหัสหมวดเงิน1	Primary Key
PIFstBudget	Currency		จำนวนเงินหมวด1 (ที่ยัง ไม่มีการเพิ่มลด)	
AddBudget	Currency		จำนวนเงินที่รับโอน	ต้องบวกเพิ่ม ยอดเงินหมวด1
LessBudget	Currency		จำนวนเงินที่โอนไป	ต้องตัดยอดเงิน หมวด1

จากตารางที่ ก-5 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbDepartBudget

6. TbTransferMain การโอนเงิน เป็นตารางที่ใช้สำหรับเก็บรายละเอียด ของการ โอนเงิน ตามหมวดต่างๆ ของคณะวิศวกรรมศาสตร์ ซึ่งมีรายละเอียด ดังนี้

ตารางที่ ก-6 ตารางพจนานุกรมข้อมูลของตาราง TbTransferMain

Field	Type	Length	Description	Remark
TransferID	Text	2	รหัสโอน	Primary Key
TransferDate	Date		วันที่โอน	
TransferDetail	Text	50	รายละเอียดการ โอนเงิน	
TransferBudget	Currency		จำนวนเงินหมวด1	

จากตารางที่ ก-6 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbTransferMain

7. TbTransferFrom เป็นตาราง ที่ใช้สำหรับเก็บรายละเอียด ของการโอนเงิน ไปตาม หมวดต่างๆ ของคณะวิศวกรรมศาสตร์ที่ต้องตัดยอดลง ซึ่งมีรายละเอียด ดังนี้

ตารางที่ ก-7 ตารางพจนานุกรมข้อมูลของตาราง TbTransferFrom

Field	Type	Length	Description	Remark
TransferID	Text	2	รหัสโอน	Primary Key
FromDepartID	Text	2	รหัสหน่วยงาน (DepartID)	หน่วยงานที่โอน เงินไปต้องตัดยอด ลดลง Foreign Key
FromPIFstID	Text	4	รหัสหมวดเงิน 1 (PIFstID)	หมวดเงินของ หน่วยงานที่โอน เงินไปต้องตัดยอด ลดลง Foreign Key

จากตารางที่ ก-7 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbTransferFrom

8. TbTransferTo เป็นตารางที่ใช้สำหรับเก็บรายละเอียด ของการรับโอนเงินตามหมวดต่างๆ ของคณะวิศวกรรมศาสตร์เพื่อเพิ่มยอดขึ้น ซึ่งมีรายละเอียด ดังนี้

ตารางที่ ก-8 ตารางพจนานุกรมข้อมูลของตาราง TbTransferTo

Field	Type	Length	Description	Remark
TransferID	Text	2	รหัสโอน	Primary Key
ToDepartID	Text	2	รหัสหน่วยงาน (DepartID)	หน่วยงานที่รับโอน เงินยอดต้องเพิ่มขึ้น Foreign Key
ToPIFstID	Text	4	รหัสหมวดเงิน 1 (PIFstID)	หมวดเงินของ หน่วยงานที่รับ โอน เงินเงินยอดต้อง เพิ่มขึ้น Foreign Key

จากตารางที่ ก-8 แสดงรายละเอียด Field, Type, Length, Description, Remark ของการเก็บข้อมูลในตาราง TbTransferTo

จากข้อมูล Data Dictionary ที่กล่าวจะทำให้ทราบถึงการออกแบบของตารางแต่ละตารางเพื่อนำไปประยุกต์ใช้ในการออกแบบโปรแกรมฐานข้อมูลระบบการจัดการงานบัญชีคณะวิศวกรรมศาสตร์ต่อไป



ภาคผนวก ข

การติดตั้งเว็บเซิร์ฟเวอร์

การติดตั้งเว็บเซิร์ฟเวอร์ที่สนับสนุนภาษา PHP จึงจะทำให้ประมวลผลเว็บแอปพลิเคชัน และแสดงผลในรูปแบบ HTML Page บนเว็บเบราว์เซอร์ได้ โดยเว็บเซิร์ฟเวอร์ที่สนับสนุน PHP มีหลายผลิตภัณฑ์ เช่น Apache, IIS, SUN ONE เป็นต้น โดยในที่นี้จะแสดงตัวอย่างการใช้ Apache โดยให้รันบน WMServer

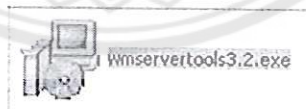
การติดตั้งโปรแกรม เป็น Web Server บน Windows XP

ใน WMServer Tools 3.2 จะมี

- PHP compiler สำหรับการ คอมไพล์ PHP
- MySQL สำหรับงานฐานข้อมูล
- Apache สำหรับเป็นเว็บ Server

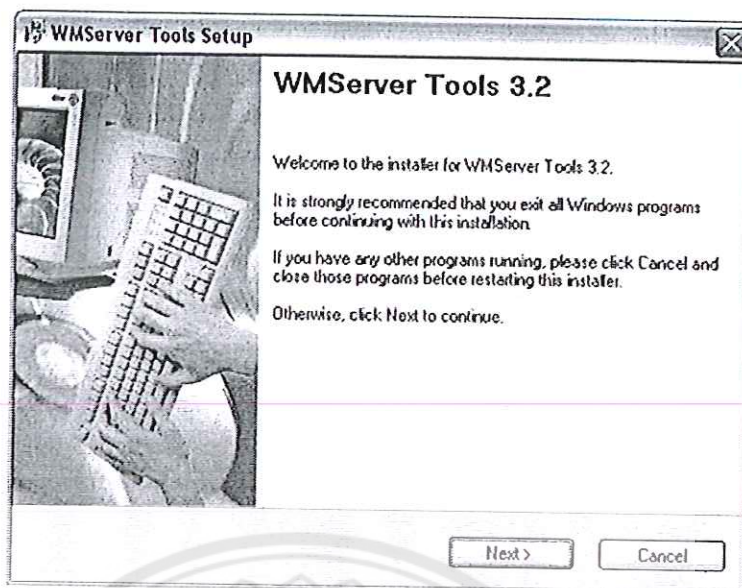
เหตุที่ใช้ตัวนี้เพราะ WMServer Tools นั้นสามารถเรียกให้โปรแกรมนี้ทำงานเมื่อใดก็ได้ แตกต่างกับ Appserv เพราะจะรันตอนที่เราเปิดเครื่องมา ทำให้เครื่องทำงานหนัก โดยที่เราไม่ได้ใช้ Appserv หากมี Appserv บนเครื่องให้ Uninstall ออกก่อน เพราะมันจะติดกัน หากลง IIS เพื่อรัน ASP ก็ให้ยกเลิก และ Uninstall ออกด้วย ซึ่งการเขียนเว็บด้วย PHP นั้น ต้องมีตัวคอมไพล์ ภาษา PHP ก่อน โดยที่ใช้ software ที่ชื่อว่า WMServer Tools 3.2 ซึ่งมีวิธีการติดตั้ง โปรแกรมดังนี้

1. เปิดไฟล์โปรแกรม Setup ดังรูปที่ ข- 1 แล้วดับเบิลคลิกขึ้นมาเพื่อทำการติดตั้ง



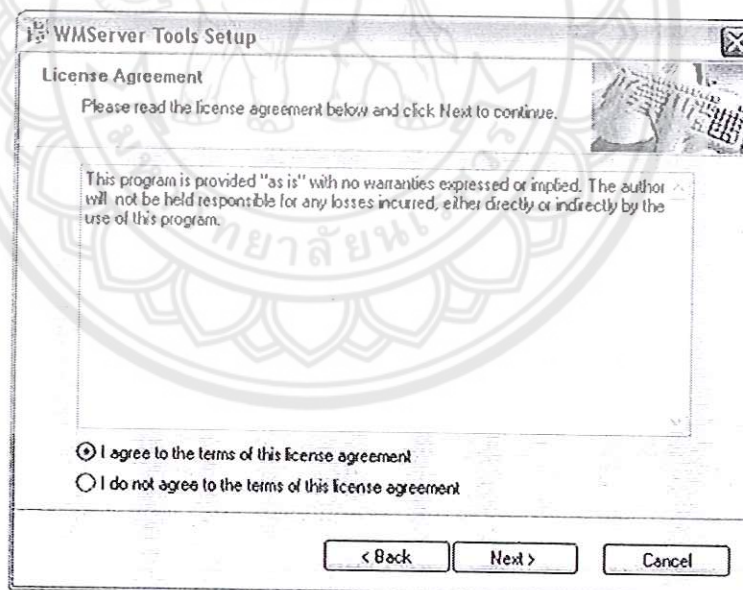
รูปที่ ข-1 ตัว Setup โปรแกรม wmserver

2. เมื่อเข้ามาหน้าต่างนี้ เป็นหน้าต่างการต้อนรับก่อนการติดตั้ง โปรแกรม ให้เราคลิก Next ดังรูปที่ ข-2

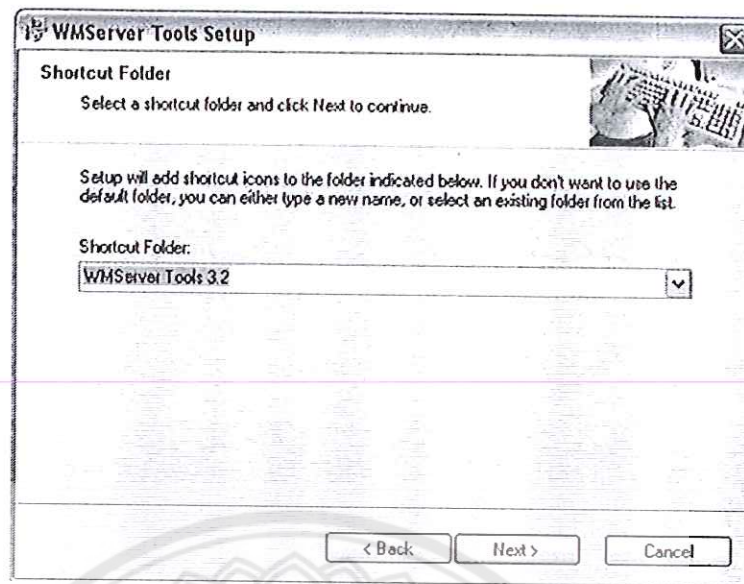


รูปที่ ข-2 หน้า setup โปรแกรม (1)

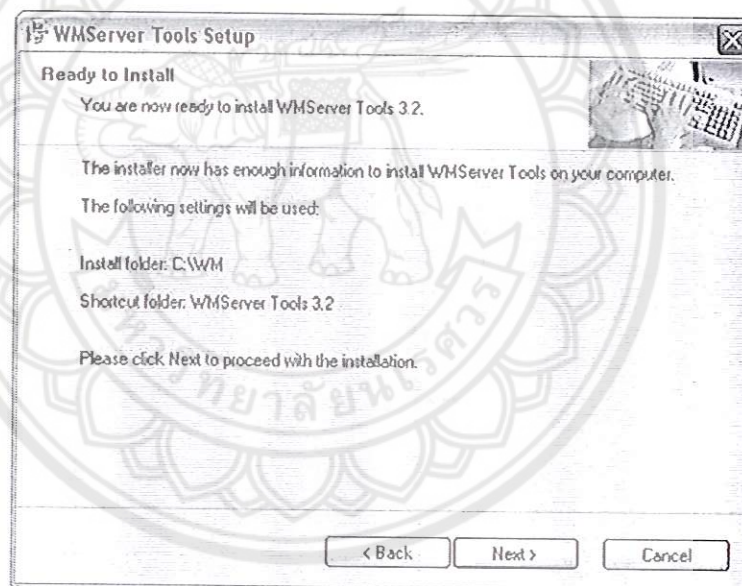
3. อ่าน License ของโปรแกรม เมื่อตกลงแล้ว ให้เลือก I agree to ... แล้วทำการคลิกที่ Next ไปเรื่อย ๆ ดังรูปที่ ข-3, ข-4, ข-5 และข-6



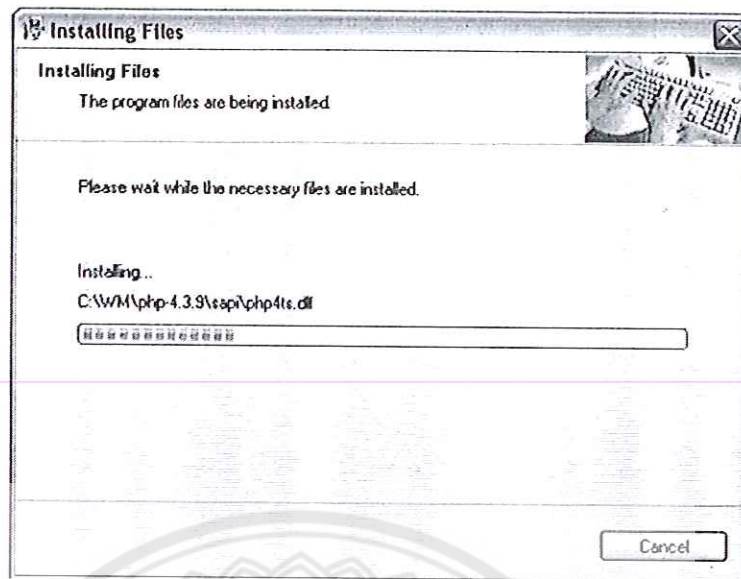
รูปที่ ข-3 หน้า setup โปรแกรม (2)



รูปที่ ข-4 หน้า setup โปรแกรม (3)

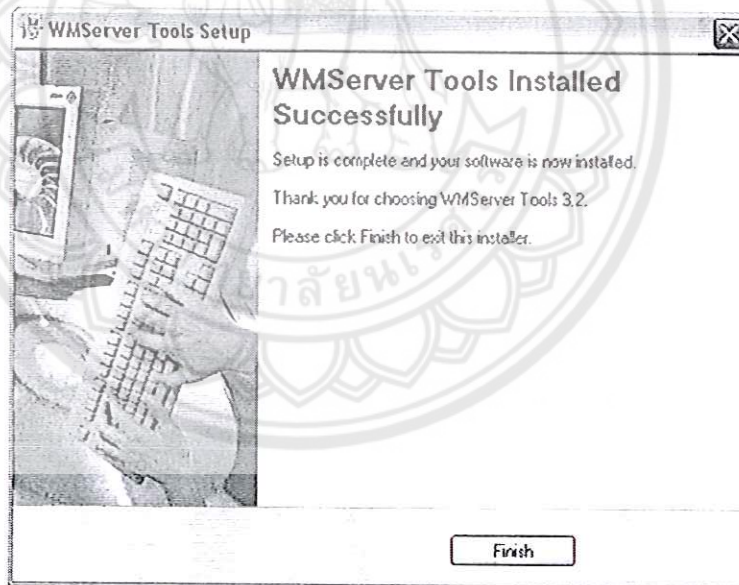


รูปที่ ข-5 หน้า setup โปรแกรม (4)



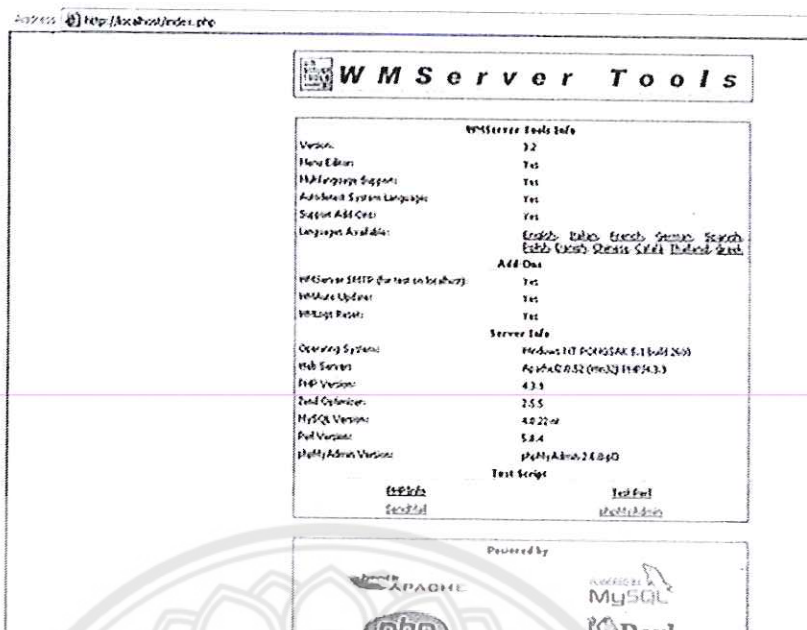
รูปที่ ข-6 หน้า setup โปรแกรม (5)

4. เมื่อติดตั้งเสร็จแล้ว ให้คลิกที่ Finish ดังรูปที่ ข-7



รูปที่ ข-7 หน้าตาโปรแกรมเมื่อทำการ setup เสร็จ

5. เมื่อคลิกที่ Finish แล้ว โปรแกรมจะทำการเรียก Web Browser เปิดหน้าแรกของโปรแกรม WMServer Tools รันขึ้นมาทดสอบ เพื่อจะแสดงให้เห็นว่าโปรแกรม Web Server ได้ทำงานแล้ว ดังรูปที่ ข-8



รูปที่ ข-8 หน้าของโปรแกรม Web Server ที่ทำงานแล้ว

6. เมื่อติดตั้งโปรแกรมแล้ว จะมีสถานะของโปรแกรม WMServer Tools รั้นอยู่ที่ Task bar มุมล่างขวามือ ดังรูปที่ ข-9



รูปที่ ข-9 ตัวไอคอน wmserver รั้นอยู่ที่ Task bar

7. ให้คลิกที่รูปสี่เหลี่ยมสีขาว สัญลักษณ์ของโปรแกรม WMServer Tools เพื่อดูเมนูต่าง ๆ ของโปรแกรม ซึ่งเมื่อคลิกแล้วจะเกิดเมนูขึ้นมา ดังรูปที่ ข-10



รูปที่ ข-10 เมนูของ โปรแกรม WMServer

8. โดยเมื่อเราเริ่มเปิดเครื่องขึ้นมา ถ้า WMServer ยังไม่ทำงาน ให้ทำการดับเบิลคลิกที่ Shoutcut โปรแกรม WMServer ที่ Desktop เพื่อทำการ Start โปรแกรมก่อนการใช้งาน ดังแสดงตามรูปที่ ข-11



รูปที่ ข-11 รูปไอคอน Wmserver Tools 3.2

9. เมื่อเปิดโปรแกรมขึ้นมาแล้ว ให้ลองเอา Mouse วางที่ Task bar ตรงรูปโปรแกรม WMServer ดู มันจะมีข้อความขึ้นมาว่า Apache และ MySQL สถานะตอนนี้เป็นยังไง ถ้าเป็นดังรูปที่ ข- 12 แสดงว่า WMServer กำลังรันอยู่ สามารถใช้งานโปรแกรมได้เลย



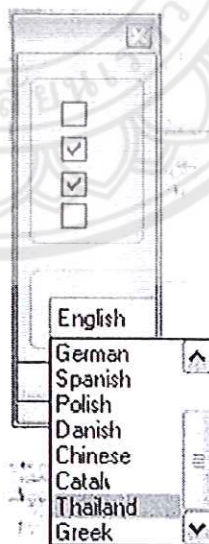
รูปที่ ข-12 Wmserver Online

10. การตั้งค่าให้โปรแกรม WMServer โดยคลิกที่ Icon WMServer ที่ Task bar แล้วทำการเลือก ที่ Setting ดังรูปที่ ข-13



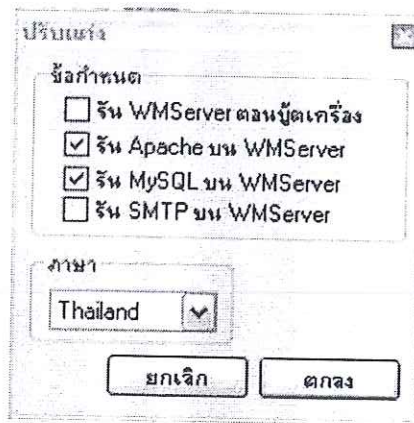
รูปที่ ข-13 เมนูการตั้งค่าโปรแกรม

11. ให้เราทำการตั้งค่าเมนูเป็นภาษาไทย โดยเลือกที่ List Box >> Thailand ดังรูปที่ ข-14



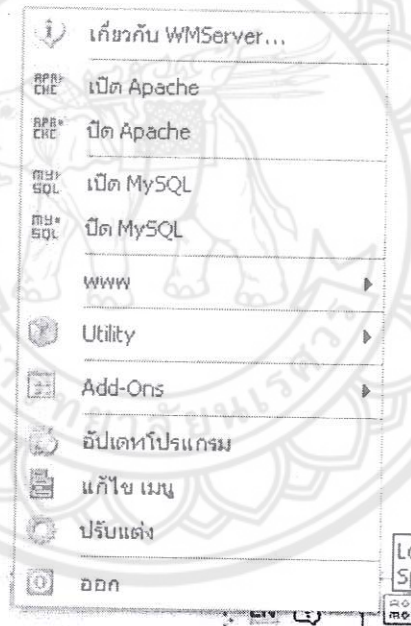
รูปที่ ข-14 เมนูการตั้งค่าให้เป็นภาษาไทย (1)

12. เมื่อเลือกเป็น Thailand เสร็จแล้ว โปรแกรมจะแสดงผลได้อย่างสมบูรณ์ ดังรูปที่ ข- 15 ซึ่งเราสามารถตั้งค่าต่าง ๆ ได้ โดยการทำเครื่องหมายเลือก Option ที่ต้องการแล้วคลิก ตกลง



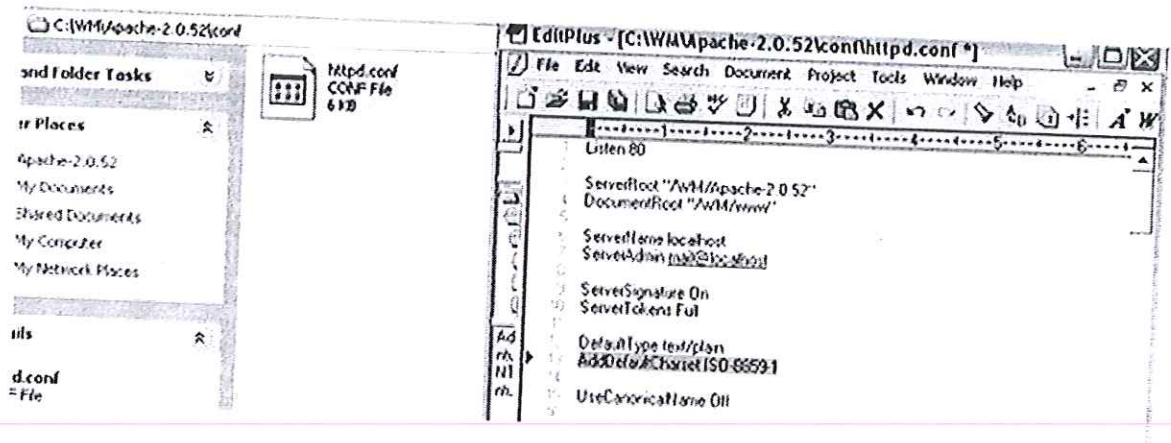
รูปที่ ข-15 เมนูการตั้งค่าให้เป็นภาษาไทย (2)

13. เมื่อคลิก ตกลง แล้วโปรแกรมและเมนูทั้งหมดจะโชว์เป็นภาษาไทย ดังรูปที่ ข-16



รูปที่ ข-16 แสดงเมนูเป็นภาษาไทย

14. จากนั้นก็ทำการแก้ไขให้ระบบ WMServer นี้ไม่มีปัญหาเรื่องการแสดงผลภาษาไทย โดยให้เข้าไปที่ C:/WM/Apache 2.0.52/conf/ แล้วทำการเปิดไฟล์ httpd.conf ด้วย text editor ขึ้นมา แล้วแก้ไขที่บรรทัดที่ 13 เพื่อแก้ไข Default type text/plain ของ web server โดยค่าปกติที่กำหนดมาให้คือ AddDefaultCharset ISO-8859-1 ซึ่งมันจะไม่ Support ภาษาไทย ดังรูปที่ ข-17



รูปที่ ข-17 การกำหนดค่าใน Text Editor (1)

15. ให้แก้ไขและเพิ่มลงไปแทนที่ AddDefaultCharset ISO-8859-1 ของเดิมที่มีอยู่ ให้ Support ภาษาไทย ซึ่งคำสั่งที่เพิ่มลงไปคือ AddDefaultCharset Windows-874 และ AddDefaultCharset TIS-620 จากนั้นทำการบันทึกไฟล์ ดังรูปที่ ข-18

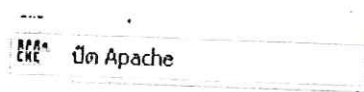
```

5  ServerAdmin mail@localhost
6  ServerAdmin mail@localhost
7  ServerAdmin mail@localhost
8  ServerAdmin mail@localhost
9  ServerSignature On
10 ServerTokens Full
11 ServerTokens Full
12 DefaultType text/plain
13 AddDefaultCharset windows-874
14 AddDefaultCharset tis-620
15 AddDefaultCharset tis-620
16 UseCanonicalName Off
17 UseCanonicalName Off
18 HostnameLookups Off
19 HostnameLookups Off
20 ErrorLog logs/error.log
21 LogLevel error
22 LogLevel error
23 PidFile logs/httpd.pid
24 PidFile logs/httpd.pid
25 Timeout 300
26 Timeout 300

```

รูปที่ ข-18 การกำหนดค่าใน Text Editor (2)

16. จากนั้นให้ทำการ Stop service ของ Apache โดยไปที่ Icon WMServer ที่ Task bar แล้วเลือก ปิด Apache ดังรูปที่ ข-19



รูปที่ ข-19 แสดงการ stop service ของ apache

17. จากนั้นให้ลองเอา Mouse ไปวางที่ Icon WMServer ดูสถานะว่า Apache Offline หรือยัง ถ้าได้แล้วจะปรากฏดังรูปที่ ข-20



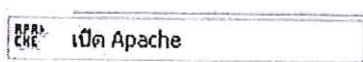
รูปที่ ข-20 MySQL Online และ Apache Offline

18. เพื่อความแน่ใจ ให้ลองเปิดเว็บเบราว์เซอร์ แล้วพิมพ์ที่ address ว่า <http://localhost> หรือ <http://127.0.0.1> ถ้าปรากฏหน้าเว็บเพจดังรูปที่ ข- 21 แสดงว่า Apache ไม่ได้ทำงาน



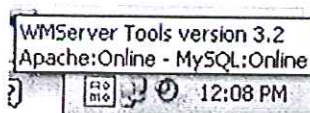
รูปที่ ข-21 แสดงหน้าจอของเว็บเบราว์เซอร์ ที่ Apache ไม่ได้ทำงาน

19. จากนั้นให้คลิกที่ Start Service ของ Apache โดยไปที่ Icon WMServer ตรง Task bar แล้วเลือกเปิด Apache เป็นอันว่าเสร็จการอัปเดตค่าคอนฟิกที่ได้ทำการแก้ไขเรื่องภาษาไทยของ Apache เรียบร้อยแล้ว ดังรูปที่ ข-22



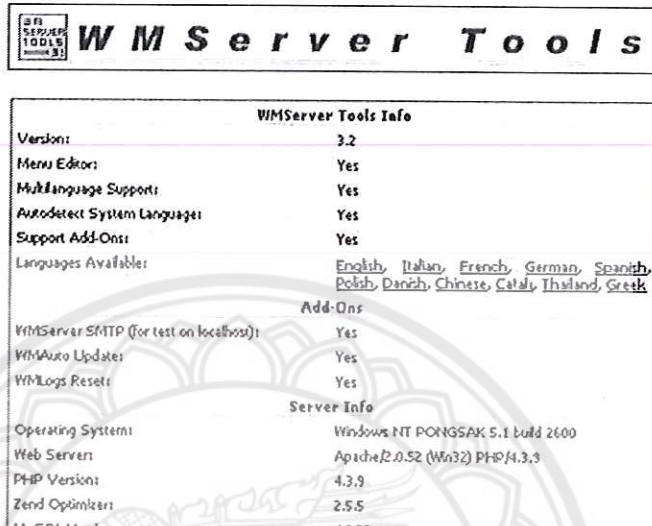
รูปที่ ข-22 Start Service ของ Apache

20. เพื่อความแน่ใจอีกครั้ง ลองเอา Mouse วางดูที่ Task bar อีกครั้งหนึ่งว่า Apache นั้น Online หรือยัง ถ้าใช้ได้ก็จะปรากฏดังรูปที่ ข-23



รูปที่ ข-23 MySQL Online และ Apache Online

21. จากนั้น ก็ให้ลองเปิดเว็บเบราว์เซอร์ โดยให้ไปพิมพ์ตรงที่ address ว่า <http://localhost> หรือ <http://127.0.0.1> ถ้าปรากฏหน้าเว็บเพจดังรูปที่ ข- 24 แสดงว่าโปรแกรม Web Server ทำงานเหมือนเดิมตามปกติ



รูปที่ ข-24 แสดงหน้าของเว็บเบราว์เซอร์ ที่ Apache ทำงาน

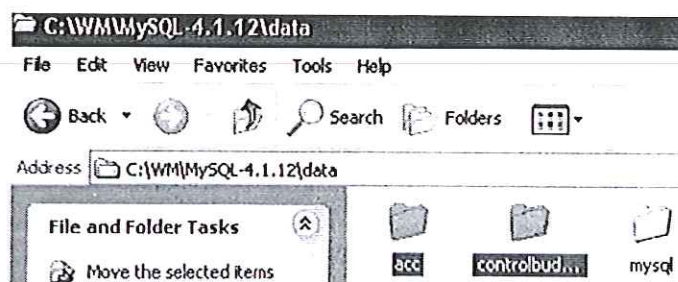
เมื่อทำการ Setup โปรแกรม ระบบการจัดการงานบัญชี ของคณะวิศวกรรมศาสตร์ เรียบร้อยแล้ว ให้ทำตามขั้นตอน ดังนี้

- ลง โปรแกรม Setup WMServer Tools 3.3 ที่มีมาให้ ถ้าเครื่องของคุณยังไม่มี
- ลง โปรแกรม Setup MyODBC ที่มีมาให้ ถ้าเครื่องของคุณยังไม่มี
- ลงโปรแกรม Setup MYSQL-Front ที่มีมาให้ ถ้าเครื่องของคุณยังไม่มี

เมื่อลงทุกอย่างเสร็จเรียบร้อยแล้ว ก็จะเป็นขั้นตอนในการเชื่อมฐานข้อมูล เพื่อให้สามารถ ใช้โปรแกรมระบบการจัดการงานบัญชีได้ โดยทำตามขั้นตอน ดังนี้

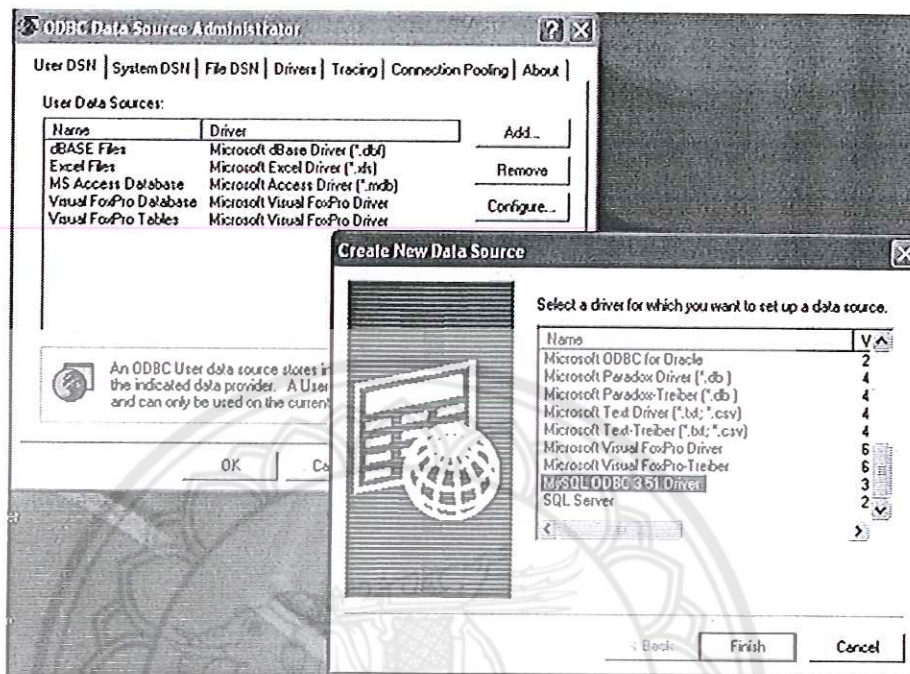
ในที่นี้ ฐานข้อมูลเรา คือ ControlBudget48 และ acc

- ทำการ Copy File ชื่อ ControlBudget48 และ acc ที่มีมาให้ ไปวางที่ Drive C-> WM-> MySQL-4.1.12 -> data ดังรูปที่ ข- 25



รูปที่ ข-25 ทำการ Copy File ชื่อ ControlBudget48 และ acc

- จากนั้นไปที่ Start -> Control Panel-> Administrative Tools-> Data Sources (ODBC)
- ที่ ODBC Data Source Administrator ให้เลือก User DSN จะปรากฏดังรูปที่ ข-26



รูปที่ ข-26 แสดงการกำหนดค่าใน ODBC

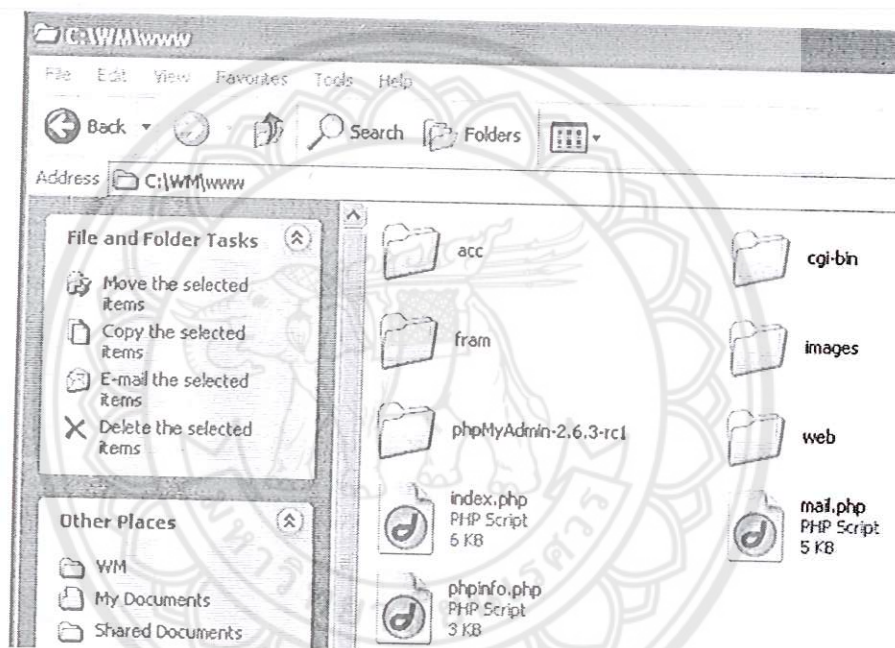
- ไปที่ Add-> MYSQL ODBC 3.5 Driver ดังรูปที่ ข- 26แล้วคลิก Finish >> ตกลง
 - จะมีช่อง Data Source Name ให้กรอก heemu
 - และในช่อง Database Name ให้กรอก ControlBudget48
- เพียงเท่านั้น Admin หรือผู้ดูแลระบบ สามารถที่จะใช้งานโปรแกรม ระบบการจัดการงานบัญชีคณะวิศวกรรมศาสตร์ ได้แล้ว

ภาคผนวก ค

รูปแบบการติดตั้งในส่วนของ Web Application

ในที่นี้จะเป็นการสร้าง Web Application โดยเลือกใช้ภาษาสคริปต์ PHP และเลือกภาษาที่จัดการกับฐานข้อมูลด้วย MySQL โดยทำงานร่วมกับเว็บเซิร์ฟเวอร์ที่เป็น Apache การติดตั้งโปรแกรมในส่วนของเว็บเพื่อนำไปใช้งานมีขั้นตอน ดังนี้

1. Copy File ที่มีชื่อว่า acc และ web ที่มีมาให้ไปไว้ใน C:\WM\www ดังรูปที่ ค-1



รูปที่ ค-1 ทำการ Copy Folder acc และ web

2. เปิดโปรแกรม WMServer เพื่อให้ Apache ทำงาน และ เพื่อให้เว็บเซิร์ฟเวอร์ประมวลผลได้เว็บเพจได้

เพียงเท่านี้ผู้บริหาร หัวหน้าภาคฯ และ Admin ก็สามารใช้งานในส่วนของ Web Application ได้ ซึ่งสามารถเปิดดูข้อมูลเกี่ยวกับระบบการจัดการงานบัญชี ของคณะวิศวกรรมศาสตร์ บนเว็บไซต์ได้

ประวัติผู้เขียนโครงการ



ชื่อ นางสาวนุชจิรา ใจตุรงค์
ภูมิลำเนา 200/33 หมู่ 5 ตำบลท่าวังทอง อำเภอเมือง จังหวัดพะเยา
56000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพะเยาพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร

e-mail : nick_17_5@hotmail.com



ชื่อ นางสาวแคทริยา พิศปิงคำ
ภูมิลำเนา 89 หมู่ 8 ตำบลท่าผา อำเภอกะลา จังหวัดลำปาง
52130

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนลำปางกัลยาณี
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร

e-mail: pae_e@hotmail.com