การพัฒนาระบบสนับสนุนการตัดสินใจด้วยเทคนิคการจำแนกประเภทข้อมูล

THE DEVELOPMENT OF DECISION SUPPORT SYSTEM BY CLASSIFICATION TECHNICAL

โดย

อนาวิล เอี่ยมเจริญ

ANAWIN EAMCHAROEN

อาจารย์ที่ปรึกษา

รศ.วรพจน์ กรีสุระเดช

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2549

THE DEVELOPMENT OF DECISION SUPPORT SYSTEM BY CLASSIFICATION TECHNICAL

ANAWIN EAMCHAROEN

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1/ 2006

หัวข้อ	การพัฒนาระบบสนับสนุนการตัดสินใจ ด้วยเทคนิคต้นไม้ตัดสินใจ
นักศึกษา	นายอนาวิล เอี่ยมเจริญ
รหัสนักศึกษา	47066140
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547
อาจารย์ที่ปรึกษา	รศ.วรพจน์ กรีสุระเดช

บทคัดย่อ

เป็นที่ยอมรับในปัจจุบันว่าสารสนเทศมีความสำคัญมากต่อการทำงาน เทคนิควิธีการในการวิเคราะห์ข้อมูลเพื่อให้ได้มาซึ่งสารสนเทศ จึงมีความสำคัญต่อการตัดสินใจทางธุรกิจ ปัจจุบันมีการพัฒนาเทคนิควิธีการค้นหาสารสนเทศซึ่งซ่อนอยู่ในฐานข้อมูลที่เรียกว่าการทำเหมืองข้อมูล หรือ Data mining ซึ่งช่วยให้สามารถเข้าถึงสารสนเทศด้วยเทคนิคต่างๆ โดยการสร้างแบบจำลอง (Model) และค้นหารูปแบบ (Pattern) ความสัมพันธ์ของข้อมูลเพื่อวิเคราะห์คาดการณ์ล่วงหน้าเพื่อสนับสนุนการตัดสินใจในการดำเนินธุรกิจขององค์กร

ปัจจุบันบริษัทต่างๆล้วนมีระบบสาระสนเทศซึ่งมีการเก็บข้อมูลต่างๆไว้ในฐานข้อมูลเป็นจำนวนมาก ไม่ว่าจะเป็นข้อมูล ลูกค้า พนักงาน และการสั่งซื้อต่างๆ ซึ่งสามารถนำวิธีการทางด้าต้าไมน์นิ่งมาใช้ในการค้นหาสาระสนเทศซึ่งเป็นประโยชน์ต่อการตัดสินใจดำเนินการทางธุรกิจ จึงนำข้อมูลมาวิเคราะห์และพัฒนาโปรแกรมทาง Business Intelligence สนับสนุนความต้องการสารสนเทศของบริษัทเป็นกรณีศึกษา

โดยในเอกสารฉบับนี้ได้ศึกษาหาความรู้ที่จำเป็นต้องใช้ในการพัฒนาโปรแกรมทาง Business Intelligence ศึกษาฐานข้อมูลของบริษัท ใช้ระบบจัดการฐานข้อมูล Oracle 10g สำหรับกรณีศึกษานี้ค้นหาสาระสนเทศด้วยวิธี Classification โดยใช้เทคนิคต้นไม้ตัดสินใจและใช้อัลกอลิทึ่ม SPRINT (Scalable RaRallelizable Induction of decision Trees) เพื่อจำแนกประเภทข้อมูลของบริษัท เพื่อเป็นประโยชน์ในการศึกษาการพัฒนาโปรแกรมทาง Business Intelligence และสนับสนุนความต้องการสารสนเทศของบริษัทต่อไป

Title	The development of decision support system by decision tree technique
Students	Mr. Anawin Eamcharoen
Student ID.	47066140
Degree	Master of Science
Programme	Information Science
Academic Year	2005
Special Project Advisor	Assoc. Prof. Worapoj Kreesuradej

ABSTRACT

In present day importance of information technology is accepted. Analysis of data for information is importance to business decision. Today we have the development of technical to find information that hiding in database which calls data mining. Data mining can help us to access information by many technical. By create model find pattern of information and relation between them to predict and support business decision.

Today all companies have information system and a lot of data in their database. This can improve information by data mining technical to support business decision. We study data to analyze and develop business intelligence software to support business decision. For this reason we take knowledge about Oracle 10g database management system and classification technical to studying the development of data mining application.

In this document we learn how to develop business intelligence software. We use Oracle 10g database management system and classify information by decision tree technical call SPRINT (Scalable RaRallelizable Induction of decision Trees) for knowledge in development business intelligence application and decision support system.

กิตติกรรมประกาศ

ในการพัฒนาระบบสนับสนุนการตัดสินใจสำหรับกรณีศึกษาจนสำเร็จนี้ทางผู้จัดทำขอขอบพระคุณ รศ.ดร.วรพจน์ กรีสุระเดช เป็นอย่างสูง ที่ให้ความรู้คำแนะนำและเป็นที่ปรึกษาในการพัฒนาระบบงาน ขอขอบคุณเพื่อนๆทุกคนที่ช่วยให้คำแนะนำเกี่ยวกับการใช้งานเครื่องมือในการพัฒนาโปรแกรมต่างๆ ทั้งที่ช่วยสนับสนุนด้าน software และขอบพระคุณ คุณพ่อ คุณแม่ที่เป็นกำลังใจทำให้ผู้จัดทำสามารถพัฒนาระบบจนสำเร็จ

ขอบคุณครับ

อานวิล เอี่ยมเจริญ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการดำเนินงานของบริษัทต่างๆ สารสนเทศมีความสำคัญอย่างมากทั้งในการดำเนินงานประจำวันและในการตัดสินใจเพื่อวางแผนการทำงานหรือนโยบายเพื่อกำหนดเป้าหมายของบริษัท แต่การเก็บรวบรวมข้อมูลต่างๆเพื่อนำมาใช้ในการดำเนินงานหรือสนับสนุนการตัดสินในนี้จำเป็นต้องมีค่าใช้จ่ายในการเก็บรักษาข้อมูล การใช้งานข้อมูลที่มีอยู่ให้คุ้มค่าจึงมีความสำคัญ ดังนั้นการค้นหาสารสนเทศซึ่งซ่อนอยู่ในข้อมูลที่เก็บรวบรวมไว้ซึ่งสามารถเพิ่มพูนความรู้ที่เป็นประโยชน์ในการสนับสนุนการดำเนินงานของบริษัทได้ ทำให้บริษัทมีสารสนเทศเพิ่มขึ้นซึ่งช่วยให้บริษัทสามารถดำเนินงานได้อย่างมีประสิทธิภาพมากขึ้น เทคนิควิธีการค้นหาสารสนเทศที่ซ่อนอยู่ในข้อมูลหรือที่เรียกว่า Data Mining จึงเข้ามามีบทบาทในการดำเนินธุรกิจมากขึ้น โดยอยู่ในรูปแบบของโปรแกรมทาง Business Intelligence การศึกษาเทคนิควิธีการทาง Data Mining จึงมีความสำคัญ และเทคนิคทาง Data Mining วิธีหนึ่งที่ได้รับความนิยมคือ Classification ซึ่งเป็นวิธีการจำแนกประเภทข้อมูลโดยการแบ่งข้อมูลออกเป็นกลุ่มข้อมูลตามลักษณะของข้อมูลซึ่งกระบวนการทำ Classification ด้วยเทคนิค Decision tree มีอยู่ด้วยกัน 2 เทคนิคคือ SLIQ และ SPRINT ซึ่งในกรณีศึกษานี้ได้เลือกเทคนิค SPRINT (Scalable RaRallelizable Induction of decision Trees) มาพัฒนาเพราะเป็นเทคนิคที่ช่วยให้สามารถแบ่งกลุ่มข้อมูลที่มีขนาดใหญ่ได้ซึ่งเหมาะกับการนำไปใช้ในธุรกิจที่มีข้อมูลจำนวนมาก

จากเหตุผลดังกล่าวจึงพัฒนาโปรแกรมสนับสนุนการตัดสินใจเพื่อเป็นประโยชน์ในการสนับสนุนการตัดสินใจทางธุรกิจเป็นกรณีศึกษาโดยใช้วิธี Classification และใช้อัลกอลิทึม SPRINT ในการพัฒนาระบบ เพื่อเป็นแนวทางในการนำ Data Mining มาประยุกต์ใช้ในธุรกิจและเป็นประโยชน์ต่อการศึกษาการพัฒนาแอพพลิเคชั่นทาง Data Mining ต่อไป

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

โครงการพัฒนาระบบงานมีวัตถุประสงค์และเป้าหมายดังต่อไปนี้

1. ศึกษาและทำความเข้าใจแนวคิดและขั้นตอนการทำ Data Mining
2. ศึกษาการทำ Classification ด้วยเทคนิค Decision tree โดยใช้อัลกอลิทึ่ม SPRINT (Scalable RaRallelizable Induction of decision Trees)
3. ศึกษาการใช้งานระบบจัดการฐานข้อมูล Oracle 10g
4. ศึกษาข้อมูลของบริษัทและออกแบบระบบเพื่อสนุบสนุนการตัดสินใจของบริษัทอัลติม่าเพื่อเป็นแนวทางในการนำ Data Mining มาประยุกต์ใช้ในบริษัทผลิตพลาสติก
5. ศึกษาการพัฒนาโปรแกรมทาง Business Intelligence

1.3 ขอบเขตของปัญหาพิเศษ

โครงการพัฒนาระบบงานนี้ เป็นการศึกษาการพัฒนาระบบสนับสนุนการตัดสินใจตามแนวความคิดทาง Data mining โดยศึกษาวิธีการ Classification โดยใช้เทคนิค Decision Tree และใช้อัลกอลิทึ่ม SPRINT ในการพัฒนา

1.3.1 ข้อมูลที่นำมาวิเคราะห์

ข้อมูลที่ใช้ในการวิเคราะห์เป็นข้อมูลของธนาคารในเยอรมันซึ่งเป็นข้อมูลตัวอย่างในการทำ Classification เกี่ยวกับการตรวจสอบ Credit ลูกค้าประกอบไปด้วยข้อมูลทั้งหมด 20 field จำนวน 1000 record ซึ่งได้แสดงรายละเอียดของข้อมูลไว้ในบทที่ 4

1.3.2 หน้าที่การทำงานของระบบ

ระบบจะวิเคราะห์ข้อมูลตามหลักการทาง Data Mining เพื่อสร้างแบบจำลองต้นไม้เพื่อจำแนกกลุ่มตามเทคนิค Classification ด้วยอัลกอลิทึ่ม SPRINT โดยระบบมีลักษณะการทำงานดังนี้

1. ระบบนำข้อมูลจากฐานข้อมูลแบบ RDBMS มาใช้ในการวิเคราะห์ผ่านทางระบบจัดการฐานข้อมูล Oracle 10g ซึ่งเป็น

2. ระบบวิเคราะห์ข้อมูลโดยจำแนกประเภทข้อมูลตามหลักการ Classification โดยใช้เทคนิค Decision tree และอัลกอลิทึ่ม SPRINT (Scalable RaRallelizable Induction of decision Trees)

3. ระบบวิเคราะห์แบบจำลองต้นไม้ โดยการแบ่งข้อมูลออกเป็น 2 ส่วนสำหรับการสร้างแบบจำลอง และสำหรับการทดสอบแบบจำลอง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ได้จากการทำปัญหาพิเศษนี้สามารถแบ่งออกเป็นหัวข้อดังต่อไปนี้

ประโยชน์ของผู้จัดทำปัญหาพิเศษ

1. ได้รับความรู้จากการศึกษาเทคนิค Data Mining ในการนำมาวิเคราะห์หาสารสนเทศที่ซ่อนอยู่ในข้อมูล
2. ได้รับความรู้จากการศึกษาระบบจัดการฐานข้อมูลที่ใช้กันอย่างแพร่หลาย
3. ได้รับความรู้จากการศึกษาการพัฒนาโปรแกรมทาง Business Intelligence
4. ได้รับความรู้ในการพัฒนาโปรแกรมภาษาจาวา และ การออกแบบโดยใช้ UML Diagram

ประโยชน์ต่อผู้ใช้โปรแกรม

1. ได้รับสารสนเทศเพิ่มขึ้นจากข้อมูลที่เก็บไว้ในฐานข้อมูล
2. สนับสนุนการตัดสินใจทางธุรกิจ

1.5 ขั้นตอนในการดำเนินงาน

การพัฒนาระบบงานสำหรับกรณีศึกษานี้ประกอบไปด้วยขั้นตอนดังต่อไปนี้

1. ศึกษาหลักการและกระบวนการในการทำ Data Mining
2. ศึกษาอัลกอลิทึ่ม SPRINT (Scalable RaRallelizable Induction of decision Trees) เพื่อนำมาใช้ในการพัฒนาระบบ
3. ศึกษาฐานข้อมูล เพื่อนำมาวิเคราะห์ด้วยเทคนิคทาง Data Mining
4. ศึกษาซอร์ฟแวร์ที่นำมาใช้ในการพัฒนาระบบ ได้แก่ ศึกษาการใช้งานระบบจัดการฐานข้อมูล Oracle 10g และ Visual Basic .NET 2003
5. เก็บรวบรวมเอกสารและข้อมูลต่างๆที่ใช้ประกอบการทำงานและที่เกี่ยวข้อง
6. เตรียมข้อมูลให้เหมาะสมในการนำมาใช้ในการวิเคราะห์โดยพิจารณาปรับปรุงแก้ไขข้อมูลหรือคัดเลือกข้อมูลที่ไม่สมบูรณ์ออก
7. ขั้นตอนการวิเคราะห์และออกแบบระบบสนับสนุนการตัดสินใจของบริษัทผลิตชิ้นส่วนพลาสติกอัลติม่า และเป็นการกำหนดเป้าหมายในการพัฒนาโปรแกรมด้วย
8. ขั้นตอนการพัฒนาระบบสนับสนุนการตัดสินใจตามที่ได้ออกแบบไว้
9. ขั้นตอนทดสอบโปรแกรม และบอกถึงความสามารถทั้งหมดที่เป็นไปได้ของโปรแกรม รวมถึงข้อจำกัดและขจัดปัญหาที่เกิดขึ้นกับระบบงาน
10. ขั้นตอนการทำเอกสารประกอบประกอบการใช้งานโปรแกรมระบบงาน และเอกสารอ้างอิงในการศึกษาเพื่อทำปัญหาพิเศษ
11. สรุปผลการศึกษา

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

รายละเอียดทางด้านอุปกรณ์คอมพิวเตอร์

1. คอมพิวเตอร์

รายละเอียดทางด้านโปรแกรม

1. ระบบปฏิบัติการ Window XP
2. Oracle 10g DBMS
3. Microsoft Visual Studio .NET 2003
4. Microsoft .NET Framework SDK v1.1

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Data Mining

ดาต้าไมน์นิ่งหมายถึงเครื่องมือที่มีประสิทธิภาพซึ่งใช้ในการค้นหาสารสนเทศที่มีประโยชน์จากข้อมูลที่เก็บไว้ในฐานข้อมูล ดาต้าไมน์นิ่งประกอบด้วยขั้นตอนและเทคนิควิธีการต่างๆเพื่อให้ได้มาซึ่งสารสนเทศที่เป็นประโยชน์ จึงได้ศึกษานิยามของดาต้าไมน์นิ่ง ขั้นตอนการทำดาต้าไมน์นิ่ง เทคนิคต่าง ๆ ของดาต้าไมน์นิ่ง และการประยุกต์การใช้งานดาต้าไมน์นิ่ง เพื่อเป็นประโยชน์ในการพัฒนาแอพพลิเคชั่นต่อไป

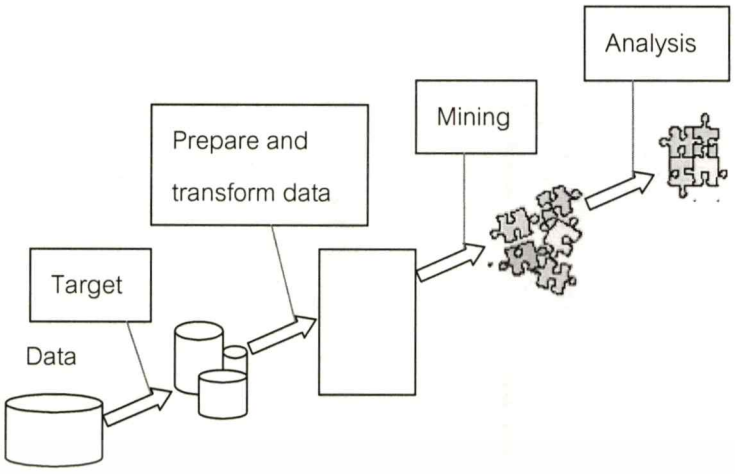
2.1.1 นิยามของดาต้าไมน์นิ่ง

ดาต้าไมน์นิ่ง (Data Mining) เป็นกระบวนการหรือขั้นตอนในการค้นหาสารสนเทศซึ่งซ่อนอยู่ในข้อมูลที่เก็บอยู่ในฐานข้อมูลที่มีขนาดใหญ่ ความรู้ที่ได้จากการค้นหาอาจเป็นความรู้ที่ไม่เคยทราบมาก่อนว่ามีประโยชน์ หรือมีความสัมพันธ์กันอย่างไร ทำให้เราทราบถึงข้อมูลเพิ่มมากขึ้นจากข้อมูลเดิมที่เราเก็บไว้อยู่แล้ว นอกจากนี้ดาต้าไมน์นิ่งที่ดีควรมีความถูกต้องหรือใกล้เคียงความเป็นจริงมากที่สุด เพื่อให้สามารถนำไปใช้สนับสนุนการวางแผนกลยุทธ์และการตัดสินใจทางธุรกิจได้ ซึ่งเปรียบเหมือนกับการที่เราสกัดทองออกมาจากหินหรือทราย ดังนั้นเราอาจเรียกดาต้าไมน์นิ่งว่า Knowledge Mining

2.1.2 ขั้นตอนการทำดาต้าไมน์นิ่ง

ขั้นตอนการทำค้นหาสารสนเทศหรือที่เรียกอีกอย่างหนึ่งว่า Knowledge Discovery in Database เป็นขั้นตอนในการสร้างแบบจำลองของกลุ่มข้อมูล เพื่อสร้างความเข้าใจในแนวโน้ม รูปแบบ และความเกี่ยวข้องกันของกลุ่มข้อมูล เพื่อใช้ในการทำนายบนข้อมูลนั้น แบ่งออกเป็น 4 ขั้นตอน คือ

1. Problem definition การกำหนดปัญหาหรือวัตถุประสงค์ของการทำดาต้าไมน์นิ่ง
2. Data gathering and preparation การเตรียมข้อมูลให้พร้อมสำหรับการทำดาต้าไมน์นิ่ง
3. Model building and evaluation การสร้างโมเดลสำหรับทำนายและปรับปรุงโมเดล
4. Knowledge deployment การเผยแพร่ความรู้หรือการนำไปใช้งาน



รูปที่ 2.1 ขั้นตอนการทำดาต้าไมน์นิ่ง

2.1.2.1 Problem definition

เป็นขั้นตอนที่สำคัญที่สุด เพราะเป็นการแปลงจุดมุ่งหมายทางธุรกิจให้อยู่ในรูปที่สามารถนำไปใช้ในการทำ Data Mining ได้ เช่น จากปัญหาทางธุรกิจที่ว่าทำอย่างไรจึงจะขายสินค้าได้มากแปลงเป็นผู้บริโภคกลุ่มไหนที่มีแนวโน้มว่าจะซื้อสินค้า A มาก แต่ก่อนที่จะ Mining ได้เราต้องมีข้อมูลการซื้อสินค้า A ในอดีตก่อนจึงสามารถเตรียมข้อมูลสำหรับ Mining ได้

2.1.2.2 Data gathering and preparation

ในขั้นตอนนี้เป็นการเตรียมข้อมูลเพิ่มเติมในส่วนที่จำเป็นต่อการทำ Mining และจัดรูปแบบข้อมูลให้พร้อม เพราะข้อมูลอาจต้องเปลี่ยนแปลงให้เหมาะสม เช่น Date_of_Birth แปลงเป็น Age เนื่องจากข้อมูลมีลักษณะดังกล่าวทำให้การค้นหาสารสนเทศทำได้ยาก Oracle Data Mining จึงถูกฝังไว้ใน Oracle Database เพื่อให้ง่ายต่อการจัดการซึ่งในขั้นตอนนี้สามารถแบ่งออกเป็น 3 ขั้นตอนย่อยดังนี้

1. Data Selection การคัดเลือกข้อมูลเพื่อการระบุถึงแหล่งข้อมูลที่นำมาใช้ที่จำเป็นต่อการนำมาวิเคราะห์ข้อมูลเบื้องต้น รวมถึงจะต้องมีความเข้าใจเกี่ยวกับลักษณะและตัวแปรของข้อมูลที่จะนำมาทำดาต้าไมน์นิ่งนอกจากการทำความเข้าใจกับชนิดของข้อมูลแล้ว ยังมีสิ่งสำคัญที่ควรพิจารณาในการเตรียมข้อมูล คือ

1. *ระดับข้อมูลระดับข้อมูลที่ต้องการ* ข้อมูลที่เก็บอยู่ในฐานข้อมูลมีหลายระดับตั้งแต่ระดับรายละเอียด จนมาถึงระดับผลสรุป ทั้งนี้ การจะใช้ข้อมูลระดับใดขึ้นกับวัตถุประสงค์ในการวิเคราะห์นั้น ๆ เช่น การวิเคราะห์การใช้โทรศัพท์ของลูกค้าของบริษัทแห่งหนึ่ง ถ้าหากกำหนดวัตถุประสงค์เพื่อศึกษาพฤติกรรมการใช้โทรศัพท์ของลูกค้า ดังนั้น ข้อมูลที่เราสนใจซึ่งเกี่ยวข้องกับลูกค้า เช่น เบอร์โทรศัพท์ ต้นทาง เบอร์โทรศัพท์ปลายทาง เวลาที่โทร ระยะเวลาที่ใช้ในการ

โทรแต่ละครั้ง หรือ ข้อมูลที่ไม่ใช่ข้อมูลสรุปบางครั้งอาจจะมีปริมาณที่มากเกินไป ทำให้จัดการได้ยาก และทำให้เกิดจำนวนการคอมไบเนชั่น (Combination) สูง เช่น Market Basket Analysis ที่เกิดจากการรวมกันของสินค้าภายในร้านขายปลีก ซึ่งมีข้อมูลของรายการสินค้าจำนวนมาก ดังนั้น การนำเอาหน่วยวัดในการจัดเก็บสินค้าในคลัง หรือ SKU (Stock Keeping Unit) เข้ามาช่วย โดยการรวมกลุ่มให้สินค้าประเภทเดียวกันอยู่ในกลุ่มเดียวกัน ทำให้เกิดการคอมไบเนชั่นกัน

2. *ความไม่สอดคล้องของข้อมูลที่มาจากหลายแหล่ง* การทำดาต้าไมน์นิ่ง บางครั้งต้องอาศัยข้อมูลจากแหล่งข้อมูล ซึ่งแต่ละแหล่งอาจจะเก็บข้อมูลเดียวกันในรูปแบบที่แตกต่างกันไป เช่น การวิเคราะห์ข้อมูลทางโทรศัพท์ เพื่อหาเบอร์โทรศัพท์ที่ใช้ฝากข้อความเข้าไปรษณีย์เสียง (Voice Mail) ในแต่ละเมือง โดยแต่ละเมืองมีการจัดเก็บที่แตกต่างกัน เช่น เมืองหนึ่งเก็บเบอร์โทรศัพท์ที่ใช้โทรเข้าไปรษณีย์เสียงเพียงเบอร์ต้นทาง และเบอร์ปลายทาง ในขณะที่อีกเมืองเก็บเบอร์ที่ไม่รู้ด้วยเบอร์ปลายทาง ส่วนอีกเมืองเก็บด้วยเบอร์ที่โทรเข้าไปรษณีย์เสียงจริง ๆ ดังนั้น จึงจำเป็นต้องทำข้อมูลเหล่านี้ให้ออกมาในรูปแบบมาตรฐานเดียวกันก่อน เพื่อที่จะได้ใจถึงความแตกต่างในการเก็บข้อมูลของแต่ละแหล่ง

3. *การจัดเก็บข้อมูลแตกต่างกัน* เนื่องจากมีการจัดเก็บในคอมพิวเตอร์แตกต่างกัน ทำให้ข้อมูลที่เรานำมาใช้ในการวิเคราะห์มีผลกระทบเกิดขึ้น เช่น ข้อมูลที่นำมาใช้ในการวิเคราะห์ส่วนมากจัดเก็บด้วยภาษา COBOL หรือ RPG ข้อมูลที่เป็น Text จะถูกเก็บเป็น EBCDIC และข้อมูลตัวเลขจะเก็บเป็น Packed Decimal ในขณะที่ระบบดาต้าไมน์นิงใช้ภาษา C หรือ C++ เก็บข้อมูล Text ในลักษณะของ ASCII และข้อมูลตัวเลขเก็บเป็น Integer หรือ Floating Point เป็นต้น ดังนั้น จะต้องมีการแปลงข้อมูลให้อยู่ในลักษณะที่สอดคล้อง

4. *ข้อมูลที่เป็นข้อความ* ข้อมูลที่จัดเก็บเป็น Text นอกจากจะประกอบด้วยข้อมูลที่ไม่จำเป็นสำหรับการวิเคราะห์แล้ว บางครั้งอาจจะก่อให้เกิดความสับสนได้ เช่น "_no" กับ "no_" หรือ "VOR2J0" กับ "VOR 2J0" ซึ่งจริง ๆ แล้วเป็นค่าเดียวกัน แต่ในการวิเคราะห์ข้อมูลของโปรแกรมจะมองว่าเป็นค่าที่แตกต่างกัน ดังนั้น วิธีการแก้ไขข้อมูลเหล่านี้ทำได้โดย การสร้างตารางสำหรับเก็บข้อมูลที่ถูกต้อง และแทนที่ข้อมูลที่นำมาวิเคราะห์ด้วยรหัส เช่น ในฐานข้อมูลเชิงสัมพันธ์ (Relational Database) มีการแทนที่ข้อมูลที่เป็น Product_Name ด้วย Product_Code ซึ่งเป็นค่าที่ unique ในตาราง

2. Data Preprocessing การประมวลข้อมูลเป็นการนำเอาข้อมูลที่จะใช้ในการทำดาต้าไมน์นิ่งมาทำให้เป็นข้อมูลที่มีคุณภาพดีก่อนที่จะนำไปใช้งานต่อไป โดยเป็นการตรวจสอบว่าข้อมูลที่ได้เลือกไว้ในขั้นตอนการคัดเลือกข้อมูลมีความเหมาะสมหรือไม่ เช่น ข้อมูลแบบ Categorical ใช้วัดการกระจายของข้อมูลเพื่อทำความเข้าใจข้อมูลได้ดียิ่งขึ้น โดยอาศัยเครื่องมือทางด้านการสร้างภาพนามธรรม (Visualize) แสดงข้อมูล เช่น กราฟแท่ง ส่วนข้อมูลแบบ

Quantitative ที่เป็นตัวเลขวัดโดยการหาค่าสูงสุดต่ำสุด ค่าเฉลี่ย ค่ามัธยฐาน และตัววัดทางสถิติอื่น ๆ ซึ่งการประมวลข้อมูลก่อนนี้ประกอบด้วย

1. *การทำความสะอาดข้อมูล (Data Cleaning)* ทำให้ข้อมูล มีความสมบูรณ์ถูกต้องและสอดคล้องกัน เป็นการเพิ่มค่าที่ขาดหายไป (Missing Values) การระบุ Noisy Data ค่าความผิดพลาด หรือความแปรปรวนที่เกิดขึ้นจากการเก็บรวบรวมข้อมูล การป้อนข้อมูลเข้าสู่ระบบ และการรับส่งข้อมูล ความไม่สอดคล้องกันจากการตั้งชื่อ แล้วจึงทำการปรับปรุงค่าข้อมูลให้มีความสอดคล้องกัน เช่น ข้อมูล ในฟิลด์ขาดหายไปอาจจะแทนค่าข้อมูลที่ขาดหายไปด้วย Unknown หรือถ้าหากข้อมูลขาดหายไปเป็นจำนวนมากและข้อมูลนั้นไม่สำคัญมากนักอาจจะทำการตัดฟิลด์นั้นทิ้งไป

2. *การรวมข้อมูล (Data Integration)* รวบรวมข้อมูลมาจาก หลาย ๆ แหล่ง แล้วทำการตรวจหาและขจัดความขัดแย้ง และความซ้ำซ้อนของข้อมูล เช่น ฐานข้อมูล A เก็บรหัสพนักงานใช้ชื่อแอททริบิวต์ คือ EmpID และ ฐานข้อมูล B เก็บรหัสพนักงานเหมือนกันแต่ใช้ชื่อแอททริ-บิวต์ว่า E_ID เมื่อนำข้อมูลมารวมกันก็จะทำให้เกิดความขัดแย้ง และความซ้ำซ้อนของข้อมูล

3. Data Transformation การแปลงรูปแบบข้อมูล เป็นขั้นตอนที่ทำการรวบรวมข้อมูลหรือเปลี่ยนแปลงข้อมูลเพื่อให้อยู่ในรูปแบบที่เหมาะสมกับอัลกอริทึมที่ใช้ในการทำดาต้าไมน์นิงของงาน ซึ่งความเหมาะสมของข้อมูลก็ขึ้นอยู่กับโมเดลที่เราจะใช้งาน ตัวอย่างของโมเดลที่จะใช้งานไม่สามารถทำการคำนวณข้อมูลที่เป็นตัวอักษรได้ ก็จะต้องแปลงตัวอักษรไปเป็นตัวเลขก่อน เช่น การแปลงระดับการศึกษา ปริญญาตรี ปริญญาโท และปริญญาเอก ไปเป็นตัวเลข 1, 2, 3 เพื่อให้สอดคล้องกับโมเดลที่จะใช้งาน

2.1.2.3 Model building and evaluation

สำหรับขั้นตอนนี้เป็นการนำเทคนิคต่างๆทาง Data Mining มาใช้ในการค้นหาสารสนเทศสร้างเป็นโมเดลสำหรับทำนายจากอัลกอลิทึ่มที่สนับสนุนเทคนิคนั้นๆ และปรับปรุงโมเดลที่ได้เพื่อให้สามารถทำนายได้ใกล้เคียงความเป็นจริงมากที่สุด จากการวิเคราะห์ทำให้ได้โมเดลมากมายในการทำนาย โดยสามารถแบ่งกลุ่มตามการดำเนินการได้ดังต่อไปนี้

1. Predictive Modeling เป็นการนำข้อมูลมาใช้ในการสร้างโมเดล เพื่อนำไปใช้ในการทำนายค่า แบ่งออกได้เป็น 2 เทคนิค คือ

1. Classification เป็นการทำนายข้อมูลในอนาคตว่าข้อมูลที่ต้องการพิจารณา ควรจะอยู่ในกลุ่มใด โดยมีการแบ่งประเภทกลุ่มไว้อยู่ก่อนแล้ว เช่น การทำนายว่าเป็นลูกค้าที่อยู่ในกลุ่มที่ควรส่งจดหมายแนะนำสินค้าและบริการใหม่ไปให้หรือไม่

2. Forecasting เป็นโมเดลที่ใช้ในการทำนายแนวโน้มข้อมูลที่เป็นตัวเลขในอนาคต เช่น การ

พยากรณ์อากาศ การทำนายหุ้น

2. Database Segmentation เป็นการแบ่งข้อมูลออกเป็นกลุ่มย่อย ๆ โดยที่ข้อมูลภายในแต่ละกลุ่ม มีลักษณะที่เหมือนกันหรือใกล้เคียงกัน โดยที่เรายังไม่เคยรู้มาก่อน เช่น ใช้ในการแบ่งกลุ่มของลูกค้าว่ามีจำนวนกี่กลุ่ม

3. Link Analysis เป็นการหาความสัมพันธ์ของข้อมูลในแต่ละเรคคอร์ด หรือกลุ่มของเรคคอร์ดในฐานข้อมูล เช่น การหาความสัมพันธ์ของสินค้าว่าลูกค้ามักจะซื้อสินค้าอะไรไปควบคู่กันในการซื้อครั้งหนึ่ง (Association Rule) การศึกษาการซื้อสินค้าในระยะยาว (Sequential Pattern Discovery) หรือการศึกษาแนวโน้มยอดขายในช่วงเวลาแต่ละสัปดาห์ หรือแต่ละเดือน (Similar Time Sequence Discovery)

4. **Deviation Detection** เป็นเทคนิคที่ใช้แสดงข้อมูลที่มีลักษณะผิดปกติไปจากข้อมูลทั่วไป แบ่งเป็น 2 ประเภท

1. Visualization เป็นเทคนิคที่ใช้ในการแสดงข้อมูลในรูปแบบต่าง ๆ เช่น แผนที่ รูปภาพ กราฟสามมิติ ซึ่งมีประสิทธิภาพในการสื่อสารค่อนข้างมาก
2. Statistics เป็นการใช้วิธีทางสถิติเข้ามาช่วยตรวจจับข้อมูล

2.1.2.4 Knowledge deployment

เมื่อได้โมเดลที่เหมาะสมกับข้อมูลแล้ว ก็เป็นการนำโมเดลไปใช้ Mining ข้อมูลนำไปสู่การพัฒนาโปรแกรม Business Intelligence เพื่อสนับสนุนการตัดสินใจทางธุรกิจ ในปัจจุบันนี้ยังมีแอพพลิเคชั่นที่ช่วยสนับสนุนการพัฒนาโปรแกรมทาง Data Mining เกิดขึ้นมากมาย ระบบจัดการฐานข้อมูล Oracle 10g เป็นระบบจัดการฐานข้อมูลหนึ่งซึ่งได้รับความนิยมซึ่งได้รวบรวมอัลกอลิทึ่มที่ใช้ในเทคนิคต่างๆทาง Data Mining ไว้ใน Oracle Data Mining feature สนับสนุนการพัฒนาโปรแกรม Business Intelligence

2.2 SPRINT Algorithm

SPRINT ย่อมาจากคำว่า Scalable PaRallelizable INduction of decision trees เป็นอัลกอริทึมที่สามารถใช้ได้กับข้อมูลแบบตัวเลข (numeric attribute) และข้อมูลที่จัดเป็นหมวดหมู่ (categorical attribute) มีขั้นตอนหลักๆเหมือนอัลกอริทึมของ Decision Tree ส่วนใหญ่คือ

1. Growth attribute เป็นขั้นตอนการสร้าง tree จาก training data

2. Prune phase เป็นขั้นตอนการ prune เพื่อเพิ่มความถูกต้องให้แก่ tree สำหรับขั้นตอน test data

SPRINT เป็นอัลกอลิทึ่มที่ถูกพัฒนาขึ้นมาช่วยแก้ปัญหาข้อจำกัดของหน่วยความจำที่ใช้ในการทำ training phase และมีความเร็ว และรับรองข้อมูลขนาดใหญ่ได้มากขึ้น โดยทั่วไปอัลกอลิทึ่มของ Decision tree มักมีข้อจำกัดในเรื่องขอจำนวนข้อมูลที่สามารถทำงานได้ เนื่องจากเก็บข้อมูลไว้ในหน่วยความจำทั้งหมดในขณะสร้าง tree ซึ่ง SPRINT ถูกออกแบบมาให้สามารถทำงานในลักษณะคู่ขนานได้ (parallelization) ซึ่งทำให้สามารถรองรับการทำงานกับข้อมูลจำนวนมากได้ดี

หลักการของอัลกอลิทึ่มใน Decision tree คือทำการแบ่ง training data ออกเป็นส่วนๆของ tree โดยการแตกกิ่ง (split branch) เพื่อนำไปสู่ class ซึ่งเป็น attributed หนึ่งของ training set เรียกว่า classifying attribute ในที่สุด จุดมุ่งหมายคือสร้างโมเดล tree ของ classifying attribute โดยขึ้นอยู่กับ attribute อื่นๆ ซึ่งเป็นการแบ่งกิ่งก้านของ tree ออกไปนี้จำทำไปเรื่อยๆจนกว่าจะได้ leaf node เป็น class ซึ่ง non-leaf node หรือ internal node คือ split point เป็น attribute ของข้อมูล training data ที่ผ่านวิธีการเลือกแล้ว โมเดลที่ได้สามารถนำไปใช้ในการหา class ของข้อมูลที่ยังไม่จัด class ได้ในอนาคต SPRINT ใช้วิธีการของ GINI index

2.2.1 Growth Phase

Tree ถูกสร้างขึ้นโดยวิธีการ recursive partitioning คือการแบ่ง tree (split) ออกเป็นส่วนๆ (partition) ที่ node test โดยทำซ้ำไปเป็นรอบๆจนกว่าแต่ละส่วนจะมีข้อมูลอยู่ในคลาสเดียวกัน โดยมีขั้นตอนการทำงานดังนี้

```
Partition(Data S)
        If(all points in S are of the same class) then
                return;
        for each attribute A do
                evaluate splits on attribute A;
        use base split found to partition S into S1 and S2
        Partition(S1)
        Partition(S2)
```

ซึ่งมีหลักที่สำคัญอยู่ 2 ประการที่จะมีผลต่อประสิทธิภาพของ tree-growth phase คือ

1. วิธีการในการหาจุดแบ่ง (split point) ของ node test
2. วิธีการเลือก split point และจะแบ่งข้อมูลอย่างไร

สำหรับ SPRINT มีโครงสร้างต่างจากอัลกอลิทึ่ม Decision tree อื่นๆ อย่าง CART และ C4.5 ตรงที่ทำการเรียงลำดับข้อมูลเพียงครั้งเดียว (one-time sort) ในตอนเริ่มต้น (growth phase) เท่านั้น

2.2.1.1 Data Structure

โครงสร้างข้อมูลในรูปแบบของ SPRINT จะประกอบไปด้วย Attribute list และ Histograms

1. **Attribute list** SPRINT เมื่อเริ่มต้นจะทำการสร้าง attribute list สำหรับแต่ละ attribute ในข้อมูลแต่ละรายการใน Attribute list เรียกว่า Attribute records ตัวอย่างเช่น Training data ประกอบด้วย Attribute Index of Record (rid), Age, Status และ Class โดยที่ Attribute Class เป็น Attribute เป้าหมาย (Target Attribute) ในตอนเริ่มต้นเมื่อสร้าง list นั้น attribute ที่เป็น continuous attribute จะถูกเรียงลำดับด้วย Attribute value ซึ่งเป็นการเรียงลำดับเพียงครั้งเดียวเท่านั้นเมื่อเริ่มสร้าง list และ Attribute list ที่เกิดขึ้นจะประกอบไปด้วย Attribute rid, Target Attribute และ Attribute ที่เลือกมาทดสอบทีละ Attribute จนครบทุก Attribute ดังนั้นถ้ามี Attribute ที่เลือกมาทดสอบ n Attribute จะมี Attribute list ทั้งหมด n Attribute list ด้วย

ตารางที่ 2.1 ตัวอย่าง Training data

rid	Age	Status	Class
0	23	Single	Good
1	17	Marriage	Good
2	43	Marriage	Good
3	68	Single	Bad
4	32	Divorce	Bad
5	20	Single	Good

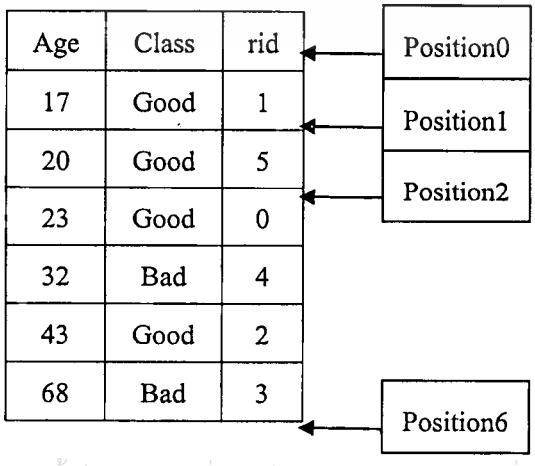
ตารางที่ 2.2 ตัวอย่าง Age Attribute List

Age	Class	rid
17	Good	1
20	Good	5
23	Good	0
32	Bad	4
43	Good	2
68	Bad	3

ตารางที่ 2.3 ตัวอย่าง Status Attribute List

Status	Class	rid
Single	Good	0
Marriage	Good	1
Marriage	Good	2
Single	Bad	3
Divorce	Bad	4
Single	Good	5

2. **Histograms** สำหรับ Continuous attribute จะมี histogram สำหรับแต่ละ node ของ decision tree ที่กำลังถูกพิจารณาการ splitting โดยจะมี C_{above} และ C_{below} ที่ใช้ในการแสดงจำนวน class ของ Attribute record ที่ยังไม่ถูกประมวลผล ส่วน categorical attribute จะมี histograms ที่เรียกว่า Count Matrix เราจะใช้ histograms ในการหา split point ด้วยวิธีการของ GINI index



รูปที่ 2.2 การแบ่งตำแหน่ง histogram

Position0

	Good	Bad
C_{below}	0	0
C_{above}	4	2

Position1

	Good	Bad
C_{below}	1	0
C_{above}	3	2

Position2

	Good	Bad
C_{below}	2	0
C_{above}	2	2

Position3

	Good	Bad
C_{below}	3	0
C_{above}	1	2

...

Position6

	Good	Bad
C_{below}	4	2
C_{above}	0	0

รูปที่ 2.3 การหา Histogram ของข้อมูลประเภทตัวเลข

สำหรับ categorical attribute จะหา histograms ด้วยวิธี Count Matrix ดังนี้

Status Attribute List

Status	Class	rid
Single	Good	0
Marriage	Good	1
Marriage	Good	2
Single	Bad	3
Divorce	Bad	4
Single	Good	5

⇩

Count matrix	Good	Bad
Single	2	1
Marriage	2	0
Divorce	0	1

รูปที่ 2.4 การหา histogram ด้วย Count Metrix

2.2.1.2 การหา Split points

ขณะที่ทำการสร้าง tree จุดหมายของแต่ละ node คือ การหา split point ที่ดีที่สุดที่จะเป็นตัวแบ่ง training record ออกเป็นแต่ละ leaf ค่าที่ได้ของ Split point จะเป็นเครื่องบ่งชี้ถึงประสิทธิภาพในการแบ่ง Class โดยในวิธีการของ SPRINT จะใช้ GINI index ในการหาค่า split point

GINI index

<u>สูตร</u>

(1) $gini(S) = 1-\sum p_j^2$

เมื่อ

S เป็น data set ที่มีตัวอย่าง n class

p_j เป็นค่าความถี่ของ class j ใน S

ตัวอย่างเช่น

มี 2 classes คือ Pos และ Neg และ p(Pos-elements),n(Neg-elements)

$P_{pos} = p/(p+n)$ $P_{neg} = n/(n+p)$

$gini(S) = 1- P_{pos}^{2} - P_{neg}^{2}$

dataset S เมื่อ split ออกเป็น S_1 และ S_2

$gini_{split}(S) = (p_1+n_1)/(p+n)* gini(S_1)+(p_2+n_2)/(p+n) * gini(S_2)$

Split point ที่ดีที่สุดคือ มีค่า split น้อยที่สุด

ขั้นตอนในการหา split point ในการสร้าง tree มี 2 ขั้นตอนดังนี้

1. หา attribute ที่จะเป็น test node คือ เหมาะสมในการที่จะใช้เป็น split point ในการวิเคราะห์ข้อมูล โดยใช้สูตรของ Gini ที่กล่าวมาแล้วข้างต้น โดยเลือก attribute ที่มีค่า $gini_{split}$ น้อยที่สุด
2. สร้างจุดแบ่งโดยดูจากค่า $gini_{split}$ ที่ต่ำที่สุดมาเป็นตัวกำหนดจุดแบ่ง โดยวิธีการในการกำหนดจุดแบ่งนั้นขึ้นอยู่กับประเภทของ attribute ดังนี้
   2.1 การแบ่งข้อมูล attribute ที่เป็นประเภทตัวเลข (numeric/continuous attribute) เป็นการแบ่งลักษณะ 2 ทาง (binary split) จากที่กล่าวมาข้างต้น $A \leq v$ โดย v เป็นตัวเลขที่เป็นไปได้ของ attribute A ที่ถูกเรียงลำดับแล้ว ซึ่งจะอยู่ในรูปแบบ $v_1, v_2, \ldots, v_n$ เมื่อได้ค่า best split point v_i ให้นำค่า $v_s + v_{s+1} / 2$ เป็น best split point
   2.2 การแบ่งข้อมูล attribute ที่เป็นประเภทจัดหมวดหมู่ (categorical attribute) โดยให้ S(A) เป็นเซตของค่าที่เป็นไปได้ของ attribute A เมื่อ $X \subset$ domain(A)ดังนั้นจำนวนซับเซตที่เป็นไปได้เท่ากับ 2^n

ตัวอย่างการสร้าง tree ด้วย SPRINT

ถ้าข้อมูลประกอบด้วย

1. อายุ (Age)
2. สถานภาพสมรส (Status)
3. ลักษณะลูกค้า (Class)

Training data

rid	Age	Status	Class
0	23	Single	Good
1	17	Marriage	Good
2	43	Marriage	Good
3	68	Single	Bad
4	32	Divorce	Bad
5	20	Single	Good

Initial เรียงตาม rid จากนั้นสร้าง attribute list ได้ดังนี้

Age attribute list

Age	Class	rid
17	Good	1
20	Good	5
23	Good	0
32	Bad	4
43	Good	2
68	Bad	3

Status attribute list

Status	Class	rid
Single	Good	0
Marriage	Good	1
Marriage	Good	2
Single	Bad	3
Divorce	Bad	4
Single	Good	5

รูปที่ 2.5 ตัวอย่างการสร้าง Tree

นำ attribute list มาตรวจสอบประเภทของ attribute ที่จะทดสอบว่าเป็นประเภทใด ในกรณีของ Age attribute list เป็นประเภท Continuous attribute แล้วหาค่า histograms ที่เป็นไปได้ทั้งหมดคือ Age ≤7, Age ≤0, Age ≤3, Age ≤2, Age ≤3, Age ≤8

Age ≤7

	Good	Bad
C_{below} (Age ≤7)	1	0
C_{above} (Age>17)	3	2

Gini(Age ≤7) = $1-((1/1)^2+(0/1)^2) = 0$

Gini(Age>17) = $1-((3/5)^2+(2/5)^2) = 0.48$

$Gini_{split} = (1/6)*(0)+(5/6)*(0.48) = 0.4$

Age ≤0

	Good	Bad
C_{below} (Age ≤0)	2	0
C_{above} (Age>20)	2	2

Gini(Age ≤0) = $1-((2/2)^2+(0/2)^2) = 0$

Gini(Age>20) = $1-((2/4)^2+(2/4)^2) = 0.5$

$Gini_{split} = (2/6)*(0)+(4/6)*(0.5) = 0.333$

Age ≤3

	Good	Bad
C_{below} (Age ≤3)	3	0
C_{above} (Age>23)	1	2

Gini(Age ≤3) = $1-((3/3)^2+(0/3)^2) = 0$

Gini(Age>23) = $1-((1/3)^2+(2/3)^2) = 0.444$

$Gini_{split} = (3/6)*(0)+(3/6)*(0.444) = 0.222$

รูปที่ 2.6 ตัวอย่างการหา Histogram

Age $\leq$2

	Good	Bad
C_{below} (Age $\leq$2)	3	1
C_{above} (Age>32)	1	1

Gini(Age $\leq$2) = $1-((3/4)^2+(1/4)^2) = 0.375$

Gini(Age>32) = $1-((1/2)^2+(1/2)^2) = 0.5$

$Gini_{split} = (4/6)*(0.375)+(2/6)*(0.5) = 0.416$

Age $\leq$3

	Good	Bad
C_{below} (Age $\leq$3)	4	1
C_{above} (Age>43)	0	1

Gini(Age $\leq$3) = $1-((4/5)^2+(1/5)^2) = 0.32$

Gini(Age>43) = $1-((0/1)^2+(1/1)^2) = 0$

$Gini_{split} = (5/6)*(0.32)+(1/6)*(0) = 0.266$

Age $\leq$8

	Good	Bad
C_{below} (Age $\leq$8)	4	2
C_{above} (Age>68)	0	0

Gini(Age $\leq$8) = $1-((4/6)^2+(2/6)^2) = 0.444$

Gini(Age>68) = 1

$Gini_{split} = (6/6)*(0.444)+(0/6)*(1) = 0.444$

รูปที่ 2.7 (ต่อ)

ในกรณีของ Status attribute list เป็นประเภท Categorical attribute แล้วหาค่า histograms ได้ทั้งหมด 2^n เมื่อ n คือจำนวนค่าใน attribute

n = 3 (Single, Marriage, Divorce)

Count matrix	Good	Bad
Single	2	1
Marriage	2	0
Divorce	0	1

Gini(Single) = $1-((2/3)^2+(1/3)^2) = 0.444$

Gini(Marriage) = $1-((2/2)^2+(0/2)^2) = 0$

Gini(Divorce) = $1-((0/1)^2+(1/1)^2) = 0$

Gini(Single, Marriage) = $1-((4/5)^2+(1/5)^2) = 0.32$

Gini(Single, Divorce) = $1-((2/4)^2+(2/4)^2) = 0.5$

Gini(Marriage, Divorce) = $1-((2/3)^2+(1/3)^2) = 0.445$

$Gini_{split}$(Single) = $(3/6)*(0.444)+(3/6)*(0.445) = 0.4445$

$Gini_{split}$(Marriage) = $(2/6)*(0)+(4/6)*(0.5) = 0.33$

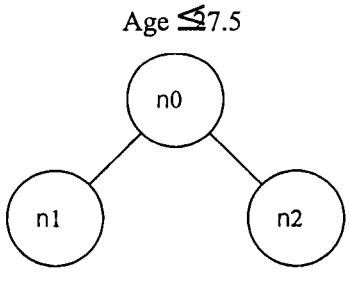
$Gini_{split}$(Divorce) = $(1/6)*(0)+(5/6)*(0.32) = 0.266$

$Gini_{split}$(Single, Marriage) = $(5/6)*(0.32)+(1/6)*(0) = 0.266$

$Gini_{split}$(Single, Divorce) = $(4/6)*(0.5)+(2/6)*(0) = 0.333$

$Gini_{split}$(Marriage, Divorce) = $(3/6)*(0.445)+(3/6)*(0.444) = 0.4445$

จาก การหา Gini index ทั้ง 2 histograms จะได้ Split point ซึ่งมีค่า Gini Index น้อยที่สุด คือตำแหน่ง Age $\leq$3 ดังนั้น Best split point คือ (23+32)/2 = 27.5



รูปที่ 2.8 การแบ่ง tree รอบที่1

table ใน node 1

rid	Age	Status	Class
0	23	Single	Good
1	17	Marriage	Good
5	20	Single	Good

Table ใน node 2

rid	Age	Status	Class
2	43	Marriage	Good
3	68	Single	Bad
4	32	Divorce	Bad

รูปที่ 2.9 แถวที่เหลือจากการแบ่ง tree รอบที่1

จะเห็นว่า Attribute เป้าหมาย (Class) ใน node 1 เป็น Good หมดแล้วจึงไม่ต้องแบ่งต่อ ส่วน node2 ยังไม่เป็นประเภทเดียวกันทั้งหมด จึงต้องแบ่งต่อได้ดังนี้

Table ใน node 2

rid	Age	Status	Class
2	43	Marriage	Good
3	68	Single	Bad
4	32	Divorce	Bad

Age attribute list

Age	Class	rid
32	Bad	4
43	Good	2
68	Bad	3

รูปที่ 2.10 การแบ่ง tree รอบที่2

Age ≤ 32

	Good	Bad
C_{below} (Age ≤ 32)	0	1
C_{above} (Age>32)	1	1

Gini(Age ≤ 32) = $1-((0/1)^2+(1/1)^2) = 0$

Gini(Age>32) = $1-((1/2)^2+(1/2)^2) = 0.5$

$Gini_{split} = (1/3)*(0)+(2/3)*(0.5) = 0.333$

Age ≤ 43

	Good	Bad
C_{below} (Age ≤ 43)	1	1
C_{above} (Age>43)	0	1

Gini(Age ≤ 43) = $1-((1/2)^2+(1/2)^2) = 0.5$

Gini(Age>43) = $1-((0/1)^2+(1/1)^2) = 0$

$Gini_{split} = (2/3)*(0.5)+(1/3)*(0) = 0.333$

Age ≤ 68

	Good	Bad
C_{below} (Age ≤ 68)	1	2
C_{above} (Age>68)	0	0

Gini(Age ≤ 68) = $1-((1/3)^2+(2/3)^2) = 0.444$

Gini(Age>68) = 1

$Gini_{split} = (3/3)*(0.444)+(0/3)*(1) = 0.444$

รูปที่ 2.11 การหาค่า Histogram

Status attribute list

Status	Class	rid
Single	Bad	3
Marriage	Good	2
Divorce	Bad	4

n = 3 (Single, Marriage, Divorce)

Count matrix	Good	Bad
Single	0	1
Marriage	1	0
Divorce	0	1

Gini(Single) = $1-((0/1)^2+(1/1)^2) = 0$

Gini(Marriage) = $1-((1/1)^2+(0/1)^2) = 0$

Gini(Divorce) = $1-((0/1)^2+(1/1)^2) = 0$

Gini(Single, Marriage) = $1-((1/2)^2+(1/2)^2) = 0.5$

Gini(Single, Divorce) = $1-((0/2)^2+(2/2)^2) = 0$

Gini(Marriage, Divorce) = $1-((1/2)^2+(1/2)^2) = 0.5$

$Gini_{split}$(Single) = $(1/3)*(0)+(2/3)*(0.5) = 0.333$

$Gini_{split}$(Marriage) = $(1/3)*(0)+(2/3)*(0) = 0$

$Gini_{split}$(Divorce) = $(1/3)*(0)+(2/3)*(0.5) = 0.333$

$Gini_{split}$(Single, Marriage) = $(2/3)*(0.5)+(1/3)*(0) = 0.333$

$Gini_{split}$(Single, Divorce) = $(2/3)*(0)+(1/3)*(0) = 0$

$Gini_{split}$(Marriage, Divorce) = $(2/3)*(0.5)+(1/3)*(0.444) = 0.481$

รูปที่ 2.11 (ต่อ)

จาก การหา Gini index ทั้ง 2 histograms จะได้ Split point ซึ่งมีค่า Gini Index น้อยที่สุด คือตำแหน่ง Status Marriage ได้ tree ดังนี้

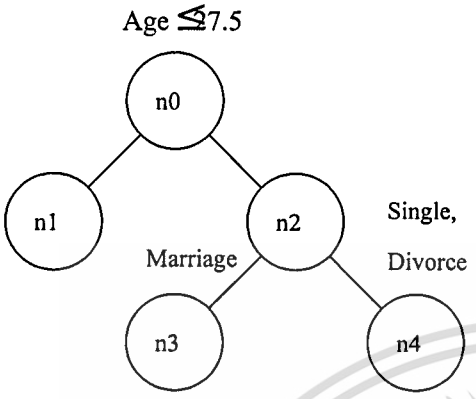


table ใน node 3

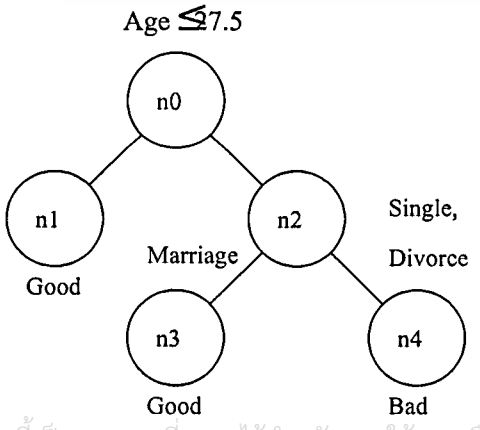
rid	Age	Status	Class
2	43	Marriage	Good

table ใน node 4

rid	Age	Status	Class
3	68	Single	Bad
4	32	Divorce	Bad

รูปที่ 2.12 แถวที่เหลือจากการแบ่ง tree รอบที่2

จะเห็นว่าทั้ง node 3 เป็นประเภทเดียวกันทั้งหมดและ node 4 เป็นประเภทเดียวกันทั้งหมดจึงหยุดการ Split



รูปที่ 2.13 Tree ที่แบ่งเสร็จแล้ว

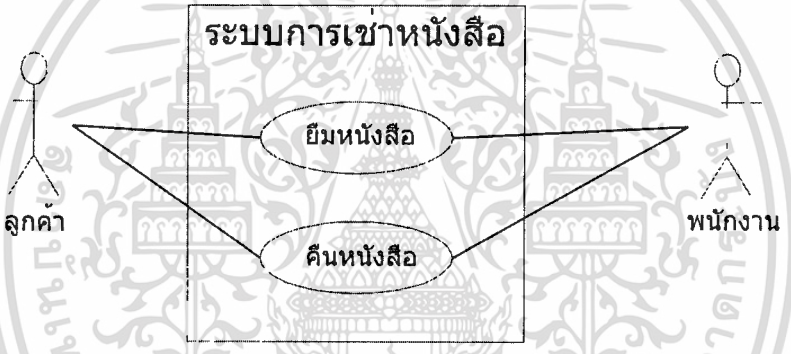
2.3 ยูเอ็มแอล (UML,Unified Modeling Language)

ยูเอ็มแอล เป็นภาษาสัญลักษณ์รูปภาพ สำหรับใช้ในการสร้างโมเดลเชิงวัตถุ ยูเอ็มแอลมีลักษณะของเมต้าโมเดล (metamodel) คือเป็นโมเดลที่เอาไว้อธิบายโมเดลอื่นๆอีกที สำหรับระบบงานของการพัฒนาซอฟต์แวร์นั้น จำเป็นที่จะต้องมีภาษาที่สามารถอธิบายสถาปัตยกรรมของระบบในมุมมองต่างๆได้

ประเภทไดอะแกรมของ UML

2.3.1 ยูสเคสไดอะแกรม (Use Case Diagram)

ยูสเคสไดอะแกรมใช้แสดงความสัมพันธ์ระหว่าง แอ็กเตอร์และยูสเคส ข้อดีของยูสเคสไดอะแกรมคือเราจะเห็นได้อย่างชัดเจนว่าขอบเขตของระบบที่เรากำลังสนใจอยู่มีอยู่แค่ไหน



รูปที่ 2.14 แสดงยูสเคสไดอะแกรม

1 สัญลักษณ์ที่ใช้ในยูสเคสไดอะแกรม

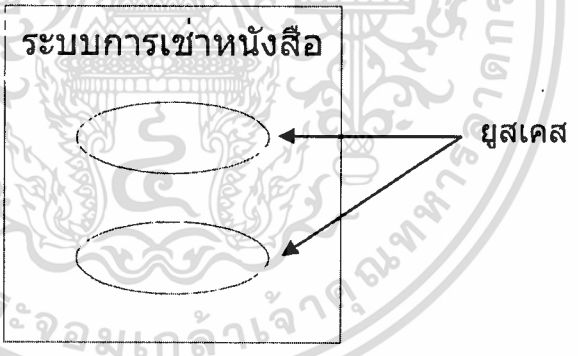
- **ระบบ (Systems)** สิ่งที่ผู้พัฒนาทำการพัฒนาเรียกว่าระบบ งานที่เป็นระบบนั้นไม่จำเป็นต้องเป็นงานที่เกี่ยวกับคอมพิวเตอร์เสมอไป ระบบในยูสเคสไดอะแกรมจะถูกแทนด้วยรูปกล่องสี่เหลี่ยม ซึ่งบรรจุยูสเคสอยู่ภายในและมีชื่อของระบบเขียนอยู่ข้างในกล่องสี่เหลี่ยม

รูปที่ 2.15 แสดงสัญลักษณ์ของระบบ

- ยูสเคส (Use Case) คือตัวระบบที่เรากำลังสนใจอยู่ซึ่งจะบอกว่าระบบจะทำอะไร ยูสเคสแทนด้วยรูปวงรีมีชื่อยูสเคสอยู่ข้างใน และทุกยูสเคสจะอยู่ในกรอบสี่เหลี่ยม (ระบบ)

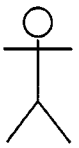
คุณสมบัติของยูสเคส มีดังนี้

1. ต้องถูกกระทำโดยแอ็กเตอร์
2. รับข้อมูลจากแอ็กเตอร์ และส่งให้แอ็กเตอร์



รูปที่ 2.16 แสดงสัญลักษณ์ของยูสเคส

- แอ็กเตอร์ (Actor) คือผู้กระทำกับระบบ โดยจะเป็นคนหรือไม่ก็ได้ นั่นคือแอ็กเตอร์เป็นผู้ที่กระทำให้เกิดเหตุการณ์ แอ็กเตอร์จะเป็นส่วนที่อยู่นอกระบบ (ระบบควบคุมไม่ได้) และสามารถทำการส่งข้อมูล หรือรับข้อมูล หรือแลกเปลี่ยนข้อมูลข่างสารกับระบบ นอกจากนี้ระบบยังสามารถทำตัวเป็นแอ็กเตอร์ได้อีกด้วย เช่นกรณีที่เชื่อมต่อกับอีกระบบหนึ่งภายนอก แอ็กเตอร์แทนด้วยรูปคน ทุกแอ็กเตอร์จะต้องกระทำกับยูสเคส อย่างน้อยหนึ่งยูสเคสเสมอ



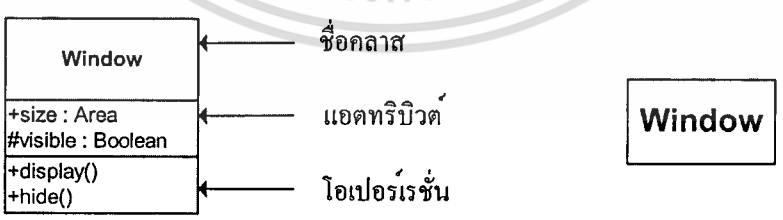
รูปที่ 2.17 แสดงสัญลักษณ์แอ็กเตอร์

- เส้นแสดงความสัมพันธ์ (Relationship) มีความสัมพันธ์มากมายระหว่างแต่ละยูสเคส หรือระหว่างแอ็กเตอร์และยูสเคส ดังนี้

1. **Association** จะมีการกำหนดถึงบทบาทของแต่ละแอ็กเตอร์ในยูสเคสที่มีความสัมพันธ์ร่วมกัน
2. **Extends** การที่นำเอายูสเคสเดิมที่มีอยู่แล้วมาเพิ่มการทำงานบางอย่าง
3. **Generalization** เป็นการถ่ายทอดคุณสมบัติ หรือพฤติกรรมบางอย่างจากยูสเคสหนึ่งไปยังอีกยูสเคสหนึ่ง หรือจากแอ็กเตอร์หนึ่งไปยังอีกแอ็กเตอร์หนึ่ง
4. **Include/Uses** ยูสเคสที่ถูกยูสเคสอื่นๆเรียกใช้งานมากกว่าหนึ่งยูสเคสขึ้นไปซึ่งยูสเคสหนึ่งๆอาจจำเป็นต้องอาศัยการทำงานของยูสเคสอื่นๆ

2.3.2 คลาสไดอะแกรม (Class Diagram)

เป็นไดอะแกรมที่ใช้อธิบายความสัมพันธ์ระหว่างคลาส คลาสเป็นการนำเอากลุ่มของออบเจ็กต์มาอธิบายความหมาย การกำหนดคลาสจะแทนด้วยสัญลักษณ์รูปสี่เหลี่ยมผืนผ้า โดยแบ่งเป็น 3 ส่วน



รูปที่ 2.18 แสดงสัญลักษณ์ของคลาส

1) ชื่อคลาส จะขึ้นต้นด้วยตัวอักษรตัวใหญ่แบบหนา และจะเอียงในกรณีที่เป็น *Abstract Class*

2) แอตทริบิวต์ (Attribute) เป็นการบอกถึงคุณสมบัติของคลาส ประกอบด้วยองค์ประกอบย่อยดังนี้

- ชนิดการเข้าถึง

Public (+) วัตถุทั้งหมดภายในระบบสามารถเข้าถึงได้

Private (-) วัตถุที่สามารถเข้าถึงต้องเป็นวัตถุที่อยู่ในคลาส หรือสับคลาสเดียวกัน

Protect (#) วัตถุที่สามารถเข้าถึงจะต้องเป็นวัตถุที่อยู่ในคลาสเดียวกันเท่านั้น

- ชื่อของแอตทริบิวต์

- ประเภทของแอตทริบิวต์ ซึ่งจะอยู่ต่อจากเครื่องหมายโคล่อน (:)

- ค่าเริ่มต้นของแอตทริบิวต์ ซึ่งอาจมีหรือไม่ก็ได้แต่ถ้ามีจะอยู่ต่อจากเครื่องหมายเท่ากับ

3) โอเปอเรชั่น (Operation) คือพฤติกรรมที่สามารถกระทำกับออบเจ็กต์ได้ประกอบด้วยองค์ประกอบย่อย ดังนี้

- ชนิดของการเข้าถึง

1. Public (+) วัตถุทั้งหมดภายในระบบสามารถเข้าถึงได้
2. Private (-) วัตถุที่สามารถเข้าถึงต้องเป็นวัตถุที่อยู่ในคลาส หรือสับคลาสเดียวกัน
3. Protect (#) วัตถุที่สามารถเข้าถึงจะต้องเป็นวัตถุที่อยู่ในคลาสเดียวกันเท่านั้น

- ชื่อของโอเปอร์เรชั่น

- พารามิเตอร์ (Parameter)

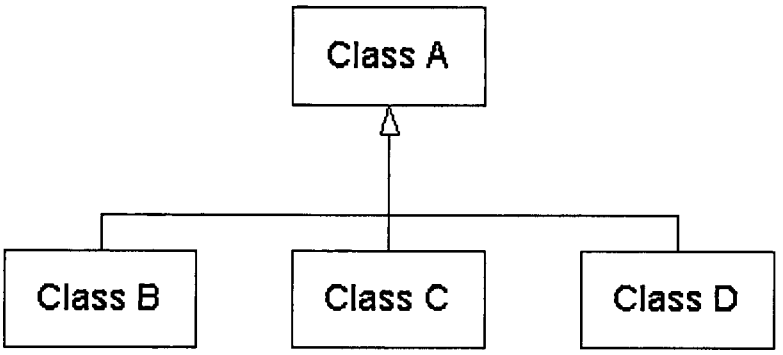
- ประเภทของค่าที่ส่งคืน (Return Type)

ความสัมพันธ์ระหว่างคลาส (Relationships)

1) Dependency ความสัมพันธ์แบบนี้เกิดขึ้นเมื่อคลาสหนึ่งมีการเปลี่ยนแปลงแล้วส่งผลกระทบต่ออีกคลาสหนึ่ง สัญลักษณ์ คือลูกศรที่มีลักษณะที่เป็นเส้นประ โดยที่คลาสที่อยู่ตรงส่วนหางของลูกศรจะขึ้นอยู่กับคลาสที่อยู่ตรงส่วนหัวของลูกศร

รูปที่ 2.19 แสดงความสัมพันธ์แบบ Dependency

2) Generalization เป็นการถ่ายทอดคุณสมบัติหรือพฤติกรรมบางอย่างจากคลาสหนึ่งไปยังอีกคลาสหนึ่ง สัญลักษณ์ คือเส้นตรงที่มีปลายข้างหนึ่งเป็นลูกศรหัวโปร่ง



รูปที่ 2.20 แสดงความสัมพันธ์แบบ Generalization

3) Association เป็นความสัมพันธ์ที่อธิบายถึงการเชื่อมโยงระหว่างคลาส การแสดงความสัมพันธ์จะประกอบด้วย

1. Association Name ชื่อของความสัมพันธ์
2. Multiplicity การกำหนดปริมาณของคลาส หรืออบเจ็กต์ที่สัมพันธ์กันอยู่

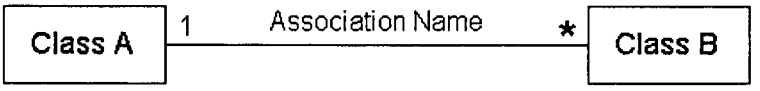
ตารางที่ 2.4 แสดง Multiplicity

1	มี 1 คลาส หรือ ออบเจ็กต์
N	มี n คลาส หรือ ออบเจ็กต์
0..1	ไม่มี หรือมีได้เพียง 1 คลาส หรือ ออบเจ็กต์
1..*	มีได้ตั้งแต่ 1 คลาส หรือ ออบเจ็กต์ขึ้นไป
*	ไม่มี หรือ มีได้ไม่จำกัด

รูปแบบของความสัมพันธ์

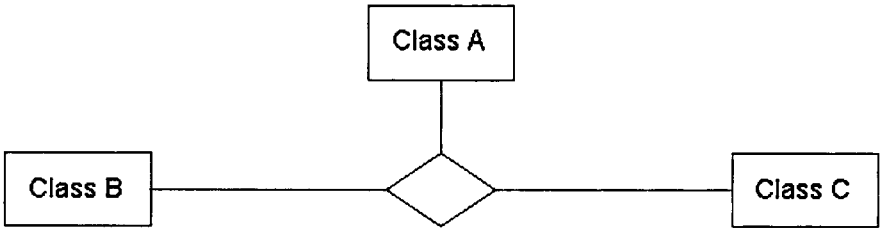
1) Normal Association รูปแบบของ Normal Association มีดังนี้

- Binary Association เป็นความสัมพันธ์ระหว่างคลาส 2 คลาส



รูปที่ 2.21 แสดงความสัมพันธ์แบบ Binary Association

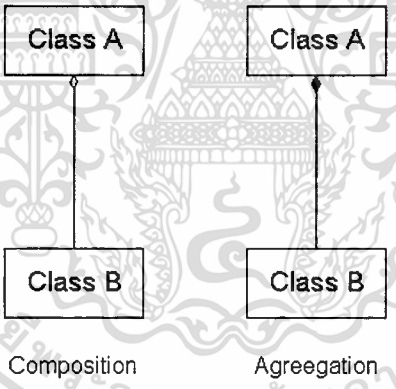
- N-ary Association ความสัมพันธ์ระหว่างคลาสจำนวน 3 คลาสหรือมากกว่า 3 คลาสขึ้นไป โดยความสัมพันธ์จะใช้สัญลักษณ์สี่เหลี่ยมขนมเปียกปูน



รูปที่ 2.22 แสดงความสัมพันธ์แบบ N-ary Association

2. Aggregation เป็นความสัมพันธ์ในแง่ของการรวมกันหรือการประกอบกัน ซึ่งองค์ประกอบที่นำมาประกอบกันนั้นจะมีหรือไม่ก็ได้ สามารถแสดงด้วยเส้นตรงโยงระหว่างคลาสโดยมีสัญลักษณ์ข้าวหลามตัดโปร่ง ติดอยู่ระหว่างปลายเส้นความสัมพันธ์กับคลาส

3. Composition เป็นความสัมพันธ์ในแง่ของการรวมกันหรือการประกอบกัน ซึ่งองค์ประกอบที่นำมาประกอบนั้นเป็นปัจจัยสำคัญที่ไม่มีไม่ได้ สามารถแสดงด้วยเส้นตรงโยงระหว่างคลาสโดยมีสัญลักษณ์ข้าวหลามตัดทึบ ติดอยู่ระหว่างปลายเส้นความสัมพันธ์กับคลาส



รูปที่ 2.23 แสดงสัญลักษณ์ของความสัมพันธ์แบบ Composition และ Aggregation

2.3.3 บีเฮฟเยอร์ไดอะแกรม (Behavior Diagram)

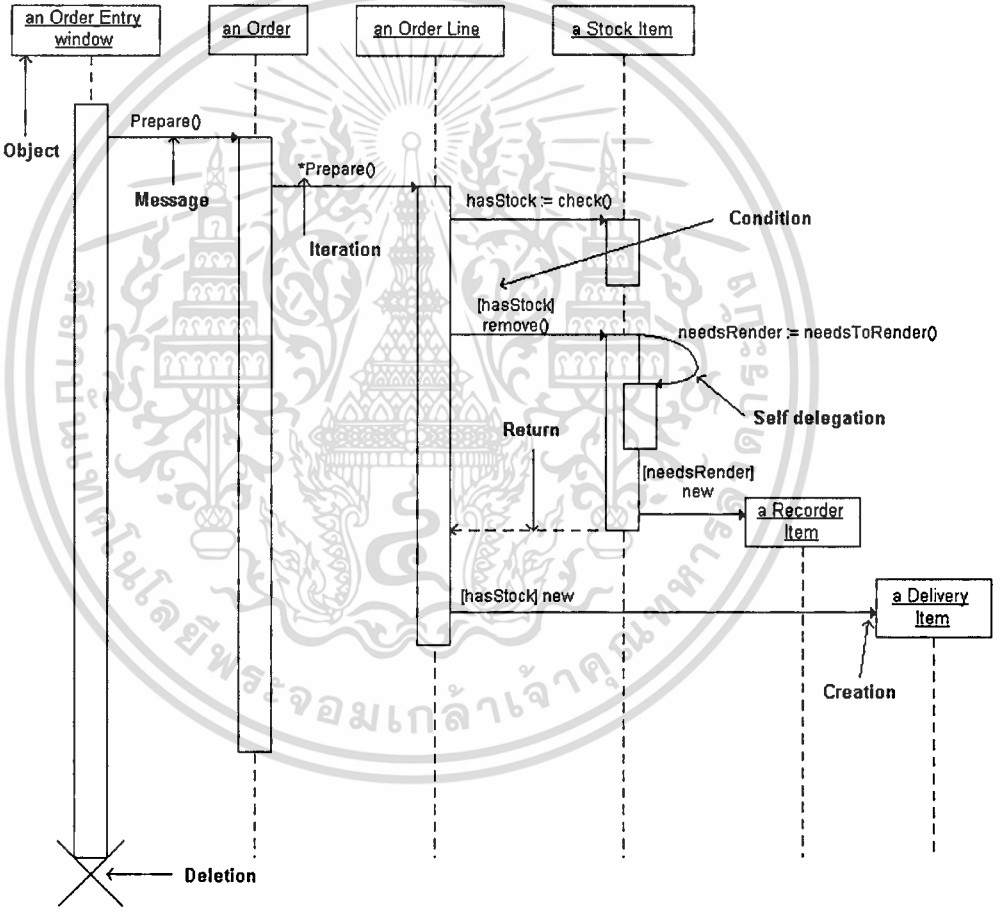
ไดอะแกรมที่บอกถึงพฤติกรรมของตัวระบบเส้นที่ใช้ในการส่งข้อมูล (Message) มีรูปแบบ ดังนี้

1. เส้นทั่วไป (→) ใช้ส่งข้อมูลแบบทั่วๆไปไม่เฉพาะเจาะจง
2. เส้นซิงโครนัส (⟶) เป็นเส้นที่ส่งข้อมูลไปแล้วจำเป็นต้องรอการตอบกลับ
3. เส้นอะซิงโครนัส (⟍) เป็นเส้นที่ส่งข้อมูลไปแล้วไม่จำเป็นต้องรอการตอบกลับ
4. เส้นส่งกลับจากการเรียกฟังก์ชั่น (⇢) มักใช้คู่กับเส้นทั่วไป เมื่อมีค่าที่ต้องการส่งกลับมา

บีเฮฟเยอร์แอ็กชั่นไดอะแกรม ประกอบด้วยไดอะแกรม ดังนี้

1. อินเทอรีแอ็กชั่นไดอะแกรม (Interaction Diagram) เป็นไดอะแกรมที่แสดงพฤติกรรมการแลกเปลี่ยนข้อมูล (Message) ของออบเจ็กต์ต่างๆ ซึ่งประกอบด้วย

2. ซีเควนซ์ไดอะแกรม (Sequence Diagram) ไดอะแกรมนี้แสดงให้เห็นว่าวัตถุแต่ละตัวติดต่อสื่อสารกัน และมีขั้นตอนการทำงานอย่างไร โดยเน้นไปที่แกนเวลา ซีเควนซ์ไดอะแกรมประกอบด้วย 2 แกน คือแกนนอนเป็นแกนที่แสดงขั้นตอนการทำงานหรือการส่งเมสเสจระหว่างวัตถุ และแกนตั้งเป็นแกนเวลา

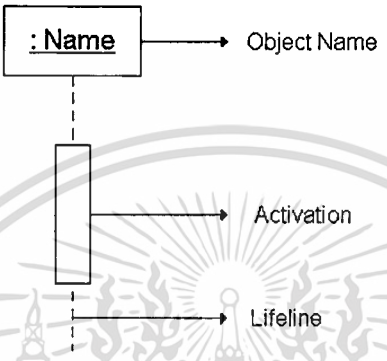


รูปที่ 2.24 แสดงซีเควนซ์ไดอะแกรม

ซีเควนซ์ไดอะแกรม มีองค์ประกอบอยู่ 3 ส่วน คือ

1. ออบเจ็กต์ (Object) ประกอบด้วย

- Object Name ประกอบด้วย ชื่อออบเจ็กต์: ชื่อคลาส (:ชื่อออบเจ็กต์อาจละไว้ได้) และขีดเส้นใต้
- Lifeline แสดงถึงชีวิตของวัตถุ
- Activation ช่วงเวลาที่วัตถุกำลังปฏิบัติงาน

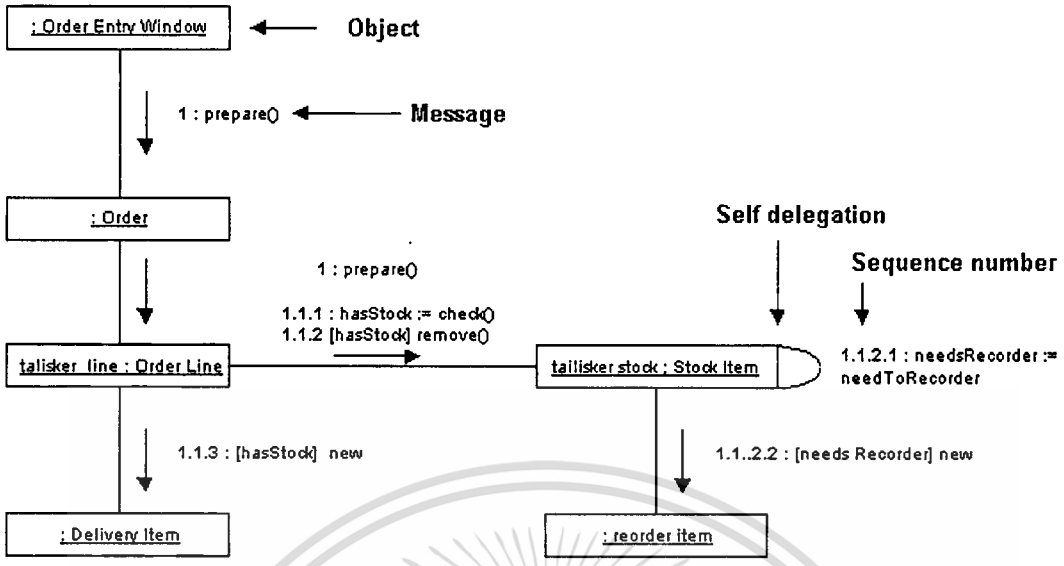


รูปที่ 2.25 แสดงส่วนประกอบของออบเจ็กต์

2. เมสเสจ (Message) เป็นการติดต่อระหว่างออบเจ็กต์

3. ช่วงเวลา (Time) การแสดงเวลาของซีเควนซ์ไดอะแกรมเป็นลักษณะจากบนลงล่าง เมสเสจที่อยู่ด้านบนจะเป็นส่วนที่เกิดขึ้นก่อนเมสเสจที่อยู่ด้านล่าง

3. คอลแลบอเรชั่นไดอะแกรม (Collaboration Diagram) จะแสดงขั้นตอนการทำงานของยูสเคสเช่นเดียวกับซีเควนซ์ไดอะแกรม แต่วิธีการเขียนจะต่างกัน คือ คอลแลบอเรชั่นไดอะแกรมจะไม่แสดงถึงแกนเวลาอย่างชัดเจน



รูปที่ 2.26 แสดงคอลแลบอเรชั่นไดอะแกรม

คอลแลบอเรชั่นไดอะแกรมประกอบไปด้วย

1. ออบเจ็กต์ ประกอบด้วย ชื่อของออบเจ็กต์/บทบาท:ชื่อคลาส และขีดเส้นใต้

2. ลิงค์ เส้นเชื่อมกันระหว่างวัตถุ ซึ่งขั้นแสดงขั้นตอนการทำงานตามทิศทางของลูกศร โดยมีตัวเลขกำกับไว้เพื่อบอกขั้นตอนการทำงาน

2.3.4 สเตตชาร์ตไดอะแกรม (Statechart Diagram)

บอกถึงสถานะต่างๆ ของคลาสในระบบว่ามีสถานะอะไรบ้าง จะเปลี่ยนสถานะเมื่อเกิดเหตุการณ์อะไร

สัญลักษณ์ในเสตทชาร์ตไดอะแกรม

1. จุดเริ่มต้น (Initial)

2. จุดสิ้นสุด (Final)

3. Transition Line เส้นที่เชื่อมโยงสเตท เราสามารถใส่รายละเอียดบางอย่างเข้าไปได้

Trigger event / action

เช่น อาจมีการใส่เหตุการณ์ที่ทำให้เกิด Transition ได้เรียกเหตุการณ์นั้นว่า Trigger event

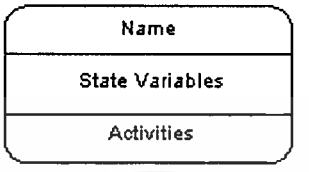
4. สเตท (State) State

รายละเอียดในสเตท

- ชื่อของสถานะการทำงาน (Name)

- ตัวแปรของสถานะการทำงาน (State Variables) ค่าที่ใช้ในสเตท เช่น ต้องการทราบว่าสเตทนี้ทำงานมาแล้วกี่ครั้งจะใช้ State Variable ช่วยในการนับ

- กิจกรรม (Activities) อธิบายถึงพฤติกรรมการทำงานของเสตท

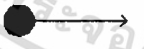


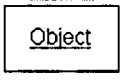
รูปที่ 2.27 แสดงรายละเอียดภายในสเตท

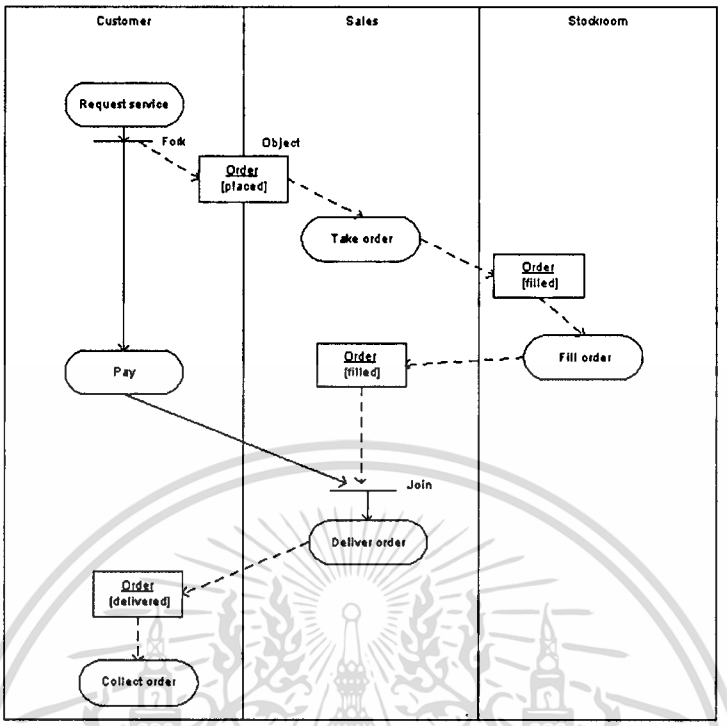
2.3.5 แอ็กทิวิตี้ไดอะแกรม (Activity Diagram)

เป็นการแสดงขั้นตอนการทำงานเช่นเดียวกับซีเควนซ์ไดอะแกรม แต่จะเน้นไปที่งานย่อยของวัตถุและสามารถแสดงรายละเอียดของกิจกรรมระหว่างออบเจ็กต์ต่างๆ ได้แอ็กทิวิตี้ไดอะแกรมจะเปลี่ยนสถานะเองได้ไม่ต้องมีเหตุการณ์ที่กำหนดไว้ในไดอะแกรมแต่จะเปลี่ยนสถานะเองตามกระบวนการทำงานแอ็กทิวิตี้ไดอะแกรมจะมีลักษณะเดียวกับ Flowchart นอกจากนั้นไดอะแกรมยังแบ่งเป็นสวิมเลนซึ่งแบ่งกลุ่มแอ็กทิวิตี้ออกเป็นเลนๆ ตามออบเจ็กต์

สัญลักษณ์ที่ใช้ในแอ็กทิวิตี้ไดอะแกรม

1. จุดเริ่มต้น (Initial) 
2. จุดสิ้นสุด (Final) 
3. Fork มีงานหลายงานที่มีการทำงานไปพร้อมกัน
4. Join แสดงการทำงานที่ต้องทำกิจกรรมเดิมให้เสร็จสิ้นก่อนจึงเริ่มกิจกรรมต่อไป
5. แอ็กทิวิตี้ (Activity) 

6. ออบเจ็กต์ (Object) 

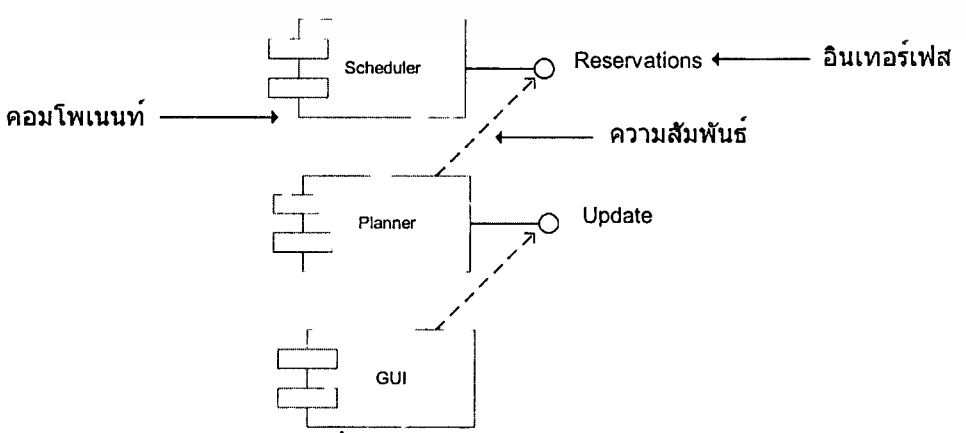


รูปที่ 2.28 แสดงแอ็กทิวิตี้ไดอะแกรม

2.3.6 อิมพลีเมนต์เตชั่นไดอะแกรม (Implement Diagram)

เป็นสถาปัตยกรรมที่เกิดจากการมองระบบในความเป็นจริง ประกอบไปด้วย 2 ไดอะแกรม คือ

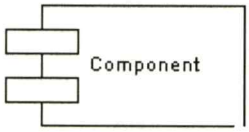
1) **คอมโพเนนต์ไดอะแกรม (Component Diagram)** เป็นการอธิบายถึงซอฟต์แวร์ต่างๆ ที่เป็นคอมโพเนนต์ของระบบว่ามีความสัมพันธ์และเชื่องต่อกันอย่างไร สัญลักษณ์ถูกแทนด้วยสี่เหลี่ยมที่ประกอบด้วยสี่เหลี่ยนเล็กๆ อีก 2 รูปติดอยู่ที่ขอบด้านซ้าย



รูปที่ 2.29 แสดงสัญลักษณ์ของคอมโพเนนต์

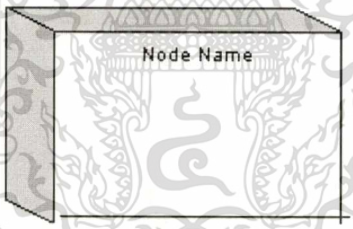
องค์ประกอบหลักของคอมโพเนนตีไดอะแกรม

1. คอมโพเนนต์ (Component)
2. อินเทอร์เฟส (Interface)
3. ความสัมพันธ์ (Relationship)

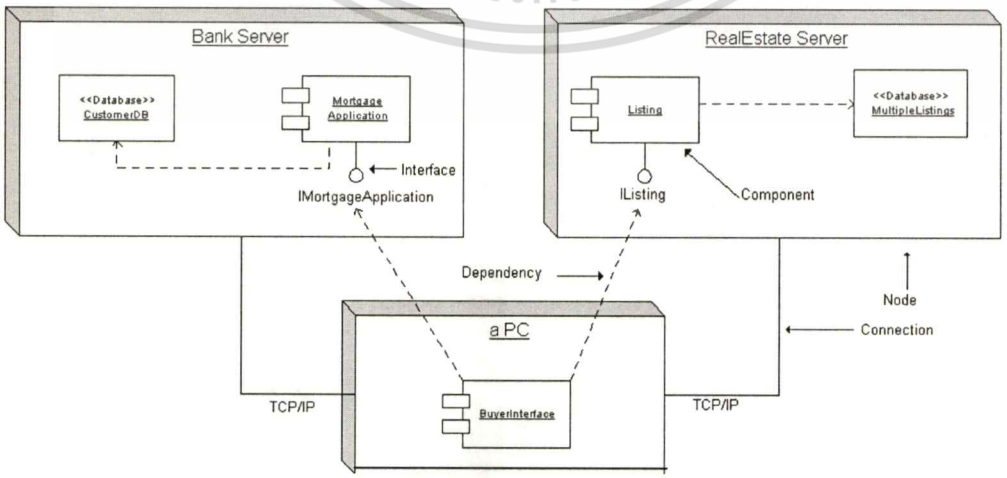


รูปที่ 2.30 แสดงองค์ประกอบหลักของคอมโพเนนต์ไดอะแกรม

2) ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram) เป็นไดอะแกรมแสดงการเชื่อมต่ออุปกรณ์ฮาร์ดแวร์ในระบบ และมักใช้ร่วมกับคอมโพเนนต์ไดอะแกรม โดยภายในฮาร์ดแวร์ (โหนด) จะประกอบไปด้วยซอฟต์แวร์คอมโพเนนต์ การอธิบายความสัมพันธ์จะเหมือนกับคอมโพเนนต์ไดอะแกรมสัญลักษณ์ในการวาด ดีพลอยเมนต์ไดอะแกรมจะแทนด้วยรูปลูกบาศก์ 3 มิติ ภายในบรรจุชื่อที่แสดงถึงประเภทของโหนดใด



รูปที่ 2.31 แสดงสัญลักษณ์ของฮาร์ดแวร์ (โหนด)



รูปที่ 2.32 แสดงดีพลอยเมนต์ไดอะแกรม

2.4 การจัดการฐานข้อมูลด้วยภาษา SQL

ภาษา SQL เป็นภาษาที่ใช้ในการจัดการกับฐานข้อมูลซึ่งมีคำสั่งทั้งหมด 3 ประเภท ได้แก่ คำสั่งที่ใช้สำหรับการกำหนด (Data Definition Language), คำสั่งที่ใช้สำหรับการทำงานกับข้อมูล (Data Manipulation Language) และ คำสั่งที่ใช้สำหรับการควบคุมการทำงาน (Data Control Language)

2.4.1 คำสั่ง Create Database

```
รูปแบบ : CREATE DATABASE db_name
```

เป็นคำสั่งที่ใช้ในการสร้าง database ขึ้นมาใหม่ ซึ่งในที่นี้ชื่อ db_name

2.4.2 คำสั่ง Drop Database

```
รูปแบบ : DROP DATABASE db_name
```

เป็นคำสั่งที่ใช้ในการลบ database ออก ซึ่งในที่นี้ชื่อ db_name

2.4.3 คำสั่ง Create Table

```
รูปแบบ : CREATE TABLE table_name (
                Column 1        data_type,
                Column 2        data_type,
                …               …
                Column N        data_type,
                PRIMARY KEY (column_name),
                Unique (column_name, column_name)
                )
```

CREATE TABLE เป็นคำสั่งที่ใช้ในการสร้างตารางสำหรับเก็บข้อมูลลงใน database ที่เราสร้างไว้ ในที่นี้สร้างตารางชื่อ table_name โดยกำหนดชื่อ column ต่างๆ ใน table ได้ตรง column 1, column 2, … , column N และ กำหนดชนิดของข้อมูลใน column นั้นๆ ตรง data_type สำหรับคีย์หลักของตารางนี้ซึ่งใช้ในการอ้างถึงข้อมูลในแถวต่างๆสามารถกำหนดให้ column ใด

เป็นคีย์หลักใน PRIMARY KEY ส่วน column ที่ไม่ใช่คีย์หลักแต่ต้องการให้ข้อมูลใน column นั้นไม่ซ้ำสามารถกำหนดใน Unique

2.4.4 คำสั่ง Select

```
รูปแบบ : SELECT column_name
          FROM table_name
          WHERE where_condition
          GROUP BY group_column
          HAVING having_condition
          ORDER BY column_name
```

SELECT เป็นคำสั่งที่ใช้ในการสืบค้นข้อมูลใน database โดย column_name จะเป็นชื่อ column ที่ต้องการสืบค้นหากต้องการสืบค้นหลายๆ column พร้อมกันสามารถใช้ "," คั่นระหว่างชื่อ column และถ้าต้องการทุก column สามารถใช้ " * " แทนชื่อ column ทั้งหมด

FROM เป็นการระบุชื่อ table ที่ต้องการสืบค้น table_name จะเป็นชื่อ table ที่ต้องการซึ่งสามารถใช้ "," คั่นได้เช่นเดียวกัน

WHERE เป็นการกำหนดเงื่อนไข where_condition คือเงื่อนไขของแถวข้อมูลที่ต้องการสืบค้น

GROUP BY เป็นการจัดกลุ่มของข้อมูลใน column ที่เหมือนกันไว้ด้วยกัน group_column เป็นชื่อ column ที่ต้องการจัดกลุ่ม

HAVING เป็นการกำหนดเงื่อนไขซึ่งจะใช้ได้ต่อเมื่อมีการใช้ GROUP BY โดยที่ having_condition จะเป็นการกำหนดเงื่อนไขของการ select

ORDER BY เป็นการเรียงลำดับข้อมูลที่สืบค้นออกมา column_name เป็นชื่อ column ที่ต้องการใช้เรียงลำดับ

2.4.5 คำสั่ง Insert

```
รูปแบบ : INSERT INTO table_name VALUE (value 1, value 2, … , value N)
```

INSERT เป็นคำสั่งที่ใช้ในการเพิ่มข้อมูลลงในตาราง table_name เป็นชื่อตารางที่ต้องการใส่ข้อมูล value 1, value 2, … , value N เป็นข้อมูลที่จะใส่ลงไปโดยเรียงตามตำแหน่ง column ของตารางนั้นๆ

2.4.6 คำสั่ง Update

รูปแบบ : UPDATE table_name SET column_name = expression

UPDATE เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงข้อมูลที่มีอยู่แล้วในตาราง table_name เป็นชื่อตารางที่ต้องการเปลี่ยนแปลง column_name เป็นชื่อ column ที่ต้องการเปลี่ยนแปลง

2.4.7 คำสั่ง Grant

รูปแบบ : GRANT priv_type ON table_name TO

GRANT เป็นคำสั่งที่ใช้ในการกำหนดสิทธิในการจัดการให้กับผู้ใช้โดย priv_type จะเป็นสิทธิที่จะให้แก่ผู้ใช้ table_name จะเป็นชื่อตารางที่ผู้ใช้มีสิทธิในการจัดการ user_name เป็นชื่อของผู้ใช้ที่ต้องการให้สิทธินั้น

2.4.8 คำสั่ง Revoke

รูปแบบ : REVOKE priv_type ON table_name FROM

REVOKE เป็นคำสั่งที่ใช้ในการถอนสิทธิในการจัดการของผู้ใช้โดย priv_type จะเป็นสิทธิที่จะถอน table_name จะเป็นชื่อตารางที่จะถอนสิทธิในการจัดการ user_name เป็นชื่อของผู้ใช้ที่ต้องการถอนสิทธินั้น

บทที่ 3
ระบบสนับสนุนการตัดสินใจด้วยเทคนิคต้นไม้ตัดสินใจ

3.1 ภาพรวมของระบบ

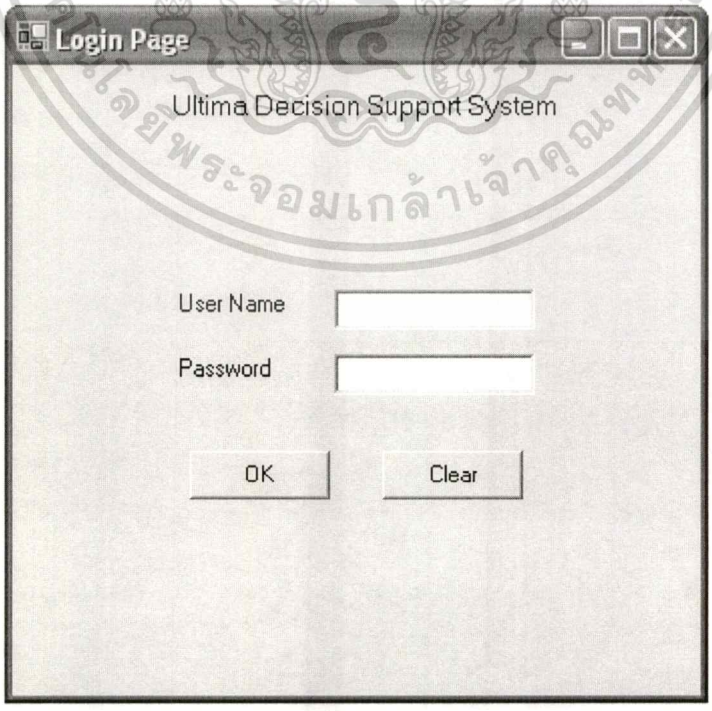
ในการพัฒนาโปรแกรมสนับสนุนการตัดสินใจสำหรับกรณีศึกษานี้พัฒนาโดยใช้ Visual Basic .NET 2003 มีการออกแบบระบบในส่วนของ Interface โดยจำแนกผู้ใช้งานระบบออกเป็น 2 ประเภท คือ admin ที่คอยดูแลสิทธิการเข้าใช้งานระบบทั้งการเพิ่ม ลบผู้มีสิทธิเข้าใช้ระบบ และ manager ซึ่งเป็นผู้มีสิทธิในการใช้งานระบบสนับสนุนการตัดสินใจ และออกแบบการทำงานของระบบโดยใช้ UML Diagram ดังต่อไปนี้

3.2 การวิเคราะห์ออกแบบส่วนหน้าจอ

หน้าจอของระบบมีทั้งหมด 11 หน้าจอ แต่ละหน้าจอมีลักษณะดังนี้

3.2.1 หน้าจอ Login

เป็นหน้าจอสำหรับเข้าสู่ระบบตรวจสอบผู้ใช้ว่ามีสิทธิในการเข้าใช้หรือไม่ โดยรับข้อมูลชื่อผู้ใช้ และ รหัสผ่าน ปุ่ม Clear ไว้สำหรับลบชื่อผู้ใช้และรหัสผ่านที่ค้างอยู่ error message จะแสดงเมื่อผู้ใช้ใส่ชื่อผู้ใช้ที่ไม่มีในระบบหรือใส่รหัสผ่านผิด

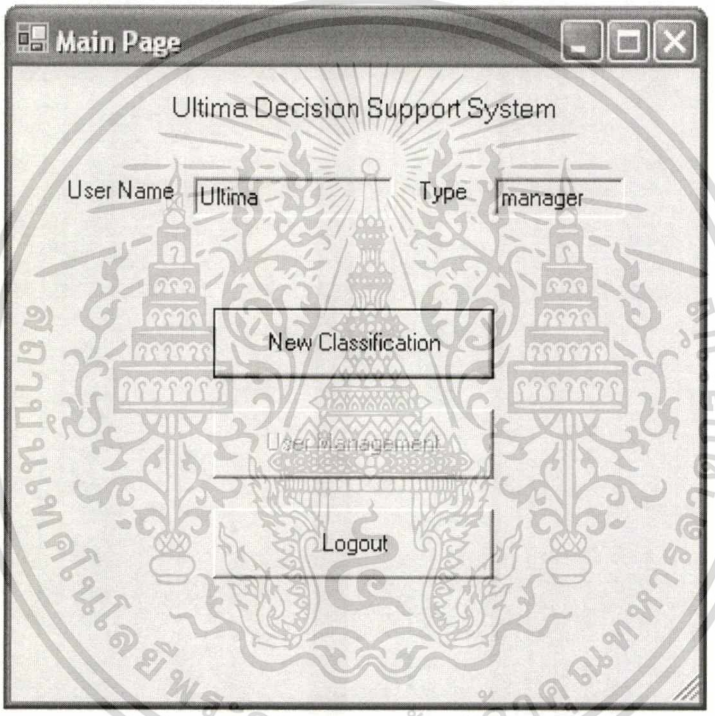


รูปที่ 3.1 หน้าจอ Login

3.2.2 หน้าจอหลัก

เป็นหน้าจอสำหรับเลือกการทำงาน มีการแสดงชื่อและประเภทผู้ใช้ โดยแบ่งผู้ใช้ออกเป็น 2 ประเภท คือ Manager มีสิทธิเข้าใช้ New Classification และ Admin มีสิทธิเข้าใช้ User Management

1. New Classification Button จะแสดงหน้าจอ Select Table
2. User Management Button จะแสดงหน้าจอ User Management
3. Logout Button จะ clear ข้อมูล Username, Password และ User type และแสดงหน้าจอ login

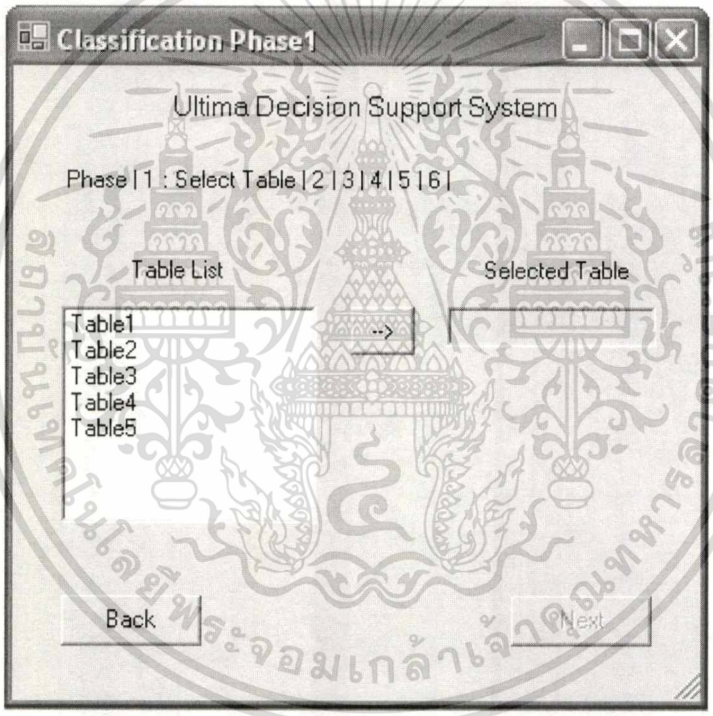


รูปที่ 3.2 หน้าจอหลัก

3.2.3 หน้าจอ Select Table

เป็นหน้าจอสำหรับเลือกตารางจากฐานข้อมูลที่จะนำมาจำแนกประเภทข้อมูล แสดงขั้นตอนในการทำ Classification สำหรับโปรแกรมนี้มี 6 ขั้นตอนด้วยกัน

1. Table List Box แสดงรายชื่อตารางที่มีในฐานข้อมูล
2. Select Table Text Box แสดงชื่อตารางที่เลือก
3. Select Button สำหรับเลือกชื่อตารางที่ต้องการนำมาจำแนกประเภท
4. Back Button จะแสดงหน้าจอหลัก
5. Next Button สามารถใช้งานได้เมื่อเลือกตารางแล้วจากนั้นแสดงหน้าจอ Select Target attribute

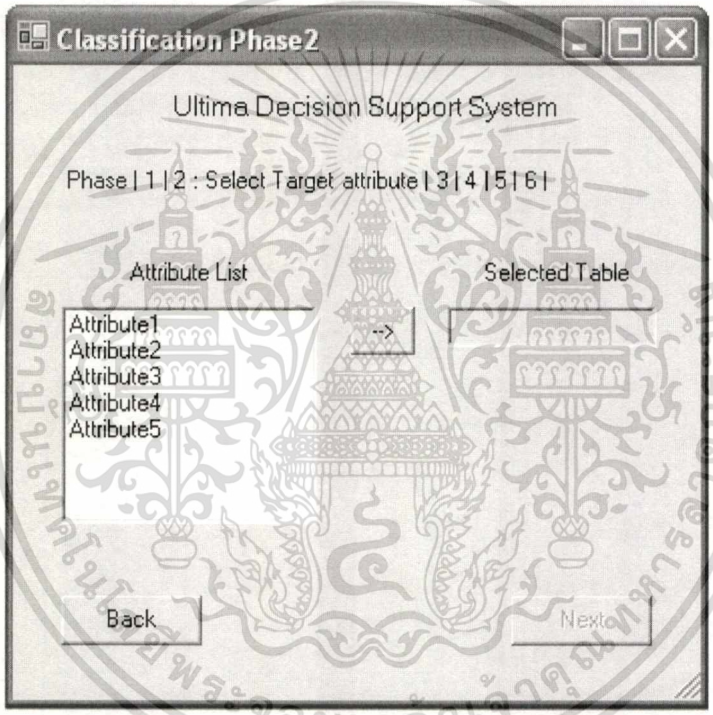


รูปที่ 3.3 หน้าจอ Select Table

3.2.4 หน้าจอ Select Target attribute

เป็นหน้าจอสำหรับเลือก Target attribute ซึ่งเป็นเป้าหมายในการจำแนกประเภทข้อมูล

1. Attribute List Box แสดงรายชื่อ attribute ที่มีในตารางที่เลือกไว้
2. Target attribute Text Box แสดงชื่อ attribute ที่เลือกเป็นเป้าหมาย
3. Select Button สำหรับเลือก attribute ที่ต้องการให้เป็นเป้าหมาย
4. Back Button จะแสดงหน้าจอ Select Table
5. Next Button สามารถใช้งานได้เมื่อเลือก attribute แสดงหน้าจอ Select Target attribute

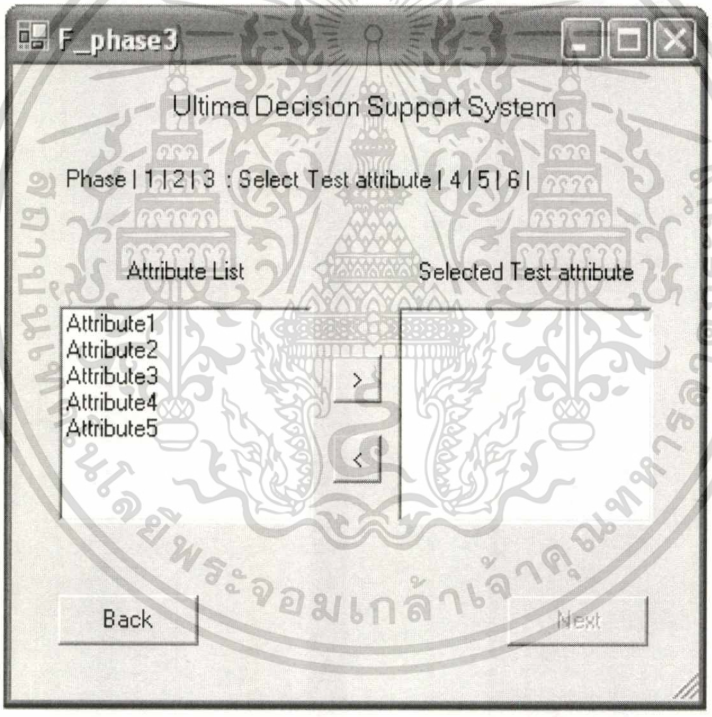


รูปที่ 3.4 หน้าจอ Select Target attribute

3.2.5 หน้าจอ Select Test attribute

เป็นหน้าจอสำหรับเลือก Test attribute ซึ่งเป็น attribute ที่นำมาทดสอบจำแนกประเภท

1. Attribute List Box แสดงรายชื่อ attribute ที่มีในตารางที่เลือกไว้ ยกเว้น Target attribute
2. Test attribute List Box แสดงรายชื่อ attribute ที่เลือกมาทดสอบ
3. Select Button สำหรับเลือก attribute ที่ต้องการนำมาทดสอบ
4. Delete Button สำหรับลบ attribute ที่ไม่ต้องการนำมาทดสอบ
5. Back Button จะแสดงหน้าจอ Select Target attribute
6. Next Button สามารถใช้งานได้เมื่อเลือก attribute อย่างน้อย 1 attribute แสดงหน้าจอ Classification Parameter



รูปที่ 3.5 หน้าจอ Select Test attribute

3.2.6 หน้าจอ Classification Parameter

เป็นหน้าจอสำหรับเลือกตั้งค่าที่จำเป็นในการสร้าง Model tree

1. Training data Text Box กำหนดปริมาณข้อมูลที่นำมาใช้ทดสอบ Training data ที่นำมาใช้ในการสร้างแบบจำลองต้นไม้โดยอยู่ระหว่าง 60-100%

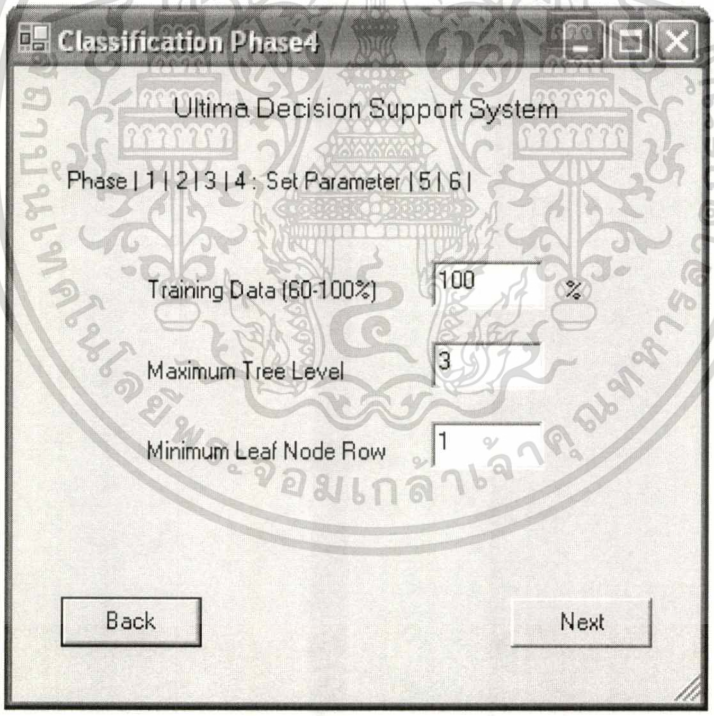
2. Maximum Tree Level Text Box กำหนดจำนวนระดับชั้นมากที่สุดของแบบจำลองต้นไม้เมื่อโปรแกรมแบ่งระดับชั้นออกจนเท่ากับที่กำหนดไว้โปรแกรมจะหยุดการจำแนกประเภท

3. Minimum Leaf node row Text Box กำหนดปริมาณต่ำสุดของแถวข้อมูลในระดับชั้นล่างสุดของแบบจำลองต้นไม้เมื่อจำแนกข้อมูลจนแถวข้อมูลเหลือเท่ากับจำนวนที่กำหนดไว้โปรแกรมจะหยุดการจำแนกประเภท

4. Back Button จะแสดงหน้าจอ Select Test attribute

5. Next Button สามารถใช้ได้เมื่อกำหนดค่าต่างๆแล้วแสดงหน้าจอ Select Preview

6. มีการแสดง error เมื่อผู้ใช้กรอกข้อมูลผิดพลาด



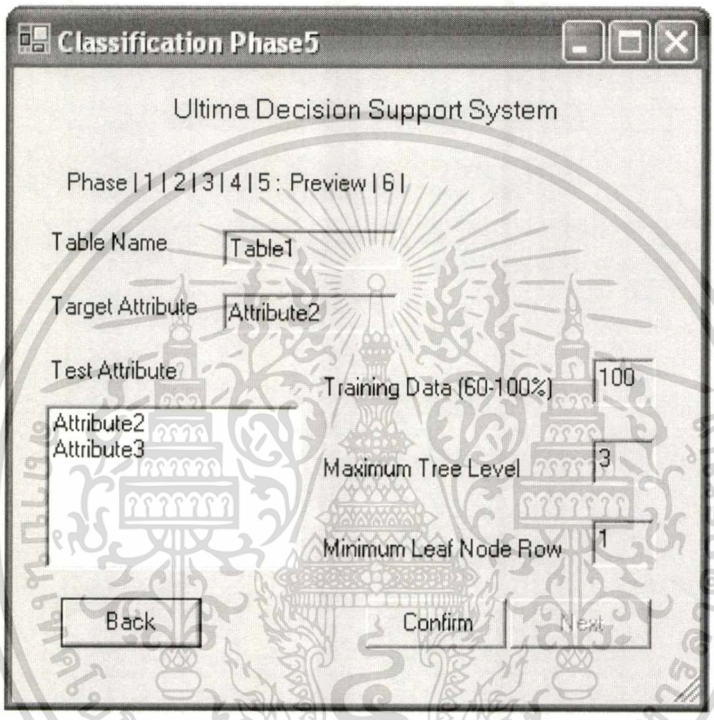
รูปที่ 3.6 หน้าจอ Classification Parameter

3.2.7 หน้าจอ Preview

เป็นหน้าจอสำหรับแสดงข้อมูลต่างๆที่เลือกไว้ในขั้นตอนที่ผ่านมาเพื่อป้องกันข้อผิดพลาดเนื่องจากการวิเคราะห์แบบจำลองต้นไม้ต้องใช้เวลานานพอสมควร

1. Back Button จะแสดงหน้าจอ Classification Parameter

2. Classify Button โปรแกรมจะเริ่มสร้างแบบจำลองต้นไม้และจำแนกประเภทจากนั้นแสดงหน้าจอ Result

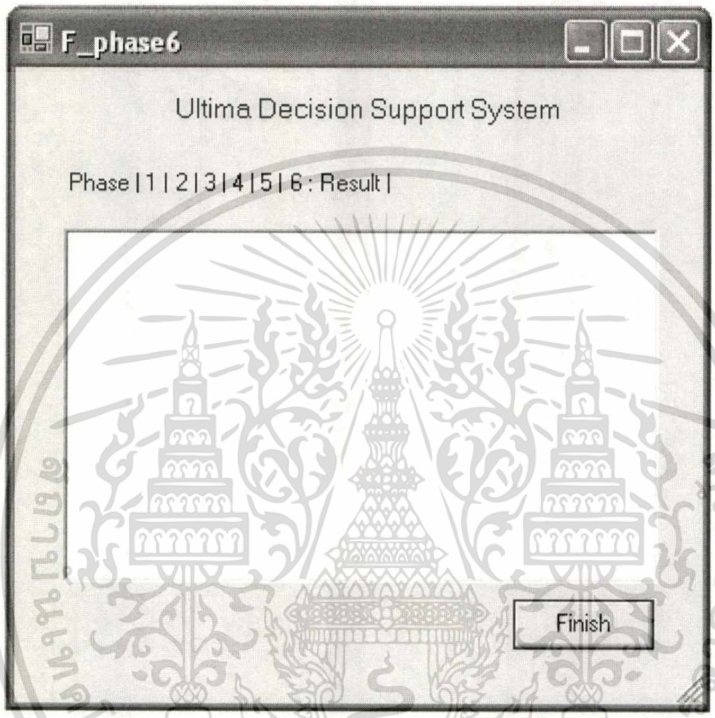


รูปที่ 3.7 หน้าจอ Preview

3.2.8 หน้าจอ Classify Result

เป็นหน้าจอแสดงผลการจำแนกประเภทโดยแสดงผลเป็นโครงสร้างต้นไม้หรือ Tree view Node แรกจะเป็น root และจะแสดงการจำแนกประเภทเป็น sub node

1. End Button จะแสดงหน้าจอหลัก

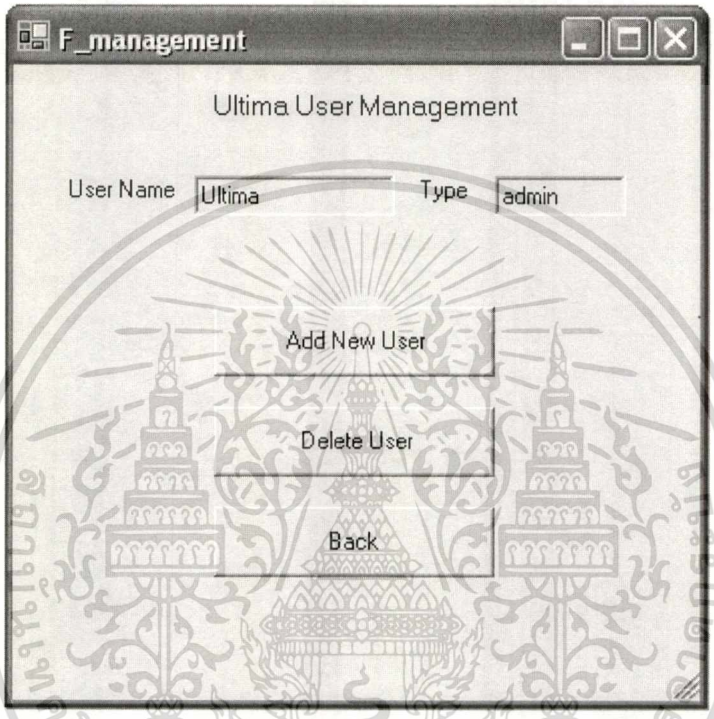


รูปที่ 3.8 หน้าจอ Result

3.2.9 หน้าจอ User Management

เป็นหน้าจอหลักในการจัดการผู้มีสิทธิในการเข้าใช้ระบบซึ่ง admin ทั้นที่มีสิทธิเข้าใช้

1. Add User Button จะแสดงหน้าจอ Add User
2. Delete User Button จะแสดงหน้าจอ Delete User
3. Back Button จะแสดงหน้าจอหลัก

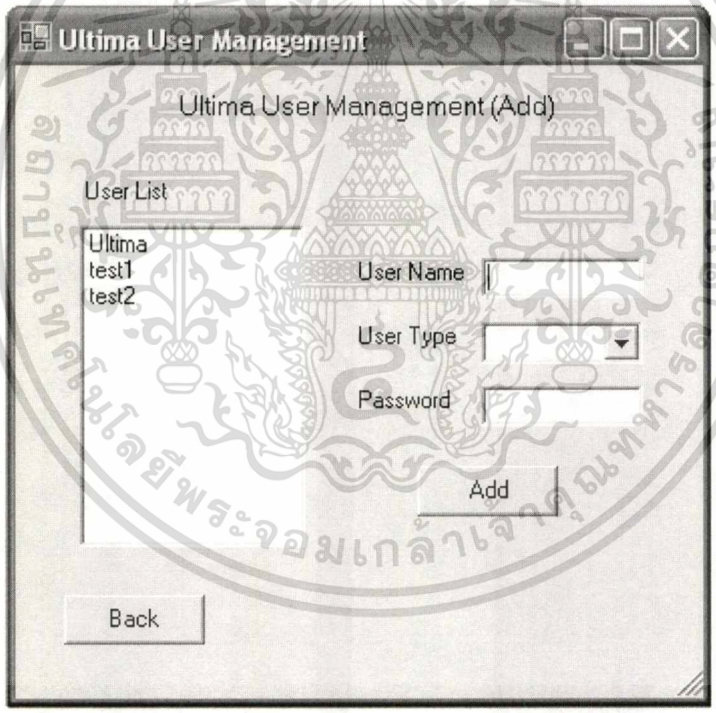


รูปที่ 3.9 หน้าจอ User Management

3.2.10 หน้าจอ Add User

เป็นหน้าจอสำหรับเพิ่มผู้มีสิทธิใช้ระบบสนับสนุนการตัดสินใจโดยมีการแสดงรายชื่อผู้มีสิทธิใช้ระบบทั้งหมด

1. User List Box จะแสดงรายชื่อผู้มีสิทธิใช้ระบบ
2. User Name Text Box สำหรับรับชื่อผู้มีสิทธิใช้ระบบ
3. User Type Combo Box สำหรับเลือกประเภทผู้มีสิทธิใช้ระบบโดยมีให้เลือก 2 ประเภท คือ admin และ manager
4. Password Text Box สำหรับรับรหัสผ่านของผู้มีสิทธิใช้ระบบ
5. Add Button สำหรับเพิ่มชื่อผู้ใช้เข้าสู่ระบบ
6. Back Button จะแสดงหน้าจอ User Management
7. มีการแสดง Error message เมื่อผู้ใช้กรอกข้อมูลไม่ครบหรือไม่ได้เลือกประเภทผู้ใช้

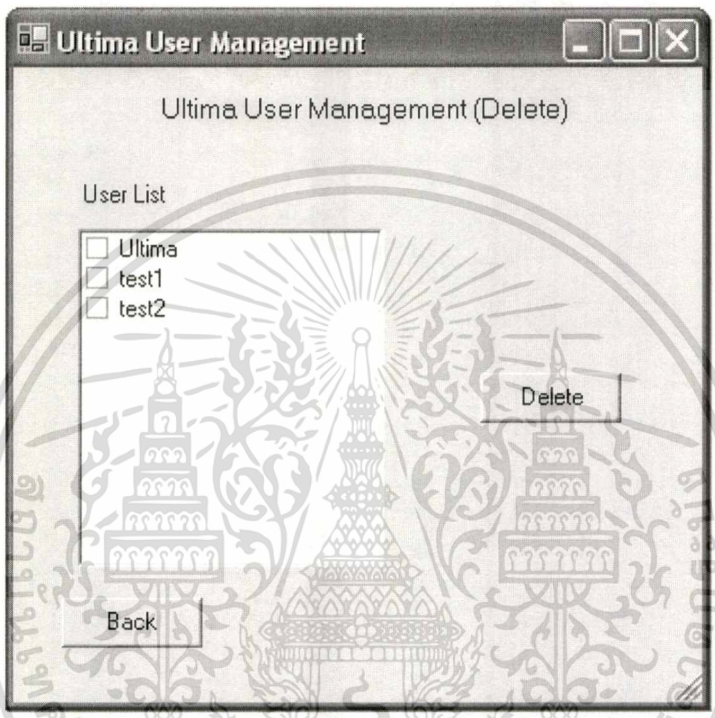


รูปที่ 3.10 หน้าจอ Add User

3.2.11 หน้าจอ Delete User

เป็นหน้าจอสำหรับลบผู้มีสิทธิใช้ระบบสนับสนุนการตัดสินใจ

1. User Check Box + List Box จะแสดงรายชื่อผู้มีสิทธิใช้ระบบและประเภทผู้ใช้
2. Delete Button สำหรับลบรายชื่อผู้ใช้ระบบที่เลือกด้วย check box
3. Back Button จะแสดงหน้าจอ User Management

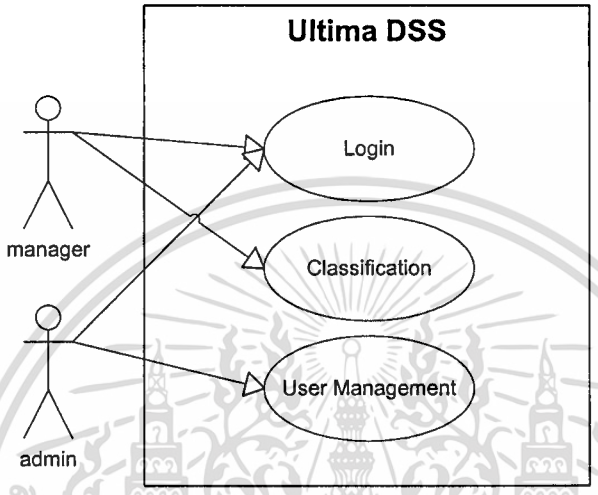


รูปที่ 3.11 หน้าจอ Delete User

3.3 การวิเคราะห์ออกแบบโปรแกรม

3.3.1 Use Case diagram

ระบบมีผู้ใช้งาน 2 ประเภทคือ Manager ใช้งานในส่วน Classification และ Admin ดูแลสิทธิการเข้าใช้ระบบ จึงมีกิจกรรมที่สามารถทำได้ทั้งหมด 3 อย่างคือ ตรวจสอบผู้มีสิทธิใช้ การใช้งานจำแนกประเภทข้อมูล และการจัดการสิทธิผู้ใช้



รูปที่ 3.12 Use Case Diagram

3.3.2 Class diagram

จากกิจกรรมทั้ง 3 อย่างใน Use case diagram ประกอบไปด้วย Class ต่างๆดังนี้

3.3.2.1 Class Login

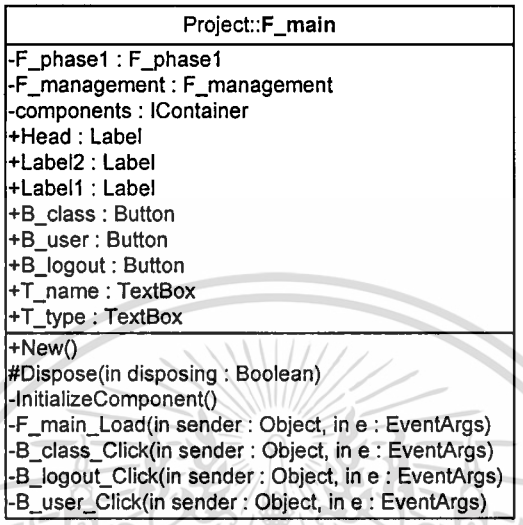
สำหรับตรวจสอบสิทธิการเข้าใช้งานระบบ

Project::F_login
-F_main : F_main +username : String +password : String +usertype : String -components : IContainer +Head : Label +Label1 : Label +Label2 : Label +T_username : TextBox +T_password : TextBox +B_ok : Button +B_clear : Button +show_error : Label
+New() #Dispose(in disposing : Boolean) -InitializeComponent() -B_clear_Click(in sender : Object, in e : EventArgs) -B_ok_Click(in sender : Object, in e : EventArgs)

รูปที่ 3.13 Class login

3.3.2.2 Class main

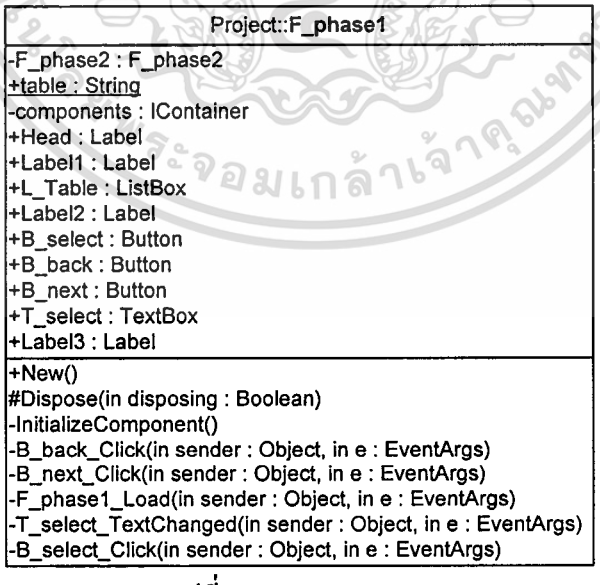
แสดงรายการที่ผู้ใช้แต่ละประเภทมีสิทธิเข้าใช้โดย manager มีสิทธิใช้งาน classification และ admin มีสิทธิเข้าใช้ user management ในการจัดการสิทธิการเข้าใช้ระบบ



รูปที่ 3.14 Class main

3.3.2.3 Class phase1

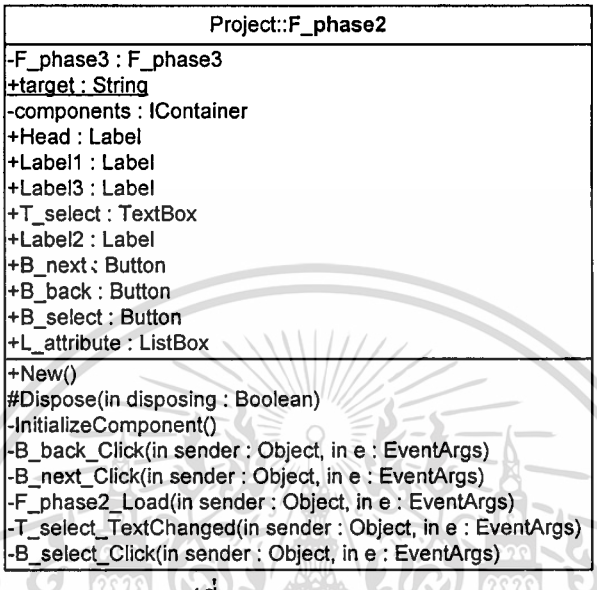
แสดงรายชื่อตารางที่สามารถนำมาจำแนกประเภทได้



รูปที่ 3.15 Class phase1

3.3.2.4 Class phase2

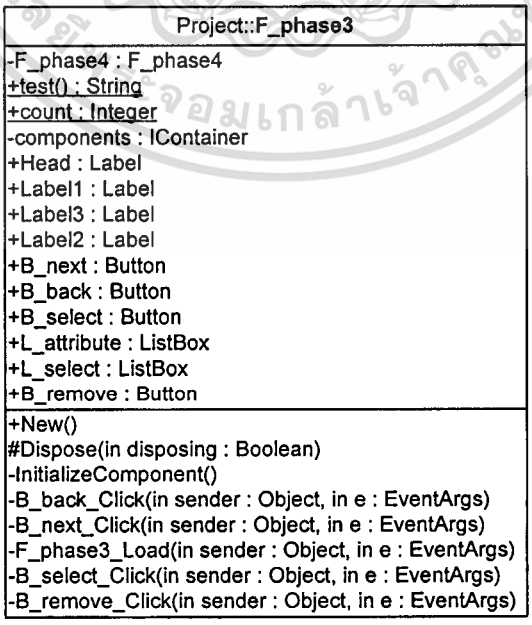
แสดงรายชื่อ attribute ที่สามารถเลือกเป็นเป้าหมาย (target attribute) สำหรับการจำแนกประเภท



รูปที่ 3.16 Class phase2

3.3.2.5 Class phase3

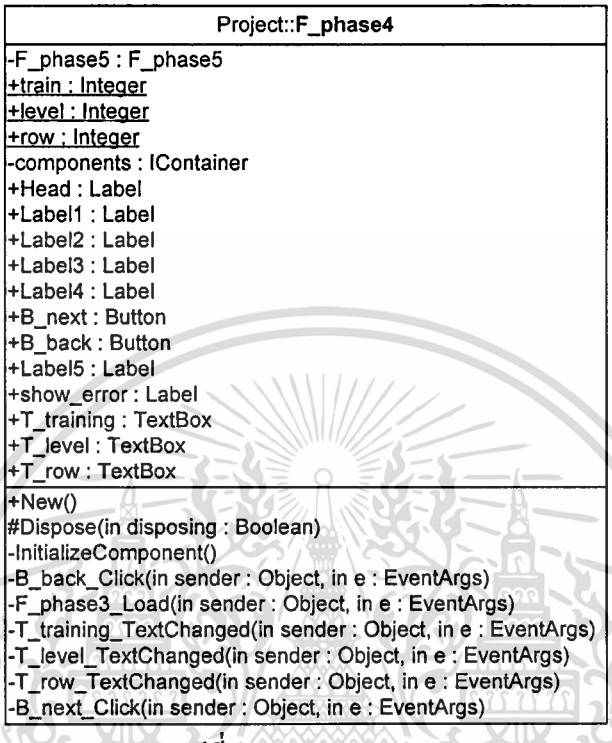
แสดงรายชื่อ attribute ที่สามารถเลือกมาทดสอบ (test attribute) ในการจำแนกประเภท



รูปที่ 3.17 Class phase3

3.3.2.6 Class phase4

แสดง parameter ที่สามารถกำหนดในการจำแนกประเภท



รูปที่ 3.18 Class phase4

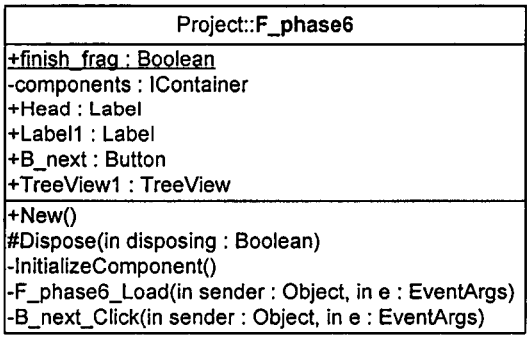
3.3.2.7 Class phase5

แสดง parameter ทั้งหมดที่กำหนดสำหรับการจำแนกประเภท

รูปที่ 3.19 Class phase5

3.3.2.8 Class phase6

แสดงผมการจำแนกประเภทโดยแสดงเป็น tree view



รูปที่ 3.20 Class phase6

3.3.2.9 Class management

แสดงรายการที่ admin สามารถใช้ในการจัดการสิทธิ

Project::F_management
-F_add : F_add -F_delete : F_delete -components : IContainer +Head : Label +T_type : TextBox +T_name : TextBox +Label2 : Label +Label1 : Label +B_back : Button +B_delete : Button +B_add : Button
+New() #Dispose(in disposing : Boolean) -InitializeComponent() -F_management_Load(in sender : Object, in e : EventArgs) -B_back_Click(in sender : Object, in e : EventArgs) -B_add_Click(in sender : Object, in e : EventArgs) -B_delete_Click(in sender : Object, in e : EventArgs)

รูปที่ 3.21 Class management

3.3.2.10 Class add

แสดงรายชื่อผู้มีสิทธิใช้ระบบและบันทึกชื่อผู้มีสิทธิใหม่

Project::F_add
-components : IContainer +Head : Label +Label1 : Label +Label2 : Label +Label3 : Label +T_name : TextBox +T_password : TextBox +B_back : Button +L_user : ListBox +Label4 : Label +B_add : Button +C_type : ComboBox +show_error : Label
+New() #Dispose(in disposing : Boolean) -InitializeComponent() -B_back_Click(in sender : Object, in e : EventArgs) -B_add_Click(in sender : Object, in e : EventArgs) -F_add_Load(in sender : Object, in e : EventArgs)

รูปที่ 3.22 Class add

3.3.2.11 Class delete

แสดงรายชื่อผู้มีสิทธิใช้ระบบและลบรายชื่อผู้มีสิทธิที่เลือกใน Check box

Project::F_delete
-components : IContainer +Head : Label +Label4 : Label +CL_user : CheckedListBox +B_back : Button +B_delete : Button
+New() #Dispose(in disposing : Boolean) -InitializeComponent() -B_back_Click(in sender : Object, in e : EventArgs) -B_delete_Click(in sender : Object, in e : EventArgs) -F_delete_Load(in sender : Object, in e : EventArgs)

รูปที่ 3.23 Class delete

3.3.2.12 Class DBmanage

สำหรับติดต่อฐานข้อมูลสำหรับคำสั่ง sql ต่างๆ

Project::DBmanage
+New() #Dispose(in disposing : Boolean) -InitializeComponent() +Connect(in sender : Object, in e : EventArgs) +Disconnect(in sender : Object, in e : EventArgs) +Sql(in sender : Object, in e : EventArgs) : String

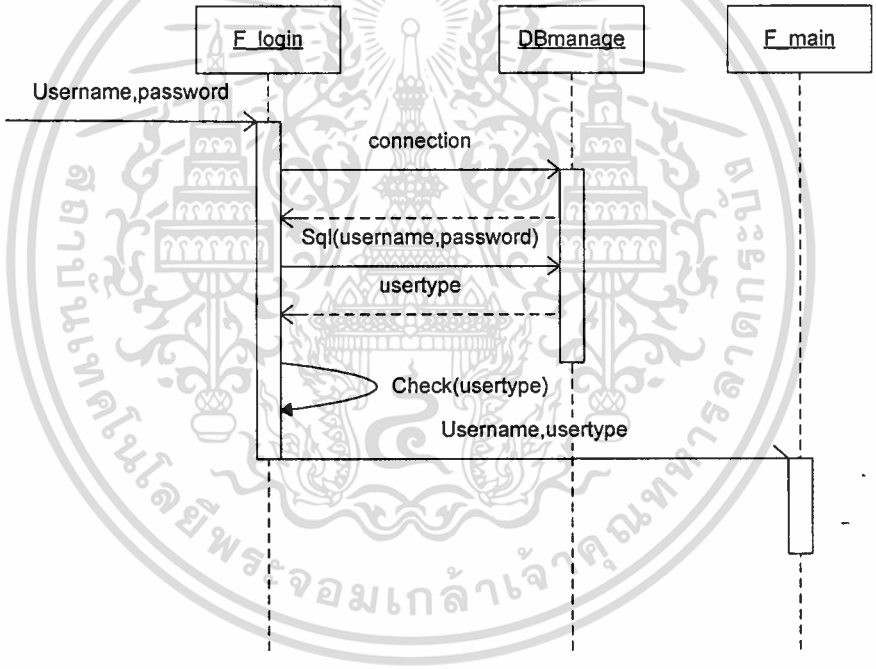
รูปที่ 3.24 Class DBmanage

3.3.3 Sequence diagram

แสดงการทำงานของกิจกรรมต่างตาม Usecase diagram แบ่งออกเป็น Sequence ต่างๆ ดังนี้

3.3.3.1 Sequence Login

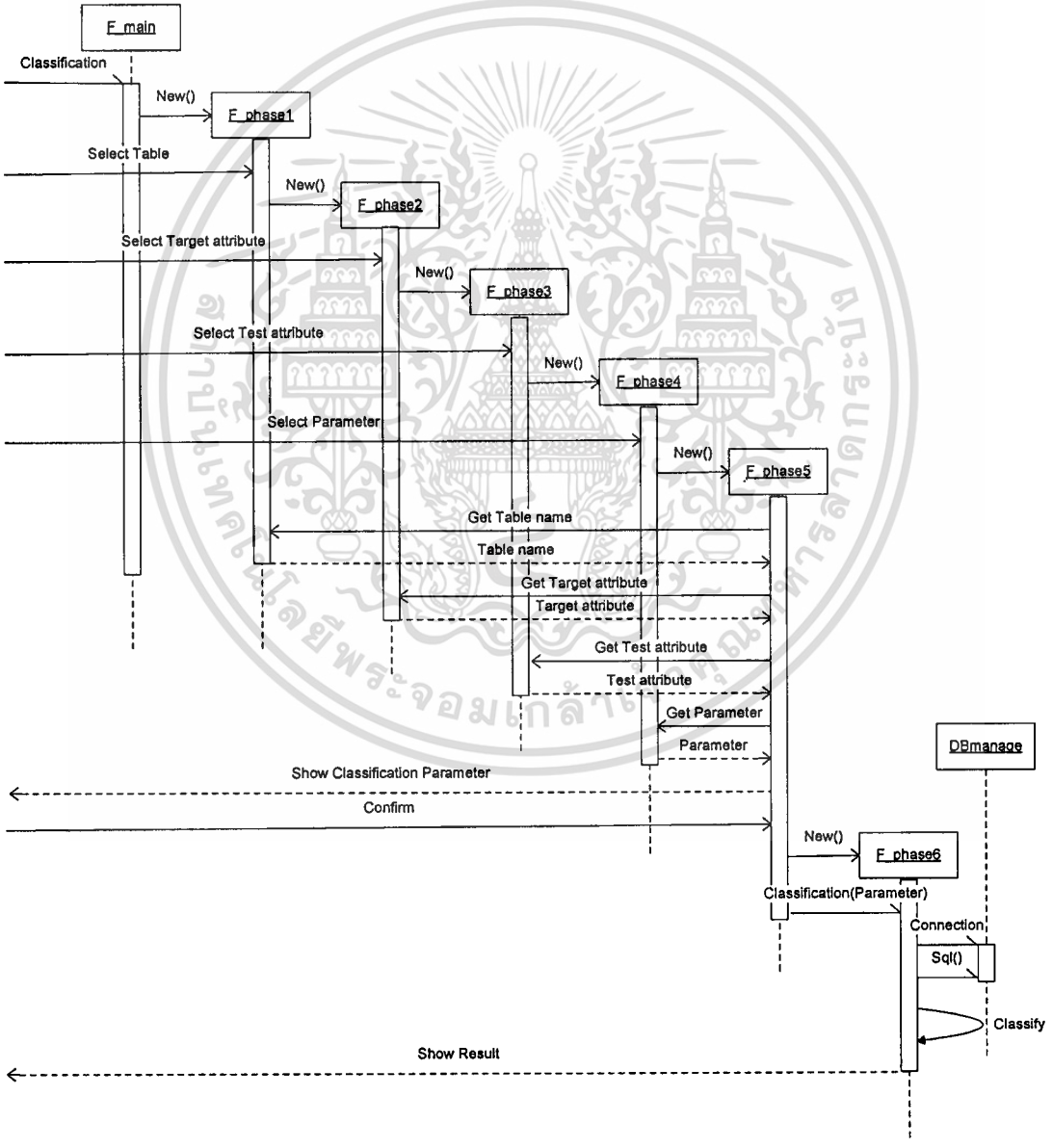
เมื่อผู้มิทธิใช้ระบบติดต่อมายัง object F_login โดยกรอก user name และ password F_login จะติดต่อกับ DBmanage โดยเรียก connection() และ select user type และส่งให้ F_login จากนั้น F_login จะตรวจสอบ user type หากมีค่าจะส่ง user name และ user type ไปยัง object F_main เพื่อแสดงรายการตามประเภทที่ผู้ใช้มีสิทธิต่อไป แต่หาก user type ไม่มีค่าแสดงว่าไม่มีผู้ใช้ในระบบจะแสดง error



รูปที่ 3.25 Sequence Login

3.3.3.2 Sequence Classification

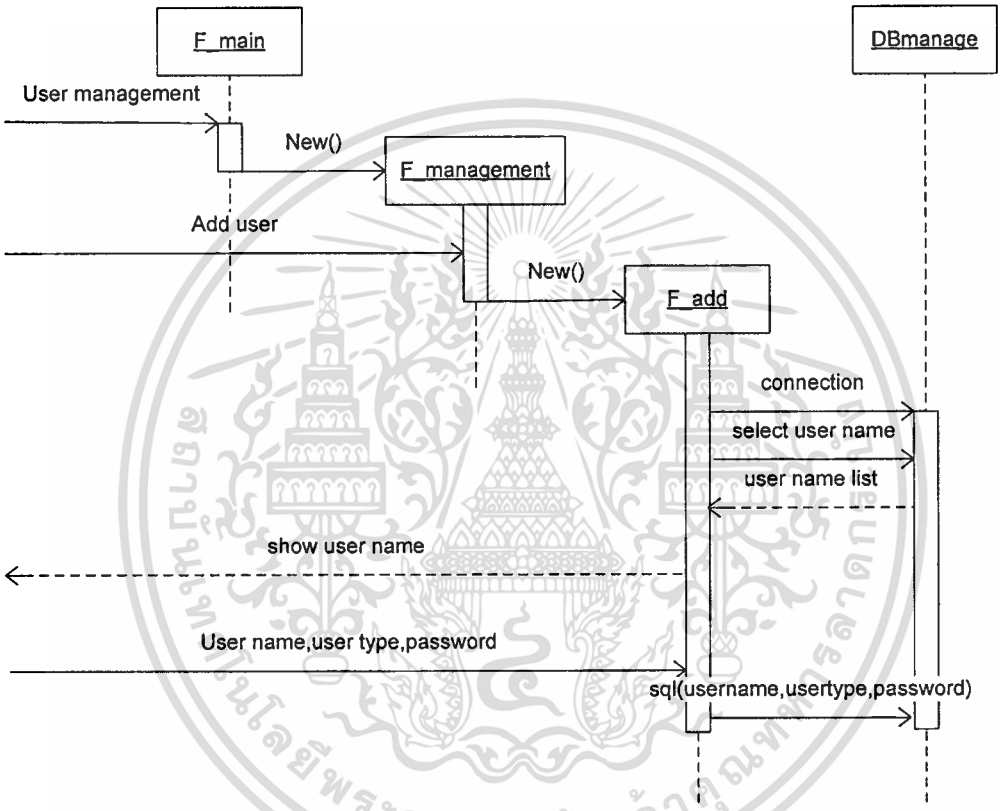
เมื่อผู้ใช้ประเภท manager เลือก classification object F_main จะสร้าง object F_phase1 ขึ้นมาแสดงรายชื่อตารางที่สามารถนำมาจำแนกประเภทได้เมื่อเลือกตารางแล้ว object F_phase1 จะสร้าง object F_phase2 ขึ้นแสดงรายชื่อ attribute ที่สามารถเลือกเป็นเป้าหมายได้เมื่อเลือก attribute เป้าหมายแล้ว object F_phase2 จะสร้าง F_phase3 เพื่อรับ parameter ต่างๆไปเรื่องๆจนครบ object F_phase5 จะแสดงรายละเอียดของ parameter ต่างๆที่เลือกมาทั้งหมดให้ยืนยันจากนั้นจะสร้าง object F_phase6 ขึ้นและส่ง parameter ทั้งหมดไป F_phase6 จะติดต่อกับ DBmanage เพื่อดึงข้อมูลมาทำการทดสอบการจำแนกประเภทและวแสดงผลการทดสอบเป็น tree view



รูปที่ 3.26 Sequence Classification

3.3.3.3 Sequence Add

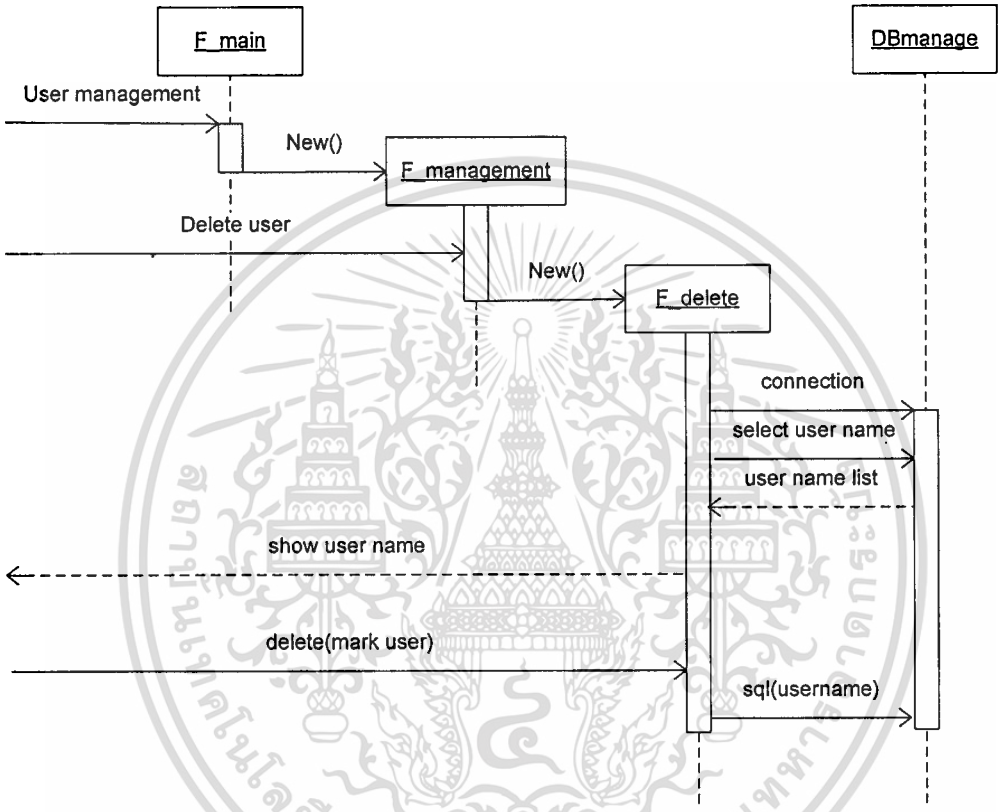
เมื่อผู้ใช้ประเภท admin เลือก user management ที่ object F_main จะสร้าง object F_management ขึ้นเพื่อแสดงหน้าจอรายการที่ admin สามารถจัดการได้เมื่อเลือก add user จะสร้าง object F_add ขึ้นมีการติดต่อ DBmanage เพื่อดึงรายชื่อผู้มีสิทธิใช้ขึ้นมาแสดง โดยผู้ใช้สามารถกรอก username usertype และ password และกด add เพื่อเพิ่มรายชื่อผู้ใช้ใหม่เข้าสู่ระบบได้



รูปที่ 3.27 Sequence Add

3.3.3.4 Sequence Delete

เมื่อผู้ใช้ประเภท admin เลือก user management ที่ object F_main จะสร้าง object F_management ขึ้นเพื่อแสดงหน้าจอรายการที่ admin สามารถจัดการได้เมื่อเลือก delete user จะสร้าง object F_delete ขึ้นมีการติดต่อ DBmanage เพื่อดึงรายชื่อผู้มีสิทธิใช้ขึ้นมาแสดง โดยผู้ใช้สามารถเลือก checkbox หน้ารายชื่อผู้ใช้และกด delete เพื่อลบรายชื่อผู้ใช้ที่เลือกได้



รูปที่ 3.28 Sequence Delete

บทที่ 4
การประยุกต์ใช้งานระบบสนับสนุนการตัดสินใจ ด้วยเทคนิคต้นไม้ตัดสินใจ

4.1 ภาพรวมฐานข้อมูล

ลักษณะของฐานข้อมูล กรณีศึกษาเป็นระบบฐานข้อมูลของธนาคารในเยอร์มันซึ่งเป็นฐานข้อมูลตัวอย่างสำหรับทำ Classification ซึ่งนำมาศึกษาการพัฒนาแอพพลิเคชั่นโดยใช้เทคนิค Decision tree ในการจำแนกประเภทข้อมูล (classification) มาพัฒนาเป็นแอพพลิเคชั่นช่วยจำแนกกลุ่มลูกค้าเพื่อสนับสนุนการตัดสินใจ จากการศึกษาข้อมูลที่เก็บไว้ในฐานข้อมูล ซึ่งประกอบไปด้วยข้อมูลต่างๆ ดังนี้ สถานะของบัญชี, ระยะเวลาการกู้, ประวัติ credits, จุดประสงค์, จำนวน Credit, บัญชีเงินฝาก, ระยะเวลาการทำงาน, เปอร์เซนต์ที่จะไม่มีรายได้, เพศและสถานะภาพสมรส, หนี้สิน/มีผู้รับรอง, อยู่อาศัยมาตั้งแต่, ทรัพย์สิน, อายุ, แผนการจ่ายเงิน, บ้าน, credits, อาชีพ , จำนวนคนที่มีแนวโน้มจะช่วยรับภาระหนี้สินได้, โทรศัพท์, ทำงานในต่างประเทศหรือไม่ และประเภท Credit จากการศึกษาพบว่าข้อมูลต่างๆที่เก็บอยู่ในฐานข้อมูลมีปริมาณมากเหมาะสมในการนำมาพัฒนาโดยใช้อัลกอลิทึ่ม SPRINT จึงนำมาใช้ในกรณีศึกษาสำหรับการพัฒนาแอพพลิเคชั่นทาง Data Mining โดยใช้เทคนิค Classification

4.2 การประยุกย์ใช้ระบบกับฐานข้อมูล

การประยุกต์ระบบเข้ากับฐานข้อมูลทำตามขั้นตอนทาง data mining ดังต่อไปนี้

4.2.1 การกำหนดปัญหาหรือวัตถุประสงค์

จากความสำคัญของสาระสนเทศที่มีต่อการดำเนินงานประจำวันและในการตัดสินใจเพื่อวางแผนการทำงานหรือนโยบายเพื่อกำหนดเป้าหมายของบริษัท การใช้งานข้อมูลที่มีอยู่ให้คุ้มค่าจึงมีความสำคัญ และเทคนิค Classification ซึ่งเป็นวิธีการจำแนกประเภทข้อมูลโดยการแบ่งข้อมูลออกเป็นกลุ่มข้อมูลตามลักษณะของข้อมูลซึ่งทำให้ได้สาระสนเทศเพิ่มขึ้นจากข้อมูลเดิมที่มีอยู่จึงเป็นวิธีการหนึ่งที่ทำให้สามารถใช้ข้อมูลที่มีอยู่ให้คุ้มค่าทั้งยังได้ข้อมูลที่ช่วยสนับสนุนการตัดสินใจ กระบวนการทำ Classification ด้วยเทคนิค Decision tree มีอยู่ด้วยกัน 2 เทคนิคคือ SLIQ และ SPRINT ซึ่งในกรณีศึกษานี้ได้เลือกเทคนิค SPRINT (Scalable RaRallelizable Induction of decision Trees) มาพัฒนาเพราะเป็นเทคนิคที่ช่วยให้สามารถแบ่งกลุ่มข้อมูลที่มีขนาดใหญ่ได้ซึ่งเหมาะกับการนำไปใช้ในธุรกิจที่มีข้อมูลจำนวนมาก

4.2.2 การเตรียมข้อมูลที่นำมาใช้ในการพัฒนาแอพพลิเคชั่น

ในกรณีศึกษาการพัฒนาระบบสนับสนุนการตัดสินใจนี้ศึกษาและเตรียมข้อมูลเพื่อใช้ในกรณีศึกษานี้โดยมีขั้นตอนการเตรียมการดังนี้

1. คัดเลือกหลักของข้อมูลในตารางโดยคัดข้อมูลหลักที่ไม่มีข้อมูลออก

2. แปลงชนิดของข้อมูลเพื่อให้เหมาะสมในการประมวลผลด้วยอัลกอลิทึ่ม SPRINT เช่น แปลงชนิดข้อมูลเป็น qualitative และ numerical

ตารางที่ 4.1 ตารางฐานข้อมูล german

	ชื่อหลัก	ประเภทข้อมูล	รายละเอียดของหลักข้อมูล	รายละเอียดข้อมูลในหลัก
1	Status	qualitative	ตรวจสอบสถานะของบัญชี	A11 : ต่ำกว่า 0 DM A12 : 0 - 200 DM A13 : มากกว่า 200 DM / มีการโอนภายใน 1 ปี A14 : ไม่ได้ตรวจสอบ
2	Duration	numerical	ระยะเวลา(เดือน)	
3	history	qualitative	ประวัติ credits	A30 : credits ทั้งหมดจ่ายในเวลา A31 : credits ทั้งหมดในธนาคารนี้จ่ายในเวลา A32 : existing credits paid back duly till now A33 : มีการจ่ายล่าช้าในอดีต A34 : critical account

ตารางที่ 4.1 (ต่อ)

4	Purpose	qualitative	จุดประสงค์	A40 : ซื้อรถ A41 : ซ่อมรถ A42 : เครื่องใช้ในบ้าน A43 : เครื่องใช้ฟุ่มเฟือย A44 : ใช้จ่ายภายในบ้าน A45 : ซ่อมแซม A46 : เพื่อการศึกษา A47 : เพื่อจุดประสงค์ส่วนตัว A48 : ฝึกอบรมใหม่ A49 : ธุรกิจ A410 : อื่นๆ
5	amount	numerical	จำนวน Credit	
6	saving	qualitative	บัญชีเงินฝาก	A61 : ต่ำกว่า 100 DM A62 : 100 - 500DM A63 : 500 - 1000 DM A64 : มากกว่า 1000 DM A65 : ไม่มีบัญชีเงินฝาก
7	employment	qualitative	ระยะเวลาการทำงาน	A71 : ว่างงาน A72 : ต่ำกว่า 1 year A73 : 1 - 4 years A74 : 4 - 7 years A75 : มากกว่า 7 years
8	dis_income	numerical	เปอร์เซนต์ที่จะไม่มีรายได้	
9	sex_status	qualitative	เพศและสถานะภาพสมรส	A91 : ชาย :หย่า/แยกกันอยู่ A92 : หญิง : หย่า/แยกกันอยู่/แต่งงาน A93 : ชาย :โสด A94 : ชาย :แต่งงาน/ม่าย A95 : หญิง:โสด

ตารางที่ 4.1 (ต่อ)

10	guarantor	qualitative	หนี้สิน/มีผู้รับรอง	A101 : ไม่มี A102 : มีหนี้สิน/มีผู้รับรอง A103 : มีผู้รับรอง
11	residence	numerical	อยู่อาศัยมาตั้งแต่	
12	property	qualitative	ทรัพย์สิน	A121 : มีกรรมสิทธิ์จริง A122 : ไม่มีกรรมสิทธิ์แต่มีสัญญาติดต่อทางธุรกิจ/ประกันชีวิต A123 : ไม่ใช่ทั้ง A121/A122 แต่มีทรัพย์สินอื่นๆ เช่น รถ A124 : ไม่ทราบหรือไม่มีทรัพย์สิน
13	age	numerical	อายุ (ปี)	
14	plan	qualitative	แผนการจ่ายเงิน	A141 : ผ่านธนาคาร A142 : จ่ายเอง A143 : ไม่ทราบ
15	housing	qualitative	บ้าน	A151 : เช่า A152 : เป็นเจ้าของ A153 : อยู่กับบุคคลอื่น
16	exist_credit	numerical	credits ที่มีในธนาคารนี้	
17	job	qualitative	อาชีพ	A171 : ว่างงานไม่ใช่พลเมือง A172 : ว่างงานเป็นพลเมือง A173 : ทำงานบริษัท A174 : บริหาร/ธุรกิจส่วนตัว ผู้บริหาร/พนักงาน

ตารางที่ 4.1 (ต่อ)

18	liable	numerical	จำนวนคนที่มีแนวโน้มจะช่วยรับภาระหนี้สินได้	
19	Telephone	qualitative	โทรศัพท์	A191 : ไม่มี A192 : มีลงทะเบียนไว้
20	foreign_worker	qualitative	ทำงานในต่างประเทศ	A201 : yes A202 : no
21	credit_status	qualitative	ประเภท Credit	1 = good 2 = bad

4.2.3 การสร้างโมเดล

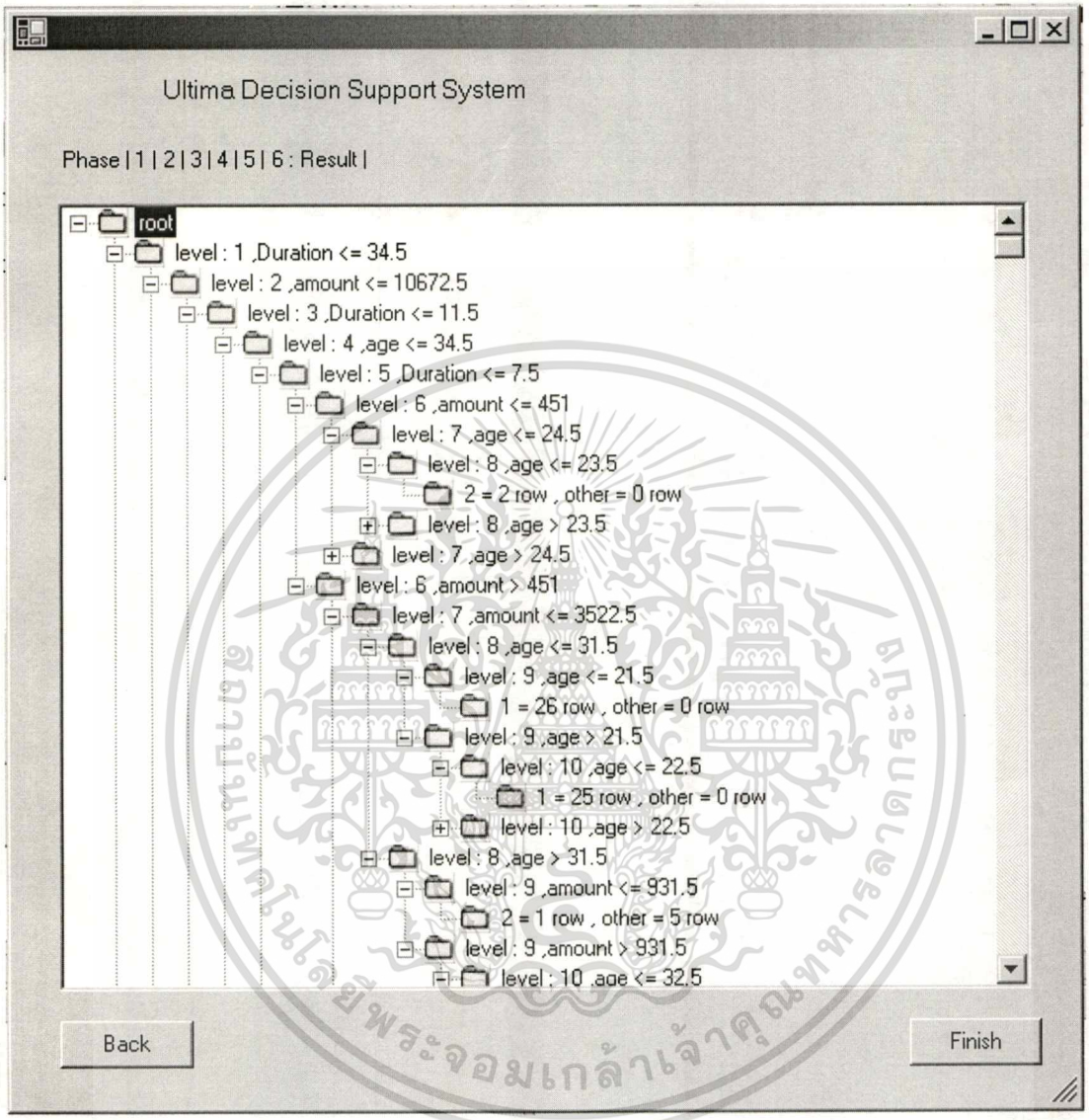
สำหรับกรณีศึกษานี้เป็นการพัฒนาระบบสนับสนุนการตัดสินใจของบริษัทอัลติม่าด้วยวิธีการจำแนกประเภทข้อมูลโดยใช้เทคนิค Decision tree เนื่องจากข้อมูลมีปริมาณมากจึงเลือกใช้อัลกอลิทึ่ม SPRINT ซึ่งมีลักษณะการทำงาน โดยการโหลดข้อมูลขึ้นมาทีละส่วนสร้างเป็น Training Data และทำการทดสอบโดยใช้ Gini Index ในการหาจุดแบ่งที่เหมาะสมการดังข้อมูลขึ้นมาทีละส่วนนี้ทำให้อัลกอลิทึ่มนี้เหมาะสมในการนำมาพัฒนาดังที่ได้กล่าวไว้ในบทที่ 2 เพราะสามารถใช้ได้กับข้อมูลที่มีปริมาณมาก

การทำงานของโปรแกรมในแต่ละรอบจะสร้าง attribute list ขึ้นมาสำหรับแต่ละ attribute ข้อมูลที่เลือกมาทดสอบแล้วตรวจสอบประเภทข้อมูลเพื่อสร้าง histogram ขึ้นตามประเภทข้อมูลนั้นแล้วคำนวน Gini index และ Gini split เพื่อหาจุดแบ่ง และเก็บค่า Gini split ที่ต่ำที่สุดในรอบนั้นพร้อมกับข้อมูลที่จำเป็นในการสร้าง Tree ไว้ เมื่อได้ค่า Gini split ที่น้อยที่สุดในรอบการทำงานนั้นก็จะสร้าง tree จากข้อมูลที่เก็บไว้แล้วตรวจสอบพารามิเตอร์ต่างๆที่กำหนดไว้สำหรับหยุดการทำงานแล้ววนทำการแตก tree ไปเรื่อยๆจนกว่าจะสำเร็จตามเงื่อนไขต่างๆที่กำหนด

4.2.4 การนำไปใช้

หลังจากพัฒนาโปรแกรมโดยใช้อัลกอลิทึ่ม SPRINT เชื่อมต่อกับฐานข้อมูล oracle 10g จากนั้นสร้างฐานข้อมูล ตาราง german ขนาด 20 หลัก(column) และนำข้อมูลจำนวน 1000 แถว (record) มาในการทดสอบการทำงานโดยเพิ่ม filed เข้าไปอีก 1 column ซึ่งใช้ในการระบุ Primary key แทนการใช้ Primary key ร่วมของแถวข้อมูลซึ่งจำเป็นในการสร้าง Decision Tree

จากการทดสอบสามารถแบ่งกลุ่มออกได้เป็น ตามลักษณะของ attribute ที่เลือกมาทดสอบและได้ผลลัพธ์ดังรูป



รูปที่ 4.1 tree ผลลัพธ์

จากการทดสอบได้ผลลัพธ์แสดงเป็น tree สามารถสรุปผลได้ยกตัวอย่าง เช่น

1. กลุ่มที่มีอายุน้อยกว่า 23.5 credit น้อยกว่า 450 และ ระยะเวลากู้น้อยกว่า 7.5 เดือนอยู่ในกลุ่ม Bad
2. กลุ่มที่มีอายุน้อยกว่า 21.5 credit อยู่ระหว่าง 450 ถึง 3522.5 และ ระยะเวลากู้น้อยกว่า 7.5 เดือนอยู่ในกลุ่ม Good
3. กลุ่มที่มีอายุระหว่าง 21.5 ถึง 22.5 credit อยู่ระหว่าง 450 ถึง 3522.5 และ ระยะเวลากู้น้อยกว่า 7.5 เดือนอยู่ในกลุ่ม Good
4. ฯลฯ

บทที่ 5
สรุปการพัฒนาและข้อเสนอแนะ

5.1 ผลการวิจัยและพัฒนา

ในการศึกษาการพัฒนาระบบสันบสนุนการตัดสินใจด้วยเทคนิคต้นไม้ตัดสินใจโดยใช้ฐานข้อมูลของบริษัทอัลติม่าเป็นกรณีศึกษาสรุปได้ดังต่อไปนี้

5.1.1 การศึกษารวบรวมข้อมูล

การศึกษาและรวบรวมข้อมูลเพื่อใช้ในการศึกษา ได้มีการศึกษาสอบถามรายละเอียดข้อมูลของบริษัทจาก ผู้บริหาร พนักงาน และผู้ดูแลระบบฐานข้อมูล ได้รับคำแนะนำจากอาจารย์ที่ปรึกษา โดยจุดมุ่งหมายของระบบคือการจำแนกประเภทลูกค้าของบริษัทเป็นกรณีศึกษา ประกอบกับการค้นคว้าข้อมูลจากห้องสมุดของสถาบันและจากเว็บไซต์ต่างๆทำให้รู้จักอัลกอลิทึ่มของ SPRINT และนำมาประยุยต์ใช้ในการพัฒนาระบบนี้

5.1.2 การวิเคราะห์และออกแบบระบบงาน

วิเคราะห์และออกแบบระบบงานโดยออกแบบ Interface ให้มีการตรวจสอบและจัดการสิทธิการเข้าใช้เพื่อความปลอดภัยของข้อมูล มีการรับ parameter ต่างๆที่จำเป็นในการทำงานของอัลกอลิทึ่ม SPRINT และแสดงผลแบบ tree view ออกแบบการทำงานของระบบด้วย UML diagram โดยใช้ Uses case diagram แสดงกิจกรรมที่ระบบมี class diagram แสดง object ต่างๆในระบบ และ sequence diagram แสดงขั้นตอนการทำงานของระบบ

5.1.3 การวิเคราะห์และการเตรียมข้อมูล

จากการศึกษารายละเอียดของข้อมูลในฐานข้อมูลของต้องมีการเตรียมข้อมูลเพื่อให้มีความเหมาะสมในการนำมาพัฒนาดังนี้

1. แปลงข้อมูลในอยู่ในรูปแบบที่เหมาะสมในการนำมาวิเคราะห์ เช่น String
2. ศึกษา attribute ในตารางต่างๆเพื่อทำความเข้าใจ
3. คัดเลือก attribute ที่มีข้อมูลไม่ครบถ้วนหรือไม่สามารถระบบความหมายได้ออก

5.1.4 คุณสมบัติของโปรแกรม

มีลักษณะสำคัญดังนี้

1. ทำงานบนระบบปฏิบัติการ Windows XP พัฒนาระบบด้วย Visual Basic .NET 2003 เชื่อมต่อกับระบบจัดการฐานข้อมูล Oracle 10g
2. วิเคราะห์จำแนกประเภทข้อมูลโดนใช้อัลกอลิทึ่ม SPRINT
3. แสดงผลการจำแนกแบบ tree view
4. มีการดูแลสิทธิการเข้าใช้ระบบ

5.2 สรุปประสิทธิภาพของโปรแกรม

สามารถกำหนดสิทธิการเข้าใช้งานระบบได้ สามารถจำแนกประเภทลูกค้าตามจุดมุ่งหมายได้ในระดับหนึ่งแต่ความถูกต้องอาจยังไม่มากนักเพราะไม่สามารถใช้ข้อมูลจากตารางประวัติได้ครบเนื่องจากตัวข้อมูลเองยังไม่ครบถ้วนการแสดงโครงสร้าง Tree ยังให้รายละเอียดได้ไม่มากนักเนื่องจากความไม่ชำนาญของผู้พัฒนาด้านภาษาโปรแกรม

5.3 ข้อเสนอแนะ

เนื่องจากเป็นกรณีศึกษาการทำงานกับฐานข้อมูลของบริษัทจริงทำให้พบปัญหาเกี่ยวกับโครงสร้างข้อมูลและความไม่ครบถ้วนของข้อมูลทำให้ใช้เวลาในการจัดการข้อมูลมาก ประกอบกับการศึกษาการโปรแกรมด้วย Visual Basic .NET 2003 ซึ่งเป็นเรื่องใหม่ที่ผู้พัฒนาต้องการเรียนรู้ทำให้ขาดความชำนาญในการใช้งานจึงเสนอแนะให้ปรับปรุงในส่วนต่างๆดังต่อไปนี้

1. ในส่วนของฐานข้อมูลควรปปรับปรุงโครงสร้างของการจัดเก็บให้ไม่เกิดความซ้ำซ้อนหรือมีความซ้ำซ้อนน้อยที่สุดเพื่อความสะดวกในการพัฒนาโปรแกรมบนฐานข้อมูลต่างๆ
2. ในส่วนของการรับ parameter ต่างๆอาจปรับปรุงให้แสดงรายละเอียดให้มากขึ้นเพื่อสนับสนุนการวิเคราะห์ให้ดียิ่งขึ้น
3. ในส่วนของการแสดงผลอาจปรับปรุงให้มีการแสดงโครงสร้างเป็นรูปภาพทำให้ผู้ใช้สามารถเข้าใจโครงสร้างได้ดียิ่งขึ้น
4. ในส่วนของ input อาจปรับปรุงให้มีความยืดหยุ่นมากขึ้นโดยการดึงรายละเอียดของตารางในฐานข้อมูลทุกตัวโดยอัตโนมัติไม่เฉพาะเจาะจงตารางใดตารางหนึ่ง

บรรณานุกรม

Boon Thau Loo. "SPRINT" **A Scalable Parallel Classifier for Data Mining.**

Johannes Gehrke, Raghu Ramakrishnan, Venkatesh Ganti. 1998. "RainForest" **A Framework for Fast Decision Tree Construction of Large Datasets.** Proceeding of the 24th.

John Shafer, et al. 2003. "SPRINT" **A Sclable Parallel Classifier for Data Mining.**

Nathan Rountree. 1999. **Further Data Mining: Building Decision Trees.**