ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การพัฒนาระบบสารสนเทศในงานบริการผู้ป่วยของโรงพยาบาล

Information System Development for Patient's Medical Service of General Hospital

โดย

นาวาโทบุญเรือง เกิดอรุณเดช ร.น.

รหัส 41067068

วัน เดือน ปี 09 ส.ค. 2550
เลขทะเบียน 01757
เลขเรียกหนังสือ วพ. บ548 ก 2543
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร.ศุภมิตร จิตตะยโศธร

H001757

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาระบบสารสนเทศในงานบริการผู้ป่วยของโรงพยาบาล
นักศึกษา	นาวาโทบุญเรือง เกิดอรุณเดช ร.น.
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร. ศุภมิตร จิตตะยโศธร
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

โรงพยาบาลเป็นสถานประกอบการที่มีจุดประสงค์หลักในการบริการแก่ผู้ป่วย ซึ่งการบริการนั้นต้องประกอบด้วยข้อมูลที่ถูกต้องและรวดเร็ว เครื่องมือสำคัญที่ใช้สร้างประสิทธิภาพนี้ คือ การพัฒนาระบบสารสนเทศ ประกอบด้วยการศึกษาระบบจัดการในส่วนให้บริการต่างๆ ภายในโรงพยาบาล นับตั้งแต่ผู้ป่วยเข้าการรักษาจนออกจากโรงพยาบาล การวิเคราะห์และการออกแบบระบบที่ดีและสอดคล้องกับระบบงานเดิมเป็นกลไกหลักในการพัฒนาโดยเน้นที่ความถูกต้องและรวดเร็ว ระบบสารสนเทศที่ได้จะสามารถใช้ทดแทนงานเดิมและลดภาระงานบริการที่ต้องอาศัยการสืบค้นข้อมูลด้วยมือลงเป็นอย่างมาก และยังได้รับผลพลอยได้ในเทคนิคการแก้ปัญหา Concurrency Control ในส่วนการใช้งานฐานข้อมูลแบบ File Server ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Information System Development for Patient's Medical Service of General Hospital
Student	Cdr. Boonruang Kerdaroondej RTN.
Advisor	Asso.Prof. Dr. Suppamit Jitayasotorn.
Level of Stduy	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

The purpose of general hospital is to service the medical care and treatment for patients. To be more accurate and faster activity are the main objective of service and issue from System Development Life Cycle (SDLC) .The system covers Medical Record,Screening, Treatment, Disease Diagnosis, Patient's Cost and Reporting. It is replaced of the old system which uses batch processing and save budgets by using X – BASE on File Sever. This System will show the technique to solve the problems of Concurreney Control of X – Base too.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I

บทคัดย่อภาษาอังกฤษ..... II

สารบัญ..... III

สารบัญตาราง..... IV

สารบัญภาพ..... V

บทที่

1. บทนำ..... ...1

1.1 ความเป็นมา..... 1

1.2 วัตถุประสงค์ของการพัฒนาระบบ..... 2

1.3 ขั้นตอนการดำเนินการโครงการ..... 2

1.4 เป้าหมายของโครงการ..... 3

2. ทฤษฎีที่เกี่ยวข้องกับการใช้ระบบ..... 4

2.1 วัฏจักรของการพัฒนาระบบ (System Development Life Cycle : SDLC).....4

2.2 Structured Methodlogies..... . 5

2.3 Data Flow Diagram (DFD)..... 6

2.4 ฐานข้อมูล..... 8

2.5 การนอร์มัลไลซ์.....22

2.6 Concurrency Control..... 26

3. การพัฒนาระบบและโปรแกรม.....32

3.1 ฮาร์ดแวร์และเน็ตเวิร์ค (Hardware and Network).....32

3.2 ซอฟต์แวร์ (Software).....32

3.3 Concurrency Control Implementation..... 33

3.4 การออกแบบส่วนสัมพันธ์กับผู้ใช้.....37

3.5 การพัฒนาโปรแกรม..... 40

4. บทสรุป.....59

บรรณานุกรม.....60

ประวัติผู้เขียน.....61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 แสดงสัญลักษณ์ต่างๆ ที่ใช้ในการเขียน DFD 7
2.2 ตัวอย่างฐานข้อมูลแบบโมเดลเชิงสัมพันธ์ 12
2.3 รีเลชั่น R1 14
2.4 รีเลชั่น R2 14
2.5 รีเลชั่น R 15
2.6 ตัวอย่างความสัมพันธ์ระหว่างนักเรียนกับผู้ปกครอง 21
2.7 ตัวอย่างความสัมพันธ์ระหว่างผู้ป่วยกับรหัสโรค 21
2.8 ตัวอย่างความสัมพันธ์ระหว่างนักเรียนกับผู้ปกครอง 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

ภาพที่

2.1 แสดงระดับของข้อมูล 3 ระดับ.....9

2.2 Schema ในฐานข้อมูล 3 ระดับ.....11

2.3 โครงสร้างของระบบฐานข้อมูล.....18

2.4 แสดงการ Lock Table บนฐานข้อมูล Timestamp T และ U.....27

2.5 แสดงการขัดแย้งของการ Read/Write บน Transaction 28

2.6 แสดงการ Lock ของ Transaction ใดๆ.....28

2.7 Validation of Transaction.....29

2.8 Transaction Conflicts for Timestamp Ordering.....31

3.1 แสดงปัญหาการใช้ Concurrency Control บน X-BASE.....35

3.2 แสดงการ Lock ไฟล์ โดยใช้ Timeouts.....36

3.3 แสดงการเพิ่มประสิทธิภาพของ Concurrency Control บน X-BASE.....37

3.4 แสดง Menu Structure.....39

3.5 แสดงหน้าจอการเข้าสู่ระบบการใช้งาน.....40

3.6 แสดงรายการในเมนูหลัก.....41

3.7 แสดงรายการในเมนูหลัก.....42

3.8 หน้าจอการบันทึกข้อมูล.....43

3.9 หน้าจอการทำงานลงทะเบียนผู้ป่วยนอก/พิมพ์ใบสั่งยา.....44

3.10 เปลี่ยนรายละเอียด/ยกเลิกใบสั่งยา.....45

3.11 คืนประวัติผู้ป่วยนอก.....46

3.12 ลงทะเบียนผู้ป่วยใน.....47

3.13 น้าจอการบันทึกค่าใช้จ่าย.....48

3.14 หน้าการคืนค่าใช้จ่าย.....49

3.15 หน้าจอการบันทึกผลตรวจ Lab,-X-Ray.....50

3.16 หน้าจอการบันผลการวินิจฉัยโรค.....51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

3.17 หน้าจอรายชื่อผู้ป่วยนอกประจำวัน.....52
3.18 หน้าจอรายชื่อผู้ป่วยในประจำวัน.....53
3.19 หน้าจอการบันทึกนัดหมายผู้ป่วย.....54
3.20 หน้าจอการบันทึกการทำค้างชำระ.....55
3.21 หน้าจอการพิมพ์ใบเสร็จรับเงิน.....56
3.22 หน้าจอรายงานการสรุปแยกรหัสบริการ.....57
3.23 หน้าจอรายงานการเงิน.....58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

โรงพยาบาลเป็นสถานประกอบการที่ให้บริการด้านการดูแลรักษาผู้ป่วย ด้วยวิทยาการและความก้าวหน้าทางการแพทย์ อันได้แก่ เครื่องมือตรวจและรักษาที่เป็นมาตรฐานเดียวกันทั่วโลก รวมทั้งทีมแพทย์และพยาบาลที่ได้รับการศึกษา อบรมเฉพาะทางในสาขาการรักษาพยาบาลนั้น ๆ แม้ว่าความเจริญก้าวหน้าในกระบวนการรักษาจะดีขึ้นเพียงใดก็ตาม แต่ระบบข้อมูลผู้ป่วยยังคงบันทึกไว้บนกระดาษ และยังคงใช้การสืบค้นข้อมูลด้วยมือ ซึ่งต้องสูญเสียเวลาและสถานที่จัดเก็บเป็นอย่างมาก อีกทั้งยังประสบปัญหาความถูกต้องของการจัดเก็บและการสูญหายของเอกสารด้วย

คณะผู้บริหารโรงพยาบาล ตระหนักในปัญหาและเห็นความสำคัญของการจัดเก็บข้อมูลตลอดจนการสืบค้นข้อมูลผู้ป่วยเป็นอย่างดี ได้พยายามที่จะแก้ไขปัญหานี้ โดยปฏิบัติเป็น 4 แนวทาง คือ

1.การปรับปรุงระบบเอกสารเดิมให้รัดกุมและสืบค้นได้ง่ายขึ้น โดยใช้ทรัพยากรเดิมและนำเครื่องคอมพิวเตอร์มาช่วยในการสืบค้นและเก็บข้อมูลบางส่วน ซึ่งต้องผสมผสานระบบเดิมกับเครื่องคอมพิวเตอร์ การดำเนินการนี้เป็นเพียงการนำเครื่องคอมพิวเตอร์มาช่วยเหลือเหมือนเครื่องพิมพ์ดีดและเก็บข้อมูลแบบตู้เอกสารเท่านั้น ระบบต่าง ๆ ที่เกี่ยวข้องยังคงใช้บุคลากรเช่นเดิม

2.การพัฒนาระบบสารสนเทศภายในองค์กร เป็นการจ้างบุคลากรด้านระบบคอมพิวเตอร์ไว้ภายในองค์กรมีหน้าที่พัฒนาระบบสารสนเทศใช้ในกิจการของหน่วย ระบบสารสนเทศที่ได้ จะขึ้นอยู่กับมุมมองของผู้บริหารและความเชี่ยวชาญของนักวิเคราะห์ระบบเป็นหลัก

3.การจัดซื้อโปรแกรมสำเร็จรูป ในระบบสารสนเทศที่ต้องการ เพื่อติดตั้งและใช้งานภายในหน่วยโดยตรวจสอบและศึกษาวิธีการใช้จากผู้ขาย และมอบหมายบุคคลหรือคณะบุคคลภายในองค์กรทดสอบและใช้งานให้เป็นไปตามเป้าหมาย

4.การจัดจ้างหน่วยงานหรือผู้เชี่ยวชาญภายนอก เป็นผู้พัฒนาระบบสารสนเทศให้ภายใต้กฎเกณฑ์และการดำเนินงานขององค์กร โดยองค์กรจัดเจ้าหน้าที่ประสานงานและอำนวยความสะดวก

ทั้ง4 แนวทางเป็นรูปแบบที่โรงพยาบาลในประเทศไทยเลือกใช้ภายใต้ข้อกำหนดและขอบเขตแตกต่างกัน ปัจจัยที่มีผลอย่างยิ่งต่อการตัดสินใจของคณะผู้บริหารโรงพยาบาล คือ เวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และงบประมาณ

เวลา เป็นปัจจัยที่แสดงการเร่งด่วนของกิจกรรมที่ได้รับการพิจารณา หากเป็นกิจกรรมที่มีเวลาจำกัดการเลือกใช้แนวทางจัดซื้อโปรแกรมสำเร็จรูปจะมีความเหมาะสมเป็นอย่างยิ่ง รองลงไปจะเป็นแนวทางที่ 1 ,4 และ 2 ตามลำดับ

งบประมาณ เป็นปัจจัยที่สำคัญมากในเชิงธุรกิจ เพราะเป็นตัวกำหนดต้นทุนและราคาค่าบริการ สำหรับหน่วยงานที่มีงบประมาณจำกัด อาจทำให้ทางเลือกลดน้อยลง แต่ถึงอย่างไรก็ดี ไม่ได้หมายความว่าจะได้ระบบที่ดี ถ้างบประมาณไม่จำกัด

การพิจารณาทางเลือกภายใต้งบประมาณที่เหมาะสมและเสร็จเรียบร้อยตามเวลาที่กำหนดเป็นเรื่องที่ต้องวิเคราะห์อย่างละเอียดและถี่ถ้วน การดำเนินการในแนวทางที่ 1 แม้ว่าจะเป็นหนทางที่ประหยัดที่สุดและอาจจะเหมาะสมกับเวลา แต่งานที่ได้เป็นเพียงการแก้ปัญหาเฉพาะหน้า ไม่ใช่การสร้างระบบสารสนเทศเหมือนในแนวทางที่ 2,3 และ 4 ประเด็นของปัญหาจึงอยู่ที่ว่า จะพัฒนาระบบสารสนเทศอย่างไรให้เหมาะสมกับเวลาและงบประมาณ

หลักการของระบบสารสนเทศที่พัฒนาขึ้นนี้ ต้องการสร้างระบบคอมพิวเตอร์ที่ใช้งบประมาณต่ำและสามารถทำงานได้เทียบเท่าระบบคอมพิวเตอร์ที่ใช้งบประมาณสูง โดยพยายามเสนอแนวคิดและเทคนิคการแก้ปัญหาที่เป็นจุดอ่อนของระบบ File Server ใน Concurreney Control ระบบฐานข้อมูลและองค์ประกอบของโปรแกรมอื่น ๆ ที่ทำให้สามารถประยุกต์ใช้โปรแกรมและฐานข้อมูลนี้กับงานขนาดใหญ่ที่ต้องการความเชื่อถือของข้อมูลในระดับสูงได้

1.2 วัตถุประสงค์ของการพัฒนาระบบ

1.2.1 เพื่อเสนอแนวคิดและเทคนิคในการนำระบบคอมพิวเตอร์แบบ File Server มาใช้ในระบบโปรแกรมที่ใช้ฐานข้อมูลขนาดใหญ่ และการแก้ปัญหา Concurreney Control ของระบบฐานข้อมูล

1.2.2 เพื่อพัฒนาระบบสารสนเทศต้นแบบในการบริการผู้ป่วยของโรงพยาบาลทั่วไป

1.2.3 เพื่อพัฒนาระบบบริการผู้ป่วยให้ถูกต้องและรวดเร็ว

1.3 ขั้นตอนการดำเนินการโครงการ

1.3.1 ศึกษาระบบงานเดิม

1.3.2 ศึกษาถึงความเป็นไปได้ในการพัฒนาระบบ

1.3.3 วิเคราะห์และออกแบบระบบ

1.3.4 ศึกษาการพัฒนาระบบงานบนเครื่อง PC ภายใต้ระบบ LAN (Local Area Network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้ Novell Netware หรือ Window NT เป็น Operating System, FOXBASE/ FOXPRO เป็นระบบจัดการฐานข้อมูลและMicrosoft FOXPRO เป็น Application Developer

1.3.5 เขียนโปรแกรมและทดสอบระบบ

1.4 เป้าหมายของโครงการ

1.4.1 ระบบงานใหม่ต้องมีความรวดเร็วและถูกต้องในการให้บริการผู้ป่วย และง่ายต่อการปฏิบัติงาน

1.4.2 ได้ระบบที่สามารถตอบสนองต่อความต้องการของผู้ใช้ได้อย่างรวดเร็ว

1.4.3 สามารถสืบค้นข้อมูลสารสนเทศผู้ป่วยได้รวดเร็วยิ่งขึ้น

1.4.4 ได้บทสรุปการแก้ปัญหา Concurreney Control ในการทำงานแบบ File Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้องกับการใช้ระบบ

ในการพัฒนาระบบสารสนเทศสำหรับงานบริการผู้ป่วยของโรงพยาบาลได้นำทฤษฎีในการออกแบบต่างๆ มาช่วยในการทำงาน ซึ่งมีรายละเอียดดังต่อไปนี้

2.1 วัฏจักรของการพัฒนาระบบ (System Development Life Cycle : SDLC)

SDLC Methodology ที่ใช้ในการวิเคราะห์พัฒนาโครงสร้างของระบบ ซึ่งได้มีการรวมหลักการไว้เป็น 3 Phase ดังนี้

Phase 1 : การวิเคราะห์ระบบ (System Analysis) เป็นระบบปัจจุบัน

Phase 2 : รายละเอียดในการวิเคราะห์และออกแบบ (Detail Analysis and Design)

Phase 3 : การสร้างระบบ (Implementation)

ภายในแต่ละ Phase จะประกอบไปด้วยขั้นตอนต่างๆดังนี้

Phase ที่ 1 : System Analysis ใน Phase นี้จะประกอบไปด้วย 4 ขั้นตอนคือ

ขั้นตอนที่ 1 : การกำหนดปัญหา (Problem Definition) เป็นการกำหนดขอบเขตของปัญหาสถานการณ์ต่างๆและจุดมุ่งหมาย ซึ่งสิ่งที่จะได้จากขั้นนี้คือ

-การรวบรวมและการตรวจสอบ วัตถุประสงค์และเป้าหมาย

-อธิบายปัญหา

-Context Diagram

-กำหนดขอบเขตการศึกษา

-เริ่มทำ Data Dictionary

ขั้นตอนที่ 2 : วางแผนการศึกษาระบบ (Plan for Study) ซึ่งเป็นขั้นตอนของการวางแผนงานว่างานที่จะต้องทำมีอะไรบ้าง และเป็นอย่างไร

ขั้นตอนที่ 3 : ข้อมูลทั่วไปภายใต้ขอบเขตที่ศึกษาและความสัมพันธ์ในข้อมูลนั้นเริ่มจากการมองภาพกว้างๆของสิ่งแวดล้อมที่ต้องการศึกษา และพยายามที่จะวิเคราะห์ข้อมูลที่มีอยู่ ภายในนั้นออกมา (General Information and Its Interactive)

ขั้นตอนที่ 4 : ทำความเข้าใจในระบบปัจจุบัน (Understanding the Existing System) Phase ที่ 2: Detail Analysis and Design ใน Phase นี้ประกอบด้วย 4 ขั้นตอน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์จะต้องพยายามที่จะเข้าใจว่า ข้อมูลอะไรที่ User ต้องการที่จะใช้ในการทำงาน โดยข้อมูลทั้งหมดในระบบนี้เราจะเรียกว่า Specification ซึ่งในขั้นตอนนี้จะเป็นการวิเคราะห์โครงสร้างที่ใช้ในการตัดสินใจ จำพวก ข้อจำกัด ข้อกำหนด และกฎต่างๆ จะเป็น ขั้นตอนการเตรียมเสนอ ข้อสรุปของระบบ

ขั้นตอนที่ 6 : การออกแบบระบบใหม่ (Designing New System) ขั้นตอนการออกแบบนี้เป็นการรวบรวมข้อมูลต่างๆ เพื่อที่จะนำข้อมูลเหล่านี้ไปออกแบบเพื่อให้ระบบข้อมูลมีความถูกต้องที่สุด

ขั้นตอนที่ 7 : การออกแบบควบคุมระบบใหม่ (Design New System Controls)

ขั้นตอนที่ 8 : การเปรียบเทียบค่า (Economic Cost Comparison) จะเป็นการวิเคราะห์ค่าและประโยชน์ของทรัพยากรของระบบ

ขั้นตอนที่ 9 : การนำเสนอระบบ (Selling System) ขั้นตอนนี้จะเป็นขั้นตอนของการนำเสนอ ผลงานที่เราได้กระทำการออกแบบและวิเคราะห์

Phase ที่ 3 : Implementation ใน Phase นี้มี 1 ขั้นตอน คือ

ขั้นตอนที่ 10 : การทำระบบใหม่หลังจากที่ได้ทำการวิเคราะห์และออกแบบแล้ว ในขั้นตอนนี้จะมีการทำโปรแกรม (Coding) การทดสอบโปรแกรม การบำรุงรักษาระบบและการฝึกอบรมการใช้ระบบต่อไป

2.2 Structured Methodologies

Structured Methodologies เป็นวิธีการสร้างระบบสารสนเทศที่แสดงให้เห็นถึงการออกแบบที่มีขั้นตอนตามลำดับอย่างเหมาะสม,ถูกวิธี และมีเอกสารประกอบที่เหมาะสม

- ความจำเป็นในการใช้ Structured Methodologies

การใช้ Structured Methodologies นั้นทำให้เกิดการพัฒนาทางด้านการสื่อสารระหว่างนักวิเคราะห์ด้วยกันเองและระหว่างนักวิเคราะห์กับผู้ใช้ เมื่อมีการนำ Structured Methodologies มาใช้อย่างเหมาะสมจะทำให้เกิดระบบที่มีการผิดพลาดน้อยที่สุดและยังสามารถผลิตเอกสารที่ทำให้เกิดความเข้าใจ,กระชับ,และสามารถเข้าไปหาได้และท้ายที่สุดยังช่วยในการหาคำตอบที่สามารถแก้ปัญหาในระบบให้กับผู้ใช้ได้อย่างมีประสิทธิภาพ

-Top - Down Approach : Functional Decomposition

หลักการของ Structured Methodologies คือ แนวความคิด Top - Down ซึ่งกำหนด Hierarchy ขององค์ประกอบภายในระบบโดยที่องค์ประกอบต่างๆสามารถแบ่งย่อยเป็นหน่วยต่างๆ และแต่ละหน่วยก็สามารถแบ่งออกเป็นหน่วยย่อยได้อีก ซึ่งเรียกกระบวนการนี้ว่า Functional

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Decomposition และเรียกหน่วยที่มีหน้าที่เฉพาะอย่างในระบบว่า Module โดยงานในระดับบนจะครอบคลุมงานที่อยู่ภายใต้ทั้งหมด

2.3 Data Flow Diagram (DFD)

แนวทางการออกแบบระบบงานวิธีหนึ่งที่นิยมกันอย่างแพร่หลายและสามารถแสดงความสัมพันธ์ในรายละเอียดให้ชัดเจนคือ วิธีการออกแบบโดยใช้แผนภาพแสดงทิศทางการไหลของข้อมูล (Data Fiow –

Diagram หรือ DFD) เป็นแผนภาพที่แสดงเฉพาะว่ามีงานอะไรบ้างที่ทำอยู่ในระบบงานโดยไม่สนใจว่างานนั้นๆมีวิธีการอย่างไร นอกจากนั้นยังแสดงให้เห็นถึงข้อมูลที่อยู่รอบๆระบบและมีข้อมูลอะไรบ้างที่อยู่เก็บไว้ในระบบ ซึ่งจะใช้สัญลักษณ์ต่างๆ ดังแสดงในตาราง 2.1 ส่วนประกอบของ DFD

- Process เป็นส่วนประกอบของ DFD ที่แสดงถึง ส่วนของงานในระบบ ที่มี Input เข้ามาใน Process แล้วมีการเปลี่ยนแปลง Input เป็น Output ส่งออกไปจาก Process สัญลักษณ์ที่ใช้แทน Process มีหลายสัญลักษณ์ด้วยกันได้แก่ วงกลม,สี่เหลี่ยม,สี่เหลี่ยมมุมโค้ง

- Flow เป็นส่วนประกอบที่แสดงถึง เส้นทางเข้า หรือออกของข้อมูล หรือเอกสาร ดังนั้น flow จะอธิบายถึงการเคลื่อนไหวของข้อมูลต่างๆ สัญลักษณ์ที่ใช้แทน flow คือเส้นตรงที่มีหัวลูกศร

- Store เป็นส่วนประกอบที่แสดงถึง การเก็บรวบรวมข้อมูล หรือเป็นที่พักของข้อมูล สัญลักษณ์ที่ใช้แทน Store อาจมีได้หลายรูปแบบด้วยกันคือ เส้นสองเส้นขนานกัน,เส้นตรงสองเส้นขนานกันปิดท้ายด้วยเส้นตรงด้านหนึ่ง หรือ เส้นตรงสองเส้นขนานกันปิดท้ายทั้งสองด้านด้วยเส้นโค้ง

- External Entities เป็นส่วนประกอบที่หมายถึง บุคคล,กลุ่มของบุคคล,หน่วยงาน หรือ องค์กรต่างๆ ที่อยู่ภายนอกระบบที่เรากำลังสนใจวิเคราะห์ และออกแบบ หรือ อยู่ภายนอกการควบคุมของระบบ สัญลักษณ์ที่ใช้แทน External Entities มีหลายสัญลักษณ์ด้วยกันได้แก่ วงรี,สี่เหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงสัญลักษณ์ต่างๆ ที่ใช้ในการเขียน DFD

ประเภท	สัญลักษณ์
Source / Destination	
Process	
Data Store	
Data Flow	

- ข้อดีของการเลือกใช้ DFD

DFD ช่วยให้นักวิเคราะห์สามารถสรุปข้อมูลที่เกี่ยวกับระบบ เข้าใจถึงปัจจัยสำคัญของระบบและระบุส่วนต่างๆ ของการทำงานที่ซ้ำซ้อน เข้าใจถึงความสัมพันธ์ระหว่างส่วนต่างๆ ของระบบและประกอบกันเป็นระบบ พัฒนาระบบได้อย่างมีประสิทธิภาพ

DFD เป็นเอกสารร่วมที่ช่วยให้นักวิเคราะห์และผู้ใช้สามารถเข้าใจระบบ และตรวจสอบความถูกต้องได้ทั้ง 2 ฝ่าย

ในการตรวจสอบเรื่องเวลาที่ใช้ในแต่ละขบวนการนั้น นักวิเคราะห์สามารถใช้ DFD เป็นเครื่องมือช่วยในการทราบถึงขอบเขตในการพัฒนารูปแบบระบบ

ลำดับขั้นการจัดการทำแผนภาพการไหลของข้อมูล

1) ให้เขียนแผนภาพโดยใช้หลักการเขียนจากบนลงล่าง (Top down) หรือจากใหญ่ไปหาเล็กโดย จะต้องรู้ว่าหน่วยงานภายนอกที่เกี่ยวข้องกับระบบทั้งหมดมีอะไรบ้างข้อมูลจะเคลื่อนไปในระบบ อย่างไร จะมีแฟ้มข้อมูลอะไรบ้าง ซึ่งสิ่งต่างๆ เหล่านี้จะต้องอยู่ภายใต้ระบบงานที่กำลังจะทำ

2) เขียนแผนภาพพื้นฐานนั้นมาก่อนซึ่งก็คือ แผนภาพระดับ Context Diagram และทบทวนว่าได้ ครอบคลุมระบบงานที่กำลังทำอยู่หรือไม่จากนั้นค่อยทำแผนภาพย่อยเพื่อประกอบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) บันทึกรายละเอียดในแต่ละขั้นตอโดยอาจมีหมายเหตุเพิ่มเติมเพื่อให้ผู้ อ่านเข้าใจได้ง่าย

4) ทบทวนว่าการบันทึกรายละเอียดนั้น ทุกค่าได้บันทึกไปมีความหมายชัดเจนดีแล้วหรือยัง หากว่า ยังไม่ชัดเจนดี ควรทำการแก้ไขให้ชัดเจนยิ่งขึ้น

ในการแสดงขั้นตอนการทำงานของระบบงาน และนอกจากนี้ยังมีข้อควรระวังซึ่งก่อให้เกิดในการออกแบบระบบงาน 3 ประการดังมีรายละเอียดคือ

แบลคโฮล (Black Hole) คือ ขั้นตอนการทำงาน (Process) ที่มีแต่ข้อมูลเข้า (Input) แต่ไม่มีผลลัพธ์ (Output) ของการทำงานออกมา

มิราเคิล (Miracle) คือ ขั้นตอนการทำงาน (Process) ที่ไม่มีข้อมูลเข้า (Input) แต่กลับมีผลลัพธ์ (Output) ของการทำงานออกมา

เกรโฮล (Gray Hole) คือ ขั้นตอนการทำงาน (Process) ที่มีข้อมูลเข้า (Input) ทั้งหมดไม่เพียงพอที่จะทำให้เกิดผลลัพธ์ (Output) ของการทำงานออกมา

2.4 ฐานข้อมูล

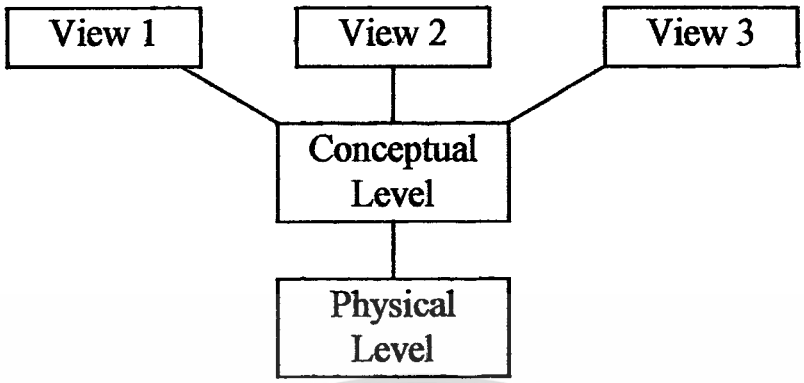
ฐานข้อมูลคือ การรวบรวมเอาข้อมูลที่มีความเกี่ยวโยงกันมาไว้รวมกัน อย่างมีระบบในที่เดียวกัน โดยผู้ใช้ฐานข้อมูลแต่ละคนจะสามารถใช้ข้อมูลได้ตามจุดประสงค์ของการประยุกต์ใช้งานโดยที่ไม่ต้องสนใจว่าการจัดเก็บข้อมูลที่แท้จริงเป็นอย่างไร แต่จะสนใจเพียงว่าจะเรียกข้อมูลขึ้นมาใช้อย่างถูกต้องมีประสิทธิภาพได้มากที่สุด ซึ่งฐานข้อมูลได้จัดแบ่งระดับของข้อมูลออกเป็น 3 ระดับคือ (ดังแสดงในภาพที่ 2.1)

1) ระดับภายใน (Internal หรือ Physical Level) ซึ่งเป็นระดับที่ต่ำที่สุดซึ่งได้แก่ ระดับ ของการจัดเก็บข้อมูลจริงๆ

2) ระดับหลักการ(Conceptual Level) ซึ่งในระดับการมอง Entity และความสัมพันธ์ระหว่าง Entity ที่มีอยู่ในระบบรวมทั้งกฎเกณฑ์ ต่างๆ เกี่ยวกับข้อมูลและผู้มีสิทธ์จะใช้ ฯลฯ ข้อมูลในระบบนี้จะอยู่ในความสนใจของ DBA (Database Administrator) เพราะจะเป็นผู้ออกแบบและควบคุมการใช้ฐานข้อมูล

3) ระดับภายนอก (External หรือ View Level) เป็นระดับที่สูงสุดอันเป็นระดับข้อมูลที่จะมองเห็นจากการใช้งานของผู้ใช้แต่ละคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 แสดงระดับของข้อมูล 3 ระดับ

ความหมายของฐานข้อมูลอีกนัยหนึ่งก็คือ ฐานข้อมูลเป็นการเอาไฟล์ข้อมูลที่ใช้ในองค์กรที่อยู่กระจัดกระจายตามที่ต่างๆ ที่อาจจะมีข้อมูลที่ใช้ซ้ำกัน ทำให้เปลืองเนื้อที่ในการจัดเก็บข้อมูล และเป็นข้อมูลที่จัดเก็บอาจจะเกิดความขัดแย้งกัน กล่าวคือ เมื่อมีการเปลี่ยนแปลงข้อมูลอาจจะเกิดการหลงลืมในการเปลี่ยนแปลงให้ครบทุกไฟล์ จากปัญหาของการประมวลผลแบบไฟล์ทำให้เกิดแนวความคิดของระบบฐานข้อมูลขึ้น ซึ่งการประมวลฐานข้อมูล จะสามารถช่วยขจัดปัญหาของการประมวลผลด้วยระบบไฟล์ เช่น ลดความซ้ำซ้อน และไม่สอดคล้องของข้อมูล ช่วยลดความซ้ำซ้อนในการเข้าถึงข้อมูล สามารถใช้ข้อมูลร่วมกันได้ ข้อมูลมีความปลอดภัย ข้อมูลมีความคงสภาพ และข้อมูลที่จัดเก็บมีความเป็นอิสระของข้อมูล

1) ลดความซ้ำซ้อนและไม่สอดคล้องของข้อมูล ข้อมูลที่เป็นข้อมูลชนิดเดียวกันจะถูกเก็บไว้ที่เดียวกันทำให้มีการเปลี่ยนแปลง หรือปรับปรุงข้อมูลก็สามารถทำได้กับข้อมูลชุดเดียวกันทำให้ข้อมูลมีความถูกต้องสมบูรณ์และสอดคล้องอยู่ตลอดเวลา

2) ช่วยลดความซับซ้อนในการเข้าถึงข้อมูล ในระบบฐานข้อมูลจะมีภาษาที่ใช้ในการเข้าถึงข้อมูลที่ทำให้ผู้ใช้สามารถเข้าใจได้ง่ายและในระบบฐานข้อมูลจะมี ซอฟต์แวร์ที่เรียกว่าระบบการจัดการฐานข้อมูล (Database Management System) หรือเรียกสั้นๆ ว่า DBMS เป็นตัวคอยจัดการให้

3) สามารถใช้ข้อมูลร่วมกันได้ ผู้ใช้สามารถใช้ข้อมูลที่มีอยู่ในฐานข้อมูลร่วมกันได้ ทำให้ไม่เปลืองเนื้อที่ในการจัดเก็บข้อมูลและสามารถแก้ไขข้อมูลได้ง่าย

4) ข้อมูลมีความปลอดภัย ในระบบฐานข้อมูลจะมีการกำหนด View หรือ Sub Schema ให้กับผู้ใช้แต่ละคนได้ ทำให้สามารถกำหนดสิทธิ์ในการเข้าถึงฐานข้อมูลผู้ใช้แต่ละคน โดย ผู้บริหารฐานข้อมูล (Database Administrator หรือ DBA) ซึ่งจะเป็นบุคคลเดียวหรือกลุ่มคนที่ใช้ได้ และการเข้าถึงข้อมูลจะต้องผ่าน DBMS และข้อมูลยังมีการเข้ารหัสไว้อีกด้วยเพื่อเป็นการป้องกันการเข้าถึงข้อมูลโดยไม่ผ่าน DBMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ข้อมูลมีความคงสภาพ ในระบบฐานข้อมูลจะมีการควบคุมการคงสภาพของข้อมูล (Integrity Constraint) คือเราสามารถที่จะกำหนดค่าของ Attribute แต่ละ Attribute ได้ ซึ่งจะมีลักษณะอย่างไร เช่น การลงรหัสโรคให้กับผู้ป่วยจะต้องใช้คู่มือลงรหัสโรคขององค์การอนามัยโลก (WHO) หรือ ในการบันทึกข้อมูลใดๆ ของผู้ป่วยต้องลงทะเบียนผู้ป่วยนั้นๆ มาก่อนจึงจะบันทึกข้อมูลได้ เป็นต้น

6) ข้อมูลที่จัดเก็บมีความเป็นอิสระ การจัดเก็บจริงใน Disk จะอยู่ในระดับภายใน Internal หรือ Physical Level เมื่อมีการเปลี่ยนแปลงเทคนิคการจัดเก็บ หรือ การเรียกใช้ข้อมูล ผู้ใช้ไม่จำเป็นต้องเขียนโปรแกรมขึ้นมาใหม่ ซึ่งตัว DBMS จะเป็นตัวที่จัดการเชื่อมข้อมูลระหว่างระดับภายนอกและระดับหลักการ กับเชื่อมระหว่างระดับหลักการกับระดับภายใน ซึ่งการเป็นอิสระของข้อมูลสามารถแบ่งออกเป็น 2 ชนิดคือ

6.1) ความเป็นอิสระแบบกายภาพ (Physical Data Independence) คือการเปลี่ยนแปลงวิธีการจัดเก็บข้อมูลระดับภายใน เช่น อาจจะเป็นเทคนิคการจัดเก็บข้อมูลจาก Indexed Sequential เป็นแบบ Direct Access จะไม่มีผลกระทบต่อโครงสร้างของระดับหลักการและระดับภายนอก

6.2) ความเป็นอิสระแบบตรรกะ (Logical Data Independence) คือการเปลี่ยนแปลงข้อมูลระดับ Conceptual เช่นอาจจะมีการเพิ่ม Attribute หรือ เพิ่ม Entity เข้าไปในระบบก็จะไม่มีผลกระทบต่อข้อมูลระดับภายนอกหรือระดับ View

2.4.1 Database Schema

Database Schema คือ ส่วนเค้าร่างของฐานข้อมูลที่ได้จากการออกแบบของฐานข้อมูลโดยรวม ซึ่งจะเป็นการกำหนดว่าฐานข้อมูลจะประกอบด้วย Entity อะไรบ้าง และแต่ละ Entity มีความสัมพันธ์กันอย่างไรและจะประกอบด้วย Attribute อะไรบ้าง และจะใช้ Attribute อะไรมาเป็นคีย์หลักของฐานข้อมูล โดยปกติแล้ว Database Schema ไม่ควรมีการเปลี่ยนแปลงหรือให้มีการเปลี่ยนแปลงน้อยที่สุด

Database Schema จะแบ่งออกเป็นระดับ คล้ายกับระดับของข้อมูลคือ

1) ระดับภายใน Internal หรือ Physical Schema คือการจัดเก็บข้อมูลว่าจะจัดเก็บข้อมูลจริงๆ ใน Disk

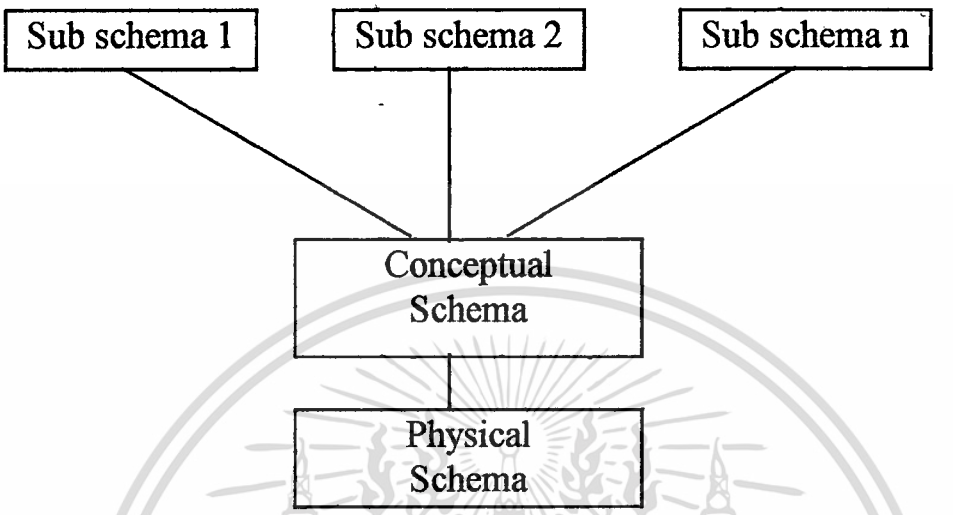
2) ระดับหลักการ (Conceptual Schema) คือระดับของการออกแบบฐานข้อมูลว่าจะจัดเก็บข้อมูลอะไรบ้าง

3) ระดับนอกสุด (Sub Schema) คือระดับในมุมมองของ User

ซึ่งจะเห็นได้ว่า ระบบฐานข้อมูลประกอบด้วย Physical Schema และ Conceptual Schema

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างละ 1 ตัว และจะมี Sub Schema หลายๆ ตัว เช่น เดียวกับระดับของข้อมูลดังแสดงในภาพที่ 2.2



ภาพที่ 2.2 Schema ในฐานข้อมูล 3 ระดับ

2.4.2 ระบบการจัดการฐานข้อมูล (Database Management Systems)

ระบบการจัดการฐานข้อมูล (Database Management Systems หรือเรียกย่อๆ ว่า DBMS) เป็นโปรแกรมที่ทำหน้าที่คอยอำนวยความสะดวกในการใช้งานระบบฐานข้อมูลให้กับผู้ใช้ (User) เพราะผู้ใช้ไม่จำเป็นจะต้องรู้เรื่องเกี่ยวกับรายละเอียดของการจัดเก็บข้อมูล DBMS มีหน้าที่ดังนี้

1) ทำหน้าที่ติดต่อกับตัวจัดการระบบไฟล์ DBMS มีหน้าที่ประสานงานกับ File Manager ในการจัดเก็บ เรียกใช้ และแก้ไขข้อมูล ซึ่ง DBMS จะทำหน้าที่แปลคำสั่ง DML ให้อยู่ในรูปแบบที่ File Manager เข้าใจ

2) การควบคุมการคงสภาพของข้อมูล ซึ่ง DBMS จะเป็นตัวทำหน้าที่ในการควบคุมให้ข้อมูลที่จัดเก็บใน Table ให้อยู่บนกรอบของข้อมูลที่กำหนดในส่วนของ Schema เช่น อายุของผู้ป่วยไม่เกินกว่า 99 ปี

3) การควบคุมระบบความปลอดภัย DBMS จะทำหน้าที่ป้องกันไม่ให้ผู้ไม่มีสิทธิใช้ข้อมูล เข้าใช้ข้อมูลได้

4) การสร้างระบบสำรองและฟื้นสภาพ เมื่อระบบหรือไฟล์ได้รับความเสียหาย DBMS จะต้องทำหน้าที่สำรองข้อมูลไว้ และทำการฟื้นสภาพให้ระบบข้อมูลกลับเข้าสู่สภาพที่ถูกต้องสมบูรณ์ได้

5) การควบคุมสภาวะพร้อมกัน (Concurrency Control) DBMS จะต้องจัดลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง ทำซ้ำ หรือจำหน่ายจ่ายแจก หรือนำไปใช้ประโยชน์ในเชิงการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าถึงข้อมูลได้อย่างถูกต้องเมื่อมีการเรียกใช้ข้อมูลพร้อมกัน

2.4.3 โมเดลเชิงสัมพันธ์ (Relational Model)

จะแสดงข้อมูลและความสัมพันธ์ระหว่างข้อมูลด้วยการรวบรวมของตาราง (Table) ซึ่งแต่ละตารางจะประกอบด้วยจำนวนของ Columns ที่มีชื่อไม่ซ้ำกัน (ซึ่งตารางนี้อาจจะเรียกอีกอย่างว่า Relation) ข้อมูลของตาราง จะถูกจัดเก็บเรียบเรียงต่อไปเป็น Row ซึ่งจะทำให้ได้ ตาราง ซึ่งมี 2 มิติ คือ Row กับ Column ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 ตัวอย่างฐานข้อมูลแบบโมเดลเชิงสัมพันธ์

เลขที่ใบสั่งยา	วันที่	เวลา	HOSPITAL NUMBER
1	25/12/43	08:00:00	43000001
2	25/12/43	08:10:00	43000003

HOSPITAL NUMBER	ชื่อ นามสกุลผู้ป่วย
43000001	นายสมชาย เชื้อไทย
43000002	นายมานพ กรุงเทพ
43000003	นางสาวจันทบุรี เมืองเก่า

ฐานข้อมูลเชิงสัมพันธ์จะแสดงในรูปแบบของตาราง (Table) ที่มี Column และ Row ซึ่ง Column แต่ละ Column จะเก็บข้อมูลชนิดเดียวกัน และ Row แต่ละ Row จะเก็บข้อมูลกลุ่มของ Column ซึ่งเราสามารถเรียกองค์ประกอบต่างๆ ได้ดังนี้

Table เป็น Relation หรือ File ในระบบประมวลผลแบบ File

Column เป็น Attribute หรือ Field ในระบบประมวลผลแบบ File

Row เป็น Tuple หรือ Record ในระบบประมวลผลแบบ File

ซึ่งเราจะนิยาม Relation คือ ตาราง 2 มิติ ที่

- แต่ละช่องของตารางจะบรรจุข้อมูลเพียงค่าเดียว
- ชื่อหัวข้อแต่ละ Column แตกต่างกันคือค่าของ Attribute

เอกสารนี้เป็น[illegible]ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ค่าข้อมูลที่อยู่ใน Column ได้แก่ ค่าของ Attribute ที่ระบุไว้ในหัวข้อ Column นั้นๆ
- การเรียงลำดับของ Column ไม่ถือว่ามีความสำคัญ
- ข้อมูลแต่ละแถวจะต้องแตกต่างกัน
- การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ

ดังนั้น ฐานข้อมูลเชิงสัมพันธ์ ได้แก่ การรวบรวมรีเลชั่นต่างๆ ที่มีความสัมพันธ์ (Relationship) ระหว่างกัน

คีย์หลัก (Primary Key) จะประกอบไปด้วยค่าของ Attribute 1 ค่า หรือมากกว่าก็ได้เพื่อที่สามารถใช้เป็นตัวแยกข้อมูลในแต่ละ Tuple ที่มีความแตกต่างกัน

คีย์นอก (Foreign Key) คือค่าของ Attribute ของ Relation หนึ่งซึ่งมีค่าซ้ำกับคีย์หลัก (Primary Key) ของอีก Relation หนึ่ง

คีย์คู่แข่ง (Candidate Key) คือ Attribute ใน Relation ที่สามารถเป็นคีย์หลักของ Relation นั้น แต่เราไม่เลือกที่จะใช้เป็นคีย์หลัก

2.4.4 ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์

1) กฎของความคงสภาพ (Integrity Rule)

กฎของการคงสภาพของโมเดลเชิงสัมพันธ์ (Relational Model) เป็นทฤษฎีที่ช่วยยืนยันความถูกต้องของความสัมพันธ์ระหว่างข้อมูลว่า รีเลชั่นใดที่เป็นไปตามกฎของความคงสภาพนี้แล้วย่อมจะมีความสัมพันธ์ระหว่างข้อมูลอย่างถูกต้องอยู่ตลอดเวลา ไม่ว่ารีเลชั่นนั้นจะมีการเปลี่ยนแปลงแก้ไขข้อมูลไปตามรูปแบบใดก็ตาม กฎของความคงสภาพมีความหมายอยู่ 2 ลักษณะ คือ กฎความคงสภาพของเอนติตี้ (Entity Integrity Rule) และกฎความคงสภาพของการอ้างอิง (Referential Integrity Rule) ดังอธิบายได้ดังนี้

1.1) กฎความคงสภาพของเอนติตี้ กล่าวว่า "แอตทริบิวต์ทุกตัวที่เป็นส่วนของคีย์หลักจะไม่อนุญาตให้มีค่าว่าง" หมายความว่า คีย์หลักทุกรีเลชั่นจะไม่สามารถเก็บค่าข้อมูลที่เป็นค่าว่างได้ เหตุผลของข้อกำหนดนี้ก็คือ เพื่อให้การเข้าถึงข้อมูลในแถวใดๆ ของรีเลชั่นมีความเป็นไปได้เสมอ เพราะว่าคีย์หลักของแถวใดมีค่าข้อมูลเป็นค่าว่างแล้ว ก็จะส่งผลให้การเข้าถึงข้อมูลในแถวนั้นไม่สามารถกระทำได้อย่างแน่นอน

1.2) กฎความคงสภาพการอ้างอิง กล่าวว่า "ถ้าเรามีรีเลชั่น R2 ดังตารางที่ 2.4 ซึ่งมี FK (Foreign Key) เป็นคีย์นอกที่อ้างอิงถึงคีย์หลัก PK (Primary Key)ในรีเลชั่น R1 ดังตารางที่ 2.3 สำหรับทุกค่าของ FK ใน R2 จะต้อง

ก. มีค่าเท่ากับค่า PK ในแถวใดแถวหนึ่งในรีเลชั่น R1

ข. มีค่าของแอตทริบิวต์ใน FK เป็นค่าว่าง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายความว่า แอตทริบิวต์ใดๆ ที่เป็นคีย์หลักของของรีเลชั่นหนึ่ง เมื่อมีการนำแอตทริ-บิวต์นั้นไปเป็นคีย์นอกของอีกรีเลชั่นหนึ่ง การเป็นคีย์นอกของแอตทริบิวต์นั้นจะต้องมีโดเมนเป็นโดเมนเดียวกันกับแอตทริบิวต์หลัก ทั้งนี้ก็เพื่อให้การนำรีเลชั่นมาใช้งานร่วมกัน (การนำรีเลชั่นมา Join กัน) กระทำได้อย่างถูกต้อง คือ ทุกแอตทริบิวต์ที่เป็นคีย์นอกจะต้องมีข้อมูลซ้ำกับข้อมูลของแอตทริบิวต์ที่เป็นคีย์หลักอย่างแน่นอน แต่อาจมีบางค่าข้อมูลของแอตทริบิวต์ที่เป็นคีย์หลักเป็นข้อมูลไม่อยู่ในโดเมนของแอตทริบิวต์ที่เป็นคีย์นอกก็ได้ นั่นคือ โดเมนชั่นของคีย์นอกจะต้องเล็กกว่าหรือเท่ากับโดเมนของคีย์หลักเสมอ

ตารางที่ 2.3 รีเลชั่น R1

คีย์หลักของ R1	คีย์อื่นๆของ R1
A	1
B	2
C	3

ตารางที่ 2.4 รีเลชั่น R2

คีย์หลักของ R2	คีย์อื่นๆของ R1
A	A
B	B

2) ฟังก์ชั่นการขึ้นต่อกัน (Functional Dependency)

ฟังก์ชั่นการขึ้นต่อกันเป็นข้อกำหนดที่ช่วยให้เราเห็นถึงความคงสัมพันธ์ของแอตทริบิวต์ต่างๆ ที่อยู่ในรีเลชั่น ทั้งนี้เพราะแอตทริบิวต์ต่างๆ ที่อยู่ในเอนติตี้เดียวกัน จะมีความสัมพันธ์กันเองโดยที่ความสัมพันธ์นี้อาจเกี่ยวข้องหรือไม่เกี่ยวกับความสัมพันธ์ที่มันมีต่อคีย์หลักของเอนติตี้นั้นก็เป็นได้ ซึ่งการที่แอตทริบิวต์เหล่านั้นมีความสัมพันธ์กันเองจะเป็นสิ่งที่เราต้องพิจารณาแยกออกเป็นรีเลชั่นย่อยๆ เพราะแอตทริบิวต์แต่ละรีเลชั่นก็ควรจะมีความสัมพันธ์กับคีย์หลักของรีเลชั่นของตนเองเท่านั้น กำหนดรีเลชั่น R ดังแสดงในตารางที่ 2.5 ถ้าที่แอตทริบิวต์ Y ของ R เป็นฟังก์ชั่นที่ขึ้นต่อแอตทริบิวต์ X ของรีเลชั่น เราสามารถเขียนแทนได้ด้วยสัญลักษณ์ R.X---->R.Y อ่านว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R.X มีฟังก์ชั่นขึ้นอยู่กับ R.Y หรือ R.X มีฟังก์ชั่นการขึ้นอยู่กับ R.Y หรือ R.Y ขึ้นอยู่กับ R.X

นิยาม R.X. มีฟังก์ชั่นการขึ้นอยู่กับ R.Y ก็ต่อเมื่อ ทุกค่าข้อมูลของแอตทริบิวต์ X ในR จะมีค่าข้อมูลของแอตทริบิวต์Y ใน R ได้เพียงค่าเดียวเสมอ โดยที่แอตทริบิวต์ X และY อาจจะมีคีย์แบบร่วม (Composite Key) ก็ได้

ตารางที่ 2.5 รีเลชั่น R

X	Y
ANT	12345
BOY	23456
ANT	12345
BOY	34567

นิยาม R.X มีฟังก์ชั่นการขึ้นอยู่กับR.Y อย่างสมบูรณ์ (R.Y Fully Functionally Dependent on R.X) ก็ต่อเมื่อ R.Y มีฟังก์ชั่นการขึ้นอยู่กับ R.X และไม่ขึ้นอยู่กับข้อมูลเพียงบางส่วนของ R.X โดยที่แอตทริบิวต์ X และ Y อาจจะเป็นคีย์แบบร่วมก็ได้

ตารางที่ 2.5 รีเลชั่น R (ต่อ)

X		Y
ANT	ant	12345
boy	boy	23456
ant	ant	12345
BOY	cat	34567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5 ภาษาของฐานข้อมูล

1)ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language) หรือเรียกย่อ ๆ ว่า DDLเป็นภาษาที่ใช้กำหนด Database Schema ซึ่งเมื่อเราคอมไพล์คำสั่งของ DDL แล้วจะได้กลุ่มของตาราง Table ที่ใช้จัดการเก็บข้อมูล ซึ่งจะอยู่จัดเก็บไว้ในไฟล์พิเศษที่เรียกว่า Data Dictionary (or Directory) ซึ่งจะเก็บข้อมูลเกี่ยวกับโครงสร้างที่ได้จากการออกแบบฐานข้อมูลนั้น ๆ และเมื่อมีการเปลี่ยนแปลงหรือเรียกใช้ข้อมูล DBMS จะต้องอาศัยข้อมูลของโครงสร้างจากไฟล์นี้เสมอ

2) ภาษาที่ใช้สำหรับการจัดการข้อมูล (Data Manipulation Language) การจัดการข้อมูลในที่นี้หมายถึง

- การดึงเรียกดูข้อมูล (Information) ที่อยู่ในฐานข้อมูล
- การ Insert ข้อมูลใหม่ (Information) เข้าไปในฐานข้อมูล
- การลบ(Delete) ข้อมูลออกฐานข้อมูล
- การปรับปรุง(Modify) ข้อมูลในฐานข้อมูล

ในส่วนของผู้ใช้จะเรียกใช้ข้อมูลโดยผ่าน DBMS โดยใช้ภาษาที่เรียกว่าภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language) หรือเรียกย่อ ๆ ว่า DML ซึ่งได้ถูกออกแบบให้ผู้ใช้ได้เข้าถึงข้อมูลหรือจัดการข้อมูลได้ง่าย ซึ่ง DML สามารถแบ่งออกได้ 2 ชนิด คือ

2.1 Procedural DML ซึ่งผู้ใช้จะต้องระบุลงไปว่าต้องการข้อมูลอะไร และจะใช้วิธีการใดซึ่งจะได้ข้อมูลมา ซึ่งจะมีลักษณะคล้ายๆ กับภาษาชั้นสูง เช่น C, Pascal หรือภาษาเฉพาะของฐานข้อมูลนั้นๆ เช่น FoxPro, Dbase เป็นต้น

2.2 Non –Procedural DML เป็นภาษาที่ผู้ใช้สามารถเรียนรู้และใช้งานได้ง่าย ถึงแม้ว่าผู้ใช้จะไม่มีความรู้เกี่ยวกับการเรียกใช้ข้อมูลเลย โดยที่ผู้ใช้เพียงแต่ระบุไปว่าต้องการข้อมูลอะไร โดยไม่จำเป็นต้องรู้วิธีการว่าจะทำอย่างไรเลย

2.4.6 โครงสร้างระบบฐานข้อมูล

ประกอบไปด้วยส่วนต่าง ๆ ดังภาพที่ 2.3 โครงสร้างของระบบฐานข้อมูล ซึ่งแสดงส่วนประกอบและการติดต่อของระบบฐานข้อมูล

1) File Manager มีหน้าที่ เป็นตัวจัดการ การจองเนื้อที่ใน Disk และเป็นโครงสร้างข้อมูลที่ใช้จัดเก็บข้อมูลใน Disk

2) Database Manager มีหน้าที่ติดต่อสื่อสานระหว่าง การจัดเก็บข้อมูลในชั้น Low – Level ในฐานข้อมูลและโปรแกรมประยุกต์ (Application Program)

3) Query Processor มีหน้าที่แปลงคำสั่งของภาษา Query ไปยังชั้น Low – Level

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

เพื่อให้ Database Manager เข้าใจ

4) DML Precompiler ทำหน้าที่แปลงคำสั่งของ DML ที่อยู่ ในโปรแกรมประยุกต์ ให้อยู่ในรูปการเรียกใช้ Procedure ปกติในภาษาหลัก (Host Language) ซึ่งตัว Precompiler โต้ตอบกับ Query เพื่อที่จะสร้าง Code ที่ถูกต้อง

5) DDL Compiler ทำหน้าที่ แปลงคำสั่งของ DDL ไปเป็นกลุ่มของตาราง (Table) ซึ่งประกอบด้วย Metadata หรือข้อมูลของข้อมูล

6) Data File ทำหน้าที่ แปลงคำสั่งของตัวมันเอง

7) Data Dictionary ทำหน้าที่เก็บ Metadata ที่เกี่ยวกับโครงสร้างของฐานข้อมูล ซึ่งจะต้องทำหน้าที่อย่างหนัก ดังนั้นจึงต้องออกแบบและสร้างอย่างดี

8) Indices ทำหน้าที่ช่วยในการเข้าถึงข้อมูลได้เร็วขึ้น

2.4.7 วิธีการออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลจะอาศัยแบบจำลองอีอาร์ (Entity Relationship Model หรือ ER Model) โดยการออกแบบฐานข้อมูลด้วยแบบจำลองอีอาร์จะแบ่งออกเป็น 2 ขั้นตอน คือ

1)เป็นการสร้างแบบจำลองอีอาร์

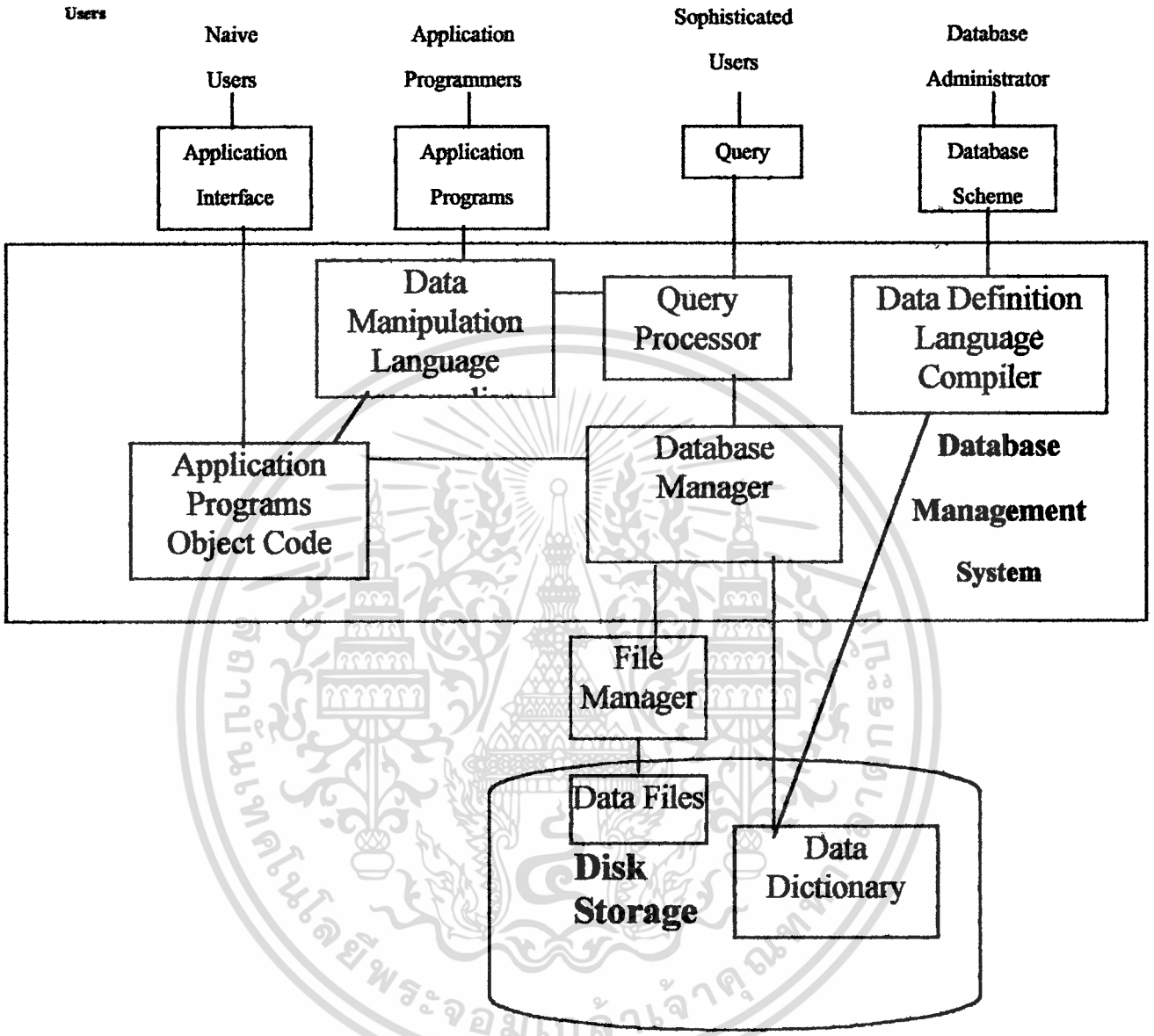
แบบจำลองอีอาร์ เป็นแผนภาพที่ถูกออกแบบมาเพื่อแสดงความสัมพันธ์

ระหว่างเอนติตี้ (Entity) ต่าง ๆ ในระบบ ซึ่งจะอยู่ในรูปแบบที่เป็นรูปธรรมมากขึ้น ในแต่ละเอนติตี้ก็จะประกอบไปด้วยแอตทริบิวต์ (Attribute)

เอนติตี้ หมายถึง สิ่งต่าง ๆ ที่เราให้ความสนใจในระบบ เช่น พนักงาน (EMPLOYEE) ,หน่วยงาน (DEPARTMENT) หรือโครงการ (PROJECT) เป็นต้น

- Regular Entity Type สัญลักษณ์ที่ใช้ คือ รูปสี่เหลี่ยม
- Weak Entity Type สัญลักษณ์ที่ใช้ คือ รูปสี่เหลี่ยมซ้อนกัน 2 รูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 โครงสร้างของระบบฐานข้อมูล

แอตทริบิวต์ หมายถึง ลักษณะประกอบของแต่ละเอนติตี้ เช่น เอนติตี้พนักงาน (EMPLOYEE) จะประกอบด้วย แอตทริบิวต์ชื่อ , แอตทริบิวต์ที่อยู่ , แอตทริบิวต์เงินเดือน หรือ แอตทริบิวต์วันเกิด เป็นต้น สัญลักษณ์ที่ใช้คือ รูปวงรี

ความสัมพันธ์ หมายถึง ความเกี่ยวพันระหว่างเอนติตี้ 2 เอนติตี้ขึ้นไป เช่น เอนติตี้หน่วยงาน(DEPARTMENT) กับเอนติตี้พนักงาน (EMPLOYEE) มีความเกี่ยวพันกัน คือ พนักงานทุกคนต้องสังกัดในหน่วยงาน เป็นต้น สัญลักษณ์ที่ใช้ คือ รูปสี่เหลี่ยมข้าวหลามตัด

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม
- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ในความเป็นจริงแล้วสัญลักษณ์ต่าง ๆ ที่ใช้ในแบบจำลองอีอาร์ ยังมีรายละเอียดมากกว่าที่กล่าวมาข้างต้น แต่จะขอกล่าวถึงความหมาย และสัญลักษณ์ที่ใช้ทั่ว ๆ ไปของแบบจำลองอีอาร์

2)เป็นการแปลงแบบจำลองอีอาร์ให้เป็นตารางฐานข้อมูล มีขั้นตอนดังนี้

2.1สำหรับแต่ละเอนติตี้ที่เป็นเอนติตี้ปกติ จะสร้างเป็นรีเลชั่น โดยมีทุกแอตทริบิวต์มาประกอบกัน คีย์หลักของรีเลชั่นอาจเกิดจากการเลือกแอตทริบิวต์ใดแอททริบิวต์หนึ่งหรือกลุ่มของแอตทริบิวต์ก็ได้

2.2 สำหรับแต่ละเอนติตี้ชนิดอ่อน จะสร้างรีเลชั่นที่เกิดจากการรวมกันของแอตทริบิวต์ โดยที่รีเลชั่นนี้จะมีคีย์หลัก หรือคีย์รวม ที่เกิดจากการรวมกันของคีย์หลักของเอนติตี้ชนิดอ่อนกับคีย์หลักของเอนติตี้ชนิดอ่อน กับคีย์หลักของเอนติตี้ที่มันต้องอ้างถึง

- สำหรับความสัมพันธ์ระหว่างเอนติตี้แบบหนึ่งต่อหนึ่ง จะสร้างรีเลชั่นจากความสัมพันธ์ดังกล่าวได้ คือ เลือกคีย์หลักของเอนติตี้ใดเอนติตี้หนึ่งมาเป็นคีย์หลักของ รีเลชั่นนี้แล้วให้คีย์หลักของอีกเอนติตี้หนึ่งมาเป็นคีย์นอกของรีเลชั่นนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวต์ก็ให้นำแอตทริบิวต์เหล่านั้นมารวมอยู่ในรีเลชั่นนี้ด้วย

- สำหรับความสัมพันธ์ระหว่างเอนติตี้แบบหนึ่งต่อกลุ่ม จะสร้างรีเลชั่นจากความสัมพันธ์ดังกล่าว โดยนำเอาคีย์หลักของเอนติตี้ฝั่งที่มีความสัมพันธ์แบบกลุ่มมาเป็นคีย์หลักของรีเลชั่นนี้แล้วให้นำเอาคีย์หลักของเอนติตี้ฝั่งที่มีความสัมพันธ์แบบหนึ่ง มาเป็นคีย์นอกของรีเลชั่นนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวต์ ก็ให้นำแอตทริบิวเหล่านั้นมารวมอยู่ในรีเลชั่นนี้ด้วย

- สำหรับความสัมพันธ์ระหว่างเอนติตี้แบบกลุ่มต่อกลุ่ม จะสร้างรีเลชั่นจากความสัมพันธ์ดังกล่าว โดยนำเอาคีย์หลักของทั้งสองเอนติตี้ มาประกอบกันเป็นคีย์หลักของรีเลชั่นนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวต์ ก็ให้นำแอตทริบิวต์เหล่านั้นมารวมอยู่ในรีเลชั่นนี้ด้วย

2.3 สำหรับเอนติตี้ใดที่มีแอตทริบิวต์ที่มีค่าข้อมูลแบบหลายค่า (Multivalue Attribute) ก็

ให้สร้างรีเลชั่นใหม่ โดยมีคีย์หลักของเอนติตี้นั้นรวมกับแอตทริบิวต์ดังกล่าวเป็นคีย์หลักของรีเลชั่นนี้

2.4 สำหรับความสัมพันธ์ระหว่างเอนติตี้ที่เกิดจากเอนติตี้มากกว่า 2 เอนติตี้ ให้สร้างรีเลชั่นของความสัมพันธ์รีเรชั่นของความสัมพันธ์นี้ โดยนำเอาคีย์หลักของทุกเอนติตี้มาประกอบกันเป็นคีย์หลักของรีเรชั่นนี้ โดยถ้าความสัมพันธ์นี้มีแอททริบิวต์ ก็ให้นำแอตทริบิวต์เหล่านั้นมารวมอยู่ในรีเลชั่นนี้ด้วย

ตารางที่มีลักษณะเป็นรีเลชั่นจะต้องมีคุณสัมบัติดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชื่อหัวข้อในแต่ละคอลัมน์มีความแตกต่างกัน อันเป็นชื่อของแอตทริบิวต์ของเอนติตี้
- ค่าข้อมูลในแต่ละคอลัมน์ คือ ค่าของแอตทริบิวต์ของเอนติตี้
- การเรียงลำดับคอลัมน์ไม่ถือว่ามีความสำคัญ
- ข้อมูลแต่ละแถวจะต้องแตกต่างกัน
- การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ

ตารางที่มีคุณสมบัติดังกล่าวจะเรียกว่า รีเลชั่น ดังนั้นเราจะได้นิยามของฐานข้อมูลเชิงสัมพันธ์ คือ ฐานข้อมูลที่เกิดการรวบรวมรีเลชั่นต่าง ๆ ที่มีความสัมพันธ์ (Relationship) ระหว่างกัน จะเรียกข้อมูลแต่ละแถวในแนวนอนของรีเลชั่นว่า ทัพเพิล (Tuple) และเรียกข้อมูลในแต่ละแถวในแนวตั้ง หรือแนวคอลัมน์ว่า แอตทริบิวต์ (Attribute) โดยที่คำว่า คีย์ (Key) จะหมายถึงข้อมูลที่เกิดจากแอตทริบิวต์ 1 ตัว หรือ หลายตัวก็ได้

แต่ละรีเลชั่นจะต้องมีสิ่งที่เรียกว่า คีย์หลัก (Primary key) คือ ข้อมูลของแอตทริบิวต์ 1 ตัว หรือมากกว่า 1 ตัวก็ได้ ที่สามารถใช้เป็นตัวเจาะจงบอกเราได้ว่ากำลังอ้างอิงถึงข้อมูลทัพเพิลใดส่วนคีย์ที่เป็นแอตทริบิวต์ของรีเลชั่นอื่นที่ซ้ำกับแอตทริบิวต์ที่เป็นคีย์หลักของรีเลชั่นหนึ่งจะเรียกว่าคีย์นอก(Foreign key) ในกรณีที่รีเลชั่นมีแอตทริบิวต์ หรือ กลุ่มของแอตทริบิวต์ที่มิได้ถูกเลือกให้เป็นคีย์หลัก เรียกว่า คีย์คู่แข่ง (Candidate key) หรือ คีย์สำรอง (Alternate key) และแอตทริบิวต์อื่น ๆ ที่เหลือที่มิได้เป็นคีย์หลักและไม่ได้เป็นส่วนใดส่วนหนึ่งของคีย์หลัก ก็จะถูกเรียกว่าเป็น Nonkey Attribute

คำว่า โดเมน (Domain) จะหมายถึง กรอบของค่าต่าง ๆ ที่เป็นไปได้ เช่น โดเมนของแอตทริบิวต์ วันที่ ก็จะหมายถึงค่าของวันที่ที่เป็นไปได้ คือ มีค่าเท่ากับ 1 ถึง 31 แต่ในการเก็บค่าข้อมูลในรีเลชั่นนั้น บางกรณีที่เรามีการกำหนดโดเมนให้แอตทริบิวต์แล้ว แต่ข้อมูลที่จะถูกเก็บเข้าไปอาจถูกบรรจุเข้าไปในภายหลัง ลักษณะนี้จะทำให้เกิด ค่าว่าง (Null Value) ขึ้นชั่วขณะก่อนที่จะมีการบรรจุค่าข้อมูลที่อยู่ในโดเมนที่กำหนดไว้เข้าไป ดังนั้น คำว่า "ค่าว่าง" จึงหมายถึงค่าที่ยังมิทราบชัดว่าแอตทริบิวต์นั้นจะมีค่าเป็นค่าใด หรือ ค่าของข้อมูลที่ไม่อยู่ในโดเมนที่ กำหนด โดยมีข้อบังคับไว้ว่าแอตทริบิวต์ที่ทำหน้าที่เป็น คีย์หลักของรีเลชั่นจะมีค่าข้อมูลเป็นค่าว่างไม่ได้เสมอ เพราะจะทำให้การเข้าถึงข้อมูลใน ทัพเพิล (Tuple) นั้นกระทำไม่ได้

เมื่อมีการจัดเก็บข้อมูลในฐานข้อมูลใด ๆ แล้ว ข้อมูลจะถูกแยกออกเป็นกลุ่มของข้อมูลเป็นชุดที่ประกอบด้วยแอตทริบิวต์ต่าง ๆ ที่มีความสัมพันธ์กัน เช่นการเก็บข้อมูลของบุคลากรในโรงเรียนก็อาจแยกเก็บเป็นกลุ่มของนักเรียน กลุ่มข้อมูลของครูอาจารย์ และกลุ่มนักการภารโรง เป็นต้น กลุ่มข้อมูลแต่ละกลุ่มนี้จะเรียกว่า เอนติตี้ (Entity) ซึ่งแต่ละเอนติตี้จะประกอบไปด้วยแอตทริบิวต์ต่าง ๆ ที่มีความสัมพันธ์กัน เช่น เอนติตี้ของนักเรียนก็จะประกอบไปด้วย ชื่อ ที่อยู่ ชั้นเรียน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการแยกจัดเก็บข้อมูลออกเป็นเอนติตี้นี้ แต่ละเอนติตี้ก็จะมีความสัมพันธ์กันความสัมพันธ์ระหว่างเอนติตี้สามารถแบ่งออกเป็น 3 ชนิด คือ

ก. ความสัมพันธ์แบบหนึ่ง ต่อ หนึ่ง หมายความถึง เมื่อเอนติตี้หนึ่งมีข้อมูลของคีย์หลักค่าหนึ่ง ค่าข้อมูลดังกล่าวก็จะมีความสัมพันธ์กับค่าข้อมูลของคีย์หลักของอีกเอนติตี้หนึ่งเพียงค่าเดียวเท่านั้น เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่าง เอนติตี้นักเรียน กับ เอนติตี้ผู้ปกครอง เป็นแบบหนึ่งต่อหนึ่งแล้ว หมายความว่า การที่เราจะอ้างอิงถึงนักเรียนคนใดคนหนึ่งก็จะสามารถอ้างอิงถึงผู้ปกครองได้เพียงคนเดียวเท่านั้น และในทางตรงกันข้ามก็ต้องเป็นจริงด้วย คือ เมื่อเราอ้างอิงถึงผู้ปกครองคนใดคนหนึ่งแล้วก็สามารถอ้างอิงถึงนักเรียนได้เพียงคนเดียวเท่านั้น ดังตารางที่ 2.6

ตารางที่ 2.6 ตัวอย่างความสัมพันธ์ระหว่างนักเรียนกับผู้ปกครอง

ชื่อนักเรียน	ชื่อผู้ปกครอง
ก	A
ข	B
ค	C

ข.ความสัมพันธ์แบบหนึ่งต่อกลุ่ม หมายถึง เมื่อเอนติตี้หนึ่งมีข้อมูลของคีย์หลักค่าหนึ่งค่าข้อมูลดังกล่าวก็จะมีความสัมพันธ์กับข้อมูลของคีย์หลักของอีกเอนติตี้หนึ่งได้หลายค่า เช่น หากเรากำหนดความสัมพันธ์ระหว่าง เอนติตี้เลขประจำตัวผู้ป่วยกับเอนติตี้ชื่อ โรคเป็นแบบหนึ่งต่อกลุ่มแล้วจะหมายความว่า การที่เราอ้างอิงถึงเลขประจำตัวผู้ป่วยหนึ่งก็จะสามารถอ้างอิงถึงรหัสโรคได้เพียงรหัสเดียวเท่านั้นและในทางตรงกันข้ามก็จะมีความหมายว่า เมื่อเราอ้างอิงถึงรหัสโรคหนึ่งแล้วก็สามารถอ้างอิงถึงเลขประจำตัวผู้ป่วยได้หลายค่า ดังตารางที่ 2.7

ตารางที่ 2.7 ตัวอย่างความสัมพันธ์ระหว่างผู้ป่วยกับรหัสโรค

เลขประจำตัวผู้ป่วย	รหัสโรค
43000001	J059
43000002	J059
43000003	V060

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.ความสัมพันธ์แบบกลุ่มต่อกลุ่ม หมายความว่า ค่าข้อมูลของคีย์หลักของแอนติตี้หนึ่งที่ต่างกันอาจอ้างอิงถึงค่าข้อมูลของคีย์หลักของอีกเอนติตี้หนึ่งได้ค่าเดียวหรือหลายค่าก็ได้ เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่างเอนติตี้นักเรียนกับเอนติตี้ผู้ปกครองเป็นแบบกลุ่มต่อกลุ่มแล้วก็จะหมายความว่า การที่เราอ้างถึงนักเรียนคนหนึ่งหรือหลายคนก็จะสามารถอ้างอิงถึงผู้ปกครองคนเดียวกันได้ และในทางกลับกัน การที่เราอ้างถึงผู้ปกครองคนหนึ่งหรือหลายคนก็จะสามารถอ้างอิงถึงนักเรียนคนเดียวกันก็ได้ ดังตารางที่ 2.8

ตารางที่ 2.8 ตัวอย่างความสัมพันธ์ระหว่างนักเรียนกับผู้ปกครอง

ชื่อนักเรียน	ชื่อผู้ปกครอง
ก	A
ข	A
ค	C
ค	D

นอกจากนี้ เอนติตี้ยังมีเอนติตี้บางประเภทที่การอ้างถึงตัวมันได้อย่างสมบูรณ์นั้นจะต้องอ้างถึงเอนติตี้อื่นเสมอ เช่นหากระบบฐานข้อมูลของบุคลากรในโรงเรียนมีเอนติตี้ของลูกนักการภารโรงด้วยแล้ว การอ้างอิงถึงเอนติตี้นี้ได้จะต้องอ้างถึงเอนติตี้นักการภารโรงด้วยเสมอเอนติตี้ประเภทนี้เราเรียกว่า เอนติตี้ชนิดก่อน (Weak Entity)

2.5 การนอร์มัลไลซ์ (Normalization)

การนอร์มัลไลซ์เป็นการออกแบบฐานข้อมูลแบบที่เป็นมาตรฐานที่สุด ออกแบบโดย Codd โดยมีจุดประสงค์ของการออกแบบก็เพื่อลดความซ้ำซ้อนของความสัมพันธ์ของข้อมูลให้เหลือน้อยที่สุด (Minimum Redundancy) ซึ่งตามมาตรฐานปกติจะอยู่ 3 ระดับ คือ

- 1 NF (First Normal Form)
- 2 NF (Second Normal Form)
- 3 NF (Third Normal Form)

โดยที่รีเลชั่นใดที่ยังไม่สอดคล้องตามรูปแบบนอร์มัล (Normal Form) ทั้งสามก็จะต้องมีการแยกรีเลชั่นนั้น ๆ ออกเป็นรีเลชั่นย่อย ๆ ต่อไปอีก (Decomposition Method) ต่อมาได้มีการออกแบบเพิ่มเติมขึ้นอีก 2 ระดับ คือ

เอกสารนี้มีลิขสิทธิ์ ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4 NF(Forth Normal Form)

- 5 NF (Fifth Normal Form)

หากรีเลชั่นใดมีมาตรฐานถึงรูปแบบนอร์มัลระดับ 5 (5NF) แล้วก็จะมั่นใจได้ว่า รีเลชั่นนั้นจะไม่มีความซ้ำซ้อนของความสัมพันธ์ของข้อมูลอย่างแน่นอน

นอกจากนี้ยังมีการออกแบบรูปแบบนอร์มัลเพิ่มเติมระหว่างรูปแบบนอร์มัลที่ 3 (3NF) และรูปแบบที่ 4 (4NF) โดย Boyce Codd Normal Form (BCNF) อีกด้วย

2.5.1 รูปแบบนอร์มัลระดับที่ 1

การปรับรีเลชั่นให้อยู่ในรูปแบบนอร์มัลระดับที่ 1 คือ การปรับจากรีเลชั่นที่ไม่นอร์มัล (Umnormalized Relation) เช่นรีเลชั่นที่มีข้อมูลของแอตทริบิวต์บางช่องมีมากกว่า 1 ค่า (มีแอตทริบิวต์ที่มีข้อมูลเป็น Repeating Group)

นิยามรีเลชั่นจะอยู่ในรูปแบบนอร์มัลระดับที่ 1 (1NF) ก็ต่อเมื่อโดเมนของแต่ละแอตทริบิวต์ ประกอบด้วยข้อมูลที่เป็นหน่วยย่อยที่สุด สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 1 ก็คือ รีเล-ชั่นยังมีความซ้ำซ้อนของความสัมพันธ์ระหว่างข้อมูลอยู่มากมาย เพราะนิยามของรูปแบบนอร์มัลระดับที่ 1 นี้ กำหนดเพียงเฉพาะว่า แต่ละแอตทริบิวต์ของรีเลชั่นจะมีโดเมนที่มีสมาชิกเป็นหน่วยที่เล็กที่สุดเท่านั้น มิได้เป็นการลดความสัมพันธ์ระหว่างข้อมูลแต่ประการใด

2.5.2 รูปแบบนอร์มัลระดับที่ 2

นิยาม รีเลชั่นจะอยู่ในรูปแบบนอร์มัลระดับที่ 2 (2NF) ก็ต่อเมื่อรีเลชั่นนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 1 แล้ว และทุกแอตทริบิวต์ที่ไม่เป็นส่วนใดส่วนหนึ่งของคีร์หลัก (Non-Key attribute) จะต้องขึ้นอยู่กับคีย์หลักของรีเลชั่นนั้นอย่างสมบูรณ์ สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 2 คือ ข้อมูลบางแอตทริบิวต์ที่ไม่ใช่คีย์หรืออาจมีความสัมพันธ์กันเองโดยที่ไม่มีความสัมพันธ์กับคีย์หลักเลย ซึ่งความสัมพันธ์ดังกล่าวนี้ถือว่าเป็นความซ้ำซ้อนประการหนึ่งของรีเลชั่นนั้น ๆ ที่จะต้องทำการลดด้วยรูปแบบนอร์มัลในระดับต่อไป

2.5.3 รูปแบบนอร์มัลระดับที่ 3

นิยาม รีเลชั่นจะอยู่ในรูปแบบนอร์มัลระดับที่ 3 (3NF) ก็ต่อเมื่อรีเลชั่นนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 2 แล้ว และทุกแอตทริบิวต์ที่ไม่เป็นส่วนใดส่วนหนึ่งของคีย์หลักจะต้องไม่เป็นฟังก์ชันที่ขึ้นต่อกันเอง โดยปกติแล้ว สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 3 คือ รีเลชั่นจะไม่มีความซ้ำซ้อนอีกต่อไปโดยที่จะสอดคล้องกับรูปแบบนอร์มัลระดับที่ 4 และ 5 ด้วย แต่ก็มีรีเลชั่นบางลักษณะที่จะต้องทำให้อยู่ในรูปแบบนอร์มัลระดับที่ 4 และ 5 ต่อไป รีเลชั่น ดังกล่าวจะมีลักษณะดังต่อไปนี้

1) เป็นรีเลชั่นที่มีหลายคีย์คู่แข่ง และเป็นคีย์คู่แข่งที่เกิดจากการรวมตัวกันของคีย์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Candidate Key เป็น Combine Key) และ

2) เป็นคีย์คู่แข่งที่มีการเหลื่อมซ้อนกัน (Overlap กัน)

2.5.4 รูปแบบบอยส์และคอดต์

นิยาม รีเลชั่นใด ๆ จะจัดอยู่ในรูปแบบบอยส์และคอดต์ (BCNF) ถ้าทุก ๆ ตัวเลือก (Determinant) เป็นคีย์คู่แข่ง (Candidate Key) หรืออาจกล่าวได้ว่าทุกแอตทริบิวต์ที่เป็นคีย์หลักของรีเลชั่นจะต้องไม่ขึ้นกับเซตย่อยของคีย์หลักตัวอื่น

2.5.5 รูปแบบนอร์มัลระดับที่ 4

นิยาม รีเลชั่นใด ๆ จะจัดอยู่ในรูปแบบนอร์มัลระดับที่ 4 (4NF) ถ้ารีเลชั่นนั้นเป็น BCNF และเป็นรีเลชั่นไม่มีการขึ้นต่อกันเชิงกลุ่ม (ต้องไม่มี Multi-Valued Dependency (MVD) เกิน 1 ตัว)

2.5.6 รูปแบบนอร์มัลระดับที่ 5

นิยาม รีเลชั่นใด ๆ จะจัดอยู่ในรูปแบบนอร์มัลระดับที่ 5 (5NF) ถ้ารีเลชั่นนั้นเป็น 4NF และผลลัพธ์ของการ Join Dependency (JD) ต้องเหมือนเดิม ไม่เกินหรือสูญหาย (Lossless Join)

การอ้างอิงถึงเอนติตี้นี้ได้จะต้องอ้างถึงเอนติตี้นักการภารโรงด้วยเสมอเอนติตี้ประเภทนี้เราเรียกว่า เอนติตี้ชนิดก่อน (Weak Entity)

2.5 การนอร์มัลไลซ์ (Normalization) การนอร์มัลไลซ์เป็นการออกแบบฐานข้อมูลแบบที่เป็นมาตรฐานที่สุด ออกแบบโดย Codd โดยมีจุดประสงค์ของการออกแบบก็เพื่อลดความซ้ำซ้อนของความสัมพันธ์ของข้อมูลให้เหลือน้อยที่สุด (Minimum Redundancy) ซึ่งตามมาตรฐานปกติจะอยู่ 3 ระดับ คือ

- 1 NF (First Normal Form)
- 2 NF (Second Normal Form)
- 3 NF (Third Normal Form)

โดยที่รีเลชั่นใดที่ยังไม่สอดคล้องตามรูปแบบนอร์มัล (Normal Form) ทั้งสามก็จะต้องมีการแยกรีเลชั่นนั้น ๆ ออกเป็นรีเลชั่นย่อย ๆ ต่อไปอีก (Decomposition Method) ต่อมาได้มีการออกแบบเพิ่มเติมขึ้นอีก 2 ระดับ คือ

- 4 NF(Forth Normal Form)
- 5 NF (Fifth Normal Form)

หากรีเลชั่นใดมีมาตรฐานถึงรูปแบบนอร์มัลระดับ 5 (5NF) แล้วก็จะมั่นใจได้ว่า รีเลชั่นนั้นจะไม่มีความซ้ำซ้อนของความสัมพันธ์ของข้อมูลอย่างแน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังมีการออกแบบรูปแบบนอร์มัลเพิ่มเติมระหว่างรูปแบบนอร์มัลที่ 3 (3NF) และรูปแบบที่ 4 (4NF) โดย Boyce Codd Normal Form (BCNF) อีกด้วย

นอกจากนี้ยังมีการออกแบบรูปแบบนอร์มัลเพิ่มเติมระหว่างรูปแบบนอร์มัลที่ 3 (3NF) และรูปแบบที่ 4 (4NF) โดย Boyce Codd Normal Form (BCNF) อีกด้วย

2.5.1 รูปแบบนอร์มัลระดับที่ 1

การปรับรีเลชั่นให้อยู่ในรูปแบบนอร์มัลระดับที่ 1 คือ การปรับจากรีเลชั่นที่ไม่นอร์มัล (Umnormalized Relation) เช่นรีเลชั่นที่มีข้อมูลของแอตทริบิวต์บางช่องมีมากกว่า 1 ค่า (มีแอตทริบิวต์ที่มีข้อมูลเป็น Repeating Group)

นิยามรีเลชั่นจะอยู่ในรูปแบบนอร์มัลระดับที่ 1 (1NF) ก็ต่อเมื่อโดเมนของแต่ละแอตทริบิวต์ ประกอบด้วยข้อมูลที่เป็นหน่วยย่อยที่สุด สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 1 ก็คือ รีเล-ชั่นยังมีความซ้ำซ้อนของความสัมพันธ์ระหว่างข้อมูลอยู่มากมาย เพราะนิยามของรูปแบบนอร์มัลระดับที่ 1 นี้ กำหนดเพียงเฉพาะว่า แต่ละแอตทริบิวต์ของรีเลชั่นจะมีโดเมนที่มีสมาชิกเป็นหน่วยที่เล็กที่สุดเท่านั้น มิได้เป็นการลดความสัมพันธ์ระหว่างข้อมูลแต่ประการใด

2.5.2 รูปแบบนอร์มัลระดับที่ 2

นิยาม รีเลชั่นจะอยู่ในรูปแบบนอร์มัลระดับที่ 2 (2NF) ก็ต่อเมื่อรีเลชั่นนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 1 แล้ว และทุกแอตทริบิวต์ที่ไม่เป็นส่วนใดส่วนหนึ่งของคีร์หลัก (Non-Key attribute) จะต้องขึ้นอยู่กับคีย์หลักของรีเลชั่นนั้นอย่างสมบูรณ์ สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 2 คือ ข้อมูลบางแอตทริบิวต์ที่ไม่ใช่คีย์หรืออาจมีความสัมพันธ์กันเองโดยที่ไม่มีความสัมพันธ์กับคีย์หลักเลย ซึ่งความสัมพันธ์ดังกล่าวนี้ถือว่าเป็นความซ้ำซ้อนประการหนึ่งของรีเลชั่นนั้น ๆ ที่จะต้องทำการลดด้วยรูปแบบนอร์มัลในระดับต่อไป

2.5.3 รูปแบบนอร์มัลระดับที่ 3

นิยาม รีเลชั่นจะอยู่ในรูปแบบนอร์มัลระดับที่ 3 (3NF) ก็ต่อเมื่อรีเลชั่นนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 2 แล้ว และทุกแอตทริบิวต์ที่ไม่เป็นส่วนใดส่วนหนึ่งของคีย์หลักจะต้องไม่เป็นฟังก์ชันที่ขึ้นต่อกันเอง โดยปกติแล้ว สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 3 คือ รีเลชั่นจะไม่มีความซ้ำซ้อนอีกต่อไปโดยที่จะสอดคล้องกับรูปแบบนอร์มัลระดับที่ 4 และ 5 ด้วย แต่ก็มีรีเลชั่นบางลักษณะที่จะต้องทำให้อยู่ในรูปแบบนอร์มัลระดับที่ 4 และ 5 ต่อไป รีเลชั่น ดังกล่าวจะมีลักษณะดังต่อไปนี้

1) เป็นรีเลชั่นที่มีหลายคีย์คู่แข่ง และเป็นคีย์คู่แข่งที่เกิดจากการรวมตัวกันของคีย์ (Candidate Key เป็น Combine Key) และ

2) เป็นคีย์คู่แข่งที่มีการเหลื่อมซ้อนกัน (Overlap กัน)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 รูปแบบบอยส์และคอดต์

นิยาม รีเลชั่นใด ๆ จะจัดอยู่ในรูปแบบบอยส์และคอดต์ (BCNF) ถ้าทุก ๆ ตัวเลือก (Determinant) เป็นคีย์คู่แข่ง (Candidate Key) หรืออาจกล่าวได้ว่าทุกแอตทริบิวต์ที่เป็นคีย์หลักของรีเลชั่นจะต้องไม่ขึ้นกับเซตย่อยของคีย์หลักตัวอื่น

2.5.5 รูปแบบนอร์มัลระดับที่ 4

นิยาม รีเลชั่นใด ๆ จะจัดอยู่ในรูปแบบนอร์มัลระดับที่ 4 (4NF) ถ้ารีเลชั่นนั้นเป็น BCNF และเป็นรีเลชั่นไม่มีการขึ้นต่อกันเชิงกลุ่ม (ต้องไม่มี Multi-Valued Dependency (MVD) เกิน 1 ตัว)

2.5.6 รูปแบบนอร์มัลระดับที่ 5

นิยาม รีเลชั่นใด ๆ จะจัดอยู่ในรูปแบบนอร์มัลระดับที่ 5 (5NF) ถ้ารีเลชั่นนั้นเป็น 4NF และผลลัพธ์ของการ Join Dependency (JD) ต้องเหมือนเดิม ไม่เกินหรือสูญหาย (Lossless Join)

2.6 Concurrency Control

ปัญหาสำคัญในการบริการข้อมูลกับ Client ของ Server คือ การสนับสนุนข้อมูลบนฐานข้อมูลที่มีผู้ใช้ร่วมกันมากว่า 1 ราย ความขัดแย้งหรือความต้องการในการใช้ข้อมูลในแหล่งเดียวกันมีโอกาสเกิดขึ้นเป็นอย่างมาก เทคนิคที่นำมาใช้จัดลำดับหรือลดความขัดแย้งนี้เรียกว่า Concurrency Control ซึ่งได้อธิบายการแก้ไขปัญหานี้ไว้ 3 วิธีคือ

1. Locks เป็นกติกาการใช้ข้อมูลของผู้ใช้ หากผู้ใช้คนใด Lock ข้อมูลได้ก่อน จะมีสิทธิใช้ข้อมูลก่อนเป็นต้น

2. Optimistic Concurrency Control เป็นเทคนิคที่อนุญาตให้ Transaction ใดๆ ที่ดำเนินการอยู่ก่อนสามารถทำต่อไปได้จนกว่าจะ Conmit และ Travsaction อื่นๆ จึงจะทำต่อเนื่องได้ ที่จัดระบบในการใช้ข้อมูลร่วมกัน

3. Timestamp ordering เป็นการจัดลำดับการใช้ข้อมูลโดยอาศัยเวลาเริ่มแรกของ Transaction เป็นเกณฑ์และดำเนินการตามเวลาก่อน-หลัง ของ Transaction นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.4 แสดงการ Lock Table บนฐานข้อมูล Timestamp T และ U

Transaction T: Bank$Withdraw (A,4) Bank$Deposit(B,4)		**Transaction U:** Bank$Withdraw (C,3) Bank$Deposit(B,3)	
Operations	Locks	Operations	Locks
OpenTransaction			
balance := A.Read()	lock A		
A.Write(balance –4)			
		OpenTransaction	
		balance := C.Read()	lock C
		C.Write(balance –3)	
balance := B.Read()	lock B		
		balance := A.Read()	wait for T's lock ,on B
B.Write(balance +4)		*	
CloseTransaction	unlocks A,B	*	
		*	locks B
		B.Write(balance +3)	
		CloseTransaction	unlocks B,C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.5 แสดงการขัดแย้งของการ READ/Write บน Transaction ใดๆ

Operations of Transactions	different	Conflict	Reason
Read	Read	No	Because the effect of a pair of Read operations does not depend on order in which they are executed
Read	Write	Yes	Because the effect of a Read and a Write operation Depends on the order of their execution
Write	Write	Yes	Because the effect of a pair of Write operations Depends on order of their executed

Locks

เป็นกติกาการ Read / Write โดยมีกฎการใช้ข้อมูลดังนี้

1. ถ้า Transaction T กำลัง Read , Transaction U จะมี Write จนกว่า T จะ Commit หรือยกเลิก

2. ถ้า Transaction T กำลัง Write , Transaction U จะมี Read / Write จนกว่า T จะ Commit หรือยกเลิก

ภาพที่ 2.6 แสดงการ Lock ของ Transaction ใดๆ

For one data item		Lock requested	
		Read	Write
Lock already	None	OK	OK
	Read	OK	Wait
	Write	Wait	Wait

Deadlocks เป็นเหตุการณ์ที่ 2 Transactions ต่างฝ่ายต่างรอกันในการใช้แหล่งข้อมูลเดียวกัน ซึ่งสมารถป้องกันได้โดยวิธีการง่ายคือ ถ้า Transaction ใด Lock Table ไว้ก่อนแล้ว Transaction อื่นๆ ก็ไม่ควร Lock Table นั้น จนกว่าจะสามารถ Lock ได้

สำหรับการแก้ไขปัญหากรณี Deadlock มีด้วยกันหลายวิธีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. The wait-for-graph เป็น Cyele การเกิด Deadlock ที่ Server จะเป็นผู้เลือก Transaction ที่เป็นมูลเหตุของปัญหา และ Break Transaction นั้นเอง

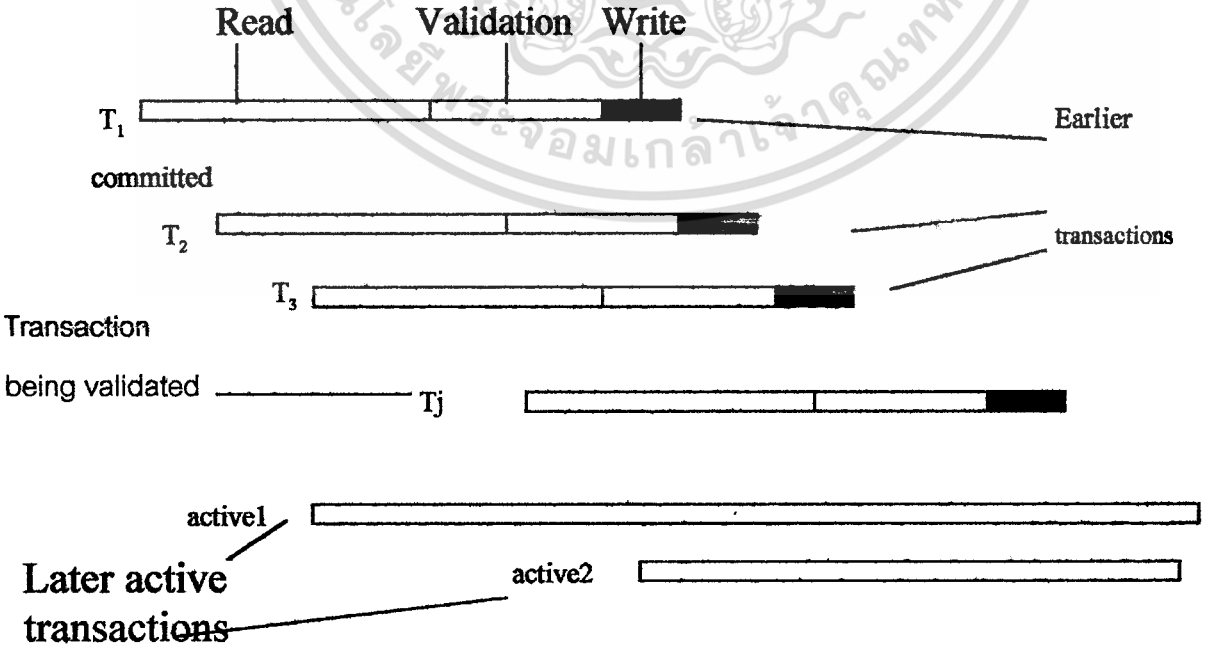
2. Timeouts เป็นวิธีการธรรมดาที่นิยมใช้ เพราะง่ายและสะดวกโดยกำหนดเวลาที่ใช้ในการ Locks หาก Locks ไม่ได้ Transaction จะถูกยกเลิกไป และในขณะเดียวกัน Transaction ที่ดำเนินอยู่หากใช้เวลามากเกินกำหนดจะถูกยกเลิกเช่นกัน

Optimistic concurrency control

เป็นเทคนิคที่พิจารณา Transaction ที่ดำเนินอยู่ (Validation) ให้ดำเนินต่อไปจนกว่าจะ Commit และเรียงลำดับ Transaction ตามตารางแสดงความสัมพันธ์ของ Transaction Ti และ Tj ดังนี้

Ti	Tj	Rule
Read	Write	1. Tj must not read data items written by Tj
Write	Read	2. Tj must not read data items written by Ti
Write	Write	3. Ti must not write data items written by Tj and Tj must not write data items written by Ti.

ภาพที่ 2.7 Validation of Transaction



เทคนิคการเลือก Validation of Transaction ที่มีอยู่ 2เทคนิค คือ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Backward Validation เป็นการเลือก Transaction จากสุดท้ายย้อนกลับมา Transaction แรกตาม อัลกอริธึม

```
Vaild := TRUE;
FOR Ti := startTn + I  TO  finishTn  DO
        IF read of Tj intersects write of Ti THEN
                Valid:= FALSE
        END
END
```

2. Forword Validation เป็นการเลือก Transaction ที่ Action จาก Transaction แรกไปยัง Transaction สุดท้าย ตามอัลกอริธึม

```
Vaild := TRUE;
FOR Tid := dctivel  TO  activeN  DO
        IF write set of  Tj intersects read set of Tid THEN
                Valid:= FALSE
        END
END
```

Timestamp ordering

เป็นเทคนิคการจัดลำดับ Transaction ที่ได้รับตามลำดับก่อนหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.8 Transaction conflicts for timestamp ordering

Rule	Tj	
1.	Write	Tj must not write a data item that has been read by any Ti where Ti > Tj this requires that Tj=> the maximum read timestamp of the data item
2.	Wirte	Tj must not write a data item that has been written by any Ti where Ti > Tj this requires that Tj > the maximum write timestamp of the Committed data item
3.	Read	Tj must not read a data item that has been written by any Ti where Ti > Tj this implies that Tj cannot read Tj < write timestamp of the Committed version of the data item

<u>เปรียบเทียบวิธีการแก้ปัญหาโดย Concurrency control Model</u>

จากการแก้ไขปัญหาโดย Concurrency control Model ทั้ง 3 วิธี มีจุดดี-ด้อยดังนี้

1. Timestamp Ordering เป็นวิธีแบบธรรมดาโดยพิจารณาการ Locks แบบ 2 จังหวะ และ

เรียงลำดับ Transaction ตามข้อมูลตามสถิติที่ได้รับ ข้อดีคือง่ายในการจัดการแต่มีปัญหาในเรื่องการจัดลำดับของ Transaction ว่าจะใช้เกณฑ์จากสื่อกลางใด เนื่องจากกลไกสำคัญเป็นเรื่องของเวลาซึ่งอาจจะประสบปัญหาเวลาที่ไม่เท่าเทียบกันได้

2. Optimistic concurrency control ถูกใช้กับ Transaction ที่ดำเนินอยู่ในการกระทำต่อไปได้จนกว่าจะ Commit ใช้ได้ดีมีข้อมูลนั้นถูกใช้น้อยครั้ง แต่เข้ามี Transaction มากจนทำให้เกิดการล่าช้าและถ้า Transaction ที่ Actuie อยู่ เกิดปัญหาขึ้น จนกระทำซ้ำ จะยังเสียเวลาทั้งระบบมากขึ้น

3. Locks ถูกใช้มาเป็นเวลานาน และนิยมใช้ใน File Server ร่วมกันเทคนิคแบบ Timestamp Ordering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การพัฒนาระบบและโปรแกรม

โครงการพัฒนาระบบนี้ เป็นการสร้างผลผลิตในระดับ Implementation โดยไม่ได้แสดงรายละเอียดทุกขั้นตอนใน SDLC แต่จะแสดงองค์ประกอบที่จำเป็นและสำคัญเพื่อให้บรรลุวัตถุประสงค์ตามที่กล่าวไว้ในบทที่ 1 ซึ่งองค์ประกอบเหล่านี้ประกอบด้วยส่วนต่างๆ ดังนี้

3.1 ฮาร์ดแวร์และเน็ตเวิร์ค (Hardware and Network)

ได้ออกแบบฮาร์ดแวร์และเน็ตเวิร์คเป็นลักษณะการทำงานแบบ File Server เนื่องจากใช้งบประมาณต่ำที่สุดในระบบการทำงานแบบเครือข่าย ที่องค์กรขนาดเล็กสามารถจัดหาได้อย่างสะดวก และองค์กรขนาดใหญ่สามารถใช้ผลการศึกษานี้ร่วมด้วยได้ เพื่อพัฒนาระบบสารสนเทศของหน่วยงานที่เหมาะสมต่อไป

ใช้เทคโนโลยีการเชื่อมต่อเน็ตเวิร์คแบบ LAN (Local Area Network) โดยมี File Server ที่มีฮาร์ดดิสก์ (Harddisk) สำหรับเก็บข้อมูลขนาดใหญ่ (มีระบบสำรองข้อมูลหรือไม่ก็ได้ เนื่องจากโครงการฯ ไม่ได้พิจารณาระบบ Fault Tolerance ร่วมด้วย และ Workstation เชื่อมต่อกับ File Server ผ่านระบบ Network สำหรับคุณลักษณะเฉพาะของ File Server และ Workstation ให้เป็นไปตามที่ Operating System (OS) กำหนดไว้ขั้นต่ำ

3.2 ซอฟต์แวร์ (Software)

แบ่งออกเป็น 3 ส่วนคือ

(1) ซอฟต์แวร์ระบบปฏิบัติการ (Operating System ; OS) สามารถเลือกใช้ OS ในระบบ LAN ได้ 2 แบบคือ Novell Netware หรือ Microsoft Windows 2000 Server สำหรับ Workstation สามารถใช้ OS ที่สามารถติดต่อกับ OS ของ File Server ได้ ในโครงการฯ ได้เลือกใช้ Microsoft Windows NT (Server) Version 4.0 สำหรับ File Server และ Microsoft Windows Version 3.11 Workgroup สำหรับ Workstation

(2) ซอฟต์แวร์จัดฐานข้อมูล (Database Management System ; DBMS) ได้เลือกใช้ฐานข้อมูลในตระกูล X-BASE ได้แก่ FOXBASE / FOXPRO ซึ่งมีลักษณะการจัดการแบบ File System และสร้าง Relation ของ Datafile ตามหลักการของ RDBMS (Relational Database

เอกสารนี้มีลิขสิทธิ์ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Management System) ติดตั้งไว้ที่ฮาร์ดดิสก์ของ File Server

(3) ซอฟต์แวร์ประยุกต์ (Application Software) ใช้ Microsoft FOXPRO เป็น Application Developer เนื่องจากเป็น DML (Data Manipulation Language) ของ FOXBASE / FOXPRO Database System ซึ่งต้องจัดระบบ Concurrency Control เองในการเขียนโปรแกรม

3.3 Concurrency Control Implementation

เนื่องจากข้อจำกัดด้านซอฟท์แวร์ระบบปฏิบัติการ(Operating System) ที่ถูกเลือกใช้ในโครงการนี้เป็น Novell Netware Version 3.11 และขีดจำกัดด้านงบประมาณ ทำให้โครงการต้องเลือกใช้ซอฟท์แวร์จัดการฐานข้อมูลตระกูล X-BASE ได้แก่ FOXBASE/FOXPRO ซึ่งมีลักษณะการจัดการแบบ File System ในลักษณะการทำงานแบบ File Server ซึ่ง Server จะสนับสนุน Application และ Data ให้กับ Workstation เพื่อประมวลผลหรือจัดการเกี่ยวกับข้อมูล และนำผลลัพธ์ไปแสดงที่ Output device หรือ Update ข้อมูลใน File Server

FOXBASE/FOXPRO ที่ใช้จัดการฐานข้อมูลในระบบ File Server นี้ เป็นตัวจัดการฐานข้อมูลที่ผู้ใช้ฐานข้อมูลทั่วไปเรียกว่า X-BASE เนื่องจากไม่สามารถจัดการการใช้ฐานข้อมูลร่วมกันของ Transaction ใดๆในระบบ Multiuser ได้ด้วยตนเอง ตัวจัดการที่ X-BASE จัดการไม่ได้นี้เรียกว่า Concurrency Control Manager

องค์ประกอบของ Concurrency Control Manager ประกอบด้วยขบวนการ 4 อย่างคือ

(1) Lock-Base Protocols เป็นขบวนการทำงานหรือติดต่อกับ Data item ใดๆบน Based ณ ขณะใดๆของ Transaction หนึ่งๆแล้วจะไม่อนุญาตให้ Transaction อื่นมากระทำกับ Data item นั้นๆได้

(2) Deadlock Prevention เป็นขบวนการป้องกันการเกิดปัญหา Deadlock ของระบบไม่ให้เกิดการรอติดต่อกับ Based แบบต่อเนื่อง และไม่สามารถยกเลิกการรอนั้นๆได้

(3) Deadlock Detection เป็นขบวนการแก้ปัญหาเมื่อเกิดปัญหา Deadlock ขึ้น เพื่อให้ระบบสามารถทำงานต่อไปได้

(4) Recovery เป็นขบวนการเรียกคืนข้อมูลกลับไปสู่สภาพเดิมเมื่อ Transaction นั้นๆไม่สามารถดำเนินไปจนสิ้นสุดขบวนการ

ทั้ง 4 ขบวนการนี้เป็นจุดบอดและจุดด้อยที่ X-BASE ใช้เป็นขอบเขตและข้อตกลงที่จะไม่พิจารณาในส่วนนี้ เพื่อผลิตซอฟท์แวร์ฐานข้อมูลให้กับผู้ใช้ที่ไม่มีความจำเป็นหรือไม่มีความเสี่ยงจากปัญหาเหล่านี้ อันจะส่งผลให้ต้นทุนในการผลิตซอฟท์แวร์ลดต่ำลงจนเป็นที่ยอมรับในผู้ใช้กลุ่มหนึ่ง แม้ว่า X-BASE จะมีข้อจำกัดในการแก้ปัญหา Concurrency Control ก็ตาม แต่ยังได้เปิดโอกาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ Programmer สามารถพัฒนา Application เพื่อแก้ปัญหานี้ สอดแทรกไว้ในโปรแกรมได้ จึงเสมือนว่ามีซอฟท์แวร์ช่วยจัดการฐานข้อมูลส่วนหนึ่งผสมอยู่ใน Application เพื่อทำให้ระบบแก้ปัญหาด้าน Concurrency Control ได้

ผู้พัฒนาโครงการได้วิเคราะห์และทดลองพฤติกรรมต่างๆของ FOXBASE/FOXPRO ทั้งที่ทำงานบน MS-DOS และ MS-WINDOWS พบว่ามีลักษณะการทำงานแบบเดียวกัน คือเป็น Automatically Generate Locked (Shared-mode) เมื่อเรียกผ่าน Database Software exeuted ที่เป็น Multiuser เช่น MFOXPLUS.EXE เป็นต้น แม้ว่าบางระบบจะต้องส่งคำสั่ง Set Exclusive off เพื่อบอกกับ System ว่าต้องการทำงานในระบบ Multiuser ก็ตาม แต่ไม่เสมอไปเนื่องจากบางระบบจะ Set Default ไว้จึงทำให้ User ที่ติดต่อกับ File Server สามารถใช้ฐานข้อมูลได้ร่วมกัน ซึ่งเมื่อ User คนใดคนหนึ่งติดต่อกับฐานข้อมูล จะมีสถานะเป็น Read และ User คนอื่นๆก็สามารถอยู่ในสถานะ Read ได้พร้อมๆกัน

ในกรณีที่เกิดการ Write ข้อมูลของ Transaction ใดๆจะกระทำได้ก็ต่อเมื่อได้ Lock ข้อมูลนั้นๆแล้ว ผู้ใช้ที่ Lock ข้อมูลได้ก่อน จึงมีสิทธิ Update ข้อมูล เป็นลำดับต่อเนื่องกันไป และในขณะที่ข้อมูลถูก Lock อยู่นั้น ผู้ใช้ที่ Read ข้อมูลอย่างเดียว ยังสามารถ Read ข้อมูลนั้นๆได้ แต่เป็นข้อมูลก่อนการ Update จนกว่า Transaction ที่ Update ข้อมูลจะ Commit (โดยการ Unlock) จึงจะทำให้ข้อมูลที่ถูก Read หลังการ Commit เปลี่ยนแปลงไป

หลักการของ Concurrency Control บน X-BASE จึงไม่รับประกัน Conflict Serializability เนื่องจากระบบอนุญาตให้ Read ข้อมูลได้ตลอดเวลาที่ต้องการ และหากต้องการ Update จึงต้อง Lock ข้อมูล ณ ขณะนั้น ซึ่งมีความเป็นไปได้ว่า Transaction ที่ Read ก่อนอาจ Lock ข้อมูลภายหลัง Transaction ที่ Read ทีหลังแต่ Lock ข้อมูลได้ก่อน จึงทำให้การ Update ข้อมูลไม่เป็นไปตามความประสงค์ของ User แสดงโดยภาพต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transaction T1	Transaction T2
Read A ;(A=5)	Read A ;(A=5)
	Lock A
	Update A ;(A=10)
	Unlock A
Lock A	
Update A ;(A=20)	
Unlock A	
Read A ;(A=20)	Read A ;(A=20)

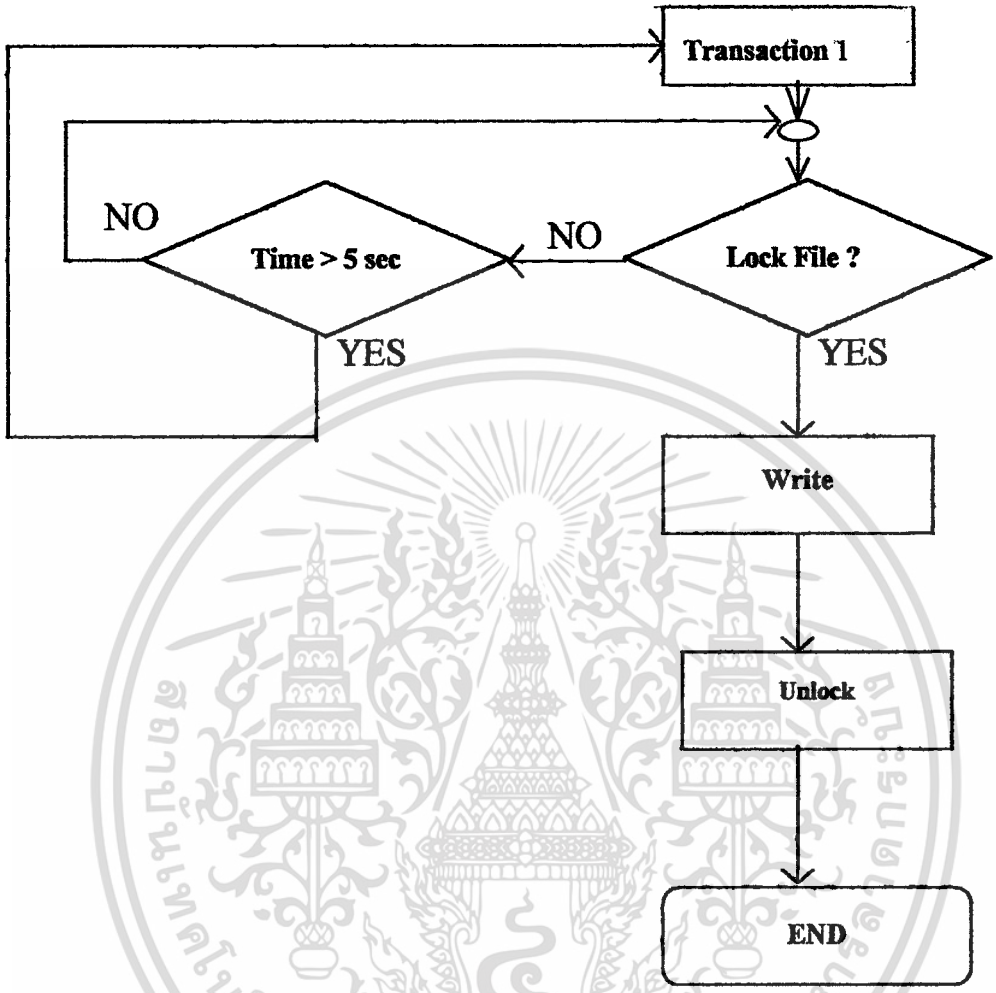
ภาพที่ 3.1 แสดงปัญหาการใช้ Concurrency Control บน X-BASE

การเพิ่มประสิทธิภาพของ Concurrency Control บน X-BASE เป็นปัจจัยหลักที่โครงการนำมาพิจารณาออกแบบและทดสอบเพื่อแก้ปัญหา จนสามารถสรุปได้ว่า เทคนิคที่ใช้เพิ่มขีดความสามารถของ Concurrency Control ที่เหมาะสมกับ Low Base คือ

(1)ใช้ Two-Phase Locking Protocol เป็นตัวจัดการการ Lock ข้อมูลบน Data item เดียวกัน และหาก Transaction ใด มีการ Write ร่วมอยู่ด้วยจะใช้ Exclusive mode (lock-X) เป็นตัว Lock ภายใต้ข้อตกลงที่ว่า Transaction ที่ Read ใดๆจะอ่านข้อมูลที่ยืนยันความถูกต้อง ณ ขณะที่ Transaction ตัวสุดท้ายได้ Unlock ข้อมูลแล้ว

(2)ใช้ Timeouts Limit เป็นตัวป้องกัน Deadlock (Deadlock Prevention) สำหรับกรณีที่ Lock ข้อมูลเพื่อ Write ไม่ได้ โดยกำหนด Timeouts Limit ไว้ที่ 5 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2 แสดงการ Lock ไฟล์ โดยใช้ Timeouts

(3)ใช้ Home Key เป็น User Attemption Control กรณีเกิด Deadlock (Deadlock Detection) แล้วไม่สามารถปลดโดยระบบได้ เมื่อใช้ Home Key แล้วให้ถือว่า Transaction นั้นๆถูกยกเลิก

(4)ไม่รับประกันการ Recovery ข้อมูล เนื่องจาก Transaction ที่ Write ข้อมูลจะต้อง Lock ข้อมูลได้ทั้งหมดก่อนการ Update และใช้ระบบไฟสำรองฉุกเฉิน (UPS & Stabilizer) ป้องกันการ Shutdown ของระบบ เพราะการ Rollback ใน Low Base ไม่สามารถกระทำได้กรณีเกิดการ System Suddenly Shutdown

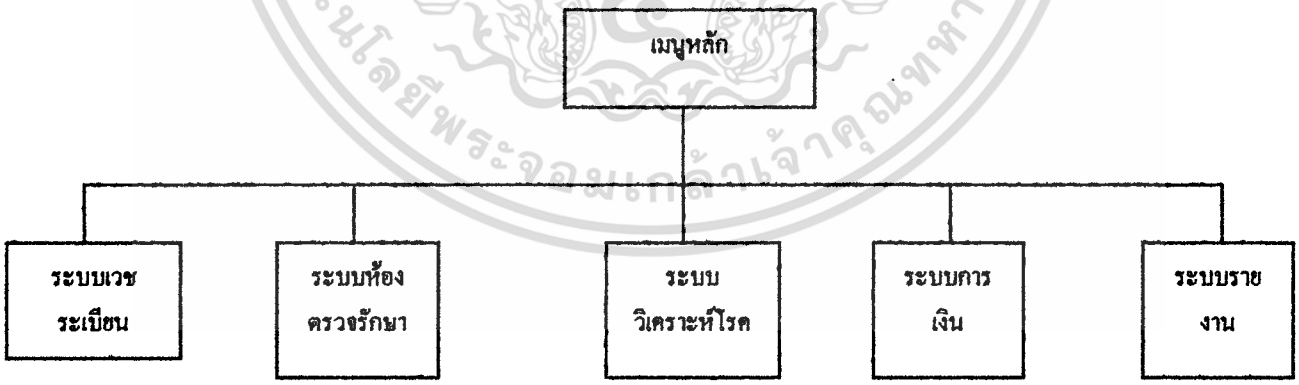
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transaction T1	Transaction T2	Transaction T3	Transaction T4	Transaction T5	Transaction T6	Transaction T7
Read A;(A=5)	Lock-X A					
	Read A;(A=5)				Lock-X A	
	Write A;(A=10)				(Timeouts Over	Lock-X A
	Unlock A				Limit Transaction	(Use Home
		Lock-X A		Read A;(A=10)	Cancel)	Key Attemption
		Read A;(A=10)				Transaction
		Write A;(A=20)				Cancel)
		Unlock A				
			Read A;(A=20)			

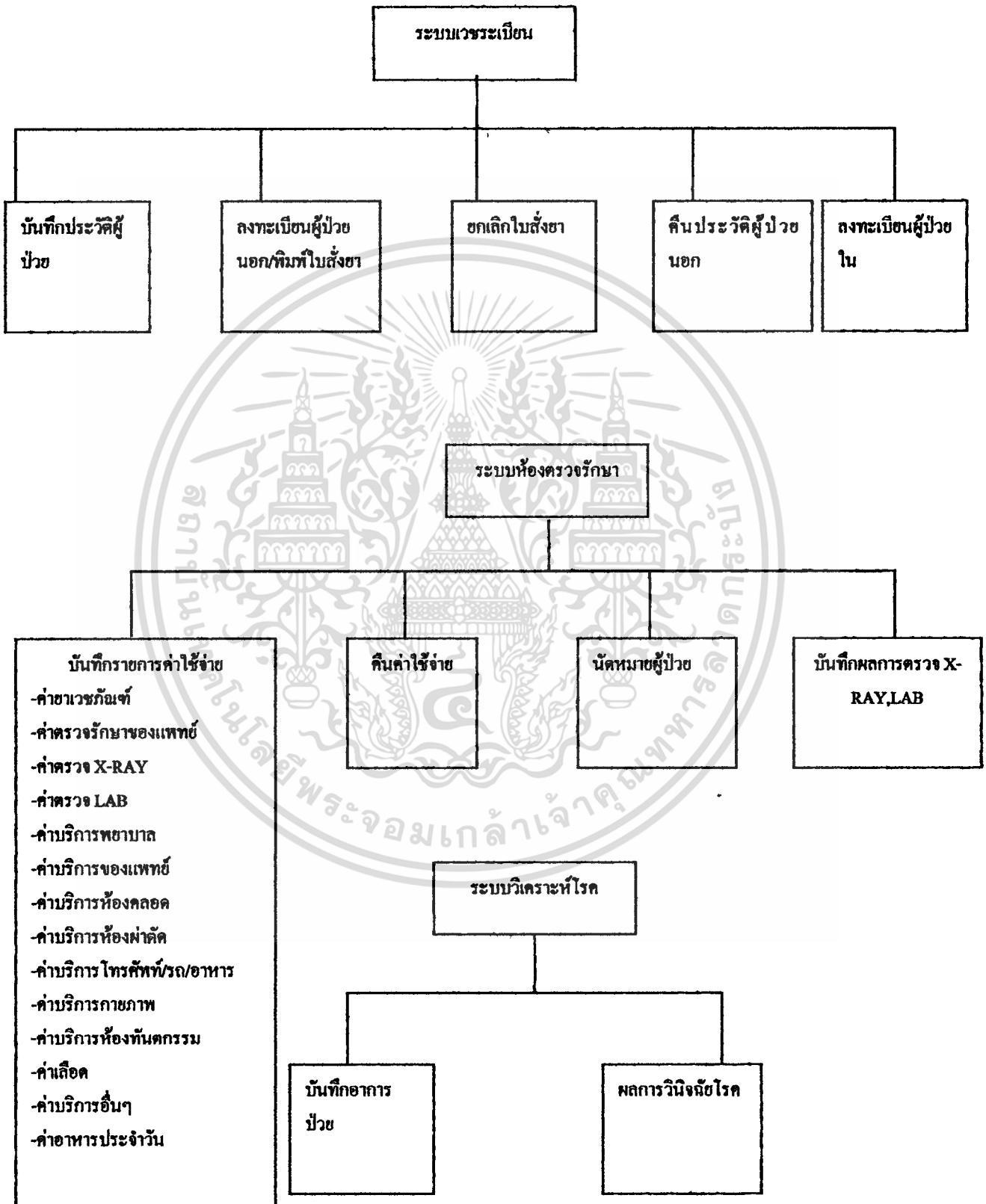
ภาพที่ 3.3 แสดงการเพิ่มประสิทธิภาพของ Concurrency Control บน X-BASE ของระบบ File Server

3.4 การออกแบบส่วนสัมพันธ์กับผู้ใช้

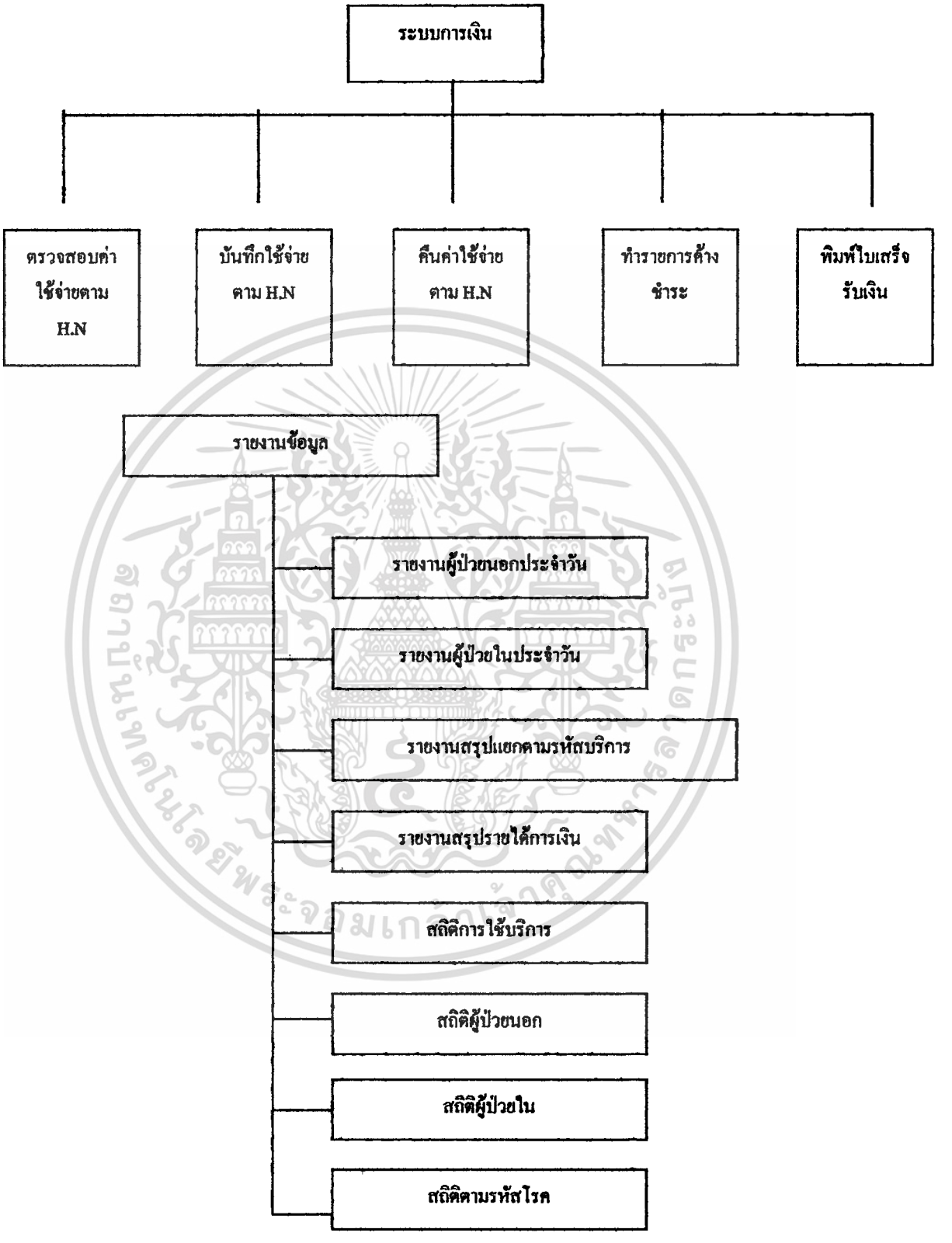
การออกแบบหน้าส่วนสัมพันธ์กับผู้ใช้จะสร้างเป็นเมนูเพื่อให้ครอบคลุมการทำงานของระบบ ดังแสดงดังภาพที่ 3.4



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.4 แสดง Menu Structure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การพัฒนาโปรแกรม

หลังจากที่ได้ออกแบบ Menu Structure แล้ว จึงทำการพัฒนาโปรแกรมโดยใช้ Microsoft Foxpro For Windows 2.5 และจัดเก็บข้อมูลโดยใช้ Foxbase ซึ่งประกอบด้วยส่วนต่างๆดังนี้

(1) การเข้าสู่ระบบการทำงาน ดังภาพนี้

ภาพที่ 3.5 แสดงหน้าจอการเข้าสู่ระบบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) เมนูหลักของระบบ ซึ่งประกอบด้วยรายการต่างๆ ดังภาพ

ภาพที่ 3.6 แสดงรายการในเมนูหลัก

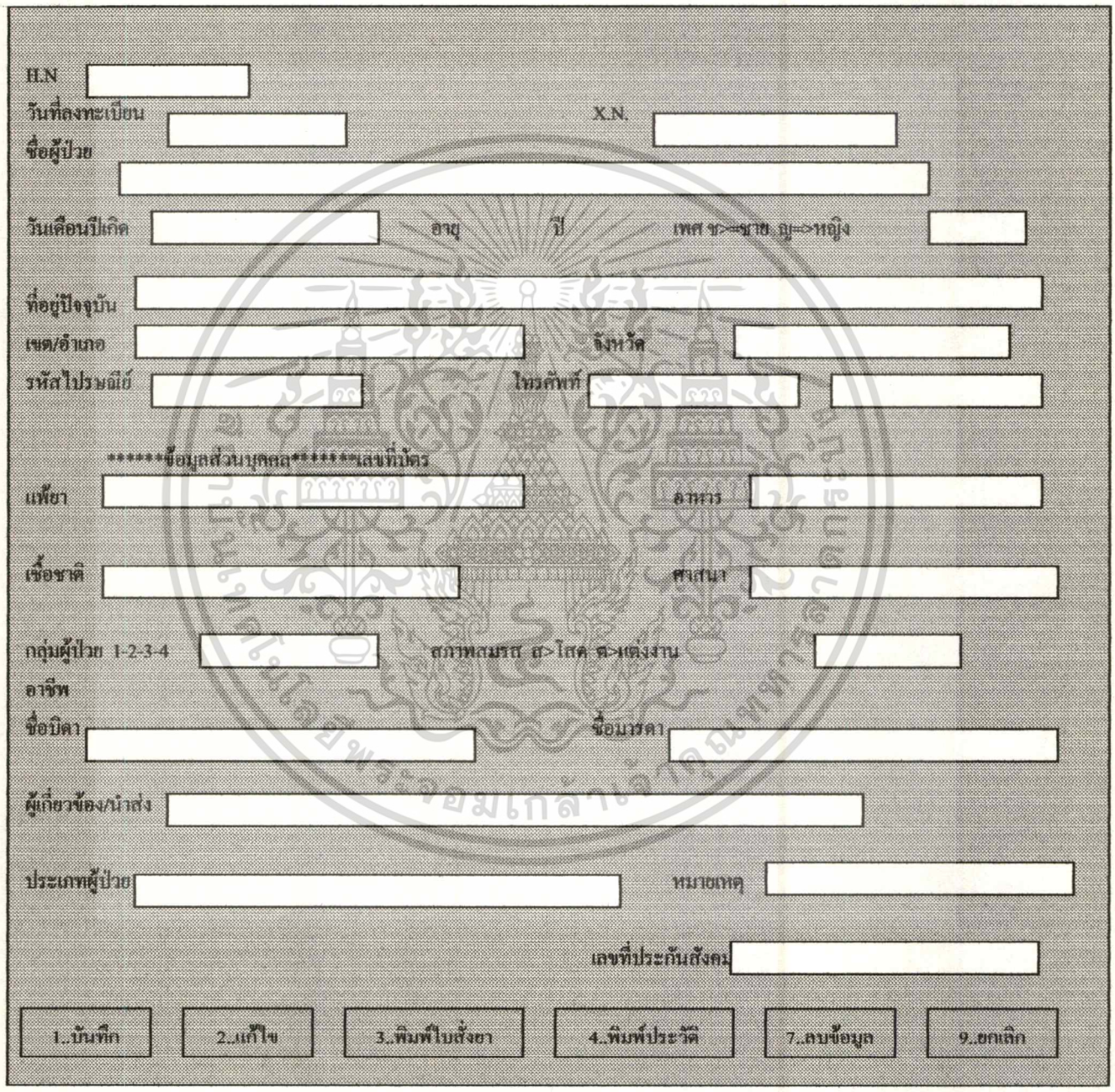
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.7 แสดงรายการในเมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 หน้าจอสำหรับการบันทึกข้อมูล

หน้าจอการบันทึกประวัติ

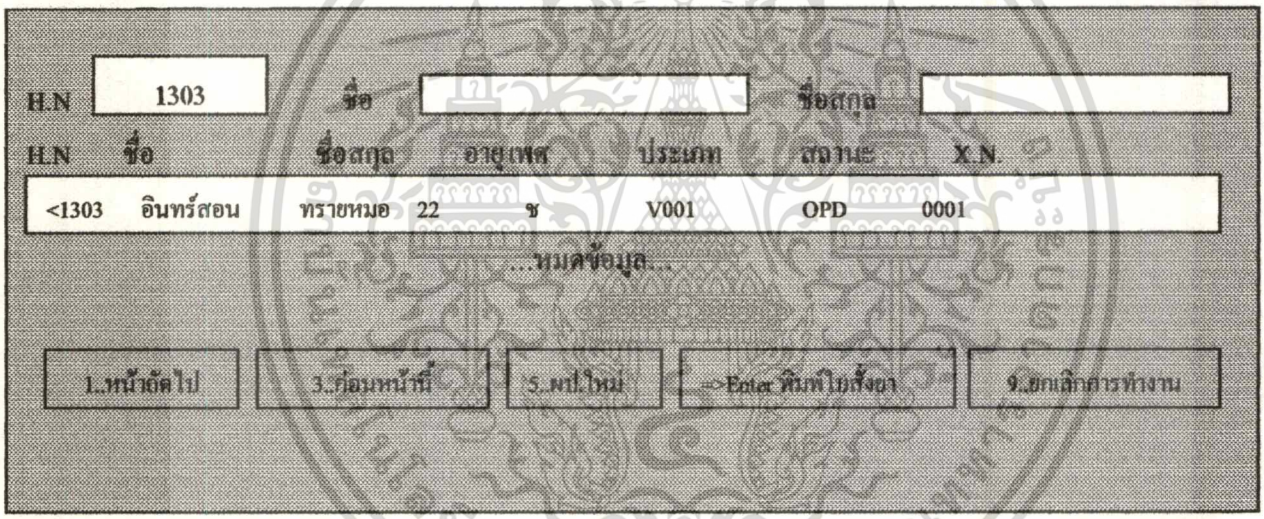


ภาพที่ 3.8 หน้าจอการบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปุ่ม 1 บันทึกข้อมูล
- ปุ่ม 2 สำหรับการแก้ไขข้อมูล
- ปุ่ม 3 สำหรับการพิมพ์ใบสั่งยา
- ปุ่ม 4 สำหรับการสั่งพิมพ์ ใบประวัติข้อมูลผู้ป่วย
- ปุ่ม 7 สำหรับการยกเลิกข้อมูลผู้ป่วย
- ปุ่ม 9 ยกเลิกการทำงาน

3.5.2 หน้าจอการทำงานลงทะเบียนผู้ป่วยนอก/พิมพ์ใบสั่งยา



ภาพที่ 3.9 หน้าจอการทำงานลงทะเบียนผู้ป่วยนอก/พิมพ์ใบสั่งยา

- ปุ่ม 1 ดูข้อมูลหน้าถัดไป
- ปุ่ม 3 ดูข้อมูลก่อนหน้านั้น
- ปุ่ม 5 สำหรับการเข้าไปยังหน้าจอบันทึกประวัติ
- กด Enter ตามรายชื่อที่มีแถบสว่าง เพื่อพิมพ์ใบสั่งยา
- ปุ่ม 9 ยกเลิกการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

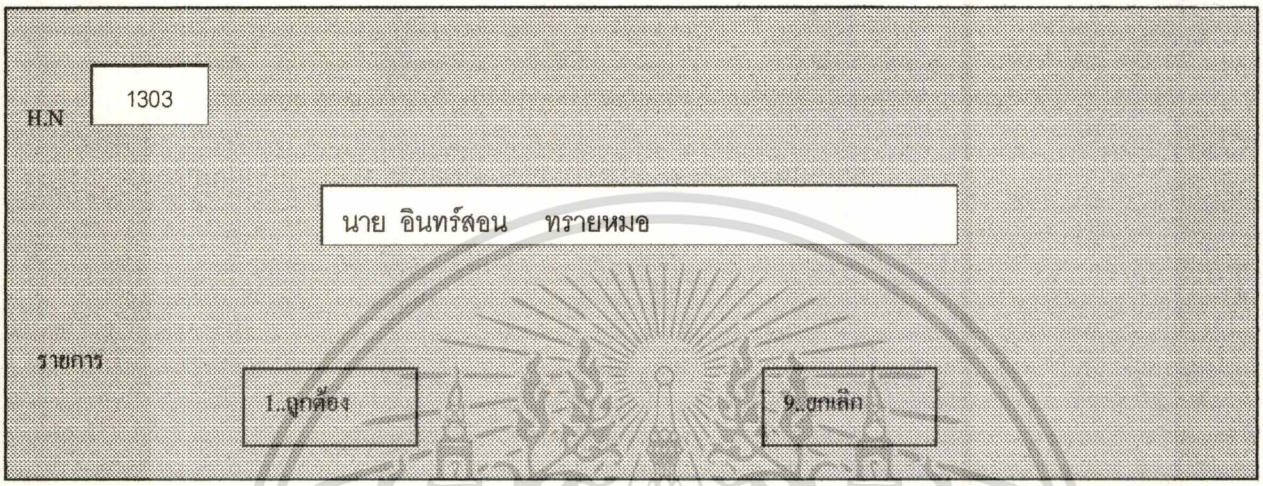
3.5.3 เปลี่ยนรายละเอียด/ยกเลิกใบสั่งยา

ภาพที่ 3.10 เปลี่ยนรายละเอียด/ยกเลิกใบสั่งยา

- ปุ่ม 1 เพื่อออกจากรายการ
- ปุ่ม 2 เพื่อแก้ไขรายละเอียดใบสั่งยา
- ปุ่ม 9 ยกเลิกใบสั่งยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 คืนประวัติผู้ป่วยนอก

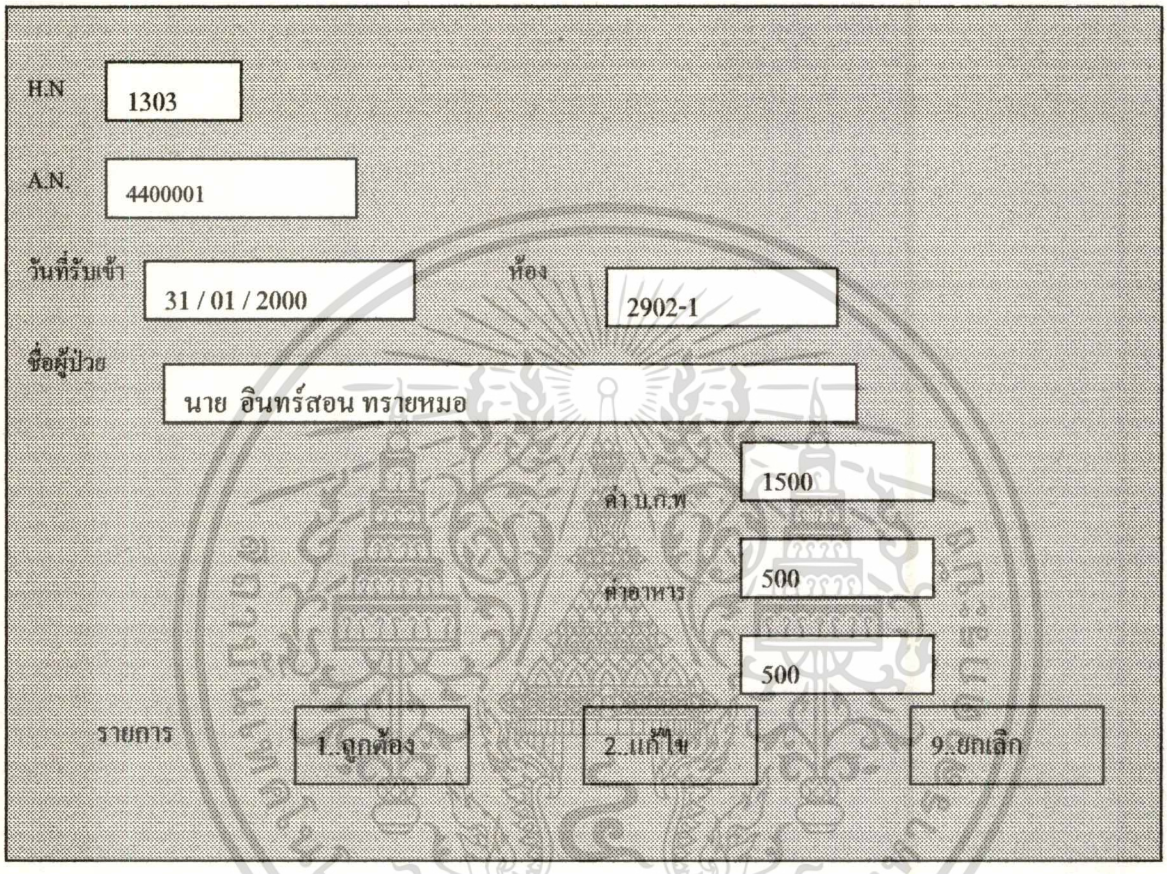


ภาพที่ 3.11 คืนประวัติผู้ป่วยนอก

- ปุ่ม 1 ต้องการคืนผู้ป่วยนอก
- ปุ่ม 9 ยกเลิกคืนผู้ป่วยนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.5 ลงทะเบียนผู้ป่วยใน



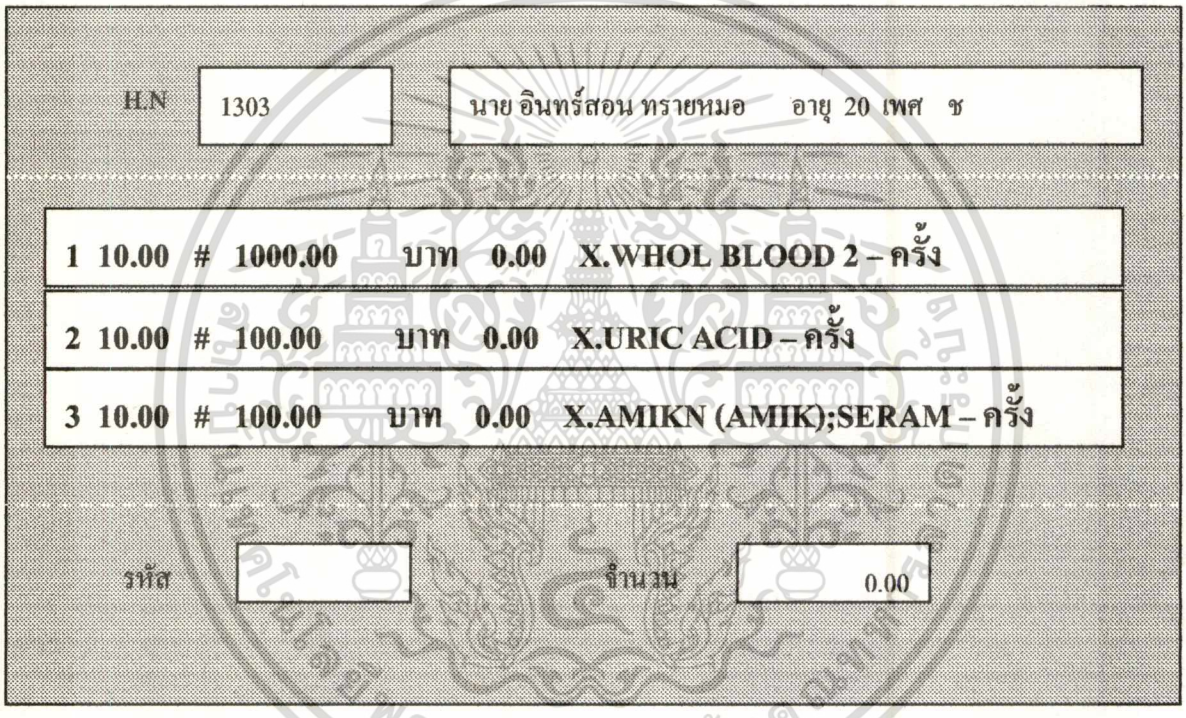
ภาพที่ 3.12 ลงทะเบียนผู้ป่วยใน

- ปุ่ม 1 บันทึกข้อมูล
- ปุ่ม 2 แก้ไขรายการ
- ปุ่ม 9 ยกเลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6. ระบบการบันทึกค่าใช้จ่าย

เป็นการบันทึกค่าใช้จ่ายผู้ป่วยในและผู้ป่วยนอก ระบบจะให้ใส่ H.N.ของผู้ป่วยที่ต้องการ เมื่อใส่ H.N. เสร็จแล้วระบบจะแสดงชื่อผู้ป่วยเป็นผู้ป่วยและมีรายละเอียดเกี่ยวกับการรักษาของผู้ป่วยนั้น เมื่อกด Enter ระบบจะขึ้นภาพต่อไปนี้



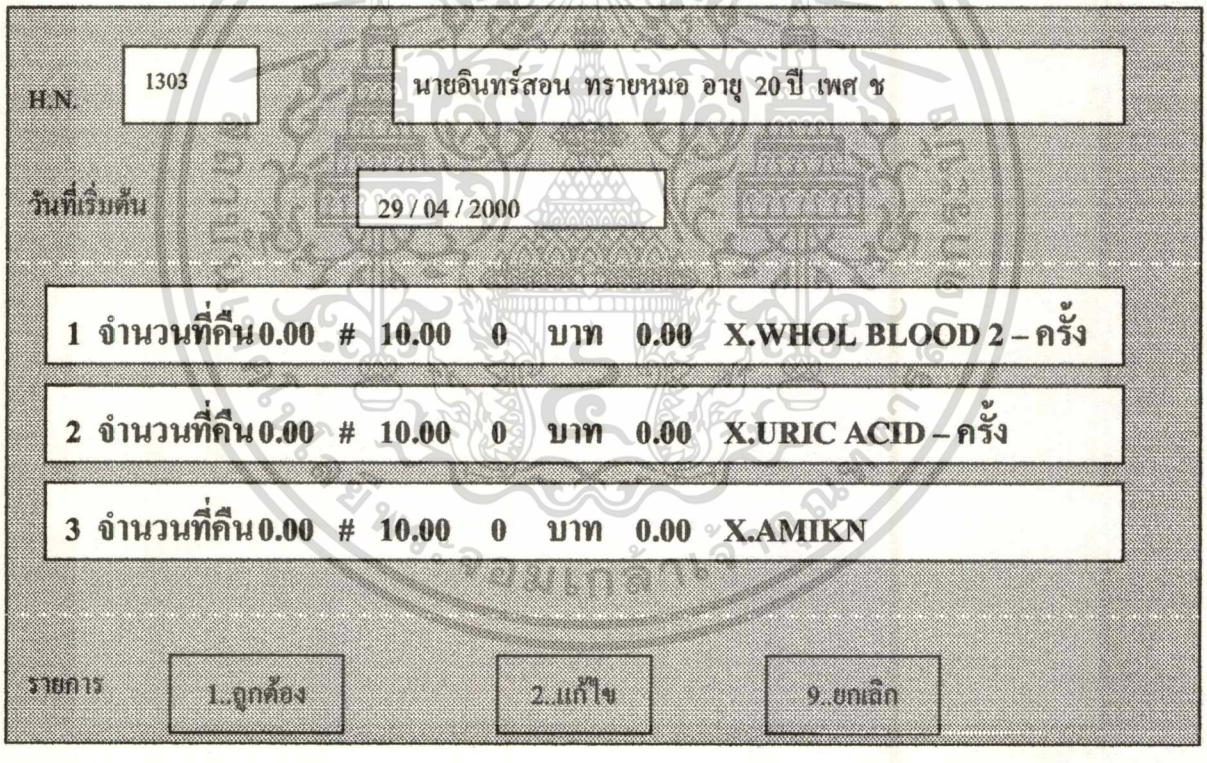
ภาพที่ 3.13 หน้าจอการบันทึกค่าใช้จ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.7. หน้าจอการคืนค่าใช้จ่าย

เมื่อมีการบันทึก รายการค่าใช้จ่ายไป หากจะต้องคืนรายการค่าใช้จ่ายออก เพื่อที่ไม่ให้ระบบคิดเงินกับผู้ป่วย จะมีขีดจำกัดในการคืน เช่น สิทธิของผู้คืนไม่พอและผู้ป่วยบางคน ได้จ่ายเงินไปแล้ว จึงทำให้ไม่สามารถที่จะคืนได้ ระบบสามารถคืนรายการย้อนหลังตามวันที่กำหนดได้ หากมีสิทธิตามที่ระบุไว้

ระบบจะให้ใส่ H.N. ของผู้ป่วยที่ต้องการ เมื่อใส่ H.N. เสร็จแล้วระบบจะแสดงชื่อผู้ป่วย ถ้าผู้ป่วยเป็นผู้ป่วยในก็จะมีรายละเอียดเกี่ยวกับ การรักษาตัวของผู้ป่วยนั้น เมื่อกด Enter ระบบจะขึ้นภาพต่อไปนี้

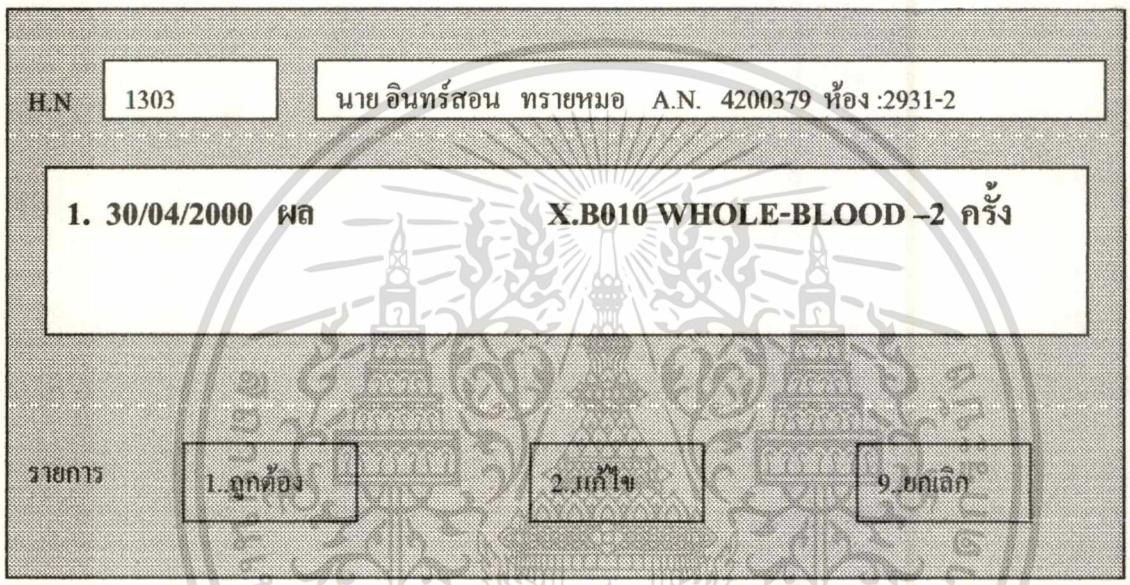


ภาพที่ 3.14 หน้าจอการคืนค่าใช้จ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.8. บันทึกผลการตรวจ LAB,X-RAY

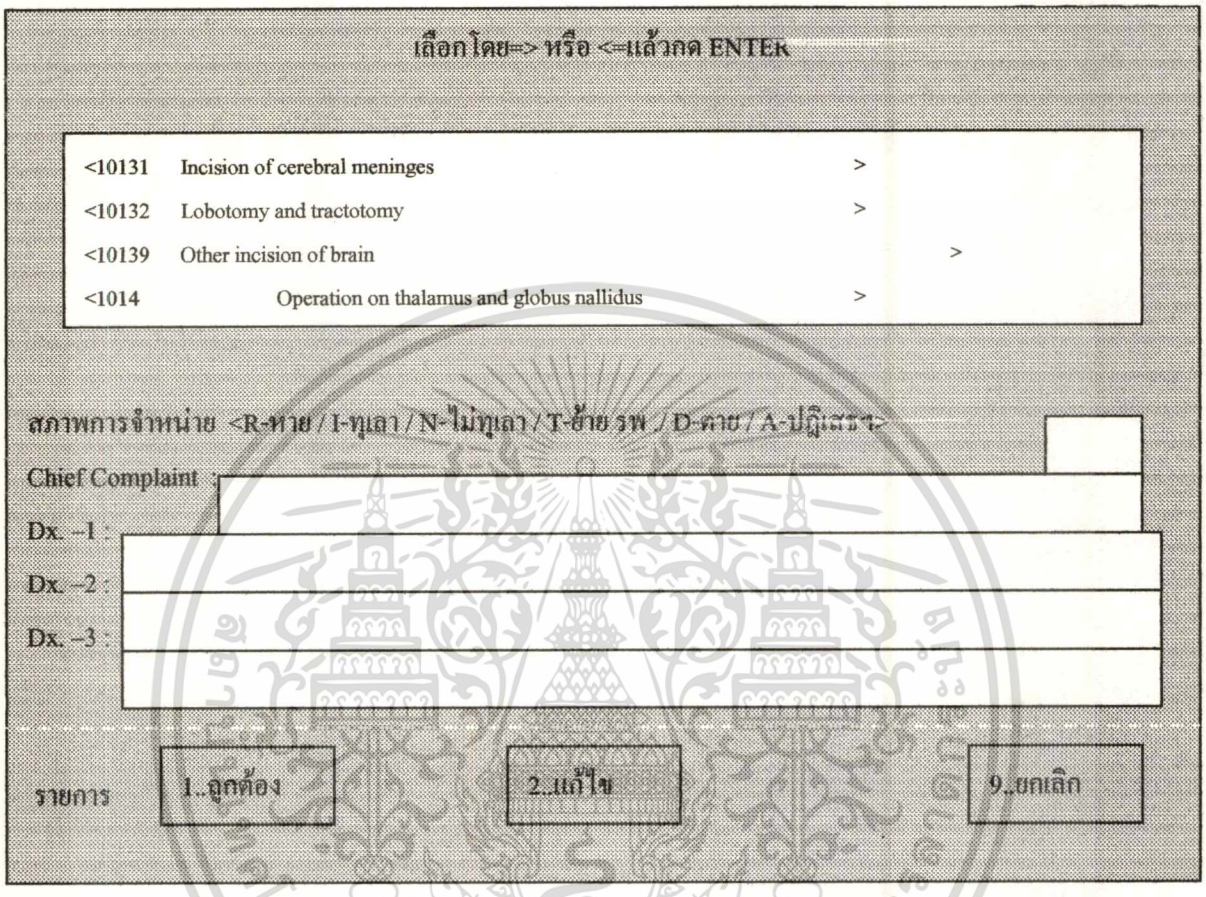
เป็นการบันทึกผลการตรวจ ให้กับผู้ป่วยโดยที่จะบันทึก ข้อความสัมพันธ์กับ H.N. ผู้ป่วย ดังนี้



ภาพที่ 3.15 หน้าจอการบันทึกผลตรวจ LAB,X-RAY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.9. บันทึกผลการวินิจฉัยโรค

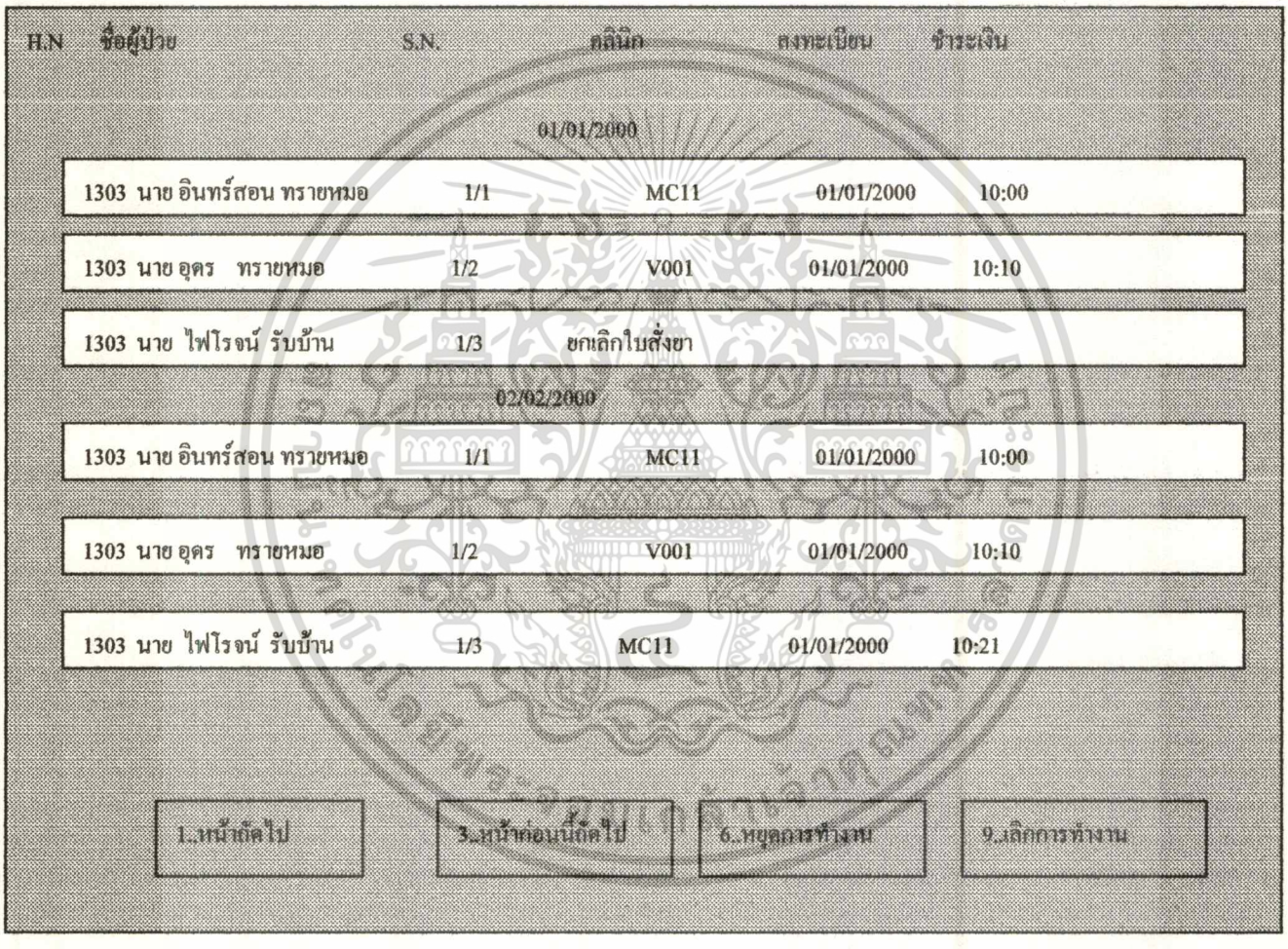


ภาพที่ 3.16 หน้าจอการบันทึกผลการวินิจฉัยโรค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.10 ระบบรายงานข้อมูลรายชื่อผู้ป่วยนอกประจำวัน

เป็นรายละเอียดรายงานรายชื่อผู้ป่วยที่มาใช้บริการที่มาลงทะเบียนผู้ป่วยนอกซึ่งระบบสามารถเรียกดูตามวันที่ๆกำหนดได้ เมื่อมีการเลือกหัวข้อระบบจะให้เลือกดตามเงื่อนไขในการดูตามรายละเอียดรูปดังนี้

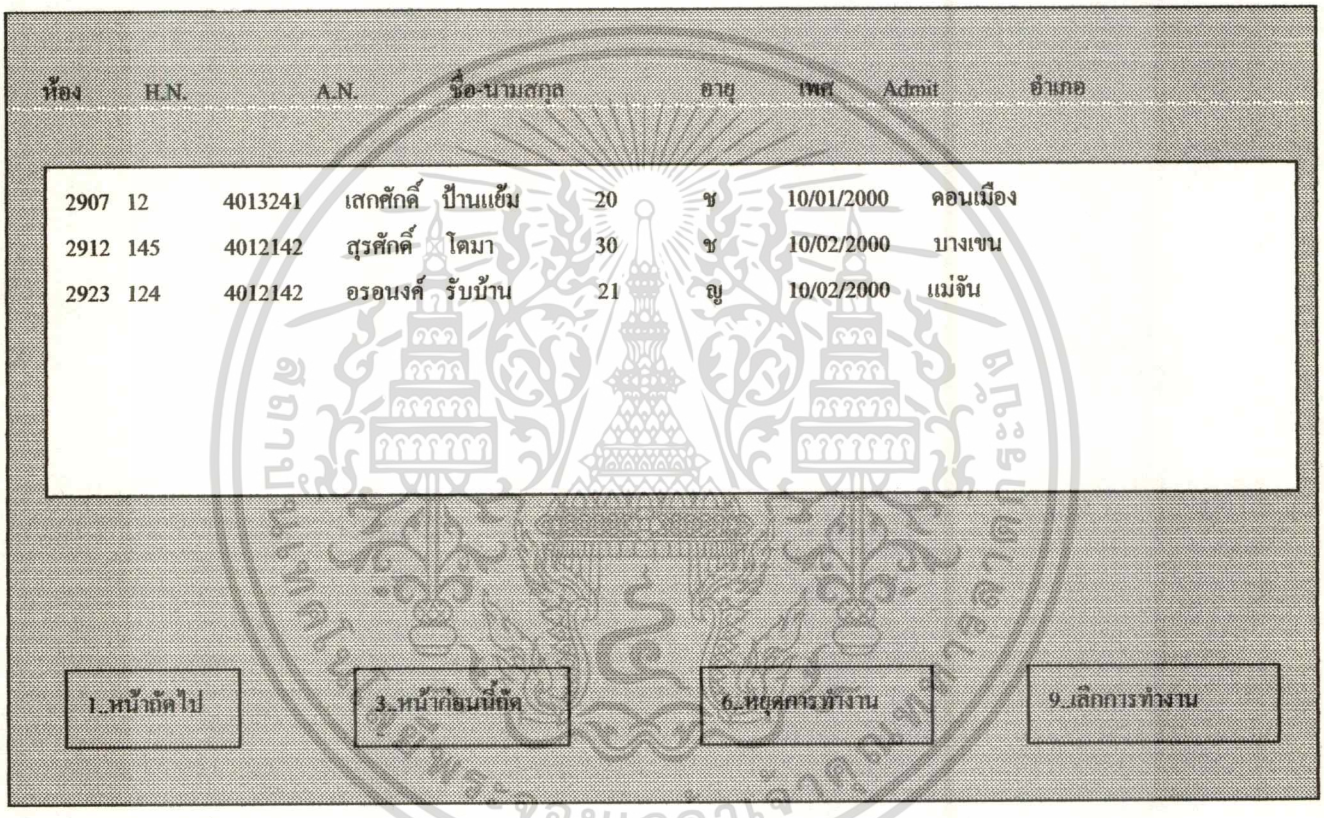


ภาพที่ 3.17 หน้าจอรายชื่อผู้ป่วยนอกประจำวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.11 ระบบรายงานข้อมูลรายชื่อผู้ป่วยในประจำวัน

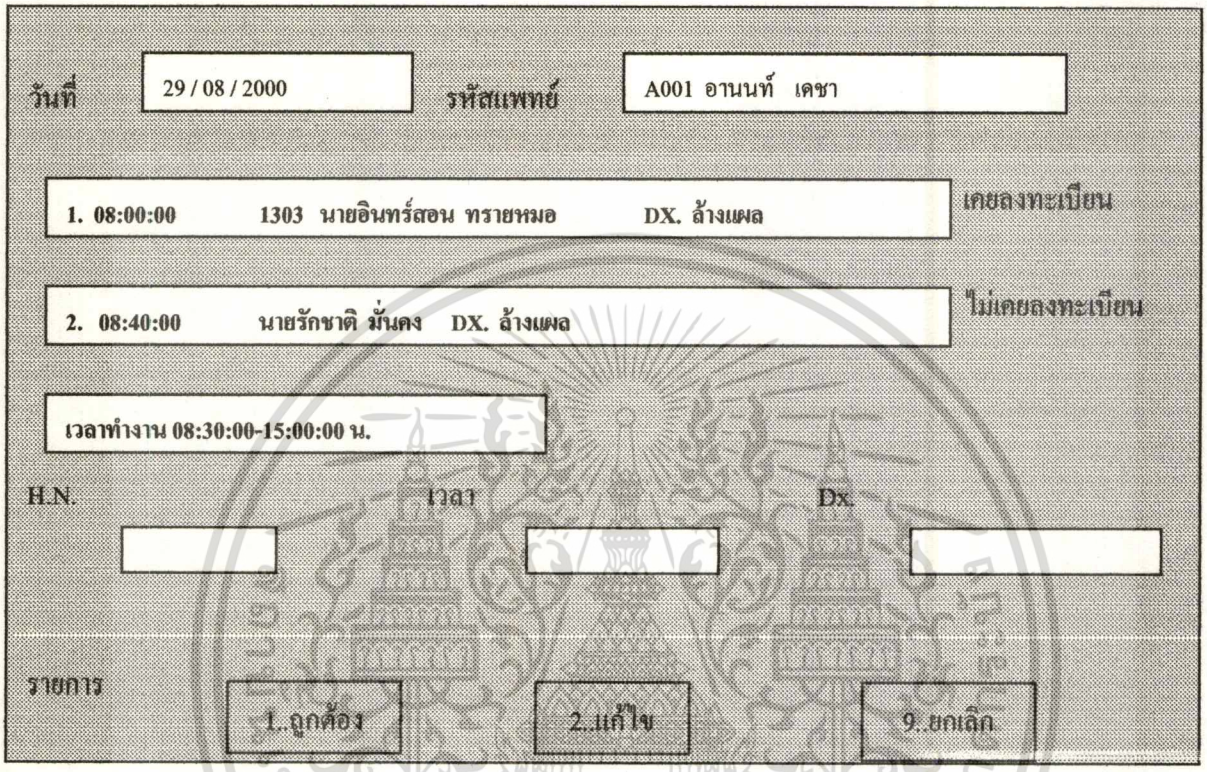
เป็นการขอดูรายชื่อผู้ป่วยในที่พักรักษาโดยระบบจะแสดงรายละเอียดเกี่ยวกับห้องพักอยู่ที่ห้องและรายการต่างๆเกี่ยวกับผู้ป่วย เมื่อเลือกหัวข้อการทำงานตามรูปภาพดังนี้



ภาพที่ 3.18 หน้าจอรายชื่อผู้ป่วยในประจำวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

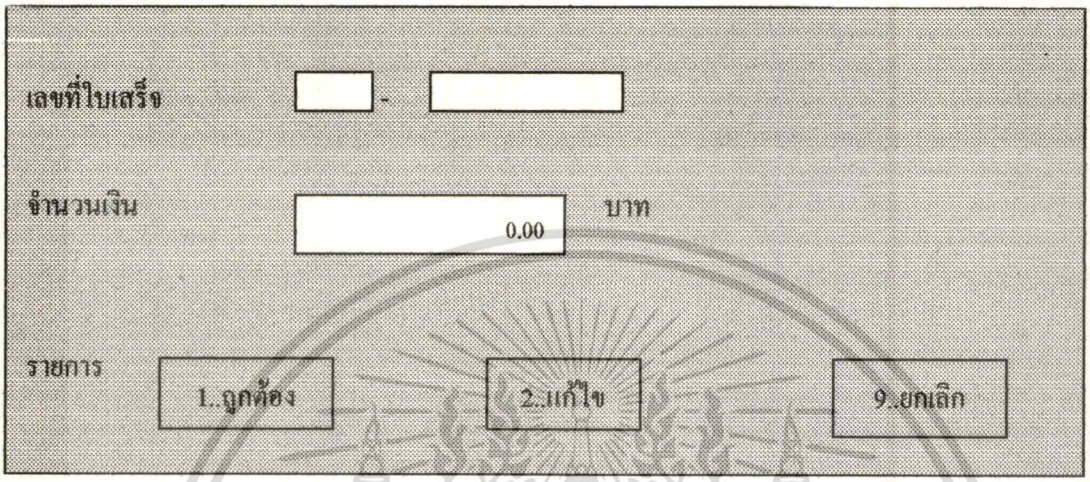
3.5.12 การทำงานการนัดหมายผู้ป่วย



ภาพที่ 3.19 หน้าจอการบันทึกนัดหมายผู้ป่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

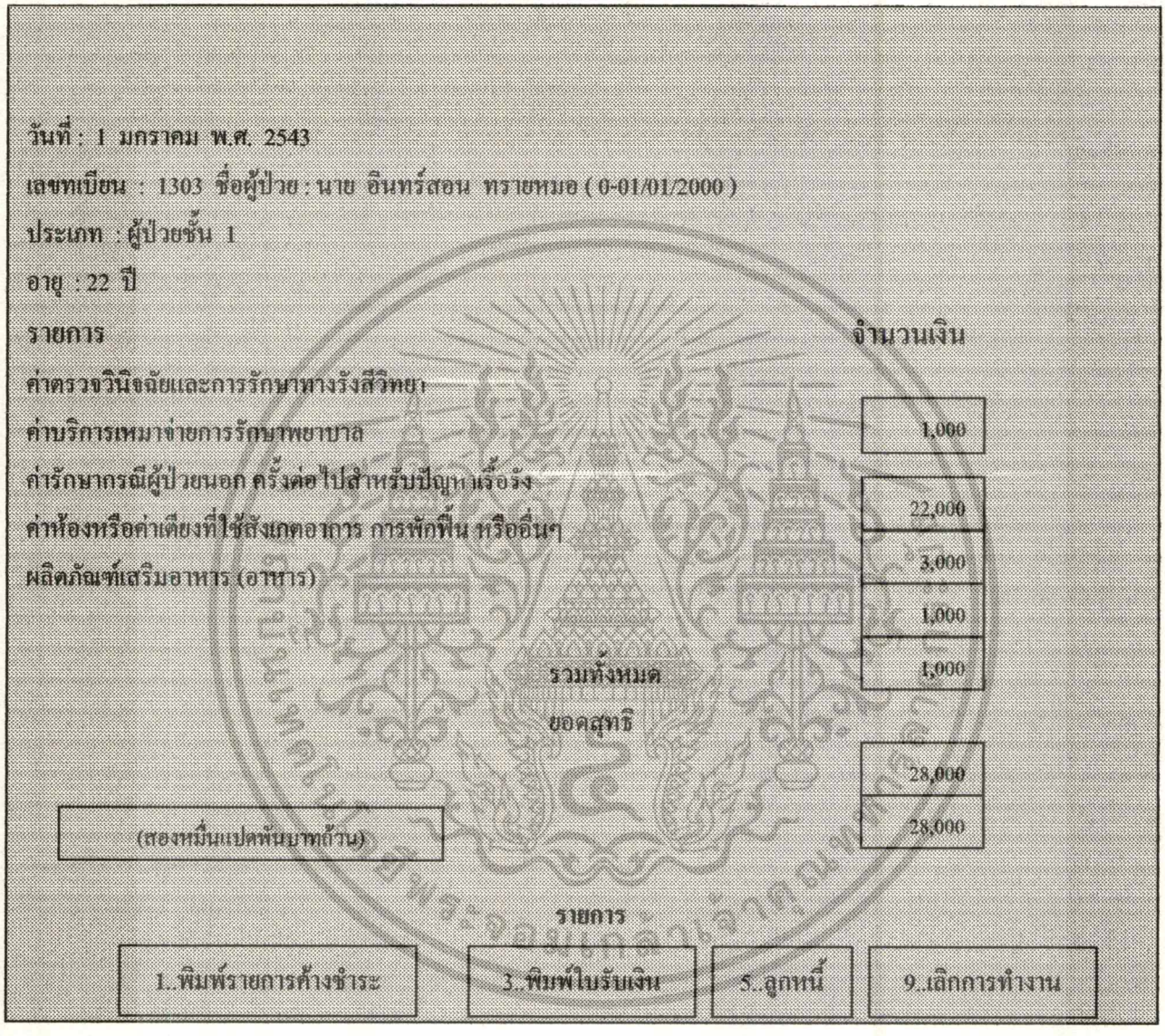
3.5.13 การทำงานการทำค้างชำระ



ภาพที่ 3.20 หน้าจอการบันทึกการทำค้างชำระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.14 หน้าจอการพิมพ์ใบเสร็จรับเงิน



ภาพที่ 3.21 หน้าจอการพิมพ์ใบเสร็จรับเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.15 รายงานการสรุปแยกตามรหัสบริการ

รหัส	ชื่อการตรวจ/บริการ	#	SOPD	SIPD	รอเก็บ
ADM	ADMISSION SET – SET	2.00	200	0	200
ADM	ADMIT SET – SET	10.00	0	0	2000
ALPH	ALPHA BED – วัน	2.00	0	0	400
BIR2	BIRD 12 hr – ครั้ง	100.00	100	0	0
	รวม 4 รายการ		300.00	.00	2600.00

ภาพที่ 3.22 หน้าจอรายงานการสรุปแยกรหัสบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.16 ระบบรายงานการเงิน

เล่มที่	เงินสด	เครดิต	รวมเงิน
OPD			
1 (6571 ฉบับ)	3,339,852	22,094	3,360,446
4 (1696 ฉบับ)	160	343,737	343,897
7 (0 ฉบับ)	0	0	0
9 (203 ฉบับ)	0	58,228	58,228
รวม OPD (8470 ฉบับ)	3,340,012	424,059	3,764,071
2 (572 ฉบับ)	5,833,813	58,921	515,1297
3 (123 ฉบับ)	1,436,246	6770	1,071,333
5 (98 ฉบับ)	0	498,282	498,282
6 (0 ฉบับ)	0	0	0
8 (3 ฉบับ)	0	11,937	11,937
รวม IPD (796 ฉบับ)	7,270,059	575,910	7,845,969
รวมทั้งหมด (9266 ฉบับ)	10610071	999969	11610040

ยกเลิกเล่ม : 1 (170 ฉบับ) เล่ม 2 : (24 ฉบับ) เล่ม 3 : (3 ฉบับ)
ยกเลิกเล่ม : 4 (52 ฉบับ) เล่ม 5 : (3 ฉบับ) เล่ม 6 : (1 ฉบับ)
ยกเลิกเล่ม : 7 (0 ฉบับ) เล่ม 8 : (2 ฉบับ) เล่ม 9 : (8 ฉบับ)
ยกเลิกทั้งหมด (264 ฉบับ)

ภาพที่ 3.23 หน้าจอรายงานการเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
บทสรุป

4.1 ผลการศึกษาและพัฒนาโปรแกรม

การศึกษาและการพัฒนาระบบสารสนเทศในงานบริการผู้ป่วยของโรงพยาบาล สรุปผลได้ดังนี้

4.1.1 แนวคิดและเทคนิคในการนำระบบคอมพิวเตอร์แบบ File Server มาใช้ในระบบโปรแกรมที่ใช้ฐานข้อมูลขนาดใหญ่ จำเป็นต้องเพิ่มประสิทธิภาพของ Concurrency Control ใน X-BASE โดยใช้ Two-Phase Locking Protocol เป็นตัวจัดการ Lock ข้อมูล และ Timeouts Limit เป็นตัวป้องกันการเกิด Deadlock รวมทั้งข้อตกลงที่จะใช้ Exclusive-Mode (Lock-X) ใน Transaction ที่มี Write รวมอยู่ด้วย จะสามารถทำให้ระบบคอมพิวเตอร์แบบ File Server สนับสนุนงานโครงสร้างฐานข้อมูลขนาดใหญ่ได้เป็นอย่างดี

4.1.2 ระบบสารสนเทศที่พัฒนาขึ้นนี้ จะเป็นต้นแบบของการขยายผลการพัฒนาระบบที่ดีและใหญ่ขึ้นอีกในภายหน้า

4.1.3 ผลการออกแบบระบบที่ดีในการพัฒนา ทำให้ระบบบริการผู้ป่วยชัดเจนถูกต้องและรวดเร็วขึ้น อันเป็นผลมาจากระบบคอมพิวเตอร์ที่ทำงานได้อย่างรวดเร็วและการวิเคราะห์ระบบใหม่ที่ลดการซ้ำซ้อนของงานลงได้เป็นอย่างมาก

4.2 ปัญหาและข้อเสนอแนะ

ในการพัฒนาระบบในครั้งนี้เนื่องจากมีเวลาค่อนข้างจำกัด จึงทำให้การแสดงข้อมูลการศึกษาดูค้อยไปบางหัวข้อ ซึ่งผู้ศึกษาได้พยายามแสดงในจุดที่เป็นหัวข้อหลักของการศึกษา หากมีโอกาสน่าจะนำผลที่ได้รับจากการศึกษานี้ไปพัฒนาระบบบน Client / Server โดยศึกษาเพิ่มเติมเกี่ยวกับ MS Visual Foxpro และการใช้ API บน Novell Netware ร่วมด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Abraham Silberschatz .et.al. 1993. **Database System Concepts.** 3 rd ed. New York : McGrow - Hill.

Date,C.J. 1993.**An Introduction to Database System.**6 th ed. Now York : Addison-Wesley.

Eliason, L. Alan. 1990. **System Development : analysis, design , and Implementation.** 2 rd ed. **New York : Harper Collins Publishers.**

Gerge Coulouris.et.al. 1994. **Distributed System : Concepts and Design.** 2 rd ed. Londm: Addison – Wesley.

Hawryszkiewycz, I.T. 1990. **Relational Database Design : An Introduction** . Australia: Prentice Hall.

Pressman S. Roger. 1997. **Software Engineering A Practitioner' s Approach** 4 th ed. New York:McGrow-Hill.

Rob, Peter and Coronel Carlos .1997 . **Database System : design , Implementation and Management.** 3 rd ed.New York : International Thomson.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นาวาโท บุญเรือง เกิดอรุณเดช ร.น.
วัน เดือนปีเกิด	25 มกราคม 2507
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	กศ.บ. (วิทยาศาสตร์-คณิตศาสตร์)
สถานที่สำเร็จการศึกษาระดับปริญญาตรี	มหาวิทยาลัยศรีนครินทร์วิโรฒ วิทยาเขตบางเขน
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2529
การทำงาน	หัวหน้าแผนกตรวจสอบและวิเคราะห์
	กองโครงการและงบประมาณ
	กองบังคับการ
	อู่ทหารเรือพระจุลจอมเกล้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้