

ระบบให้บริการข่าวออนไลน์แบบกระจายการทำงาน
ด้วยเทคโนโลยี .Net Remoting
Distributed Online News System with .Net Remoting

โดย

นายเสริมศักดิ์ ศรีโพธิ์ทอง

รหัส 45066079

อาจารย์ที่ปรึกษา

ผศ. ดร. จันทร์บุรณธ์ สถิตวิริยวงศ์

วัน เดือน ปี.....	03 ก.พ. 2550
เลขทะเบียน.....	02159
เลขเรียกหนังสือ.....	กษ. ๒๑๒๕๖ ๒๕๔๖
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ ๓๐๑."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2546
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



H002159

ชื่อหัวข้อ	ระบบให้บริการข่าวออนไลน์แบบกระจายการทำงานด้วยเทคโนโลยี .Net Remoting
นักศึกษา	นายเสริมศักดิ์ ศรีโพธิ์ทอง
อาจารย์ที่ปรึกษา	ผศ. ดร. จันทรบุรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

ในปัจจุบันข่าวสารข้อมูลต่างๆ ได้เข้ามามีบทบาทในชีวิตประจำวันเรามากขึ้น ซึ่งแหล่งข่าวที่มีอยู่นั้นจะมาจากแหล่งต่างๆ หลากหลาย มีทั้งทางหนังสือพิมพ์ โทรทัศน์ วิทยุ และทางเว็บไซต์ ในอินเทอร์เน็ต โดยเฉพาะทางอินเทอร์เน็ตนี้ ทุกคนสามารถเข้าไปหาอ่านได้ฟรี ซึ่งมีอยู่มากมายหลายเว็บไซต์ ด้วยเหตุนี้เองจึงทำให้เกิดแนวความคิดที่จะรวบรวมข้อมูลข่าวสารจากเว็บไซต์ต่างๆ เหล่านั้นมาไว้ด้วยกัน แล้วจัดให้บริการ เพื่อให้ผู้ใช้ได้รับความสะดวกและรวดเร็วมากยิ่งขึ้น รวมทั้งยังสามารถค้นหาข่าวสารต่างๆ ที่ผู้ใช้บริการต้องการได้อีกด้วย โดยระบบที่รวบรวมข่าวสารนี้จะถูกแบ่งการให้บริการออกเป็นหลายส่วน ในลักษณะของระบบแบบกระจาย ซึ่งเป็นรูปแบบการทำงานที่จะช่วยเพิ่มประสิทธิภาพให้กับระบบมากขึ้น เช่น การเพิ่มประสิทธิภาพในการให้บริการของแต่ละส่วนซึ่งสามารถทำได้ง่าย และถึงแม้ว่าจะมีบางส่วนที่ไม่สามารถให้บริการได้ แต่ส่วนอื่นๆ ก็ยังสามารถทำงานได้เป็นปกติ ในกรณีศึกษานี้จึงได้นำเทคโนโลยี .Net Remoting ซึ่งเป็นเทคโนโลยีใหม่เข้ามาประยุกต์ใช้ในการพัฒนาระบบข่าวออนไลน์แบบกระจายนี้

Title Distributed Online News System with .Net Remoting
Student Mr. Sermsak Sriphothong
Advisor Asst. Prof. Dr. Chanboon Sathitwiriawong
Level of study Master of Science in Information Technology
Major Information Science
Academic Year 2003

ABSTRACT

Now a day information is more and more important to our life. The source of information can be from various place such as newspapers, magazines, TV, radio and Internet ...etc. Especially from internet, everyone can enter to search for free with the available of tremendous websites. For this reason bring to the initiative of information collection from many website for service. Not only serve more convenient and faster to user, but also able to search for many of user's requirement information. This system is separated into many parts in a character of distributed system. This working format will improve the efficiency of system such as scalability, fault tolerance. In this case study will bring ".Net Remoting" which is a new technology for applying in the distributed online news system.

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
สารบัญ.....	III
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตของการศึกษา.....	2
1.4 ขั้นตอนในการดำเนินการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
2. ทฤษฎีที่เกี่ยวข้อง.....	3
2.1 บทนำ.....	3
2.2 โครงสร้างของ Client/server.....	3
2.2.1 ปัญหาของ client/server.....	4
2.3 Distributed Architecture.....	4
2.3.1 ประโยชน์ของ Distributed Architecture.....	5
2.3.2 DCOM.....	5
2.4 .NET Remoting.....	5
2.4.1 .NET Remoting Fundamentals.....	6
2.4.2 Serializable Classes.....	6
2.4.3 Remotable Classes.....	6
2.4.4 The Remote Component Host.....	8
2.4.5 สิ่งที่ต้องคำนึงในการออกแบบ .NET Remoting.....	9
2.4.6 Object lifetime.....	11
2.4.7 การติดต่อสื่อสารกับ remote component.....	11

III

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4.8 Hosting .NET Remoting Objects.....	12
2.4.9 Metadata.....	12
2.4.10 Configuration files.....	12
2.5 Distributed Design Patterns.....	13
2.6 Automated Deployment.....	14
2.7 เปรียบเทียบกับระบบอื่นๆในปัจจุบัน.....	15
3. การออกแบบและการพัฒนาระบบ.....	16
3.1 การออกแบบระบบโดยรวม.....	16
3.2 System Requirement.....	17
3.3 รายละเอียดภายในระบบ.....	18
3.4 Data Flow Diagram.....	26
4. การติดตั้งและใช้งานระบบ.....	33
4.1 Environment และการใช้งาน โปรแกรมฝั่ง server.....	33
4.1.1 User service.....	33
4.1.2 Search service.....	35
4.1.3 Story service.....	35
4.1.4 Alert service.....	36
4.1.5 Crawler.....	38
4.2 Environment และการใช้งาน โปรแกรมฝั่ง client.....	39
4.2.1 NewsOn.....	39
5. สรุปผลการพัฒนา.....	46
5.1 ผลการพัฒนาโครงการทั้งหมด.....	46
5.2 ประโยชน์ที่ได้รับ.....	47
5.3 ปัญหาที่เกิดขึ้น.....	47
5.4 ข้อดีของโครงการที่พัฒนา.....	47

สารบัญ (ต่อ)

	หน้า
5.5 ข้อเสียของโครงการที่พัฒนา.....	48
5.6 ข้อเสนอแนะ.....	48
บรรณานุกรม.....	49
ภาคผนวก.....	50
ประวัติผู้เขียน.....	59



บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันอินเทอร์เน็ตเข้ามามีบทบาทในชีวิตประจำวันมากซึ่งเป็นแหล่งข้อมูลขนาดใหญ่ให้คนเข้าไปศึกษาหาความรู้ต่างๆที่มีอยู่ทั่วโลก รวมไปถึงข่าวสารที่เดิมเคยมีอยู่แต่ในหนังสือพิมพ์ โทรทัศน์ วิทยุ ในปัจจุบันนี้ก็มีอยู่มากมายในเว็บไซต์ทำให้ผู้ใช้สามารถเข้าไปอ่าน ได้ตลอดเวลาที่ต้องการ

แต่เนื่องจากที่ข่าวสารต่างๆอยู่ในเว็บไซต์ต่างๆนี้เองทำให้ผู้ใช้อาจจะไม่ได้รับความสะดวกในการเข้าไปอ่านข่าวสารที่ตนต้องการเพราะไม่ทราบว่าต้องเข้าไปที่เว็บไซต์ไหนจึงจะมีข่าวที่ตนเองต้องการหรือถ้าหากต้องการจะดูข่าวในอดีตก็อาจจะไม่สามารถทำได้เนื่องจากเว็บไซต์ส่วนใหญ่จะไม่เก็บข่าวในอดีตหรืออาจจะเก็บแต่เก็บเพียงระยะเวลาไม่นาน ด้วยเหตุนี้เองทำให้มีแนวความคิดที่จะพัฒนาระบบที่รวบรวมข้อมูลข่าวสารจากเว็บไซต์ต่างเข้ามาเก็บไว้เพื่อที่จะให้บริการข้อมูลข่าวสารแก่ผู้ใช้งานให้ได้รับความสะดวกเร็วยิ่งขึ้น โดยระบบดังกล่าวจะมีการแบ่งการให้บริการออกเป็นส่วนต่างๆ ซึ่งเป็นลักษณะของระบบแบบกระจาย (Distributed system) เพื่อให้ระบบมีประสิทธิภาพมากขึ้น

1.2 วัตถุประสงค์ของการศึกษา

จากปัญหาดังกล่าว โครงการนี้จึงนำแนวความคิดของระบบการทำงานแบบกระจาย (Distributed System) มาใช้ในการพัฒนาระบบข่าวออนไลน์เพื่อให้ระบบสามารถรองรับการทำงานของผู้ใช้ได้เป็นจำนวนมาก และสามารถป้องกันการผิดพลาดของระบบคือถึงแม้ว่าจะมีส่วนบริการ (Service) ส่วนหนึ่งเกิดผิดปกติไม่สามารถให้บริการได้ ส่วนบริการส่วนอื่นๆก็ยังสามารถให้บริการได้อยู่ทำให้ผู้ใช้งานยังสามารถใช้งานระบบได้อยู่ โดยใช้เทคโนโลยี .Net Remoting มาช่วยในการพัฒนาระบบดังกล่าวซึ่งเทคโนโลยีนี้เป็นเทคโนโลยีใหม่ของ Microsoft ที่ใช้ในการพัฒนาระบบแบบกระจาย

1.3 ขอบเขตของการศึกษา

โครงการนี้จะนำเสนอแอปพลิเคชันของผู้ใช้งาน (Client Application) ซึ่งทำหน้าที่ในการติดต่อกับเซิร์ฟเวอร์ในการตรวจสอบสิทธิ์ของผู้ใช้และแสดงข่าวที่ผู้ใช้ต้องการที่หน้าจอ และเซิร์ฟเวอร์ แอปพลิเคชัน (Server Application) ซึ่งจะแบ่งออกเป็น 4 เซิร์ฟเวอร์ คือ

- 1 Alert Server ซึ่งจะให้บริการในการแจ้งข่าวที่เข้ามาใหม่ให้ผู้ใช้ทราบ
- 2 Search Server ซึ่งจะให้บริการในการค้นหาข่าวที่ผู้ใช้ต้องการ
- 3 Story Server ซึ่งจะให้บริการในการแสดงเนื้อหาข่าวที่ผู้ใช้ต้องการอ่าน
- 4 Authentication Server ซึ่งจะให้บริการในการตรวจสอบว่าผู้ใช้นั้นมีสิทธิ์เข้ามาใช้งานระบบหรือไม่

โดยทางฝั่งไคลเอนต์และเซิร์ฟเวอร์นี้จะติดต่อกันผ่านทางเครือข่ายเน็ตเวิร์ค โดยใช้ TCP โพรโตคอลและ เซิร์ฟเวอร์แต่ละตัวก็จะทำงานอยู่บนคนละเครื่องโดยใช้ TCP โพรโตคอลติดต่อกันสื่อสารกันผ่านเครือข่ายเน็ตเวิร์ค

1.4 ขั้นตอนในการดำเนินการศึกษา

1. กำหนดวัตถุประสงค์การศึกษา
2. ศึกษาวิธีการทำงานของ .Net Remoting และการนำมาใช้งานกับระบบที่จะพัฒนา
3. ออกแบบโครงสร้างการทำงานโดยรวมของระบบ
4. ออกแบบโครงสร้างและการทำงานของแต่ละเซิร์ฟเวอร์ (server)
5. ออกแบบฐานข้อมูลในการเก็บข้อมูลข่าวต่างๆ
6. พัฒนาระบบข่าวออนไลน์
7. ทดสอบการทำงานของระบบว่าถูกต้องเป็นไปตามที่ออกแบบไว้หรือไม่ รวมทั้งแก้ไขข้อบกพร่องต่างๆ
8. สรุปผลการศึกษา

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ผู้ใช้สามารถใช้งานระบบในการดูข่าวสารที่ต้องการได้สะดวกรวดเร็ว และระบบที่พัฒนาขึ้นมามีประสิทธิภาพสามารถรองรับการใช้งานของผู้ใช้ได้ และสามารถปรับเปลี่ยนให้ระบบสามารถรองรับการใช้งานที่เพิ่มมากขึ้นได้ง่าย และระบบยังสามารถปรับเปลี่ยนวิธีการติดต่อกันสื่อสารกันผ่านเครือข่ายได้สะดวก อีกทั้งยังสามารถเข้าใจวิธีการทำงานของเทคโนโลยี .Net Remoting ได้เป็นอย่างดีซึ่งจะสามารถนำไปประยุกต์แก้ไขกับระบบงานต่างๆ ได้

บทที่ 2

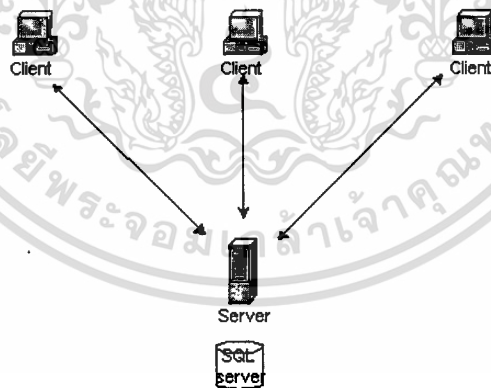
ทฤษฎีที่เกี่ยวข้อง

2.1 บทนำ

Distributed Application คือ โปรแกรมที่ถูกกระจายไปทำงานที่คอมพิวเตอร์มากกว่าหนึ่งเครื่อง โดยมีจุดประสงค์เพื่อที่จะปรับปรุงประสิทธิภาพการทำงานและการปรับเปลี่ยนขนาดให้ดีขึ้น เพื่อรองรับการทำงานของไคลเอนท์ ในเวลาเดียวกันได้มากขึ้น อีกทั้งยังสามารถลดภาระการทำงานที่เครื่องไคลเอนท์ โดยเทคนิคที่ใช้ทำ distributed application นั้นมีหลายวิธี โดยในหัวข้อสัมมนานี้จะกล่าวถึงโครงสร้างและวิธีการทำงานของ .NET remoting ซึ่งเป็นเทคโนโลยีหนึ่งที่ใช้ในการพัฒนา distributed application

2.2 โครงสร้างของ Client/server

โดยทั่วไป โปรแกรมประเภทไคลเอนท์/เซิร์ฟเวอร์จะอยู่ในเซิร์ฟเวอร์ที่อยู่บนเครือข่ายภายใน (Internal network) ซึ่งส่วนใหญ่เป็นฐานข้อมูล โดยไคลเอนท์สามารถรันโปรแกรมได้จากคอมพิวเตอร์เครื่องต่างๆ ดังรูปที่ 2.1



รูปที่ 2.1 Basic client/server interaction

เซิร์ฟเวอร์ จะมี 2 บทบาท คือ เป็น File server และ Database server
การใช้งานระบบแบบนี้มีข้อดี 2 ประการคือ (Macdonald, 2003)

1) ทุกๆไคลเอนท์ สามารถเข้ามาใช้งาน โปรแกรมเวอร์ชันเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

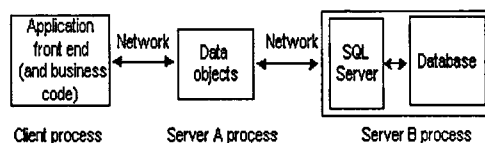
2) มีศูนย์กลางข้อมูลอยู่ที่เดียวกัน แม้ว่าการออกแบบในลักษณะนี้โปรแกรมถูกใช้งานร่วมกันในหลายๆ ไคลเอนต์แต่ที่จริงแล้วไคลเอนต์ ทำงานมากกว่าเซิร์ฟเวอร์ เพราะว่า ไคลเอนต์ อ่าน Executable และ DLL file จากเซิร์ฟเวอร์ และ load มายังไคลเอนต์ เพื่อรันโปรแกรม

2.2.1 ปัญหาของ client/server

Database ปัญหาคอขวด (bottleneck) ในกรณีที่มีไคลเอนต์จำนวนมากเข้ามาใช้ ฐานข้อมูล ในขณะที่เดียวกันจะต้องเสีย overhead มากในการสร้างและปล่อยการเชื่อมต่อ ซึ่งบ่อยครั้งจะพบว่าการออกแบบโปรแกรมไม่ดี ไคลเอนต์จะเปิดคอนเน็คชัน ค้างไว้ตลอดการทำงานของโปรแกรมซึ่งมีผลทำให้จำนวน user ที่สามารถใช้งานพร้อมกันได้จะเท่ากับจำนวนคอนเน็คชัน ที่สามารถต่อกับ database server แต่ที่จริงแล้วปัญหาคือ overhead ในการสร้างคอนเน็คชันรวมถึงการตรวจสอบสิทธิ์ของ ไคลเอนต์ (client) ซึ่งเป็นปัญหาที่ไม่สามารถหลีกเลี่ยงได้ การติดตั้ง ไคลเอนต์/เซิร์ฟเวอร์(client/server) โปรแกรม ในส่วนของเซิร์ฟเวอร์ นั้นถ้ามีการอัปเดตก็เพียงแค่นำไฟล์ใหม่ไปแทนที่ไฟล์ เดิมบนเครื่อง เซิร์ฟเวอร์ เท่านั้นแต่ในส่วนของไคลเอนต์ โปรแกรมอาจจะต้องการ คอมโพเนนต์หลายอย่างซึ่งจะต้องถูกติดตั้งไปบนเครื่อง ไคลเอนต์แต่ละไคลเอนต์ และในบางครั้งเมื่อมีการเปลี่ยนแปลงคอมโพเนนต์ (component) จำเป็นจะต้อง setup โปรแกรมใหม่ ซึ่งถ้าเป็น distributed โปรแกรม คอมโพเนนต์ต่างๆจะถูกรีจิสเตอร์ (register) ไว้ที่เครื่องที่ถูกเรียกใช้ (execute) เท่านั้น (Macdonald. 2003)

2.3 Distributed Architecture

เป็นโครงสร้างที่พัฒนาจาก ไคลเอนต์/เซิร์ฟเวอร์ให้สามารถปรับเปลี่ยนขนาดได้มากขึ้น โดยแนวความคิดหลักของ distributed application คือ โปรแกรมที่มีบางฟังก์ชันรันอยู่บนคอมพิวเตอร์เครื่องอื่น โดยฝั่งไคลเอนต์ จะติดต่อกับ เครื่องรีโมท เพื่อส่งคำสั่งหรือรับข้อมูล รูปที่ 2.2 จะแสดงถึงการออกแบบ โดยทั่วไปของ distributed application



รูปที่ 2.2 ตัวอย่าง Distributed application

ซึ่งเราสามารถแบ่งออกเป็นหลายๆคอมโพเนนต์ และกระจายไปยังหลายๆเครื่องตามจำนวนงานของแอปพลิเคชัน (Macdonald. 2003)

2.3.1 ประโยชน์ของ Distributed Architecture

อนุญาตให้ใช้งานคอมโพเนนต์บนคอมพิวเตอร์เครื่องอื่นได้ , สนับสนุน cross-platform และการกระจาย transaction และประโยชน์หลักที่สุดคือด้าน scalability คือ คอมพิวเตอร์เครื่องใหม่สามารถถูกรวมเข้ามาในระบบได้เมื่อระบบมีความต้องการประสิทธิภาพเพิ่มมากขึ้น เรื่อง stability ที่ดีขึ้นเนื่องจากเมื่อมีเครื่องไหนผิดพลาดไม่สามารถทำงานต่อไปได้ ก็ยังมีเครื่องอื่นสามารถทำงานต่อไปได้ ด้านการใช้งาน resource ที่มีอยู่จำกัดเช่น database connection โดยใช้ฟีเจอร์ object pooling เข้ามาช่วยได้

2.3.2 DCOM

เป็นเทคโนโลยีที่ใช้สร้าง distributed application ในสมัยก่อน ซึ่งมีปัญหาอยู่ 2 ปัญหา คือ garbage collection system และ binary communication protocol
Garbage collection: ใช้การ ping ในการทำ garbage collection โดยที่ไคลเอนท์ จะส่งแมสเซจไปยังคอมโพเนนต์ต่างๆ (distributed component) โดยอัตโนมัติ ถ้าเซิร์ฟเวอร์ไม่ได้รับแมสเซจ นั้น หลังจากช่วงเวลาหนึ่งอบเจกต์ จะถูก deallocate ใช้ในการจัดการ memory ซึ่งทำให้เพิ่ม network traffic ส่งผลกระทบต่อการทำงานของระบบใน network ที่ความเร็วต่ำเช่น WAN

DCOM จะรวมเซอร์วิซต่างๆเข้าไว้ด้วยกันเช่น transport level security เพื่อซ่อนการติดต่อผ่านเน็ตเวิร์ก ไว้ทำให้ง่ายในการสร้างแต่บ่อยครั้งทำให้ไม่สามารถติดต่อกับคอมพิวเตอร์ หลายๆเครื่องและไม่ผ่าน firewall ได้

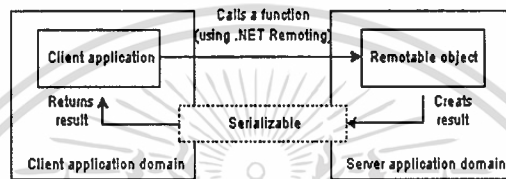
2.4 .NET Remoting

.NET Remoting ถูกพัฒนาขึ้นมาแทน DCOM ซึ่งได้แก้ปัญหาต่างๆของ DCOM ในการสร้าง distributed application โดยที่ client application ใช้งาน คอมโพเนนต์บนเครื่องรีโมท เซิร์ฟเวอร์ ได้เสมือนเป็นเครื่องตัวเอง .NET Remoting เป็นวิธีการใช้กับแอปพลิเคชันที่อยู่บนเครือข่ายภายใน(internal network) และมีการเปลี่ยนแปลงแก้ไขตัวแอปพลิเคชันได้ง่ายกว่า DCOM โดยแก้ที่ คอนฟิกไฟล์ของ โปรแกรม เช่น ต้องการเปลี่ยนแมสเซจที่ใช้ติดต่อหรือเปลี่ยน โปรโตคอล ที่ใช้ .NET Remoting ถูกออกแบบมาให้ใช้กับ peer-to-peer application

2.4.1 .NET Remoting Fundamentals

.NET Remoting ประกอบด้วย 3 คลาสที่สำคัญคือ (Macdonald, 2003)

- Serializable classes เป็นคลาส ที่จะถูก mark ด้วย <serializable> ซึ่งสามารถส่งระหว่าง โพรเซส (process), แอปพลิเคชัน และคอมพิวเตอร์ ได้
- Remotable classes ซึ่งทำให้คลาส ถูกเรียกใช้งานโดยรีโมทไคลเอนท์ได้
- Ordinary classes เป็นคลาสที่ไม่ได้ใช้ในการติดต่อสื่อสารด้วย .NET Remoting ซึ่งอาจจะ เป็น ส่วนในแอปพลิเคชัน



รูปที่ 2.3 Remotable และ Serializable types

2.4.2 Serializable Classes

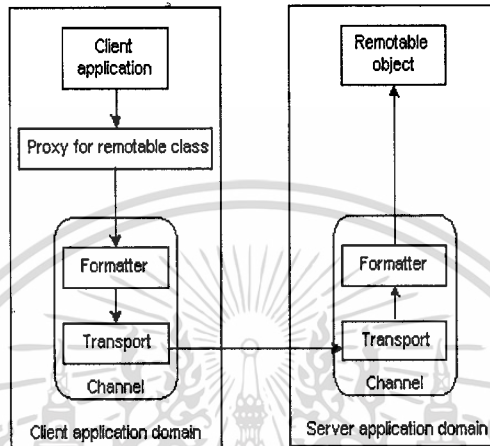
ทำหน้าที่จัดการ information package โดย .NET จะเปลี่ยน ข้อมูลทั้งหมดเป็น stream of byte และสร้างมันขึ้นมาใหม่ และ serializable object สามารถถูกส่งไปยังแอปพลิเคชัน โดเมน อื่นได้ แนวความคิดของ แอปพลิเคชัน โดเมนคล้ายกับ windows process โดยแอปพลิเคชัน โดเมน เป็นส่วนที่ถูกแยกออกมาที่ไม่สามารถติดต่อได้โดยตรง ถ้าแอปพลิเคชันเสียหาย (application crash) ในแอปพลิเคชัน โดเมนมันจะไม่กระทบกับ แอปพลิเคชัน ในแอปพลิเคชัน โดเมนอื่น แอปพลิเคชัน โดเมนมีข้อดีที่การแยกเป็น โพรเซสแต่จะลด overhead การแยกแอปพลิเคชัน โดเมนจะถูกจัดการ ด้วย .NET common language runtime (CLR)

2.4.3 Remotable Classes

ทุกๆ คลาสที่ derive มาจาก System.MarshalByRefObject จะสามารถถูกเรียกใช้จากรีโมทไคลเอนท์ได้ การติดต่อสื่อสารของ .NET remoting จะถูกควบคุมดูแลผ่านทาง proxy object โดยโปรแกรมจะติดต่อกับตัว proxy ซึ่งเป็นเสมือน remote object เมื่อเรียกใช้ method บน proxy object มันจะไปเรียก remote object ให้และจะให้ข้อมูลที่ต้องการคืนกลับมา

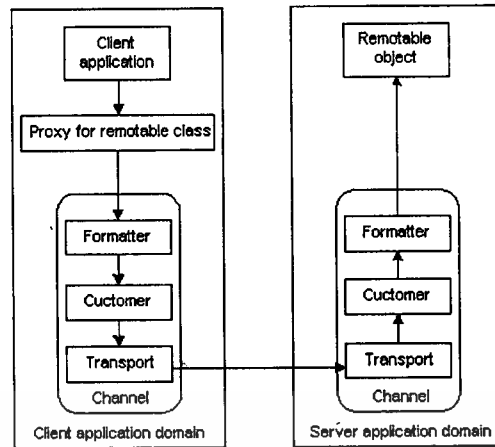
เมื่อมีการติดต่อกับ remote object proxy ไม่ได้ทำทุกอย่างที่ถูกร้องขอมาแต่มันจะติดต่อกับ formatter ซึ่งจะแปลงข้อมูลที่ไคลเอนท์ร้องขอหรือ เซิร์ฟเวอร์ตอบกลับ ให้เป็นรูปแบบที่เหมาะสม

แล้ว formatter จะติดต่อกับ transport channel เพื่อส่งข้อมูลโดยใช้โปรโตคอลที่เหมาะสม ซึ่งผู้พัฒนาระบบสามารถเปลี่ยนแปลง formatter และ channel ของโปรแกรมโดยไม่ต้องคอมไพล์โค้ดใหม่ โดยเพียงแค่แก้คอนฟิกูเรชันไฟล์ใหม่รูปที่ 2.4 จะแสดงวิธีการร้องขอแพ็คเกจบน ฟังโคลเอนท์ แล้วถูกถอดแพ็คเกจ บนฝั่งเซิร์ฟเวอร์และ เซิร์ฟเวอร์จะคืนค่ากลับไปให้โคลเอนท์ด้วยทางเดียวกัน



รูปที่ 2.4 The multilayered .NET Remoting architecture

Formatter จะทำ serialize message ให้เป็น binary หรือ SOAP ก่อนที่จะส่งแพ็คเกจออกไป transport channel หรือมี custom channel มาขึ้นระหว่าง formatter และ transport channel. นี้ก็จะจัดการ encryption, logging, compression และ transport channel จะต้องเป็น sink สุดท้ายเพื่อใช้ในการส่งข้อมูลไปยังรีโมทแอปพลิเคชัน โดเมน



รูปที่ 2.5 Interception with a custom sink

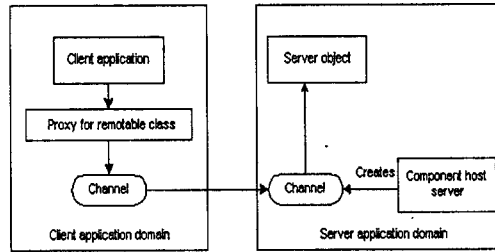
2.4.4 The Remote Component Host

นอกจากออบเจ็กต์ นั้นจะเป็น instance ของคลาส ที่ derive มาจาก MarshallByRefObject จึงจะสามารถที่จะถูกเรียกใช้งานจากระโมทโคลเอนท์ได้ เซิร์ฟเวอร์แอปพลิเคชัน ยังต้องฟังการร้องขอมาจากโคลเอนท์โดยเราสามารถสร้างเซิร์ฟเวอร์แอปพลิเคชัน ได้หลายรูปแบบ

- Windows service ซึ่งสามารถ start ขึ้นมาได้โดยอัตโนมัติเมื่อ เซิร์ฟเวอร์เปิดขึ้นมาและ ไม่มี user interface
- Console application ซึ่งเป็นข้อมูล log ที่แสดงขึ้นมาบนหน้าจอ
- Windows application ซึ่งเป็นอินเทอร์เฟส

Component host จะต้องทำงาน 2 งาน คือ

- 1) สร้าง server-side channel ซึ่งจะทำหน้าที่ฟัง client request เมื่อได้รับ client request .NET Remoting จะสร้างรีโมทออบเจ็กต์ที่โคลเอนท์ต้องการที่แอปพลิเคชันเดียวกับคอมโพเนนต์โฮสต์
- 2) หลังจากคอมโพเนนต์ ถูกสร้างขึ้น คอมโพเนนต์โฮสต์ยังทำให้เกิดการติดต่อสื่อสารระหว่างออบเจ็กต์ ในแอปพลิเคชันโดเมน ของมัน และ รีโมทโคลเอนท์อย่างไรก็ตามโคลเอนท์ไม่เคยติดต่อโดยตรงกับคอมโพเนนต์โฮสต์ รูปที่ 2.6 แสดงแผนภาพ (diagram) ของความสัมพันธ์นี้



รูปที่ 2.6 The role of the component host

.NET Remoting ใช้ thread pool ในการรอฟัง client request นั้นหมายความว่า สามารถมีไคลเอนต์ มากกว่าหนึ่ง ไคลเอนต์ สามารถสร้างหรือใช้อ็อบเจ็กต์ได้พร้อมกัน แนวความคิดของ Object Pooling ลองคิดว่า limit ของ คอมโพเนนต์ เป็นเสมือนการนั่งรถ taxi โดยมีเป้าหมายที่จะเดินทางไปสนามบิน โดยถ้าผู้โดยสารคนที่ต้องการ ไปสนามบินใช้ taxi คนละคันก็จะทำให้ไม่มีที่จอดรถเพียงพอซึ่งสถานการณ์นี้ก็เปรียบเสมือนกับการเขียนไคลเอนต์/เซิร์ฟเวอร์ แอปพลิเคชัน ไม่ได้ในจัดการเกี่ยวกับฐานข้อมูล คือไม่มีการคืน คอนเน็คชันให้คนอื่นที่ต้องการจะใช้งานได้ใช้ อีกวิธีหนึ่งคือแต่ละคนใช้ taxi แต่ครั้งเดียว โดยเมื่อถึงสนามบินก็ให้ทำลาย taxiทิ้งซึ่งก็จะทำให้ที่จอดรถที่สนามบินว่างและรถ taxi คันใหม่ก็สามารถเข้ามาได้ แต่การสร้าง taxi คันใหม่ก็อาจจะไม่ทันกับความต้องการที่จะใช้รถ ซึ่งก็เปรียบได้กับการเขียนไคลเอนต์/เซิร์ฟเวอร์ แอปพลิเคชัน ที่ดีที่คืน database connection และวิธีที่ดีที่สุดที่ใกล้เคียงกับการใช้งานจริงคือสร้างกลุ่มของ taxi ขึ้นมาจำนวนหนึ่ง (pool of taxis) เมื่อ taxi ส่งผู้โดยสารถึงสนามบินก็จะกลับมารับผู้โดยสารคนใหม่ โดย dispatcher จะจัดการดูแลให้ request ใหม่และจัดการให้ใช้รถ taxi ที่สามารถใช้งานได้ในขณะนั้น

2.4.5 สิ่งที่ต้องคำนึงในการออกแบบ .NET Remoting

Activation modes มีอยู่ 2 ประเภท (Curra. 2002; Olsen. 2002; Pinnock. 2002)

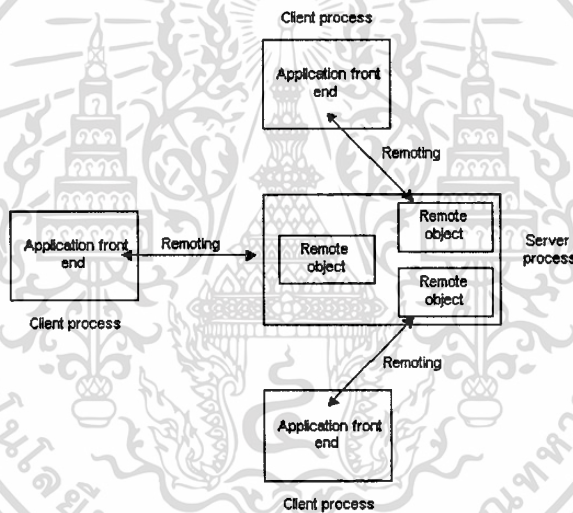
- Server Activation เป็นอ็อบเจ็กต์ที่ถูกสร้างขึ้นมาโดยเซิร์ฟเวอร์ ซึ่งจะถูกรวบรวม lifetime โดยเซิร์ฟเวอร์ โดยที่ เซิร์ฟเวอร์จะสร้างอ็อบเจ็กต์ขึ้นมาเมื่อ ไคลเอนต์เรียกใช้งานเมทอดผ่านอ็อบเจ็กต์ของไคลเอนต์แอปพลิเคชัน โดเมนไม่ใช่ขณะที่ไคลเอนต์ New object (New()) ซึ่งมีด้วยกัน 2 ประเภท

Singleton type: ณ.เวลาขณะใดขณะหนึ่งจะไม่มีอ็อบเจ็กต์มากกว่า 1 instance ถ้ามี instance เกิดขึ้นแล้ว instance นั้นจะรับทุกๆ request ที่ไคลเอนต์ร้องขอมาถ้ายังไม่มี instance เกิดขึ้น

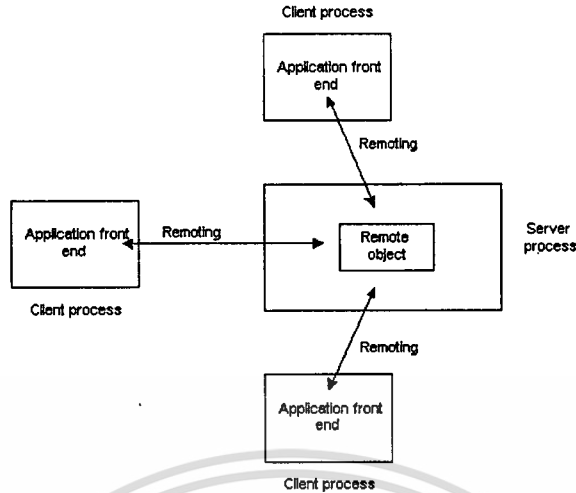
เซิร์ฟเวอร์ จะสร้าง instance ขึ้นมาใหม่และ instance นั้นจะรับ request ของไคลเอนท์ ที่มาหลังจากนั้นและมีการเก็บสถานะของการเรียกใช้โดย อ็อบเจ็กต์ประเภทนี้จะอยู่จนกระทั่งหมด lifetime ที่กำหนดขึ้น

Singlecall type: มี 1 instance ต่อ 1 client request คือ การขอใช้ method ถัดไปจะถูกให้บริการโดย instance คนละตัวกัน และไม่มีการเก็บแสดงของการเรียกใช้ ซึ่งจะอยู่แค่ในระหว่างที่มีการเรียกใช้ method นั้น

- Client Activation โดย client-activated object จะสร้างโดย ไคลเอนท์ ซึ่งแต่ละ instance จะถูกสร้างขึ้นสำหรับหนึ่งไคลเอนท์ โดยจะเกิดขึ้นเมื่อ client new instance ใหม่และมีการเก็บแสดง (state) ของการเรียกใช้งานโดยจะถูกทำลายไปต่อเมื่อไคลเอนท์ เลิกใช้งานหรือ เกิน lifetime



รูปที่ 2.7 .NET Remoting with singlecall objects



รูปที่ 2.8 .NET Remoting with a singleton object

2.4.6 Object lifetime

เมื่อ distributed system มีการใช้งานข้ามเครื่องเพราะฉะนั้น จะต้องรู้ว่าเมื่อไรจะต้องปล่อยทรัพยากร (resource) โดยใน .NET lifetime object จะถูกจัดการ โดยโปรเซสที่เรียกว่า leased-based lifetime โดย lease จะถูกสร้าง สำหรับทุกๆ อ็อบเจกต์ที่อ้างถึงถูกสร้างขึ้นนอกแอปพลิเคชัน โดเมนของมัน แต่ละ แอปพลิเคชัน โดเมนก็จะมี lease manager ของตัวเอง ซึ่งจะคอยตรวจสอบสถานะระหว่าง lease ภายนอก โดยแต่ละ lease จะมี lease time ของตัวเอง เมื่อเวลานี้หมด lease manager จะปล่อยอ็อบเจกต์นั้นและ อ็อบเจกต์นั้นจะถูกทำลายโดย garbage collector แต่ละ lease time จะเริ่มด้วยค่าดีฟอลต์ (default) แต่สามารถเปลี่ยน โดยอ็อบเจกต์ของไคลเอนท์ หรือเซิร์ฟเวอร์ ซึ่ง lifetime นี้จะใช้กับอ็อบเจกต์ ประเภท Singleton และ Client-activated (Macdonald. 2003)

2.4.7 การติดต่อสื่อสารกับ remote component

โดยรูปแบบของการติดต่อและช่องทางการติดต่อมีให้เลือกอยู่ 2 แบบ

รูปแบบของข้อมูล (Curra. 2002; Olsen. 2002; Pinnock. 2002)

- Binary formatter ทำการ serialize ข้อมูลอยู่ในรูปแบบของ binary ซึ่งจะใช้ได้กับ .NET application เท่านั้น
- Soap formatter ทำการ serialize ข้อมูลให้อยู่ในรูปของ SOAP message ซึ่งจะสามารถใช้ข้าม platform กันได้ SOAP ต้องใช้ขนาดของเมสเสจ ที่ใหญ่กว่า binary ดังนั้นทำให้ประสิทธิภาพโดยรวมลดลงช่องทางการส่งข้อมูล
- TCP Channel เป็นคอนเน็คชันที่ใช้โปรโตคอล TCP

- HTTP Channel เป็นคอนเน็คชั่น ที่ใช้โพรโทคอล TCP ซึ่งจะใช้ในการติดต่อผ่าน อินเทอร์เน็ต

ซึ่งทั้งรูปแบบการติดต่อสื่อสารและช่องทางการติดต่อสื่อสารนั้นสามารถคอนฟิก ได้ในคอนฟิก ไฟล์

2.4.8 Hosting .NET Remoting Objects

- 1 Executable file .NET Remoting object สามารถเป็น host ได้ด้วย EXE
- 2 IIS สามารถเป็น โฮสต์ ได้ด้วย Internet Information Server (IIS) โดยปกติแล้วจะให้บริการผ่านทาง HTTP channel โดยที่ configuration file จะถูก load เมื่อมีแมสเสจแรกมาถึง แอปพลิเคชัน
- 3 .NET Component service

2.4.9 Metadata

.NET Framework ใช้ metadata และ assemblies ในการเก็บข้อมูลเกี่ยวกับ component โดย .NET Remoting ใช้ metadata ในการสร้าง proxy object ซึ่งจะถูกรวบรวมที่ฝั่งไคลเอนท์ โดยมีคลาส เหมือนกับฝั่งเซิร์ฟเวอร์โดย proxy จะส่ง request ไปยังเซิร์ฟเวอร์ และ formatter ใช้ metadata ในการเปลี่ยน method เป็น payload stream

ไคลเอนท์ได้รับ metadata information ที่ต้องการใช้งานรีโมทออบเจกต์ได้ 3 ทางดังนี้

- 1 .NET Assembly ของเซิร์ฟเวอร์อ็อบเจกต์ โดย เซิร์ฟเวอร์สามารถสร้างและกระจาย metadata ให้ไคลเอนท์ ได้ไคลเอนท์อ็อบเจกต์จะอ้างถึง assembly ขณะที่กำลัง compile ไคลเอนท์อ็อบเจกต์
- 2 ผ่านทาง WSDL (Web services Description Language) file ซึ่งใช้อธิบายอ็อบเจกต์ และ method ของอ็อบเจกต์นั้น ไคลเอนท์สามารถอ่านและสร้าง SOAP request ไปยัง WSDL file
- 3 .NET client สามารถใช้ SOAPSUDS download XML schema จาก เซิร์ฟเวอร์ที่สร้าง source file หรือ assembly ที่มีแค่ metadata (Macdonald, 2003)

2.4.10 Configuration files

Configuration file ใช้ระบุข้อมูลต่างๆสำหรับ object โดยปกติแต่ละ application domain จะมีคอนฟิกไฟล์เป็นของตนเอง ซึ่งช่วยในการ location transparency คือสามารถเปลี่ยนที่อยู่ของ อ็อบเจกต์ ได้โดยไม่ต้อง compile program ใหม่ โดยทั่วไปคอนฟิกไฟล์ประกอบด้วยข้อมูลดังนี้

- Host Application information
- ชื่อของอ็อบเจกต์

- URI ของอ็อบเจ็กต์
- Channel ที่ใช้งาน
- Lease time สำหรับเซิร์ฟเวอร์

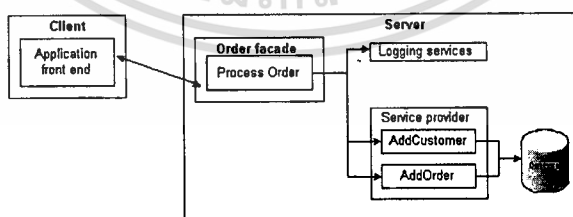
2.5 Distributed Design Patterns

มีการออกแบบอยู่ 3 รูปแบบที่สำคัญในการสร้างแอปพลิเคชัน ประเภทที่แบ่งเป็นแต่ละคอมโพเนนต์ และนำมารวมกันเป็นระบบใหญ่ (Macdonald. 2003)

- Interfaces เป็นคุณสมบัติและ method ที่ทุกๆ คอมโพเนนต์ จะต้องมาให้ โดยสามารถมองว่าอินเทอร์เฟซเป็นเสมือน control panel ของคลาส โดยจะทำหน้าที่ควบคุมและกำหนดขอบเขตการทำงานของ ผู้ใช้ที่สามารถกระทำได้ โดยจะต้องสร้างอินเทอร์เฟซ ก่อนที่จะมีการพัฒนาคอมโพเนนต์ จริงๆ ซึ่งกระบวนการนี้จะบังคับให้กำหนดวิธีการติดต่อระหว่างคลาส และแบ่งฟังก์ชันการทำงานระหว่างคลาส ก่อนที่จะเขียน code จริง และ อ็อบเจ็กต์หนึ่งสามารถมี interface ได้มากกว่า 1 แบบเพราะฉะนั้น ไคลเอนต์ สามารถเลือก interface ที่เหมาะสมเพื่อใช้ในการติดต่อ

- Factories เป็น คลาสที่สร้างคลาสอื่นๆ การใช้ factory ทำให้ไคลเอนต์สามารถสร้างรีโมทอ็อบเจ็กต์ได้ class factory สามารถตรวจสอบข้อมูลเพิ่มเติมเช่นค่าใน คอนฟิกูเรชันไฟล์ และตัดสินใจว่าจะ return proxy ที่ใช้สำหรับติดต่อกับ remote instance ซึ่งส่วนนี้จะเป็นส่วนไม่ขึ้นอยู่กับที่อยู่ของ ไคลเอนต์ทำให้ง่ายในการเปลี่ยนแปลงจาก architecture หนึ่ง ไปเป็นอีกแบบหนึ่ง และซ่อนรายละเอียดภายในจากไคลเอนต์แอปพลิเคชัน

- Facades คือ layer ระหว่าง service provider component และ ไคลเอนต์ ซึ่ง service provider จะซ่อนขั้นตอนการทำงานของเซอร์วิจไว้ภายใน เช่น การทำงานภายในของฐานข้อมูล ดังรูป 9



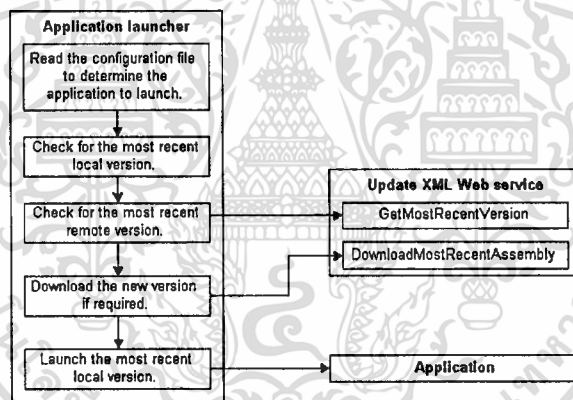
รูปที่ 2.9 Using a façade

การเพิ่ม facade ทำให้สามารถแยก business logic ออกจาก data layer ซึ่งทำให้ไม่ต้องมีการเปลี่ยนแปลง client code แต่มันจะใช้เวลาโดยรวมของการทำงานมากขึ้นเพราะว่ามีการดึงการเครือข่าย และยังสามารถเพิ่ม interception service เช่นสามารถตรวจสอบ error และ logging ฟังก์ชันต่างๆจาก data access code

2.6 Automated Deployment

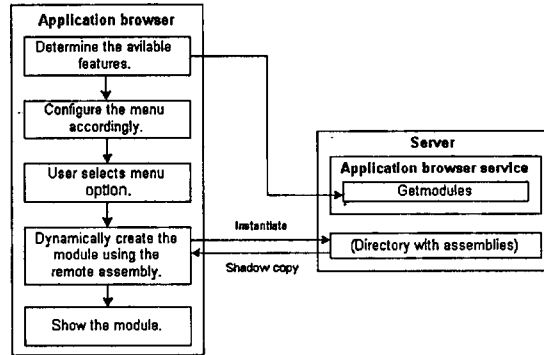
มีพื้นฐานการ deployment อยู่ 2 แบบคือ (Macdonald, 2003)

- application launcher ซึ่งจะตรวจสอบว่าเวอร์ชัน ของแอปพลิเคชัน ที่ใช้งานนั้นเป็นเวอร์ชันใหม่สุดหรือไม่ โดยจะไปค้นหาที่ XML web service ซึ่งถ้ายังไม่เป็นเวอร์ชันใหม่ launcher จะทำการส่งอีเมลไปเตือนผู้ใช้ หรืออาจจะอัปเดต แอปพลิเคชัน ให้โดยอัตโนมัติซึ่งเมื่อกระบวนการนี้สำเร็จ แอปพลิเคชันเวอร์ชันใหม่ที่สุดก็就会被สตาร์ทขึ้นมาใช้งาน รูปที่ 2.10 แสดงการทำงานของ application launcher



รูปที่ 2.10 The update process with an application launcher

- Application browser โดยจะแบ่งแอปพลิเคชันฟังก์ชัน ออกเป็นโมดูลหลายๆ โมดูลที่ทำหน้าที่คนละส่วนกันเมื่อ ผู้ใช้ login เข้ามาใช้งาน application browser จะไปค้นหาใน XML web service เพื่อตรวจสอบว่ามีโมดูลไหนที่อนุญาตให้ ผู้ใช้คนนี้ใช้งานได้บ้างซึ่งมันจะใช้ข้อมูลนี้จัดการ user interface เมื่อ ผู้ใช้พยายามที่จะใช้ module หนึ่ง application browser ก็จะดาวน์โหลดโมดูล นั้นมาจากเซิร์ฟเวอร์ซึ่งวิธีนี้จะไม่สามารถใช้ได้กับแอปพลิเคชันที่ disconnect จากเซิร์ฟเวอร์



รูปที่ 2.11 Using an application browser

ข้อแตกต่างระหว่าง application launcher และ application browser

- application browser แอปพลิเคชันถูกออกแบบเป็นแยกเป็นหลายๆส่วน เพื่อให้เป็นอิสระต่อกันเพื่อที่จะสามารถ ดาวน์โหลด ได้เฉพาะคอมโพเนนต์ที่ต้องการ
- เพราะว่า application browser มีความซับซ้อนมากทำให้การจัดการกระบวนการดาวน์โหลดภายในโปรแกรมยาก
- application browser เป็นตัวกำหนด user interface สำหรับบางระดับ เช่น application browser ใช้เป็นแบบ multiple document interface (MDI) window แต่ภายในวินโดวส์ย่อยที่ถูกสร้างขึ้นโดย MDI ใช้เป็น application launcher

2.7 เปรียบเทียบกับระบบอื่นๆในปัจจุบัน

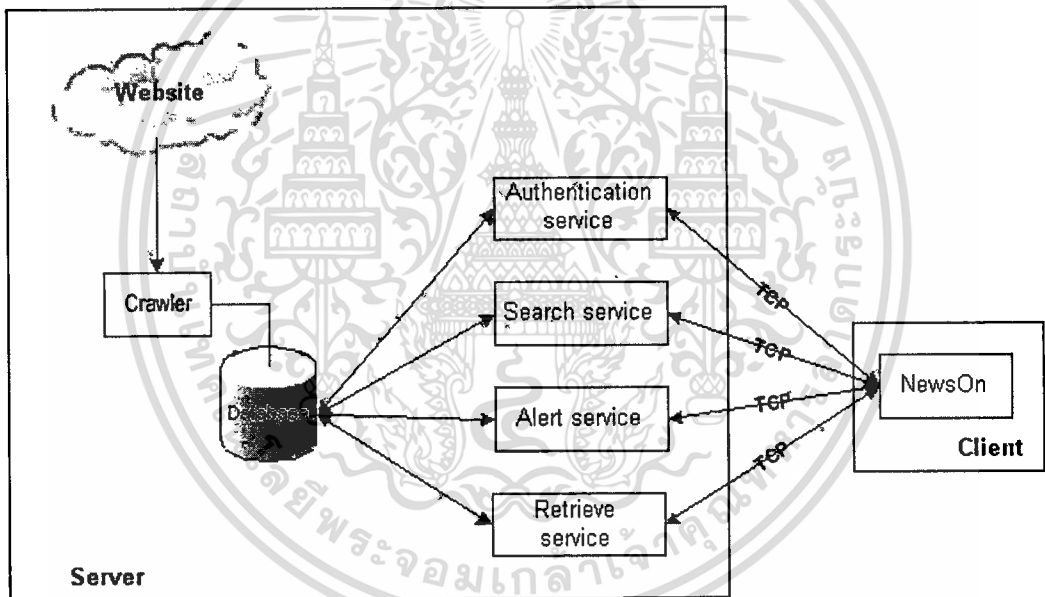
การทำงานของระบบอื่นๆที่ทำงานคล้ายกับในลักษณะนี้ส่วนใหญ่จะเป็นการดึงข้อมูลจากเว็บไซต์มาแล้วนำไปแสดงบนหน้าเว็บไซต์ของตนเองซึ่งจะมีการทำงานในลักษณะbatchคือจะมีการเปลี่ยนข้อมูลวันละครั้ง เป็นต้น ซึ่งไม่สามารถค้นหาข้อมูลที่ต้องการได้และไม่ทราบว่ามีข่าวใหม่เข้ามาเมื่อไรต้องกด refresh ที่ Browser เอง อีกทั้งข้อมูลส่วนใหญ่ยังไม่สามารถดูย้อนหลังได้ ซึ่งระบบ News online นี้สามารถที่จะแก้ปัญหาต่างๆที่กล่าวมาได้ และยังเป็นระบบแบบกระจาย ซึ่งการทำงานฝั่งเซิร์ฟเวอร์สามารถกระจายการทำงานไปยังหลายเครื่องได้

บทที่ 3

การออกแบบและการพัฒนาระบบ

3.1 การออกแบบระบบโดยรวม

โครงสร้างระบบโดยรวมจะแบ่งออกเป็นแต่ละเซอร์วิส ซึ่งแต่ละเซอร์วิสจะเป็นส่วนที่ให้บริการงานแต่ละประเภทเมื่อไคลเอนต์มีการเข้ามาใช้งานระบบโดยแต่ละเซอร์วิส สามารถที่จะให้บริการอยู่บนเครื่องเดียวกันหรือคนละเครื่องกันได้โดยจะติดต่อกันผ่านเครือข่ายเน็ตเวิร์ก ซึ่งรูปแบบของระบบโดยรวมจะออกมาดังรูปที่ 3.1



รูปที่ 3.1 โครงสร้างระบบ News online system

ส่วนประกอบต่างๆของระบบประกอบด้วย

ฝั่งเซิร์ฟเวอร์

- Crawler เป็น แอปพลิเคชัน ที่คอยจัดการดึงข่าวจากในหน้าเว็บไซต์เข้ามาเก็บไว้ในฐานข้อมูล เพื่อที่จะเป็นฐานข้อมูลสำหรับระบบ

- Authentication Service เป็นแอปพลิเคชันเซิร์ฟเวอร์ (application server) ที่ทำหน้าที่ให้บริการด้านการตรวจสอบสิทธิ์ของผู้ใช้ที่เข้ามาใช้งานว่ามีสิทธิ์ถูกต้องหรือไม่ โดยจะทำการตรวจสอบกับฐานข้อมูลของระบบ
- Search Service เป็นแอปพลิเคชันเซิร์ฟเวอร์ (application server) ที่ทำหน้าที่ให้บริการด้านการค้นหาข่าวจากคำที่ผู้ใช้ต้องการหา ซึ่งจะเข้าไปหาในฐานข้อมูลทั้งหมดของระบบว่ามีข่าวที่ต้องการหรือไม่ ถ้ามีก็จะทำการส่งหัวข้อข่าวที่ได้กลับไปให้ผู้ใช้ แต่ถ้าไม่พบข่าวที่ต้องการก็จะส่งไปบอกผู้ใช้ว่าไม่พบข่าวที่ผู้ใช้ต้องการ
- Retrieve Service เป็นแอปพลิเคชันเซิร์ฟเวอร์ (application server) ที่ทำหน้าที่บริการให้เนื้อข่าวที่ผู้ใช้ต้องการอ่าน (ดับเบิลคลิกที่หัวข้อข่าว) โดยจะนำข้อมูลของข่าวชิ้นนั้นที่เก็บอยู่ในฐานข้อมูลไปแสดงที่หน้าจอของผู้ใช้
- Alert Service เป็นแอปพลิเคชันเซิร์ฟเวอร์ (application server) ที่ทำหน้าที่บริการในการส่งข่าวไปอัปเดตที่หน้าจอของผู้ใช้เมื่อมีข่าวชิ้นใหม่เข้ามาในระบบ
- Crawler application เป็นแอปพลิเคชันเซิร์ฟเวอร์ (application server) ที่ทำหน้าที่คอยไปดึงข่าวมาจากเว็บไซต์ (website) ใน internet เพื่อให้มีข่าวใหม่เข้ามาในระบบอยู่เสมอเมื่อเว็บไซต์นั้นมีการอัปเดต (update)

ฝั่งไคลเอนท์

- NewsOn เป็นไคลเอนท์แอปพลิเคชัน (Client Application) ที่ทำหน้าที่ติดต่อระหว่างผู้ใช้งานกับ service ต่างๆ ทางฝั่ง server รวมไปถึงการแสดงผลที่ได้จาก service ต่างๆ ของระบบ

โดยการติดต่อสื่อสารระหว่างไคลเอนท์ และเซิร์ฟเวอร์นี้จะติดต่อสื่อสารกันด้วยโพรโตคอล TCP

3.2 System Requirement

Server

- CPU: Pentium III ขึ้นไป
- Ram: 128 MB ขึ้นไป
- Harddisk: 4GB ขึ้นไป
- OS: windows2000 Professional / windows2000 server / windowsXP
- Database: SQL2000 server
- .NET framework redistributed 1.0

Client

- CPU: Pentium II ขึ้นไป
- Ram: 128 MB ขึ้นไป
- Harddisk: 4GB ขึ้นไป
- OS: windows98 / windows2000 Professional / windowsXP
- .NET framework redistributed 1.0

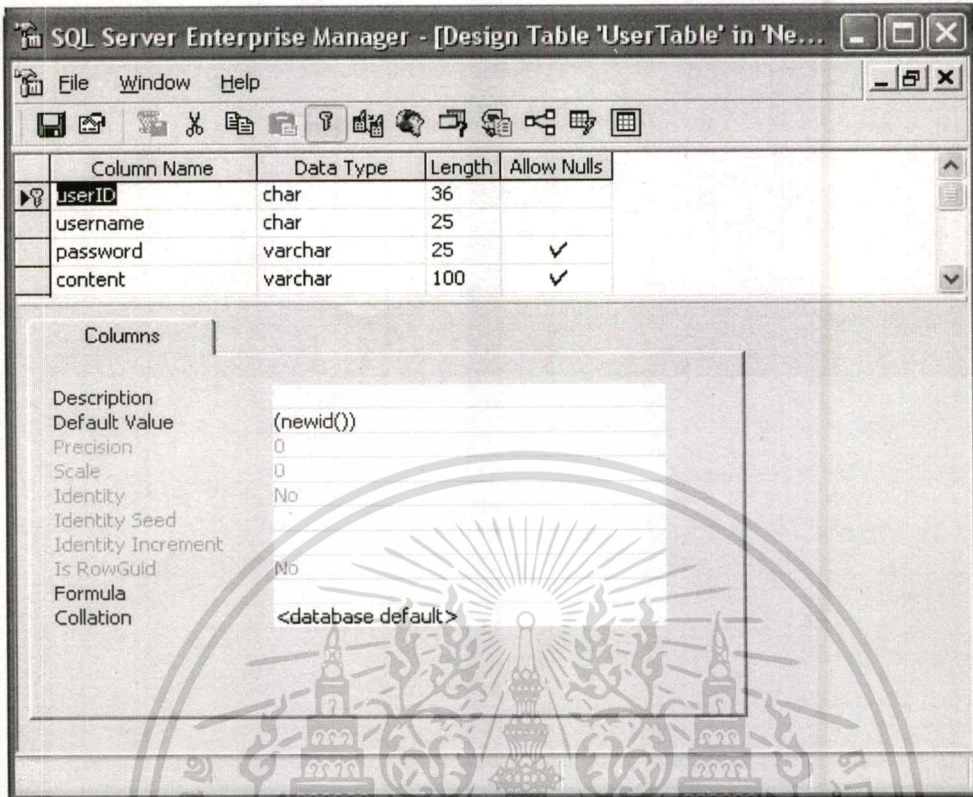
3.3 รายละเอียดภายในระบบ

Authentication Service

ทำหน้าที่ตรวจสอบว่าผู้ใช้นั้นมีสิทธิ์เข้ามาใช้งานในระบบหรือไม่โดยมีกระบวนการดังนี้
 รับ username และ password มาจาก client application (NewsOn) โดยไคลเอนท์จะส่ง username และ password มาด้วยการ encode โดยใช้ TCP protocol ในการติดต่อสื่อสารกันและ เมื่อฝั่ง server ได้รับก็จะทำการ decode username และ password เหล่านั้นแล้วจะนำไปตรวจสอบกับฐานข้อมูลชื่อ NewsDB ในตารางชื่อ usertable ซึ่งมี field ดังต่อไปนี้

```
CREATE TABLE [dbo].[UserTable] (
  [userID] [char] (36) COLLATE Thai_CI_AS NOT NULL ,
  [username] [char] (25) COLLATE Thai_CI_AS NOT NULL ,
  [password] [varchar] (25) COLLATE Thai_CI_AS NULL ,
  [content] [varchar] (100) COLLATE Thai_CI_AS NULL
) ON [PRIMARY]
```

- userID เก็บข้อมูลประเภท character ขนาด 36 ไบต์ ซึ่งใช้เก็บ unique ID ของผู้ใช้แต่ละคน โดย unique ID นี้จะสร้างมาจาก GUID โดยใช้ฟังก์ชัน newsid()
- username เก็บข้อมูลประเภท character ขนาด 25 ไบต์ ใช้เก็บ username ของผู้ใช้ที่จะใช้ในการเข้าระบบซึ่งจะมีความยาวไม่เกิน 25 ตัวอักษร
- password เก็บข้อมูลประเภท character ขนาด 30 ไบต์ ใช้เก็บ password ของผู้ใช้ที่จะใช้ในการเข้าระบบซึ่งจะมีความยาวไม่เกิน 30 ตัวอักษร
- content เก็บข้อมูลประเภท character ขนาด 100 ไบต์ ใช้เก็บรายละเอียดของผู้ใช้นั้นซึ่งจะมีความยาวไม่เกิน 100 ตัวอักษร



รูปที่ 3.2 User Table

โดยเซอรวัดนี้จะเรียกใช้ ฟังก์ชันชื่อ

Function `getuserid(ByVal v_username As String, ByVal v_password As String) As String`

ในการรับ username และ password แล้วเข้าไปตรวจสอบในฐานข้อมูล โดยจะเรียกใช้งานผ่าน

store procedure ที่ชื่อ `sp_checkuser` โดย store procedure นี้จะคืน userID เมื่อ username และ

password ถูกต้อง และจะคืนค่า null ให้ถ้าไม่มี username หรือ password ผิด และฟังก์ชัน `getuserid`

นี้จะคืนค่า userid กลับไปให้โคลอนท์เพื่อให้โคลอนท์ใช้ในการติดต่อกับ service อื่นได้ ในกรณีที่

username และ password ถูกต้องและจะคืนค่า -1 ให้กับ โคลอนท์เมื่อ username และ password ที่

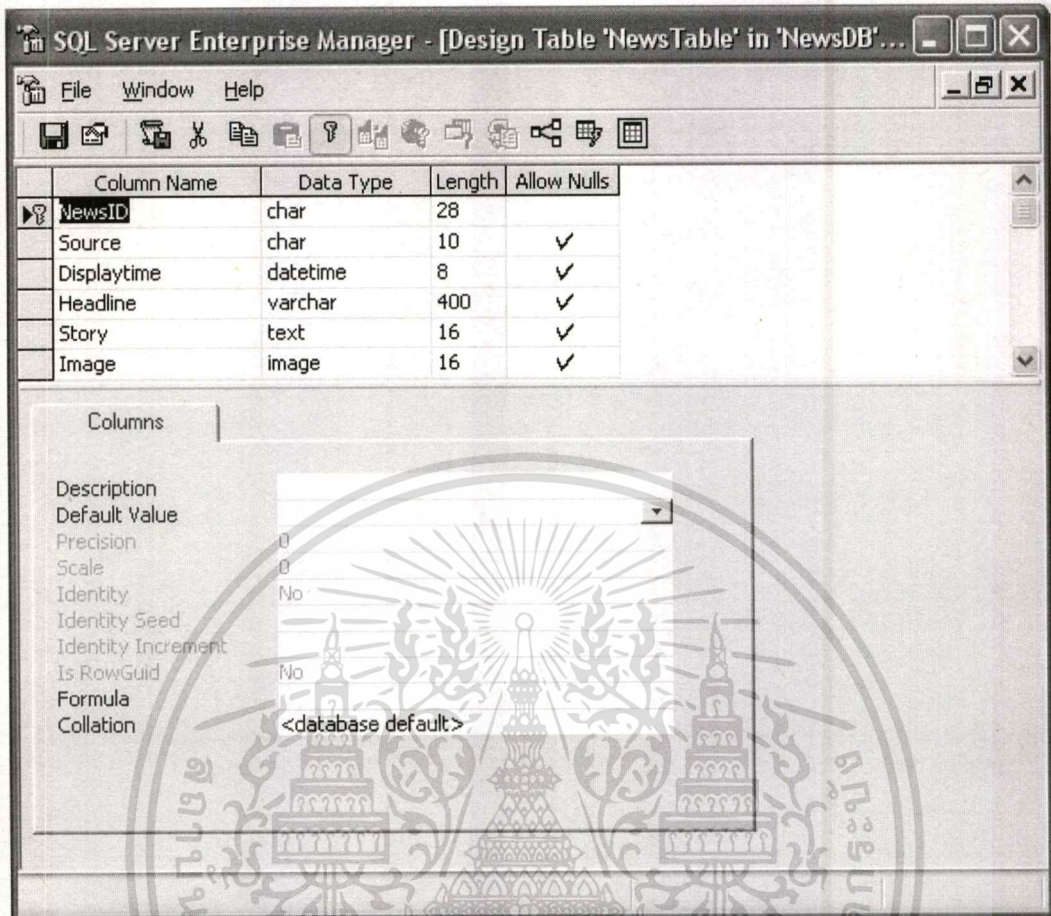
ผู้ใช้ login เข้ามาไม่ถูกต้อง

Search Service

ทำหน้าที่ค้นหาข่าวที่ผู้ใช้ต้องการค้นหาโดยจะค้นหาจากคำที่ผู้ใช้ใส่เข้ามาซึ่งมีกระบวนการทำงานดังนี้

รับ userid (ที่ได้มาจาก authentication service) และ keyword ที่ผู้ใช้ต้องการค้นหาจากไคลเอนท์ ซึ่ง userid จะเป็น unique ID คือ แต่ละคนจะไม่ซ้ำกันเพื่อนำไปใช้ตรวจสอบว่า request มาจากผู้ใช้ที่มีสิทธิ์จริงหรือไม่ และ keyword นั้นเป็นคำค้นที่ผู้ใช้ต้องการจะหาข่าวที่มีคำนั้นอยู่ในข่าวโดยที่ผู้ใช้สามารถใส่เป็นแบบ phase เช่น “การท่องเที่ยว และเศรษฐกิจ” โดยการค้นหาแบบ phase นี้จะค้นหาคำที่เหมือนในเครื่องหมาย “ ” ทั้งหมด หรือค้นหาอีกแบบคือ ค้นหาหลายๆ keyword เช่น การท่องเที่ยว เมืองไทย การค้นหาแบบนี้จะตัดคำออกเป็น 2 คำโดยแบ่งตามวรรค (space) ซึ่งจะได้ออกมาเป็น 2 คำ คือ การท่องเที่ยว and เมืองไทย โดยผลที่ได้จะเป็นข่าวที่มีคำว่า การท่องเที่ยว หรือ คำว่าเมืองไทย โดยทั้ง 2 รูปแบบในการค้นหานี้จะใช้รูปแบบการค้นหาแบบ pattern matching ของ SQL2000 server เช่นค้นหาคำว่า “กาน้ำ” ก็จะเจอทั้งข่าวที่มีคำว่า ฝากาน้ำร้อน และ ปากกาน้ำเงิน การค้นหาข่าวที่มีคำค้นที่ผู้ใช้ใส่เข้ามานี้จะไปค้นหาในฐานข้อมูลชื่อ NewsDB ในตารางชื่อ NewsTable ซึ่งมี field ดังต่อไปนี้

```
CREATE TABLE [dbo].[NewsTable] (
  [NewsID] [char] (28) COLLATE Thai_CI_AS NOT NULL,
  [Source] [char] (10) COLLATE Thai_CI_AS NULL,
  [Displaytime] [datetime] NULL,
  [Headline] [varchar] (400) COLLATE Thai_CI_AS NULL,
  [Story] [text] COLLATE Thai_CI_AS NULL,
  [Image] [image] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```



รูปที่ 3.3 NewsTable

- NewsID เก็บข้อมูลประเภท character ขนาด 28 ไบต์ ซึ่งใช้เก็บ uniqueID ของข่าวแต่ละชิ้น โดย unique ID นี้จะสร้างมาจาก SHA1 ของ Headline + displaytime โดยใช้ฟังก์ชัน ComputeHash()
- Source เก็บข้อมูลประเภท character ขนาด 10 ไบต์ ซึ่งใช้เก็บแหล่งข่าวว่าข่าวชิ้นนี้มาจากแหล่งข่าวไหน
- Displaytime เก็บข้อมูลประเภท datetime ใช้เก็บเวลาของข่าวแต่ละชิ้นว่าเป็นข่าวของวันที่เท่าไร
- Headline เก็บข้อมูลประเภท varcharacter ขนาด 400 ไบต์ ซึ่งใช้เก็บ headline ของข่าวแต่ละชิ้น
- Story เก็บข้อมูลประเภท text ใช้เก็บเนื้อหาของข่าวแต่ละชิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Image เก็บข้อมูลประเภท Image ใช้เก็บรูปภาพของข่าวในกรณีที่ข่าวนั้นมีรูปภาพ

โดยเซอร์วิสจะเรียกใช้งานฟังก์ชัน

Function Searching(ByVal userid As String, ByVal keywordstr As String, ByVal displaytime, ByVal direction) As ArrayList โดยรับพารามิเตอร์เป็น userid, keywordstr (คำที่ผู้ใช้ต้องการค้น), displaytime(เป็นเวลาที่ใช้เริ่มทำการค้นหา) และ direction(เป็นการบอกว่าต้องการค้นหาแบบไปข้างหน้าหรือย้อนหลัง)ซึ่งในการค้นหาในฐานข้อมูลจะเรียกใช้ store procedure ชื่อ sp_getheadline โดย store procedure นี้จะคืนค่า newsid, source, displaytime, headline กลับมาให้ในกรณีที่ค้นหาข่าวที่มีค่านั้นพบ และจะคืนค่า null ถ้าหากไม่พบข่าวที่มีค่าเหล่านั้น และฟังก์ชัน searching จะคืนค่าที่ได้มาจาก store procedure ส่งกลับไปให้ไคลเอนท์ในรูปแบบของ array of record

Retrieve Service

ทำหน้าที่ในการส่งเนื้อหาข่าวที่ผู้ใช้ต้องการอ่านกลับไปให้ไคลเอนท์ซึ่งมีกระบวนการดังนี้ รับ userid (ที่ได้มาจาก authentication service) และ newsID ของข่าวชิ้นที่ผู้ใช้ต้องการดูเนื้อหาข่าวมาจากไคลเอนท์ซึ่ง userid จะเป็น unique ID คือ แต่ละคนจะไม่ซ้ำกันเพื่อนำไปใช้ตรวจสอบว่า request มาจากผู้ที่มีสิทธิ์จริงหรือไม่ และ newsID ซึ่งเป็น uniqueID ของข่าวแต่ละชิ้นที่ได้มาจาก search service โดยจะนำ newsID ไปค้นหาจากฐานข้อมูล NewsDB ในตาราง NewsTable ซึ่งเป็นตารางเดียวกันกับที่ search service ใช้คือ

```
CREATE TABLE [dbo].[NewsTable] (
  [NewsID] [char] (36) COLLATE Thai_CI_AS NOT NULL,
  [Source] [char] (10) COLLATE Thai_CI_AS NULL,
  [Displaytime] [datetime] NULL,
  [Headline] [varchar] (400) COLLATE Thai_CI_AS NULL,
  [Story] [text] COLLATE Thai_CI_AS NULL,
  [Image] [image] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

โดยเซอร์วิส จะเรียกใช้งานฟังก์ชัน

Function `getstory(ByVal userid As String, ByVal Newsid As String) As String` ซึ่งรับพารามิเตอร์เป็น `userid` และ `newsid` ซึ่งในการค้นหาเนื้อหาข่าวในฐานข้อมูลจะเรียกใช้ `store procedure` ชื่อ `sp_getstory` โดย `store procedure` นี้จะคืนค่าเป็น เนื้อข่าวของข่าวชิ้นที่ผู้ใช้ต้องการดูกลับมาให้ในกรณีที่ค้นพบข่าวชิ้นนั้น ในฐานข้อมูล และในกรณีที่ไม่มีพบข่าวชิ้นนั้น ในฐานข้อมูลจะคืนค่าเป็น `null` และฟังก์ชัน `getstory` จะคืนค่า `story` ที่ได้กลับไปให้ไคลเอนท์ถ้า `store procedure` ไม่ได้คืนค่า `null` แต่ถ้า `store procedure` คืนค่ามาให้เป็น `null` ฟังก์ชัน `getstory` จะคืนค่า `-1` กลับไปให้ไคลเอนท์

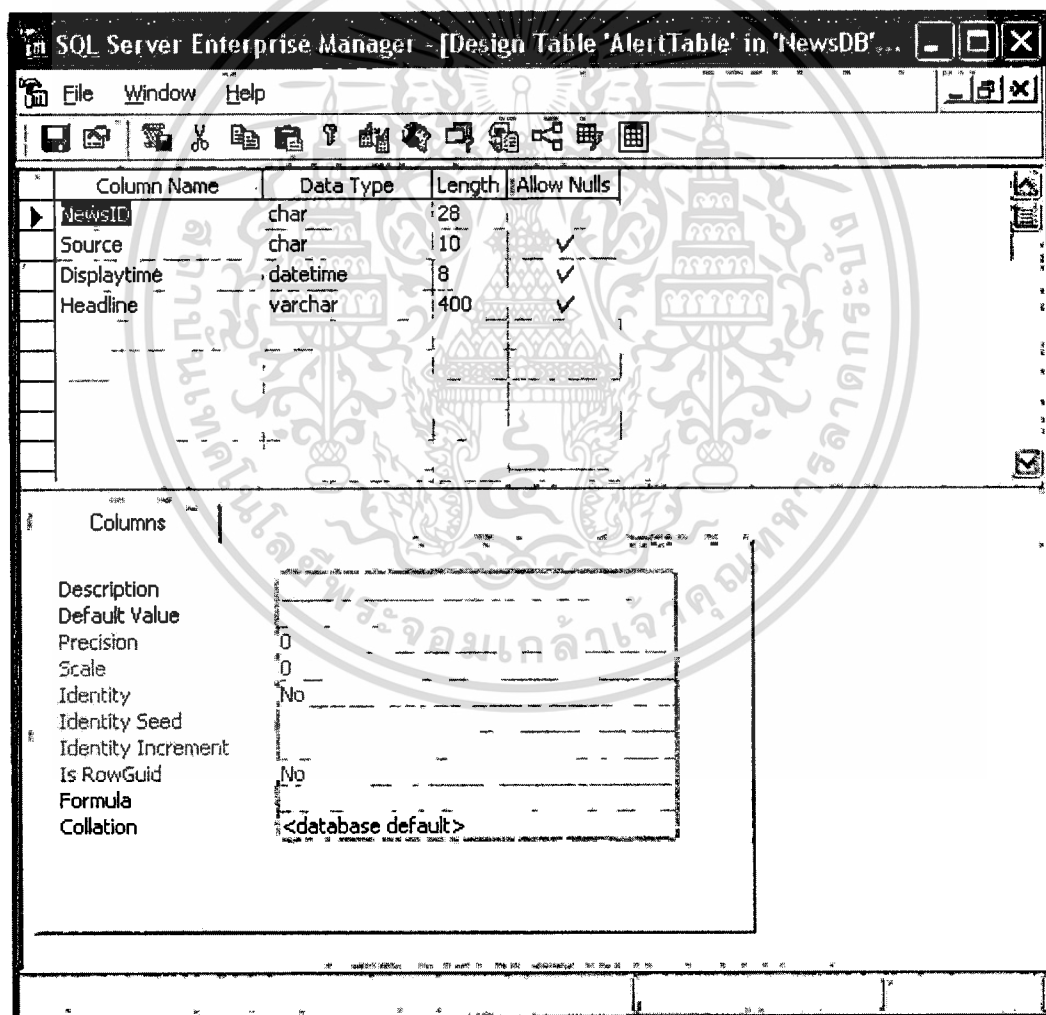
Alert Service

ทำหน้าที่คอยส่ง headline ของข่าวที่เข้ามาใหม่ไปให้ไคลเอนท์ซึ่งมีกระบวนการทำงานดังนี้ ไคลเอนท์จะต้องมีการขอ `sessionID` จาก server ก่อน โดยใช้ค่าของ `userid` หลังจากนั้นไคลเอนท์จะทำการขอ `alert record` (ข่าวใหม่) โดยเรียกใช้ ฟังก์ชัน

Function `GetAlertrecord(ByVal sessionid As String, ByVal keyword As String, ByVal updatetime As Date) As Object` ในการขอข่าวที่เข้ามาในระบบใหม่โดย server จะรับ `sessionID` ที่ไคลเอนท์ส่งมาซึ่งได้จาก `userid` ก่อนหน้านี้ , `Keyword` ซึ่งเป็นคำที่ผู้ใช้ได้ตั้งเป็นคำค้นไว้ซึ่ง `alert service` จะคอยตรวจสอบว่ามีข่าวที่ตรงกับคำค้นที่ตั้งไว้เข้ามาในระบบหรือไม่ถ้ามีก็จะส่งไปให้ไคลเอนท์ และ `updatetime` เป็นเวลาของข่าวชิ้นล่าสุดที่ตรงกับคำค้นที่ผู้ใช้ตั้งไว้ ที่ไคลเอนท์ได้รับ โดยวิธีการหาข่าวใหม่ที่เข้ามาในระบบนี้ใช้วิธีการเข้าไปตรวจสอบในฐานข้อมูลตามเวลาที่กำหนด (polling) โดยไคลเอนท์จะเป็นฝั่งที่จะคอยเข้าไป ตรวจสอบในฐานข้อมูลโดยใช้งานผ่านฟังก์ชัน `GetAlertRecord` ซึ่ง `GetAlertRecord` จะเข้าไปตรวจสอบข่าวใหม่ในฐานข้อมูลได้นั้นโดยติดต่อผ่านฟังก์ชัน Function `Alerting(ByVal keyword As String, ByVal updatetime As Date) As ArrayList` โดยฟังก์ชันนี้จะเข้าไปตรวจสอบในฐานข้อมูลด้วย `store procedure` ที่ชื่อว่า `sp_alert` ซึ่ง `sp_alert` จะเข้าไปตรวจสอบข่าวใหม่จากฐานข้อมูล `NewsDB` ในตารางชื่อ `AlertTable` ซึ่งตาราง `AlertTable` นี้จะมีเฉพาะข่าวใหม่ที่เข้าในระบบภายใน 1 ชั่วโมงเท่านั้นเพื่อที่จะได้ลดเวลาในการค้นหาข่าวใหม่จากฐานข้อมูลหลังของระบบซึ่งจะเสียเวลาถ้าหากมีข้อมูลเป็นจำนวนมาก โดยใช้ `Trigger` ในการเก็บข้อมูลลงใน `AlertTable` นี้ด้วย `Trigger` ที่ชื่อว่า `AlertNews` โดย `trigger` นี้จะทำการ `Insert` ข้อมูลใหม่ลงใน `AlertTable` เมื่อมีข้อมูลใหม่ถูก `Insert` เข้ามาใน `NewsTable` โดยการค้นหาข่าวใหม่จะใช้การค้นหาข่าวใน `AlertTable` ที่มี field `displaytime` ใหม่กว่า `updatetime` ที่ไคลเอนท์ส่งมา `store procedure` นี้จะคืนค่า `newsID` , `source` , `displaytime` , `headline` ถ้ามีข่าวที่มี `displaytime` ใหม่กว่า `updatetime` หรือคืนค่า `null` ให้ในกรณีที่ไม่มีข่าว `headline` ถ้ามีข่าวที่มี `displaytime` ใหม่

กว่า updatetime และฟังก์ชัน GetAlertRecord จะคืนค่าที่ได้เหล่านั้นไปให้ไคลเอนท์ โดย AlertTable ประกอบด้วย field ดังนี้

```
CREATE TABLE [dbo].[AlertTable] (
    [NewsID] [char] (28) COLLATE Thai_CI_AS NOT NULL ,
    [Source] [char] (10) COLLATE Thai_CI_AS NULL ,
    [Displaytime] [datetime] NULL ,
    [Headline] [varchar] (400) COLLATE Thai_CI_AS NULL
) ON [PRIMARY]
```



รูปที่ 3.4 AlertTable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- NewsID เก็บข้อมูลประเภท character ขนาด 36 ไบต์ ซึ่งใช้เก็บ uniqueID ของข่าวแต่ละชิ้น โดย unique ID นี้จะสร้างมาจาก SHA1 ของ Headline + displaytime โดยใช้ฟังก์ชัน ComputeHash()
- Source เก็บข้อมูลประเภท character ขนาด 10 ไบต์ ซึ่งใช้เก็บแหล่งข่าวว่าข่าวชิ้นนี้มาจากแหล่งข่าวไหน
- Displaytime เก็บข้อมูลประเภท datetime ใช้เก็บเวลาของข่าวแต่ละชิ้นว่าเป็นข่าวของวันที่เท่าไร
- Headline เก็บข้อมูลประเภท varcharter ขนาด 400 ไบต์ ซึ่งใช้เก็บ headline ของข่าวแต่ละชิ้น

Crawler application

ทำหน้าที่ดึงเว็บไซต์ข่าวจากอินเทอร์เน็ตมาเก็บลงในฐานข้อมูลในตารางชื่อ NewsTable

```
CREATE TABLE [dbo].[NewsTable] (
  [NewsID] [char] (28) COLLATE Thai_CI_AS NOT NULL,
  [Source] [char] (10) COLLATE Thai_CI_AS NULL,
  [Displaytime] [datetime] NULL,
  [Headline] [varchar] (400) COLLATE Thai_CI_AS NULL,
  [Story].[text] COLLATE Thai_CI_AS NULL,
  [Image] [image] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

ซึ่งเป็นตารางที่ให้ search service ค้นหาข้อมูล และให้ retrieve service ค้นหาเนื้อหาข่าวที่ผู้ใช้ต้องการ โดยประกอบไปด้วยฟังก์ชันสำคัญ 2 ส่วนคือ

Function GetInfo(ByVal range as date) as String ใช้ในการดึงข้อมูลที่ต้องการจากเว็บไซต์เข้าไปเก็บในฐานข้อมูล โดยเรียกใช้ function GetPage(ByVal URL As String, _

ByVal myCookie As String, _

ByRef htmlText As StringBuilder, _

Optional ByVal Timeout As Integer = 300, _

Optional ByVal proxyAddr As String = Nothing, _

Optional ByVal proxyPort As Integer = 0) As Integer ใช้สำหรับติดต่อกับ

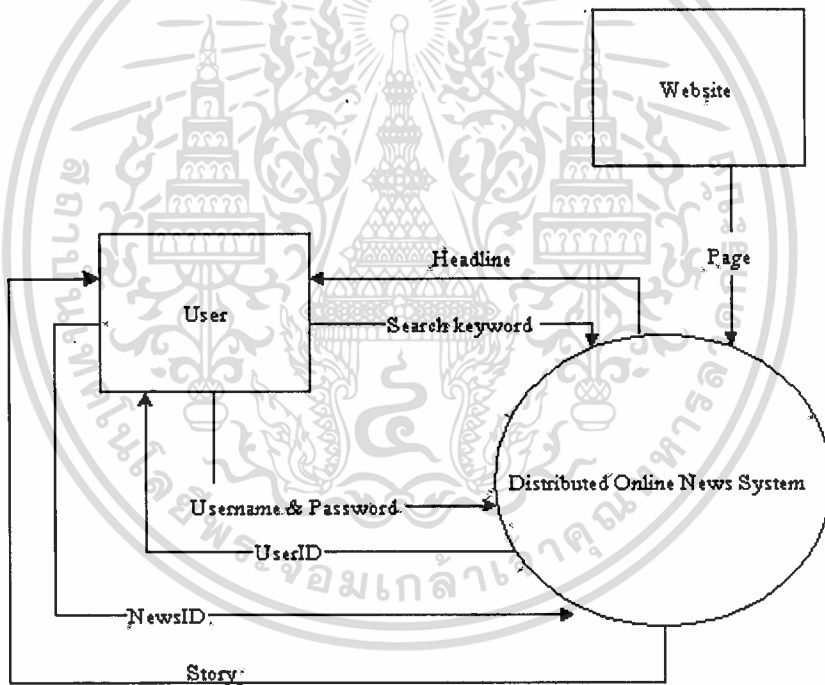
เว็บไซต์ที่ต้องการดึงมาและทำการโหลดหน้าที่ต้องการมาเก็บไว้ในตัวแปร htmltext หลังจากนั้นจะ

เรียกใช้ function ParsePage(ByVal htmlpage As String, ByVal begintax As String, ByVal endtax As String) As String เพื่อทำการ parse string ที่ได้ออกมาเป็นข้อมูลที่ต้องการ และนำข้อมูลที่ต้องการเก็บลงในฐานข้อมูล โดยใช้ function

```
Inserting(ByVal newsid As String, ByVal source As String, _
ByVal displaytime As Date, ByVal headline As String, _
ByVal story As String, Optional ByVal image As Byte() = Nothing)
```

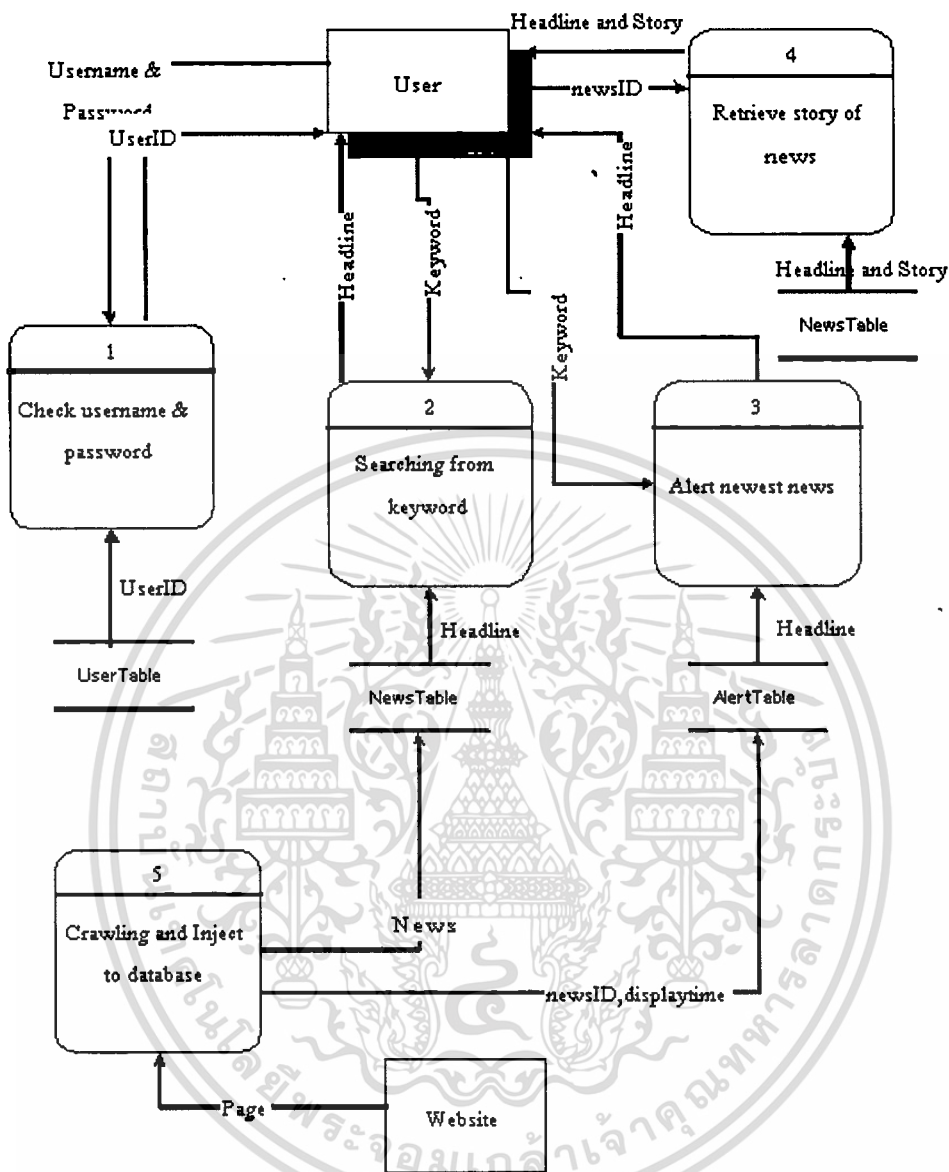
3.4 Data Flow Diagram

การทำงานในขั้นตอนต่างๆของแต่ละ service สามารถอธิบายด้วย Data Flow Diagram ได้ดังนี้



รูปที่ 3.5 Context Diagram ของ Distributed Online News System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Data Flow Diagram Level1 ของ News online System

จากรูปจะเห็นได้ว่าระบบมีโปรเซสสำคัญอยู่ 5 ส่วนซึ่งประกอบด้วย

- Check username and password เป็นส่วนที่อยู่ใน authentication service จะทำหน้าที่ตรวจสอบ username และ password ที่ส่งมาจาก Client application กับฐานข้อมูลว่าถูกต้องหรือไม่แล้วคืนค่าเป็น userid (ถ้าพบในฐานข้อมูล) หรือ error code กลับไปให้ Client application

- Searching from keyword เป็นส่วนที่อยู่ใน search service ซึ่งจะรับ keyword ที่ผู้ใช้ต้องการค้นหาจาก Client application มาค้นหาในฐานข้อมูลแล้วคืนค่าเป็น headline และ newsID ของข่าวที่มีค่าที่ต้องการค้นหาในข่าวกลับไปให้ Client application แต่ถ้าไม่พบก็จะคืนค่าไปเป็น null เพื่อให้ client application ไปบอกผู้ใช้ว่าไม่พบข่าวที่มีคำค้นที่ผู้ใช้ใส่เข้ามา
- Alert newest new เป็นส่วนที่อยู่ใน Alert service จะรับ keyword ที่ผู้ใช้ต้องการค้นหาที่ search service ได้รับมาเก็บเป็น session ของผู้ใช้แต่ละคนไว้เพื่อที่จะได้ใช้ในการส่งข่าวที่ตรงกับคำค้นที่เข้ามาในระบบใหม่ไปให้ผู้ใช้แต่ละคนได้ถูกต้อง
- Retrieve story of news จะอยู่ในส่วนของ Retrieve service โดย process นี้จะรับ newsID มาจาก client application เมื่อผู้ใช้คลิกดูข่าว แล้วจะเอา newsID นั้นไปค้นหาในฐานข้อมูลว่าเป็นข่าวชนิดใดเพื่อส่งเนื้อหาข่าวที่มี newsID ตรงกันกลับไปให้ client application
- Crawling and inject to database อยู่ในส่วนของ Crawler application ซึ่งเป็น process ที่ไม่ต้องรับข้อมูลมาจากผู้ใช้เป็นเสมือน agent ที่คอยทำหน้าที่ดึงข่าวจากเว็บไซต์เข้ามาในระบบ

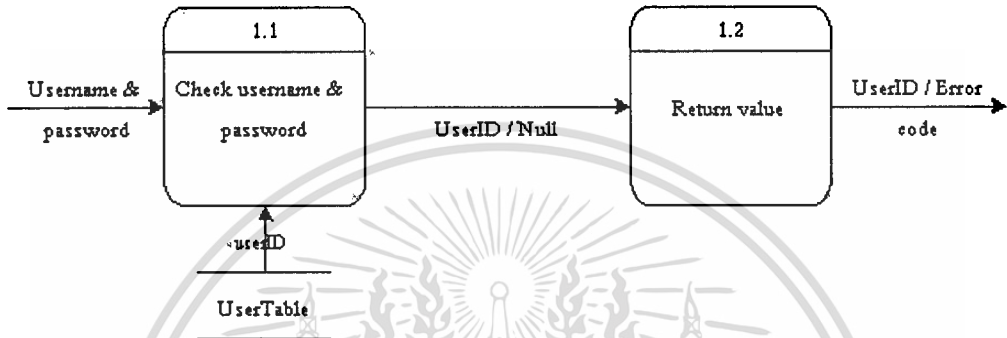
จากโปรเซสทั้งหมดในรูปที่ 3.6 นั้นแต่ละโปรเซสสามารถแบ่งออกเป็นขั้นตอนย่อยๆ ได้อีกดังต่อไปนี้

จากรูปที่ 3.7 จะมีโปรเซสย่อยๆ อยู่ 5 โปรเซสคือ

- User Interface ซึ่งทำหน้าที่รับคำสั่งจากผู้ใช้ และแสดงผลพิกซ์กลับไปให้ผู้ใช้ที่หน้าจอ
- GetUserID จะรับ username และ password จากที่ผู้ใช้ป้อนเข้ามาทาง user interface เพื่อที่จะส่งไปให้ authentication service ตรวจสอบกับฐานข้อมูลว่าผู้ใช้คนนั้นมีสิทธิ์เข้ามาใช้หรือไม่ถ้ามีสิทธิ์ก็จะได้ userID กลับมา
- Searching จะรับคำค้น (keyword) มาจาก user interface เพื่อส่งไปให้ search service ตรวจสอบกับฐานข้อมูลว่ามีข่าวที่ตรงกับคำค้นนั้นหรือไม่ ซึ่งจะได้ผลลัพธ์จากการค้นหากลับมาจาก server
- GetAlertRecord จะรับคำค้น (keyword) มาจาก user interface ซึ่งจะเป็นคำเดียวกับที่ผู้ใช้ต้องการค้น พร้อมกับ userID ของผู้ใช้ เพื่อส่งไปให้ Alert service เพื่อที่จะได้ข่าวที่เข้ามาในระบบใหม่ที่ตรงกับคำค้นนั้นกลับมา

- Getstory จะรับ newsID จาก user interface เมื่อผู้ใช้ double click เพื่อต้องการดูเนื้อหาของ headline ขึ้นนั้นเพื่อส่งไปให้ retrieve service ค้นหาฐานข้อมูลแล้วคืนเนื้อหาว่าที่ต้องการกลับมาให้

Authentication Service (Check username and password)

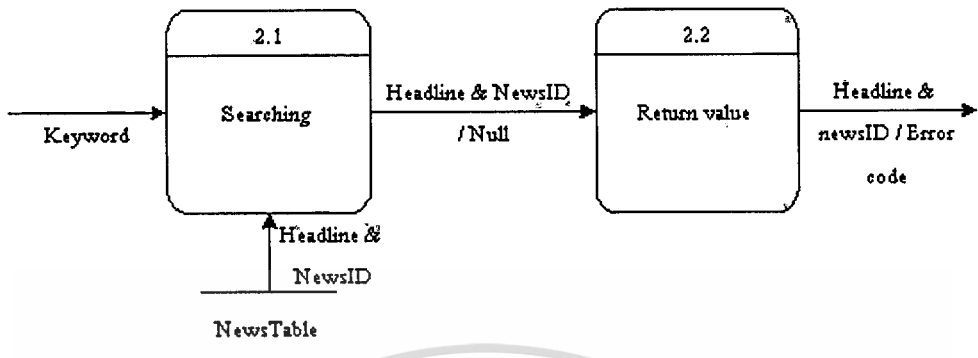


รูปที่ 3.7 Data Flow Diagram Level2 ของ Authentication service

จากรูปที่ 3.7 จะประกอบด้วยโปรเซสย่อยอยู่ 2 โปรเซสคือ

- Check username & password ซึ่งจะรับ username และ password มาจากฝั่งไคลเอนท์เพื่อตรวจสอบกับฐานข้อมูลในตารางชื่อ UserTable ว่ามีหรือไม่แล้วคืนค่า userID
- Return value จะทำหน้าที่คืนค่าที่ได้จาก process ที่ตรวจสอบข้อมูลในฐานข้อมูลกลับไปให้ฝั่งไคลเอนท์ซึ่งถ้าถูกต้องจะคืนค่าไปเป็น userID แต่ถ้าไม่พบจะคืนค่าเป็น error code (-1) กลับไป

Search service (Searching from keyword)

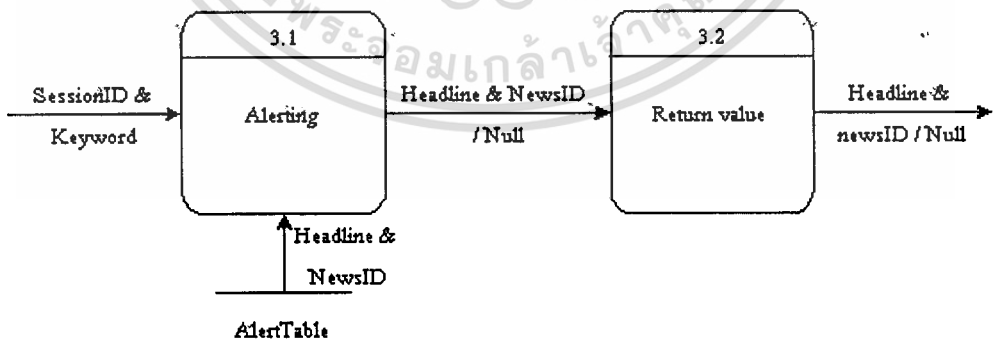


รูปที่ 3.8 Data Flow Diagram Level2 ของ Search service

จากรูปที่ 3.8 จะประกอบด้วย โปรเซสย่อยอยู่ 2 โปรเซส คือ

- Searching ซึ่งจะรับ keyword มาจากฝั่ง ไคลเอนท์เพื่อที่จะนำมาตรวจสอบกับฐานข้อมูลในตารางชื่อ NewsTable ว่ามีข่าวที่ตรงกับ keyword นั้นหรือไม่แล้วคืนค่าเป็น record set ของข่าว 30 ชิ้นแรกจากข่าวใหม่ที่สุด
- Return value ทำหน้าที่คืนค่า recordset ที่ได้กลับไปเป็น array list ของ headline, newsID กลับไปให้ไคลเอนท์

Alert service (Alert newest new)

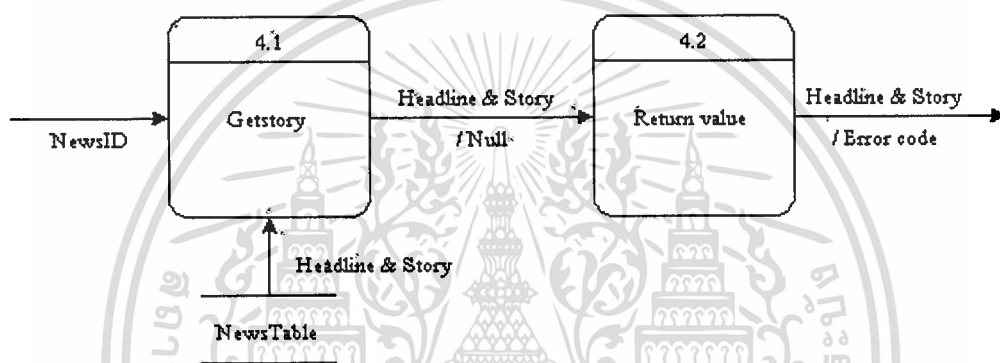


รูปที่ 3.9 Data Flow Diagram Level2 ของ Alert service

จากรูปที่ 3.9 จะประกอบด้วยโปรเซสย่อยอยู่ 2 โปรเซสคือ

- Alerting จะรับคำค้นและ userID มาจากฝั่งไคลเอนท์เพื่อที่จะสร้างเป็น session ของผู้ใช้แต่ละคนไว้ และเข้าไปตรวจสอบในฐานข้อมูลในตารางที่ชื่อ AlertTable ว่ามีข่าวที่ตรงกลับคำค้นหรือไม่แล้วคืนค่าออกมาเป็น recordset ของข่าวที่พบ
- Return value จะเก็บ recordset ที่ได้ลงใน array list ของ session ผู้ใช้แต่ละคนแล้วคืนกลับไปให้ไคลเอนท์

Retrieve service (Retrieve story of news)

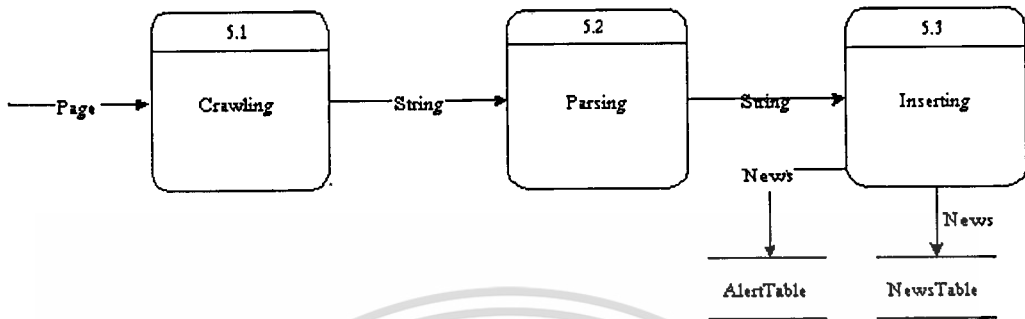


รูปที่ 3.10 Data Flow Diagram Level2 ของ Retrieve service

จากรูปที่ 3.10 จะประกอบด้วยโปรเซสย่อยอยู่ 2 โปรเซสคือ

- Getstory ซึ่งจะรับ newsID มาจากฝั่งไคลเอนท์และนำไปตรวจสอบในฐานข้อมูลในตารางชื่อ NewsTable เพื่อหาเนื้อข่าวที่มี newsID เท่ากับ newsID ที่ได้รับมาแล้วคืนค่าออกมาเป็น recordset ของข่าวที่มี newsID นั้น
- Return value ทำหน้าที่คืนค่า recordset ของข่าวที่ได้ไปเป็น string กลับไปให้ไคลเอนท์

Crawler application (Crawling and Inject to database)



รูปที่ 3.11 Data Flow Diagram Level2 ของ Crawler application

จากรูปที่ 3.11 ประกอบด้วยโปรเซสย่อยอยู่ 2 โปรเซสคือ

- Crawling ทำหน้าที่ไหลคหน้าเว็บไซต์เพื่อที่จะทำการเก็บลงฐานข้อมูล
- Parsing ทำหน้าที่นำเอา string ที่ได้มาจาก process Crawling มากรองเอาเฉพาะข้อมูลที่ต้องการเก็บลงฐานข้อมูลในตารางชื่อ NewsTable
- Inserting จะทำการสร้าง newsid ให้กับข่าวนั้นก่อนที่จะเก็บลงฐานข้อมูล โดย newsid จะถูกสร้างมาโดยการหาค่า hash ของ headline + วันที่ของข่าวนั้นๆ
- ก่อนที่จะเก็บข้อมูลลงฐานข้อมูล inserting จะทำการการตรวจสอบข้อมูลใน database ก่อนว่าข่าวนั้นมีอยู่ในฐานข้อมูลแล้วหรือไม่ได้ถ้ามีแล้วก็จะไม่เก็บเข้าฐานข้อมูล

บทที่ 4

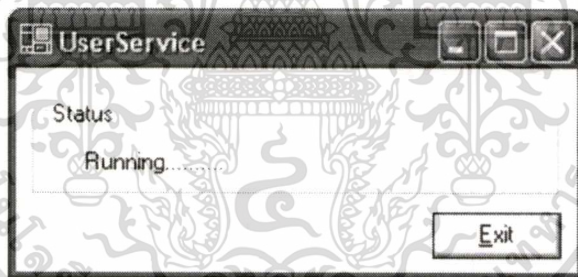
การติดตั้งและใช้งานระบบ

4.1 Environment และการใช้งานโปรแกรมฝั่ง server

- Service แต่ละตัวสามารถที่จะติดตั้งอยู่บนเครื่องคนละเครื่องกันได้ โดยแต่ละเครื่องที่จะติดตั้งจะต้องมี .NET Framework 1.0
- ฐานข้อมูลที่เป็น SQL 2000 server โดยโครงสร้างของ table, store procedure, trigger ที่ใช้งานสามารถดูได้ในภาคผนวก

4.1.1 User service

ทำหน้าที่ตรวจสอบ username และ password ของผู้ใช้โดยจะมีหน้าจอดังนี้



รูปที่ 4.1 User service

โดยจะเริ่มทำงานทันทีเมื่อถูกเรียกขึ้นมาใช้งานและจะหยุดการให้บริการเมื่อกดปุ่ม Exit ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

- เมื่อโปรแกรมเริ่มทำงานจะเปิด TCP port 10001 รอไว้ให้ไคลเอนท์เข้ามาติดต่อด้วย โดย port ที่เปิดไว้สำหรับรอรับบริการและฐานข้อมูลที่เซิร์ฟเวอร์จะต้องติดต่อด้วยสามารถเปลี่ยนได้จาก ไฟล์ชื่อ userservice.exe.config

- เมื่อได้รับ username และ password จากไคลเอนท์ก็จะไปเรียกใช้ Verifyuser class ในการตรวจสอบ
- เมื่อผ่านการตรวจสอบแล้วถูกต้องจะคืนค่า userid ซึ่งแต่ละคนจะมีค่าไม่ซ้ำกันกลับไปให้ไคลเอนท์
- เมื่อตรวจสอบกับฐานข้อมูลเสร็จก็ตัด connection กับฐานข้อมูล
- ถ้าคปุ่ม Exit port ที่ถูกเปิดไว้สำหรับรอรับบริการก็จะถูกปิดไปด้วย

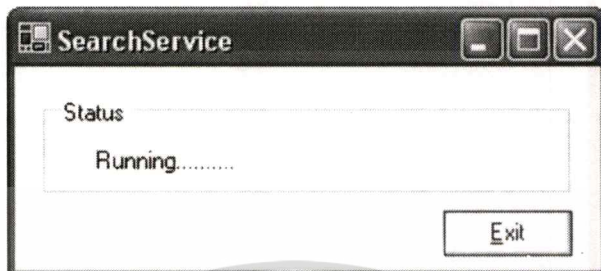
ตัวอย่าง configuration file ของ User service

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<appSettings>
  <add key="Server" value="localhost" />
  <add key="DatabaseName" value="NewsDB" />
  <add key="User" value="newsclient" />
  <add key="Password" value="123" />
</appSettings>
  <system.runtime.remoting>
    <application name="UserService">
      <service>
        <wellknown mode="Singleton" type="Verifyuser.Verifyuser,
          Verifyuser" objectUri="Verifyuser"/>
      </service>
    </application>
  </system.runtime.remoting>
</configuration>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 Search service

ทำหน้าที่ค้นหาข่าวที่ตรงกับคำค้นของผู้ใช้ระบบ โดยมีหน้าจอดังนี้



รูปที่ 4.2 Search service

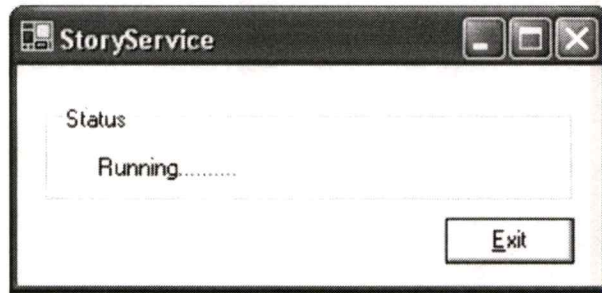
โดยจะเริ่มทำงานทันทีเมื่อถูกเรียกขึ้นมาใช้งานและจะหยุดการให้บริการเมื่อกดปุ่ม Exit ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

- เมื่อโปรแกรมเริ่มทำงานจะเปิด TCP port 10002 รอไว้ให้ไคลเอนต์เข้ามาติดต่อด้วย โดย port ที่เปิดไว้สำหรับรอรับบริการและฐานข้อมูลที่เซิร์ฟเวอร์จะต้องติดต่อด้วยสามารถเปลี่ยนได้จาก ไฟล์ชื่อ searchservice.exe.config
- จะรับ userid จากไคลเอนต์เพื่อใช้ตรวจสอบว่าผู้ที่ต้องการค้นหาข่าวผ่านการตรวจสอบมาแล้วว่ามีสิทธิ์เข้าใช้งานระบบ และรับคำค้นของผู้ใช้ โดยจะเรียกใช้ผ่าน Searchnews class
- คำค้นที่ส่งเข้ามาถ้าเป็นคำที่มีเว้นวรรคจะเอามา and แบ่งเป็น 2 คำแต่ถ้ามี "" ปิดหัวและท้าย จะนำเอาคำที่อยู่ในเครื่องหมาย "" มาค้นหาทั้งคำ โดยจะใช้การค้นหาแบบ pattern matching
- Service นี้จะคืนค่าเป็น recordset กลับไปให้ไคลเอนต์ โดยจะนำ recordset ที่ได้มาเก็บไว้ใน arraylist ของ Datarecord ซึ่งเป็น object ที่มี field ดังนี้
[no, newsID, source, displaytime, headline]
- เมื่อตรวจสอบกับฐานข้อมูลเสร็จก็ตัด connection กับฐานข้อมูล
- ถ้ากดปุ่ม Exit port ที่ถูกเปิดไว้สำหรับรอรับบริการก็จะถูกปิดไปด้วย

4.1.3 Story service

ทำหน้าที่ส่งเนื้อหาข่าวกลับไปให้ไคลเอนต์ โดยมีหน้าจอดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



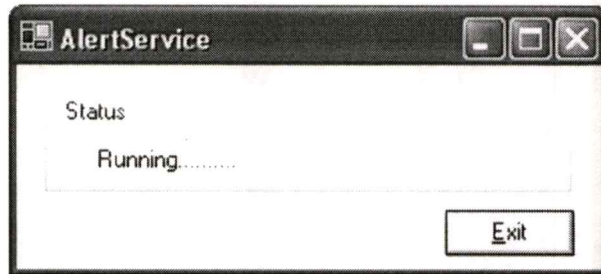
รูปที่ 4.3 Story service

โดยจะเริ่มทำงานทันทีเมื่อถูกเรียกขึ้นมาใช้งานและจะหยุดการให้บริการเมื่อกดปุ่ม Exit ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

- เมื่อโปรแกรมเริ่มทำงานจะเปิด TCP port 10003 รอไว้ให้ไคลเอนท์เข้ามาติดต่อด้วย โดย port ที่เปิดไว้ สำหรับรอรับบริการและฐานข้อมูลที่เซิร์ฟเวอร์จะต้องติดต่อด้วยสามารถเปลี่ยนได้จาก ไฟล์ชื่อ storyservice.exe.config
- ต่อมาจะรับ userid จากไคลเอนท์เพื่อใช้ตรวจสอบว่าผู้ที่ต้องการค้นหาข่าวผ่านการตรวจสอบมาแล้วว่ามีสิทธิ์เข้าใช้งานระบบ และรับ newsid ของ headline ที่ผู้ใช้ต้องการดูเนื้อหาข่าว โดยเรียกใช้งานผ่าน RetrieveStory class
- ถ้าพบ newsID ตรงกับที่ไคลเอนท์ส่งมาจะคืนค่าเป็นเนื้อหาข่าวที่ต้องการกลับไปเป็น string ส่งกลับไปให้ไคลเอนท์
- ถ้าไม่พบ newsID ที่ตรงกับที่ส่งมาจะคืนค่ากลับไปเป็น “No Story were found”
- เมื่อตรวจสอบกับฐานข้อมูลเสร็จก็ตัด connection กับฐานข้อมูล
- ถ้ากดปุ่ม Exit port ที่ถูกเปิดไว้สำหรับรอรับบริการก็จะถูกปิดไปด้วย

4.1.4 Alert service

ทำหน้าที่คอยตรวจสอบกับฐานข้อมูลว่ามีข่าวใหม่เข้ามาในระบบหรือไม่ถ้ามีก็จะส่งข่าวใหม่นั้นกลับมาให้ไคลเอนท์โดยมีหน้าจอดังนี้



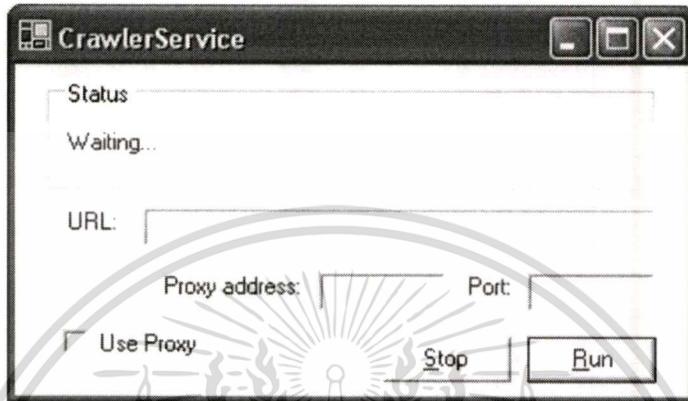
รูปที่ 4.4 Alert service

โดยจะเริ่มทำงานทันทีเมื่อถูกเรียกขึ้นมาใช้งานและจะหยุดการให้บริการเมื่อกดปุ่ม Exit ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

- เมื่อ โปรแกรมเริ่มทำงานจะเปิด TCP port 10004 รอไว้ให้ไคลเอนท์เข้ามาติดต่อด้วย โดย port ที่เปิดไว้ สำหรับรอรับบริการและฐานข้อมูลที่เซิร์ฟเวอร์จะต้องติดต่อด้วยสามารถ เปลี่ยนได้จาก ไฟล์ชื่อ alertservice.exe.config
- หลังจากนั้นเมื่อ ไคลเอนท์มีการค้นหาข้อมูล userid, keyword, และเวลาที่กำลังค้นหาข้อมูล อยู่นั้นก็จะถูกส่งมาให้ alert service ด้วย
- โดยจะนำ userid ไปตรวจสอบว่าผู้ที่ต้องการค้นหาข่าวผ่านการตรวจสอบมาแล้วว่ามีสิทธิ์ เข้าใช้งานระบบ และนำ keyword และ เวลาที่ผู้ใช้กดปุ่มค้นหา มาใช้ในการกรองว่าข่าวที่ เข้ามาใหม่นั้นมีค่าค้นที่ตรงกับที่ผู้ใช้กำลังต้องการ(ใช้การตัดคำเหมือน search service) และเป็นเวลาที่ใหม่กว่าตอนที่ผู้ใช้งานเริ่มค้นหาข่าว โดยจะใช้งานผ่าน Alertnews class
- โดยไคลเอนท์จะเรียกใช้งานผ่าน method GetAlertrecord โดยจะคอยไปตรวจดูในฐานข้อมูลว่ามีข่าวใหม่ที่ตรงกับคำค้นหรือไม่
- เมื่อค้นพบข่าวที่ตรงแล้วก็จะเอาข่าวเหล่านั้นมาเก็บไว้ใน arraylist ของผู้ใช้แต่ละคน และ ส่งกลับไปให้ ไคลเอนท์
- รอบเวลาที่ใช้ในการเข้าไปค้นหาในฐานข้อมูล (Interval time) ไคลเอนท์จะเป็นคนกำหนด
- เมื่อตรวจสอบกับฐานข้อมูลเสร็จก็ตัด connection กับฐานข้อมูล
- ถ้ากดปุ่ม Exit port ที่ถูกเปิดไว้สำหรับรอรับบริการก็จะถูกปิดไปด้วย

4.1.5 Crawler

ทำหน้าที่คอยเข้าไปดึงข่าวจากเว็บไซต์ในเครือข่ายอินเทอร์เน็ตตามเวลาที่กำหนดและเอามาเก็บลงในฐานข้อมูลเพื่อให้ผู้ใช้งานสามารถค้นหาข่าวที่ต้องการได้ โดยมีหน้าจอดังนี้



รูปที่ 4.5 Crawler

โดยจะเริ่มทำงานเมื่อมีการกดปุ่ม RUN และจะหยุดการให้บริการเมื่อกดปุ่ม STOP ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

- เมื่อมีการกดปุ่ม run โปรแกรมจะเข้าไป connect กับเว็บไซต์เป้าหมายแล้วทำการดึงหน้าของเว็บไซต์นั้นทั้งหน้าลงมาเก็บไว้เป็น string โดยใช้ CSETINet class และเรียก method Getpage ในการดึงข้อมูลมา
- หลังจากนั้นจะทำการ parse page ที่ได้มานี้ออกโดยจะเอาเฉพาะส่วนที่ต้องการจะเก็บลงฐานข้อมูลในตาราง newstable (newsID, source, displaytime, headline, story, image) เท่านั้นมาเก็บไว้ในตัวแปรต่างๆ
- เมื่อแยกได้ข้อมูลที่ต้องการแล้วก็จะเรียกใช้งาน method Inserting เพื่อใช้เก็บข้อมูลที่ดึงลงในฐานข้อมูลโดยผ่าน store procedure ชื่อ sp_insertNewstable
- โดยโปรแกรมจะทำการ ดึงข้อมูลในเว็บไซต์นั้นที่เป็นข่าวภายในวันนั้นเท่านั้นซึ่งจะทำงานไปเรื่อยๆจนกว่าจะพบข่าวที่เก่ากว่าวันนั้น
- หลังจากพบข่าวที่เก่ากว่าวันนั้นแล้วก็จะหยุดดึงและรอไปจนถึงเวลาที่ตั้งไว้ให้ดึงในรอบถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

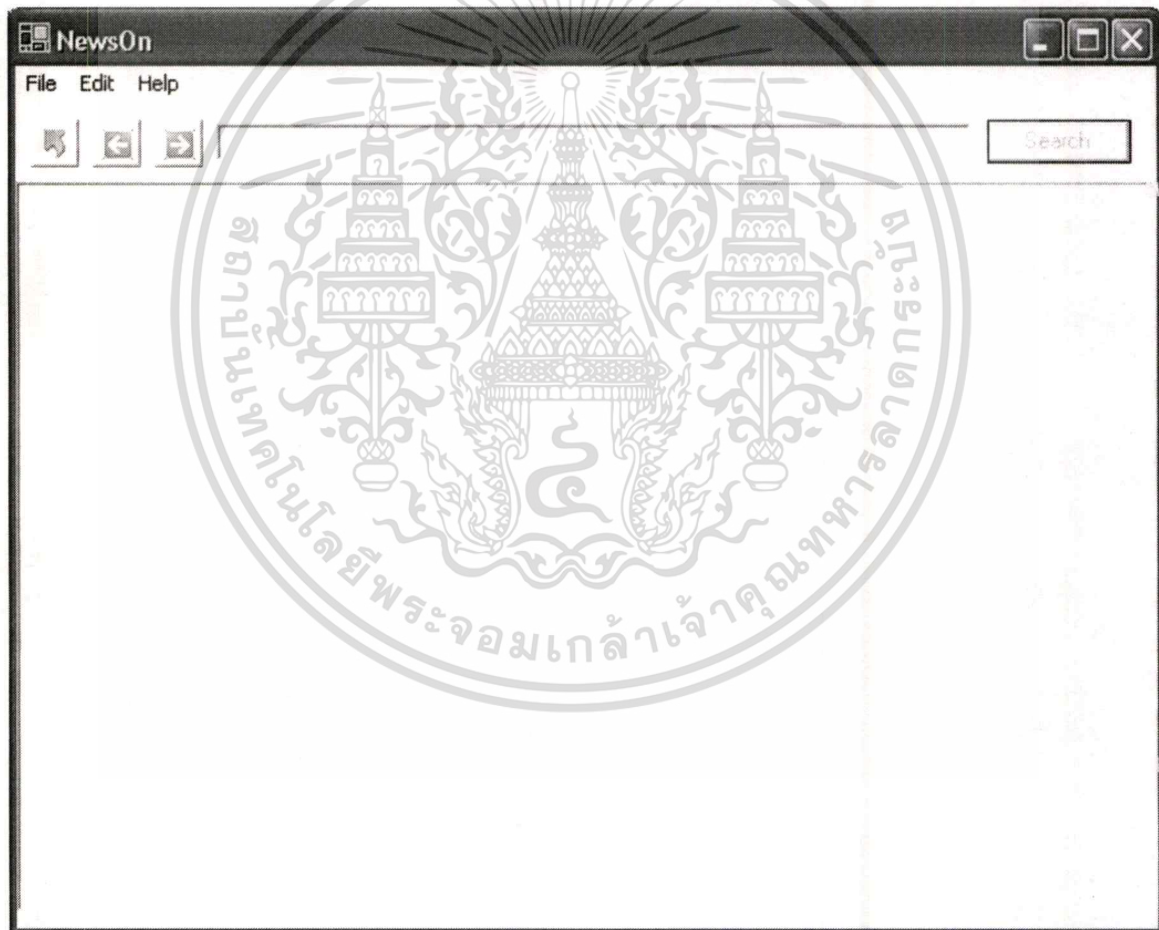
4.2 Environment และการใช้งานโปรแกรมฝั่ง client

- โปรแกรมบนฝั่ง client จะติดตั้งอยู่บนเครื่องคอมพิวเตอร์ของผู้ใช้งาน โดยบนเครื่องของผู้ใช้จะต้องมี .NET Framework 1.0

4.2.1 NewsOn

เป็นไคลเอนต์แอปพลิเคชัน ให้ผู้ใช้ ใช้ในการสืบค้นข้อมูลข่าวสารต่างๆ ซึ่งมีหน้าจอดังแสดงในรูปที่

4.6



รูปที่ 4.6 NewsOn

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

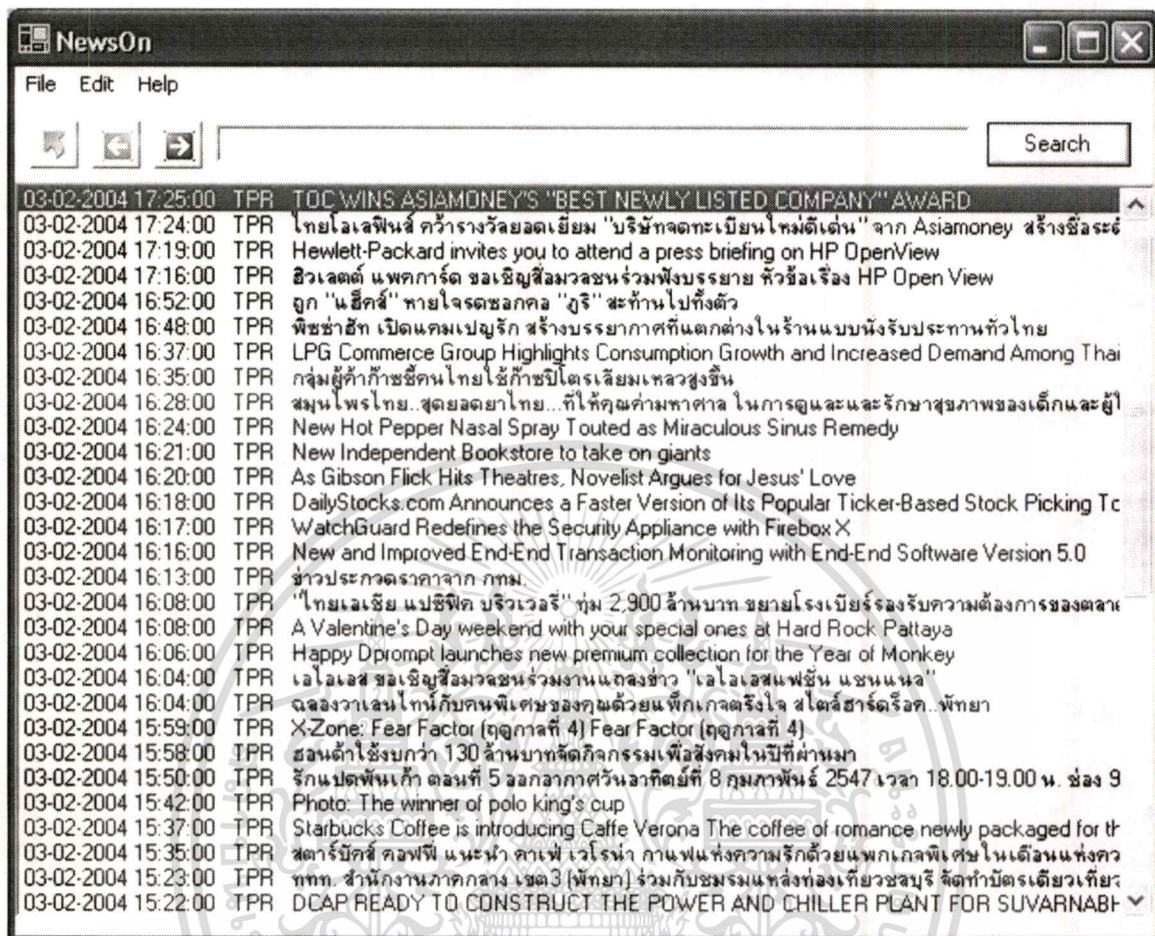
โดยมีขั้นตอนการทำงานดังนี้

- เมื่อเปิดโปรแกรม NewsOn ขึ้นมาโปรแกรมจะไปอ่านค่า configuration ที่ใช้สำหรับติดต่อกับ service ต่างๆ ว่า IP address อะไร ใช้ port อะไรเพื่อเข้ามาใช้งานระบบในไฟล์ชื่อ newson.exe.config
- หลังจากนั้นจะต้องทำการติดต่อกับเซิร์ฟเวอร์โดยเลือกเมนู File -> connect โปรแกรมจะ popup หน้าจอขึ้นมาให้ใส่ username และ password สำหรับใช้งานระบบซึ่งผู้ใช้ต้องใส่ username และ password ก่อนทุกครั้งจึงจะเข้าใช้งานระบบได้



รูปที่ 4.7 Login Screen

- โดยจะนำ username และ password ส่งให้ user service เป็นคน ตรวจสอบ เมื่อตรวจสอบผ่านแล้วก็จะส่ง userID กลับมาให้เพื่อที่จะได้ใช้ในการค้นหาข่าวต่อไปได้
- โดย default แล้วเมื่อ login ผ่าน โปรแกรมจะทำการส่งคำค้นว่า “/all” ส่งไปให้ search service ทำการค้นหาซึ่งคำค้นนี้เป็นคำพิเศษสำหรับแสดงข้อมูลทั้งหมดที่มีอยู่ในฐานข้อมูล ดังรูปที่ 4.8 และจะส่ง “/all” ไปให้ alert service เพื่อเป็นการบอกให้ alert service ทราบว่าเมื่อมีข่าวใหม่เข้ามาในระบบให้ส่งข่าวนั้นกลับมาด้วย



รูปที่ 4.8 First Screen

- ผู้ใช้งานสามารถค้นหาข่าวที่ต้องการได้โดยใส่คำค้นลงในช่อง query แล้วกด search โดยคำค้นนั้นถ้าเป็นคำที่มี เว้นวรรค ระบบจะตีความเป็นการนำคำมา and กัน คือข่าวที่ได้จะต้องมีทั้งสองคำในข่าว หรือถ้าต้องการค้นหาคำที่ต้องเว้นวรรคให้ใส่ "" ปิดหัวและท้ายคำค้นนั้น จะเป็นการบอกให้ระบบหาทั้งคำนั้นเลย
- ผลลัพธ์ที่ได้จะออกมาเป็น record ที่ประกอบด้วย วันที่ของข่าวชิ้นนั้น, แหล่งข่าว และ หัวข้อข่าวที่ค้นพบ มาแสดงบนหน้าจอ ดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NewsOn

File Edit Help

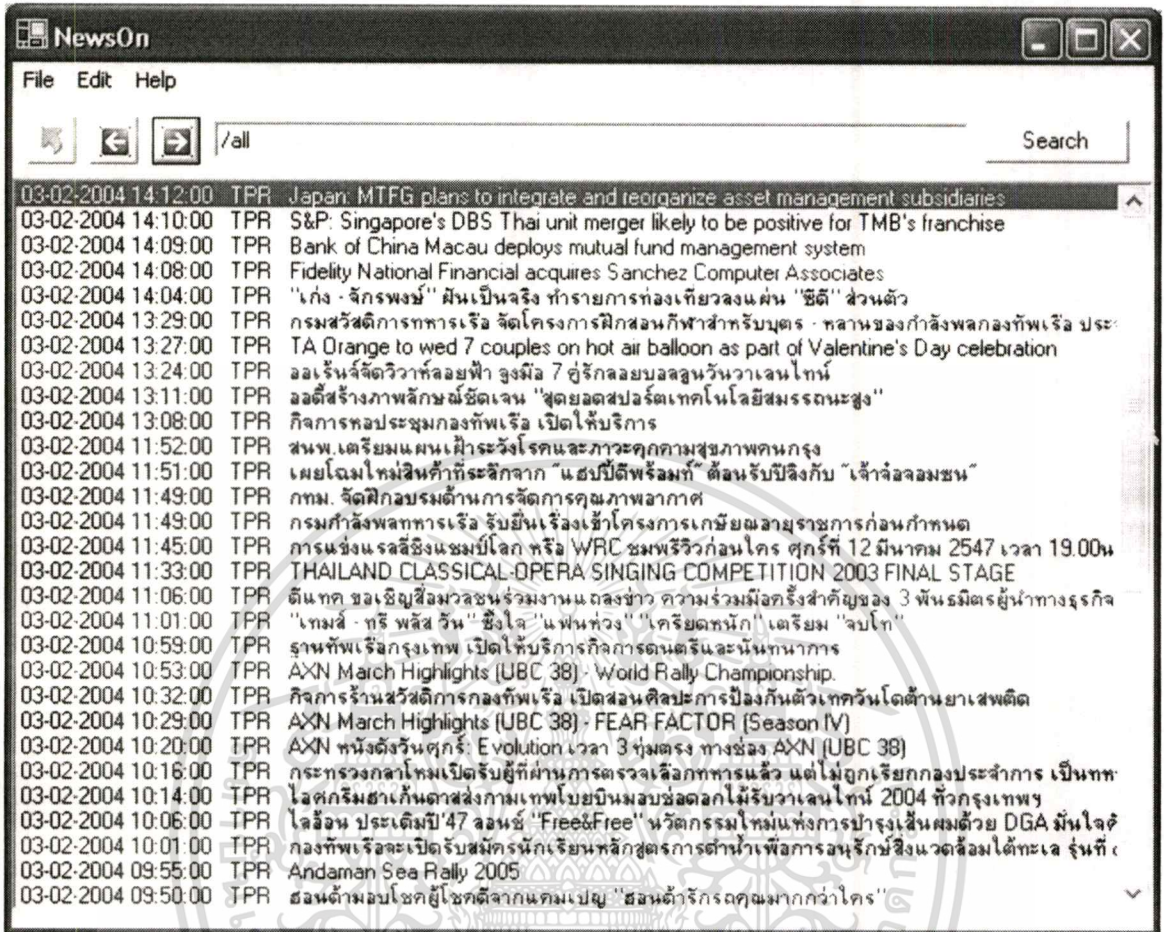
ท่องเที่ยว Search

03-02-2004 15:23:00	TPR	ททท. สำนักงานภาคกลาง เขต3 (พิกษา) ร่วมกับชมรมแหล่งท่องเที่ยวชลบุรี จัดทำบัตรเดียวเที่ยวคุ้ม
03-02-2004 15:11:00	TPR	ททท. ขอเชิญร่วมประกวดภาพยนตร์สารคดีการท่องเที่ยว
03-02-2004 14:04:00	TPR	"เก่ง - ลีกรพงษ์" มีนเป็นจริง ทำรายการท่องเที่ยวลงแผ่น "ซีดี" ส่วนตัว
02-02-2004 08:05:00	TPR	ปฏิทินการท่องเที่ยวแห่งประเทศไทยประจำปี (2 กุมภาพันธ์ 2547)
30-01-2004 07:42:00	TPR	ปฏิทินการท่องเที่ยวแห่งประเทศไทยประจำปี (30 มกราคม 2547)
29-01-2004 14:03:00	TPR	ททท. เชิญชมหนังสือส่งเสริมการท่องเที่ยวไทยเรื่องแรกเข้าฉายในบางกอกฟิล์ม
29-01-2004 11:33:00	TPR	ททท. ขอเชิญนักท่องเที่ยว ประณบึง แต่งไทย เที่ยวงานแผ่นดินสมเด็จพระนารายณ์
29-01-2004 07:58:00	TPR	ปฏิทินการท่องเที่ยวแห่งประเทศไทยประจำปี (29 มกราคม 2547)
28-01-2004 09:17:00	TPR	ททท. แจ้งทำความเข้าใจกับนักท่องเที่ยวเรื่องใช้หวัดนก
28-01-2004 08:00:00	TPR	ปฏิทินการท่องเที่ยวแห่งประเทศไทยประจำปี (28 มกราคม 2547)
27-01-2004 11:06:00	TPR	สสพ. มหิกกำลังททท. ออกชนผลิตภัณฑ์ประเทศไทยเป็นจุดหมายระดับโลกในการจัดประชุม แสดง
27-01-2004 07:51:00	TPR	ปฏิทินการท่องเที่ยวแห่งประเทศไทยประจำปี (27 มกราคม 2547)
26-01-2004 15:07:00	TPR	รัฐมนตรีช่วยว่าการกระทรวงคมนาคม (นายวิเชษฐ์) ส่งเสริมการบินเครื่องบินส่วนตัว สร้างมิติใหม่
26-01-2004 07:26:00	TPR	ปฏิทินการท่องเที่ยวแห่งประเทศไทยประจำปี (26 มกราคม 2547)
23-01-2004 14:46:00	TPR	ปฏิทินการท่องเที่ยวแห่งประเทศไทยประจำปี (24 มกราคม 2547)
23-01-2004 09:49:00	TPR	ททท. ขอเชิญนักท่องเที่ยว ประณบึง แต่งไทย เที่ยวงานแผ่นดินสมเด็จพระนารายณ์
22-01-2004 16:10:00	TPR	ม.เกษตรศาสตร์ จัดโครงการฝึกอบรมหลักสูตร "ท่องเที่ยวศึกษาการปลูกเลี้ยงกล้วยไม้" รุ่นที่ 2
20-01-2004 13:06:00	TPR	ผู้อำนวยการ ททท. สำนักงานภาคหิพิชิตรางวัลบุคคลด้านการท่องเที่ยวดีเด่นปี 2003
20-01-2004 11:21:00	TPR	ททท. พาคณะสื่อมวลชนร่วมงาน OTOP CITY และสำรวจแหล่งท่องเที่ยวใหม่
20-01-2004 08:23:00	TPR	มิติใหม่ท่องเที่ยวจริง "Romantic Destination" รวมสุดยอดของดีเมืองจริง กับโครงการ "Trang SEA
19-01-2004 11:57:00	TPR	กองทัพเรือ เปิดแหล่งท่องเที่ยว "หาดทรายแก้ว"
19-01-2004 08:02:00	TPR	"ชีวิตริมน้ำ วิถีกรุงบนรอยอดีต" สัมผัสใหม่ของการท่องเที่ยวกรุงเทพฯ
15-01-2004 17:02:00	TPR	สายการบิน SGA แถลงข่าว...เปิดนำฟ้า ส่งเสริมท่องเที่ยวเชิงอนุรักษ์
15-01-2004 14:47:00	TPR	ลดราาา จัดงานเฉลิมฉลอง 111 ปี พร้อมเปิดตัวพิกษาที่เมือง ต่อเนื่องรับนักท่องเที่ยวอย่างยิ่งใหญ่
15-01-2004 08:29:00	TPR	ปฏิทินการท่องเที่ยวแห่งประเทศไทยประจำปี (15 มกราคม 2547)

รูปที่ 4.9 Search Screen

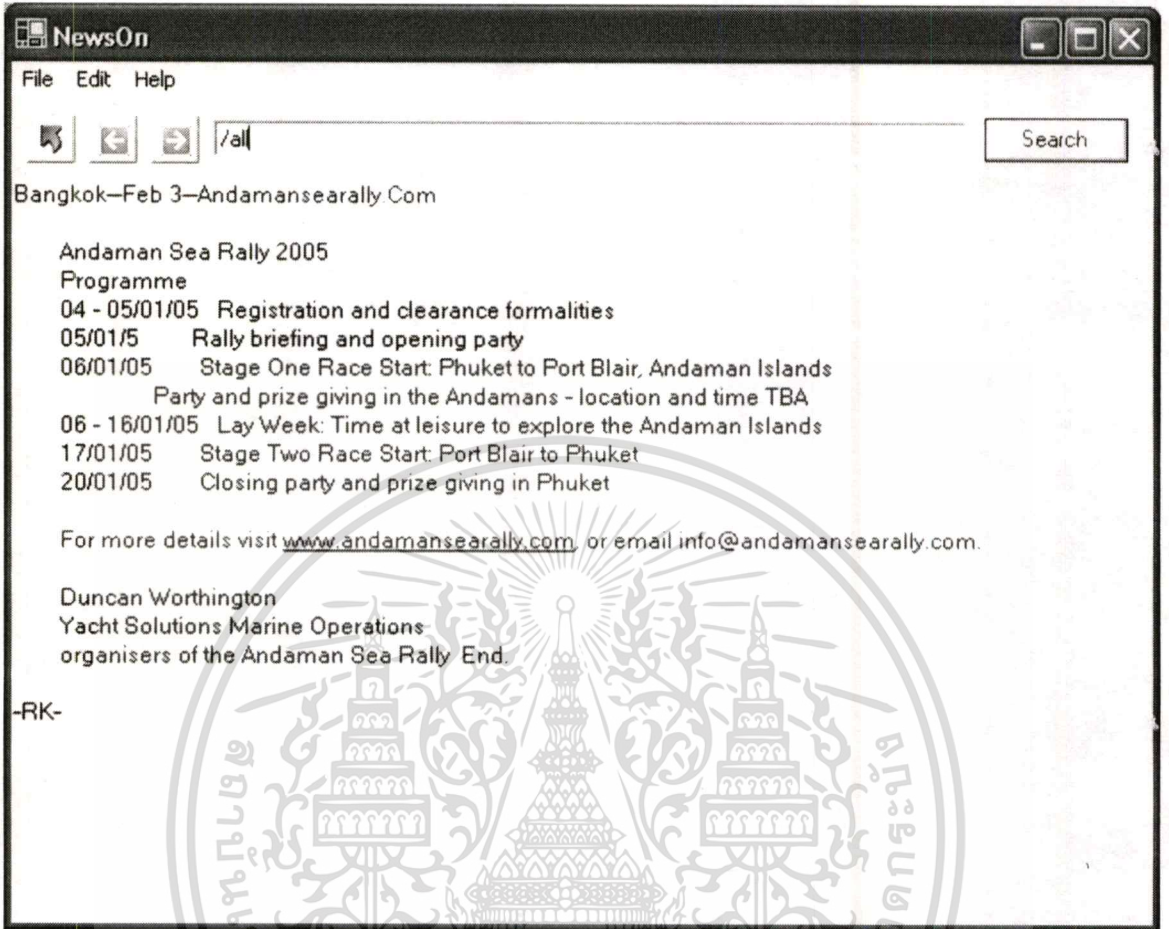
- ผู้ใช้สามารถดูหัวข้อข่าวที่อยู่ในหน้าถัดไปได้โดย click ปุ่ม next(->) และสามารถกลับมาหน้าเดิมได้โดยกดปุ่ม back(<-)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



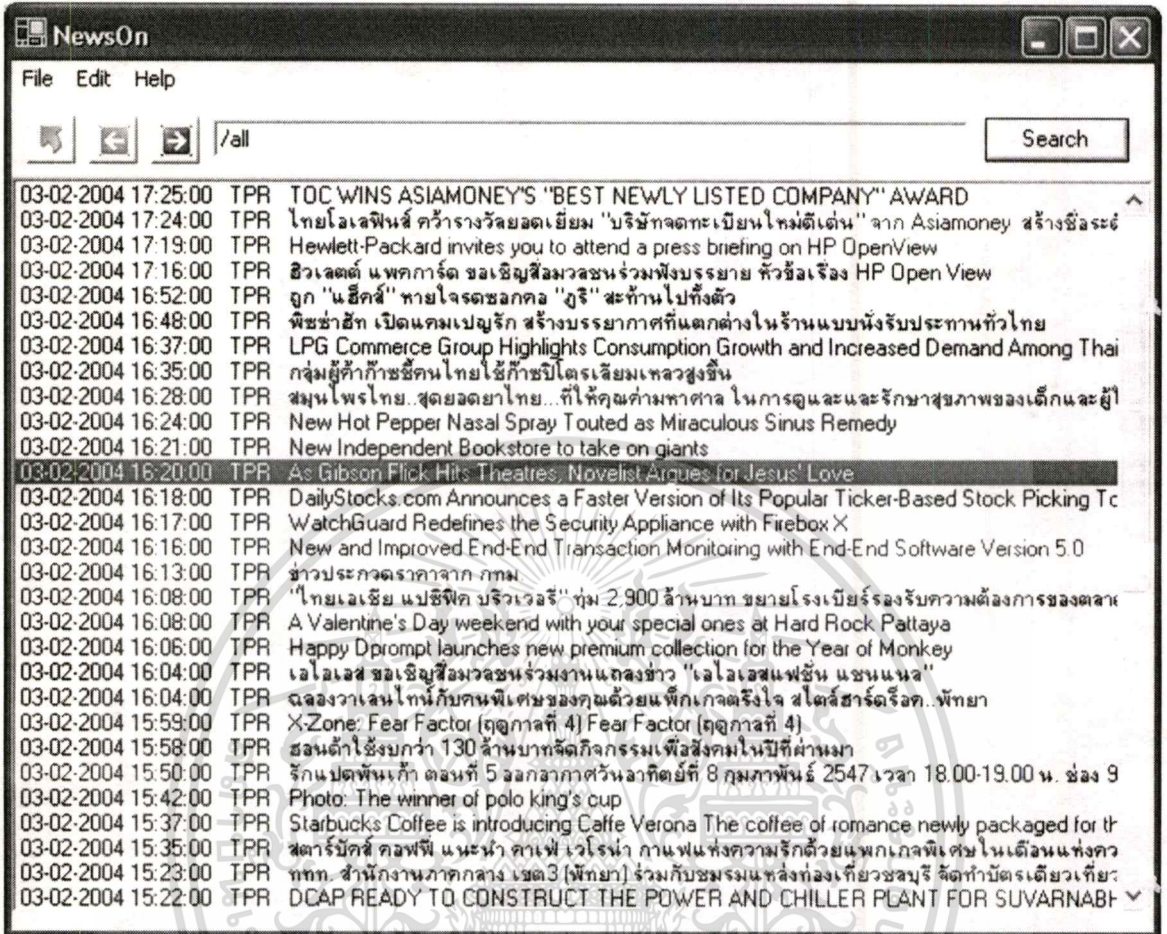
รูปที่ 4.10 Next button click Screen

- ผู้ใช้สามารถดูเนื้อหาของหัวข้อที่ต้องการได้โดย double click เข้าไปที่ หัวข้อข่าวที่ต้องการดู โปรแกรมจะเปลี่ยนจากหน้าดูหัวข้อข่าวไปเป็นหน้าที่แสดงเนื้อความข่าว ดังรูปที่ 4.11



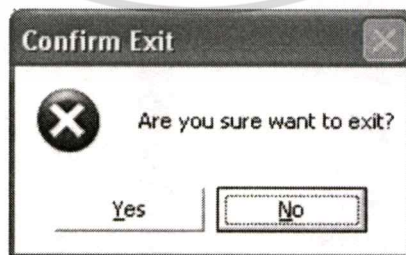
รูปที่ 4.11 Story Screen

- เมื่อต้องการกลับมาที่หน้าคู่มือข้อ่าวใหม่ให้คลิกที่ปุ่ม



รูปที่ 4.12 Back to headline page Screen

- ผู้ใช้งานสามารถออกจากระบบได้โดยเลือก menu exit



รูปที่ 4.13 Exit confirm Screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการพัฒนา

5.1 ผลการพัฒนาโครงการทั้งหมด

- 1 ทำการออกแบบระบบบริการข่าวออนไลน์ทั้งส่วนของเซิร์ฟเวอร์ และไคลเอนท์ซึ่งออกแบบให้เป็นการทำงานแบบกระจาย Distributed system โดยแบ่งออกเป็นบริการ ต่างๆดังนี้
 - User service ทำหน้าที่ตรวจสอบ username และ password
 - Search service ทำหน้าที่ค้นหาข่าวที่ต้องการจากคำค้น
 - Alert service ทำหน้าที่ส่งข่าวที่เพิ่งเข้ามาใหม่ให้กับผู้ใช้
 - Story service ทำหน้าที่ส่งเนื้อหาข่าว ไปให้ที่ผู้ใช้ต้องการดูข่าวในหัวข้อนั้น
 - NewsOn (Client application) ทำหน้าที่เป็น user interface ระหว่าง client และ server
- 2 ทำการออกแบบฐานข้อมูลที่ใช้เก็บข้อมูลเพื่อที่จะให้บริการกับผู้ใช้ รวมถึงวิธีการที่จะให้ service แต่ละตัวเข้ามาใช้งานในฐานข้อมูล
- 3 พัฒนา service ตามที่ได้ออกแบบไว้โดยใช้เทคโนโลยี .Net Remoting ของ Microsoft ซึ่งมีข้อดีคือ สามารถแยกชั้นการติดต่อสื่อสารออกจาก การพัฒนาระบบ (Coding) ออกจากกัน ทำให้การพัฒนาระบบสามารถพัฒนาได้สะดวกและง่ายขึ้น เนื่องจากสามารถที่จะใช้การติดต่อสื่อสารกับเครื่องอื่นได้เพียงแค่แก้ไขใน Configuration file ไม่ต้องแก้ไขโปรแกรมใหม่เช่น ต้องการจะเปลี่ยนจาก TCP ไปเป็น HTTP
- 4 พัฒนาการติดต่อเพื่อดึงข้อมูลระหว่าง service ต่างๆ และฐานข้อมูล โดยใช้ store procedure ซึ่งทำให้การแก้ไขการดึงข้อมูลหรือการเก็บข้อมูลลงฐานข้อมูลสามารถแก้ไขได้ที่ store procedure โดยจะไม่กระทบกับ Code ในโปรแกรมที่พัฒนา อีกทั้งยังสามารถทำงานได้เร็วกว่าการที่ให้โปรแกรมส่ง query string ที่เป็น SQL statement มา execute ที่ฐานข้อมูลอีกด้วย

5.2 ประโยชน์ที่ได้รับ

จากการศึกษาและการพัฒนาระบบงาน ทำให้ได้รู้ถึงเทคโนโลยีแบบใหม่ๆที่จะช่วยให้พัฒนาระบบที่มีประสิทธิภาพมากขึ้นและใช้ระยะเวลาในการพัฒนาน้อยลง และสามารถนำความรู้และเทคนิคที่ได้รับไปใช้เป็นแนวทางในการพัฒนาระบบอื่น ซึ่งการทำงานของระบบแบบกระจายนี้เริ่มมีการใช้งานที่แพร่หลายมากขึ้นเนื่องการเครือข่ายเน็ตเวิร์กมีการพัฒนาให้เร็วขึ้นและค่าใช้จ่ายถูกลง และยังทำให้ระบบมีเสถียรภาพมากขึ้น ไม่ว่าจะในด้าน Performance หรือ Fault Tolerance

5.3 ปัญหาและข้อจำกัดของระบบ

- 1 ข้อจำกัดเรื่อง platform ของเครื่องที่จะใช้งานระบบเนื่องจากการพัฒนาระบบนี้เครื่องที่สามารถที่จะใช้งานได้ต้องมี .Net Framework 1.0 ภายในเครื่อง ซึ่งเครื่องที่จะลงได้จะต้องเป็น Windows 98 ขึ้นไป
- 2 การติดต่อสื่อสารกันระหว่างฝั่ง client และ server นั้นอาจจะมีปัญหาเกี่ยวกับ firewall ได้ถ้าใช้การติดต่อสื่อสารด้วยโปรโตคอลที่ไม่ใช่ HTTP port 80
- 3 การที่ไปดึงข้อมูลมาเก็บลงในฐานข้อมูลจะยังไม่เป็นแบบ realtime จริงๆเนื่องจากใช้เวลาในการเข้าไปดูว่าเว็บไซต์นี้มีการเปลี่ยนแปลงหรือยังเป็นรอบๆ จึงอาจจะทำให้ได้ข้อมูลช้ากว่าบนหน้าเว็บไซต์
- 4 ข้อจำกัดในการได้ alert ของ client นั้นใช้วิธีการที่คอยเข้าไปตรวจสอบกับฐานข้อมูลซึ่งอาจจะทำให้ได้ข้อมูลช้ากว่าเวลาที่ข้อมูลถูกเก็บลงฐานข้อมูล
- 5 การค้นหาข่าวยังทำได้ไม่ดีนักเนื่องจากใน SQL 2000 server ไม่สามารถรองรับการค้นหาแบบ full-text search ภาษาไทย จึงต้องใช้การค้นหาแบบ pattern matching ซึ่งการค้นหาแบบนี้จะได้ข้อมูลที่อาจจะไม่ตรงกับคำค้นที่เราต้องการจะหาขึ้นมาด้วย

5.4 ข้อดีของโครงการที่พัฒนา

- 1 สามารถค้นหาข่าวที่ต้องการได้โดยไม่ต้องเข้าไปเปิดเว็บไซต์หน้าต่างๆ
- 2 สามารถดูข่าวย้อนหลังในอดีตที่ผ่านมาได้
- 3 สามารถได้ข่าวใหม่ที่ต้องการได้เมื่อเว็บไซต์มีการเปลี่ยนแปลง
- 4 ระบบทำงานได้มีประสิทธิภาพมากขึ้นเนื่องจากสามารถแยก service ทำงานคนละเครื่องกันได้ทำให้รองรับการทำงานได้มากขึ้น

5.5 ข้อเสียของโครงการที่พัฒนา

- 1 การรับส่งข้อมูลระหว่าง client และ server มีการ encrypt เฉพาะส่วน password จึงอาจจะมีการดักจับข้อมูลได้
- 2 ถ้า firewall ไม่ยอมให้ packet ผ่าน port ที่ระบุไว้ client จะไม่สามารถติดต่อกับฝั่ง server ได้
- 3 ใช้งานได้กับเครื่องที่เป็น Windows 98 ขึ้นไปและมี .Net Framework

5.6 ข้อเสนอแนะ

การพัฒนาในระบบในส่วนของการ alert นั้นควรจะให้การ push ข้อมูลกลับมาให้ Client เพื่อให้ได้ข้อมูลที่เป็น realtime มากที่สุดแต่จะติดปัญหาเรื่อง Firewall เนื่องจาก Firewall ส่วนใหญ่จะไม่เปิดให้ packet ภายนอกเข้ามาภายในได้ และในส่วนของการค้นหาข้อมูลควรจะใช้การ search แบบ Full-Text search ซึ่งจะได้ข้อมูลที่ถูกต้องมากกว่าแบบ pattern matching ซึ่ง developer จะต้องพัฒนาตัว search engine เองหรือ ไปใช้ search engine อื่นที่สนับสนุนภาษาไทย การพัฒนาโปรแกรมถ้าเปลี่ยนเป็นการทำงานแบบ multi thread จะสามารถเพิ่มประสิทธิภาพให้กับระบบได้มากขึ้นเนื่องจากสามารถทำงานไปพร้อมๆกันได้

บรรณานุกรม

บัญชา ปะทีละเตลิ่ง. 2545. **Visual Basic .NET**. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

Macdonald, Matthew. 2003. **Microsoft .NET Distributed Applications: Integrateing XML Web services and .NET Remoting**. Washington: Microsoft Press.

Curra, David, Olsen, Andy and Pinnock, Jon. 2002. **Visual Basic .NET Remoting Handbook**. Birmingham: Wrox Press.

Balaji, Ramesh. 2002. **15 Seconds Developing with .Net Delegates** [Online]. Available: <http://www.15seconds.com/issue/020815.htm>.

ภาคผนวก

Store procedure และ trigger ต่างๆที่ใช้ในการทำงานกับฐานข้อมูลในระบบมีดังนี้

SP_Alert: สำหรับ alert service เรียกใช้งานเพื่อตรวจสอบหาข่าวที่เข้ามาใหม่

```
CREATE PROCEDURE [dbo].[sp_alert]
```

```
@userid char(36),@str varchar(250),@displaytime datetime
```

```
AS
```

```
if exists(select userid from usertable where userid = @userid)
```

```
begin
```

```
    if @str = '/all'
```

```
        begin
```

```
            select newsid,source,displaytime,headline from newstable where newsid in
            (select newsid from alerttable where displaytime > @displaytime) order by
```

```
displaytime
```

```
        end
```

```
    else
```

```
        begin
```

```
            declare @sqlstr varchar(800)
```

```
            declare @pos int
```

```
            declare @pos1 int
```

```
            declare @pos2 int
```

```
            declare @phase int
```

```
            declare @keyword varchar(100)
```

```
            declare @cause varchar(400)
```

```
            set @str = ltrim(rtrim(@str))
```

```

set @cause = "
while len(@str) <> 0
begin
set @phase = charindex("'",@str)
if @phase = 0
begin
set @pos = charindex(' ',@str)
if @pos = 0
begin
set @keyword = @str
set @str = replace(@str,@keyword,"")
set @cause = @cause + '(headline like "%' + @keyword + '%" and
(story like "%' + @keyword + '%" )'
end
else
begin
set @keyword = substring(@str,1,@pos)
set @str = replace(@str,@keyword,"")
set @keyword = rtrim(@keyword)
set @cause = @cause + '(headline like "%' + @keyword + '%" and
(story like "%' + @keyword + '%" ) and '
end
end
else
begin
set @pos1 = charindex("'",@str)
set @pos2 = charindex("'",@str,@pos1+1)
set @keyword = substring(@str,@pos1+1,@pos2-@pos1-1)
set @str = ltrim(rtrim(replace(@str,"'" + @keyword + "'",)))
set @keyword = @keyword

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if len(@str) = 0
    set @cause = @cause + '(headline like "%' + @keyword + '%" ) and
    (story like "%' + @keyword + '%" )'
else
    set @cause = @cause + '(headline like "%' + @keyword + '%" ) and
    (story like "%' + @keyword + '%" ) and '
end
set @sqlstr = 'select newsid,source,displaytime,headline from newstable where newsid
in '
+ '(select newsid from alerttable where displaytime > "' +
convert(varchar(19),@displaytime,20)
+ '" ) and (' + @cause + ') order by displaytime'
end
exec(@sqlstr)
end
GO

```

SP_Checkuser: สำหรับ user service เรียกใช้ในการตรวจสอบ username และ password

CREATE PROCEDURE [dbo].[sp_checkuser]

@username char(25),@password varchar(30)

AS

select userid from UserTable where username = @username and password = @password

GO

SP_Getheadline: สำหรับ Search service ใช้เพื่อค้นหา headline

CREATE PROCEDURE [dbo].[sp_getheadline]

@userid char(36),@str varchar(255),@displaytime datetime,@direction varchar(10)

AS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

create table #tmptable
(newsId char(28)
,source char(10)
,displaytime datetime
,headline varchar(400) )

if exists (select userId from UserTable where userId = @userid)
begin
    if @str = 'all'
        if @direction = 'forward'
            select top 60 newsId,source,displaytime,headline from NewsTable
where displaytime <= convert(varchar(19),@displaytime,20) order by displaytime desc
        else
            begin
                insert into #tmptable (newsId,source,displaytime,headline)
                select top 50 newsId,source,displaytime,headline from NewsTable
where displaytime >= convert(varchar(19),@displaytime,20) order by displaytime
                select newsId,source,displaytime,headline from #tmptable order by
displaytime desc
                drop table #tmptable
            end
        else
            begin
                declare @sqlstr varchar(1000)
                declare @pos int
                declare @pos1 int
                declare @pos2 int
                declare @phase int
                declare @keyword varchar(100)
                declare @cause varchar(400)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set @str = ltrim(rtrim(@str))
set @cause = "
while len(@str) <> 0
begin
set @phase = charindex("'",@str)
if @phase = 0
begin
set @pos = charindex(' ',@str)
if @pos = 0
begin
set @keyword = @str
set @str = replace(@str,@keyword,"")
set @cause = @cause + '((headline like "%' + @keyword + '%" or
(story like "%' + @keyword + '%"'))'
end
else
begin
set @keyword = substring(@str,1,@pos)
set @str = replace(@str,@keyword,"")
set @keyword = rtrim(@keyword)
set @cause = @cause + '((headline like "%' + @keyword + '%" or
(story like "%' + @keyword + '%"')) and '
end
end
else
begin
set @pos1 = charindex("'",@str)
set @pos2 = charindex("'",@str,@pos1+1)
set @keyword = substring(@str,@pos1+1,@pos2-@pos1-1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set @str = ltrim(rtrim(replace(@str,"" + @keyword + "", "")))
set @keyword = @keyword
if len(@str) = 0
    set @cause = @cause + '(headline like "%' + @keyword + '%" or (story
like "%' + @keyword + '%" )
else
    set @cause = @cause + '(headline like "%' + @keyword + '%" or (story
like "%' + @keyword + '%" ) and '
end
end

if len(@cause) = 0
    if @direction = 'forward'
        set @sqlstr = 'select top 60 newsid,source,displaytime,headline from
NewsTable where displaytime <= "' + convert(varchar(19),@displaytime,20) + "' order by
displaytime desc'
    else
        begin
            set @sqlstr = 'insert into #tmptable
(newsid,source,displaytime,headline) '
            set @sqlstr = @sqlstr + 'select top 50
newsid,source,displaytime,headline from NewsTable where displaytime >= "' +
convert(varchar(19),@displaytime,20) + "' order by displaytime '
            set @sqlstr = @sqlstr + 'select newsid,source,displaytime,headline from
#tmptable order by displaytime desc '
            set @sqlstr = @sqlstr + 'drop table #tmptable'
            --set @sqlstr = 'select top 60 newsid,source,displaytime,headline from
NewsTable where displaytime >= "' + convert(varchar(19),@displaytime,20) + "' order by
displaytime'
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    if @direction = 'forward'
        set @sqlstr = 'select top 60 newsid,source,displaytime,headline from
NewsTable where ' + @cause + ' and displaytime <= ' + convert(varchar(19),@displaytime,20)+
''' order by displaytime desc'
    else
    begin
        set @sqlstr = 'insert into #tmptable
(newsid,source,displaytime,headline) '
        set @sqlstr = @sqlstr + 'select top 50
newsid,source,displaytime,headline from NewsTable where ' + @cause + ' and displaytime >= ' +
convert(varchar(19),@displaytime,20)+ ''' order by displaytime '
        set @sqlstr = @sqlstr + 'select newsid,source,displaytime,headline from
#tmptable order by displaytime desc '
        set @sqlstr = @sqlstr + 'drop table #tmptable'
    end
    exec (@sqlstr)
end
end
else
select errorcode = -1
GO

```

SP_Getstory: สำหรับ Story service เรียกใช้เพื่อดึงเนื้อหาข่าว

CREATE PROCEDURE [dbo].[sp_getstory]

@userid char(36),@newsid char(36)

AS

if exists (select userid from UserTable where userid = @userid)

select story from NewsTable where newsid = @newsid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
--else
```

```
--select errorcode= -1
```

```
GO
```

SP_InsertNewsTable: สำหรับให้ Crawler เรียกใช้เพื่อ insert ข้อมูลลงฐานข้อมูล

```
CREATE PROCEDURE [dbo].[sp_insertNewsTable]
```

```
@newsID char(28),@source char(10),
```

```
@displaytime datetime,@headline varchar(400),
```

```
@story text
```

```
AS
```

```
if not exists(select newsid from newstable where newsid = @newsid)
```

```
insert into newstable(newsId,source,displaytime,headline,story)
```

```
values(@newsid,@source,@displaytime,@headline,@story)
```

```
GO
```

SP_InsertImgNewsTable: สำหรับให้ Crawler เรียกใช้เพื่อ insert ข้อมูลลงฐานข้อมูล

```
CREATE PROCEDURE [dbo].[sp_insertImgNewsTable]
```

```
@newsID char(28),@source char(10),
```

```
@displaytime datetime,@headline varchar(400),
```

```
@story text,@image image
```

```
AS
```

```
if not exists(select newsid from newstable where newsid = @newsid)
```

```
insert into newstable(newsId,source,displaytime,headline,story,image)
```

```
values(@newsid,@source,@displaytime,@headline,@story,@image)
```

```
GO
```

Trigger: Alertnews ใช้สำหรับ Insert ข้อมูลลง AlertTable เมื่อมีข้อมูลเข้ามาใน NewsTable

CREATE TRIGGER [Alertnews] ON dbo.NewsTable

FOR INSERT

AS

insert into AlertTable(newsId,source,displaytime,headline)

select newsId,source,displaytime,headline from inserted



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	เสริมศักดิ์ ศรีโพธิ์ทอง
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	วทบ. คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าฯ ลาดกระบัง
ตำแหน่งหน้าที่	วิศวกรระบบ
สถานที่ทำงาน	บริษัทอินโฟเคสลิมิตเต็ด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้