

หุ่นยนต์ค้นหาข้อมูลบนอินเทอร์เน็ต

Web Application Development



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2541

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ้ยนตค้ันหาข้อมูลบนอินเทอร์เน็ต

Web Application Development

ผู้จัดทำ

- | | | |
|------------------|--------------|----------|
| 1. นาย วรเชษฐ์ | ชื่อสุทธิกุล | 38014428 |
| 2. นางสาว อัจฉรา | สุรกาญจน์ | 38014642 |



อาจารย์ที่ปรึกษา

(ดร. วรวัฒน์ ลิ้มโกกา)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ค้นหาข้อมูลบนอินเทอร์เน็ต

นายวรเชษฐ์ ชื้อสุทธิกุล 38014428

นางสาวอัจฉรา สุรกาญจน์ 38014642

ดร. วรวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษา

ปีการศึกษา 2541

บทคัดย่อ

ในปัจจุบันเว็บไซต์ในประเทศไทยมีแนวโน้มที่จะเพิ่มปริมาณขึ้นอย่างรวดเร็ว ดังนั้นความต้องการใช้งานหุ่นยนต์ค้นหาข้อมูลบนอินเทอร์เน็ตที่มุ่งเน้นการค้นหาข้อมูลที่มีขอบเขตอยู่ในประเทศไทยนั้นจึงมีความต้องการเพิ่มขึ้นเช่นกัน

โครงการ “หุ่นยนต์ค้นหาข้อมูลบนอินเทอร์เน็ต” นี้จึงถูกจัดสร้างขึ้นเพื่อรองรับความต้องการดังกล่าว โดยโครงการนี้ได้แบ่งออกเป็น 3 ส่วน คือ ส่วนแรกคือส่วนดึงข้อมูลและวิเคราะห์ข้อมูล ซึ่งจะเป็นส่วนที่ทำงานโดยโรบอท (Robot) ซึ่งเป็นเอเจนต์ (Agent) ประเภทหนึ่งบนอินเทอร์เน็ต ในการดึงข้อมูลจากเว็บไซต์ต่าง ๆ ที่อยู่ในประเทศไทย แล้วนำข้อมูลที่ได้อาวิเคราะห์จัดให้อยู่ในรูปแบบที่เหมาะสม ส่วนที่สองคือ ส่วนที่ทำการติดต่อระหว่างผู้ใช้บริการกับเซิร์ฟเวอร์ การทำงานในส่วนนี้จะทำผ่านโปรแกรมที่เรียกว่า โปรแกรมเซิร์ฟเล็ดมีหน้าที่ในการส่งและรับข้อมูลระหว่างผู้ใช้บริการและเซิร์ฟเวอร์ ส่วนสุดท้ายคือส่วนจัดเก็บและค้นหาข้อมูลซึ่งในส่วนนี้จะเป็นส่วนที่จัดเก็บข้อมูลที่ได้จากโรบอทในรูปแบบที่พร้อมที่จะทำการค้นหาตามเงื่อนไขที่ผู้ใช้บริการต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web Application Development

Mr. Worachet Suesuthikul

Ms. Atchara Surakarn

Dr. Voravat Limpoka Advisor

Abstract

In the present , the more Thailand 's website is very expanded ,the more search engine which search information in Thailand is very important.

The project “Web Application Development” develop with this reason. The structure of this project has three parts. First part is Search and Analyze Information that has important responsibility to find the information from web site in Thailand and later convert these information into proper form. Second part is Communication between user and server via program called Servlet. Data passing in this part are words for search and options in searching. The last part of this project is Database and Query Information, Database in this search engine is desinged by using ER-Model and using in Oracle Server in JAVA to create database system, so server can query data in demand of user from Oracle server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน

- อาจารย์ ดร.วรวัฒน์ ลิ้มโกศา ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการนี้ที่ได้ให้ความดูแลเอาใจใส่ แนะนำ และช่วยเหลือซึ่งต้องขอขอบพระคุณเป็นอย่างมาก
- บิดา มารดา ที่เป็นแรงใจ คอยเติมพลังใจเวลาที่ขาดหาย
- เพื่อน ๆ GANG 4D ร่วมทุกข์ ร่วมสุขมาตลอด 911 วัน
- ห้องภาค สถานที่ทำงานที่เย็นน้ำและหลับสบาย
- ภาควิชาที่ยังอนุญาตให้ทำงานในยามค่ำคืนได้

วรเชษฐ ชื้อสุทธิกุล
อัจฉรา สุรกาญจน์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

| | หน้าที่ |
|--|---------|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญตาราง | VII |
| สารบัญรูปภาพ | VIII |
| บทที่ 1 บทนำ | 1 |
| 1.1 วัตถุประสงค์ของโครงการ | 1 |
| 1.2 ขอบเขตการทำงานของโครงการ | 2 |
| 1.3 วิธีการดำเนินงาน | 2 |
| 1.4 ประโยชน์ที่ได้รับ | 3 |
| บทที่ 2 ทฤษฎีและหลักการ | 4 |
| 2.1 ส่วนข้อมูลและวิเคราะห์ข้อมูล | 5 |
| 2.1.1 ความรู้เกี่ยวกับเอเจนต์ | 6 |
| 2.1.2 คุณสมบัติของเอเจนต์ | 7 |
| 2.1.3 คุณสมบัติและการทำงานของเว็บโรบอท | 8 |
| 2.1.4 กฎของโรบอท | 9 |
| 2.1.5 สิ่งที่ต้องปฏิบัติในการสร้างโรบอท | 12 |
| 2.1.6 ภาษาที่ใช้เขียนโรบอท | 14 |
| 2.1.7 คุณสมบัติในเรื่องพอร์ตเทเบิ้ลและความปลอดภัยของจาวา | 14 |
| 2.2 ส่วนเชื่อมต่อระหว่างผู้ใช้บริการและส่วนจัดเก็บข้อมูล | 15 |
| 2.2.1 เซิร์ฟเล็ต | 15 |
| 2.2.2 การติดต่อกันของเซิร์ฟเล็ตกับฐานข้อมูล | 16 |
| 2.2.3 การใช้งาน Java servlet API โดย web application | 17 |
| 2.2.4 การเขียนเซิร์ฟเล็ต | 17 |
| 2.3 ส่วนจัดเก็บและค้นหาข้อมูล | 19 |
| 2.3.1 ระบบฐานข้อมูล | 19 |
| 2.3.2 ระบบจัดการฐานข้อมูล | 20 |
| 2.3.3 โมเดลข้อมูล | 20 |
| 2.3.4 ข้อดีของการเก็บข้อมูลและฐานข้อมูล | 21 |
| 2.3.5 หลักการของอีอาร์โมเดล | 22 |
| 2.3.6 ออราเคิล | 24 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|--|----|
| 2.3.7 ภาษาเอสคิวแอล | 24 |
| 2.4 ส่วนหน้าจอรับข้อมูลและแสดงผล | 26 |
| 2.4.1 โครงสร้างของเอกสารแบบเอชทีเอ็มแอล | 28 |
| 2.4.2 เซด | 28 |
| 2.4.2.1 TITLE | 28 |
| 2.4.2.2 ISINDEX | 29 |
| 2.4.2.3 NEXTID | 29 |
| 2.4.2.4 LINK | 29 |
| 2.4.2.5 BASE | 29 |
| 2.4.2.6 META | 30 |
| 2.4.3 บอด้ | 30 |
| 2.4.3.1 Section Headings | 30 |
| 2.4.3.2 Marking Paragraphs with HTML | 30 |
| 2.4.3.3 Line Breaks | 30 |
| 2.4.3.4 IMG | 30 |
| 2.4.3.5 Hypertext Anchors | 31 |
| 2.4.3.6 Lists | 31 |
| 2.4.3.7 Horizontal Ruled Line | 31 |
| 2.4.3.8 Special Characters in HTML | 31 |
| 2.4.3.9 Charaater Emphasis Modes | 32 |
| 2.4.3.10 Special Text Formatting Modes | 32 |
| 2.4.3.11 TABLES | 33 |
| 2.4.3.12 FORMS | 33 |
| บทที่ 3 ความรู้เกี่ยวกับเซิร์ฟเอนจิน | 36 |
| 3.1 ประเภทของเซิร์ฟเอนจิน | 36 |
| 3.2 การออกแบบเซิร์ฟเอนจิน | 36 |
| 3.3 ตัวอย่างของเซิร์ฟเอนจินที่ได้รับความนิยม | 41 |
| 3.3.1 Alta Vista | 41 |
| 3.3.2 HotBot | 41 |
| 3.3.3 Yahoo | 42 |
| บทที่ 4 การออกแบบและการสร้าง | 43 |
| 4.1 การกำหนดฟังก์ชันการทำงานของตัวค้นหาข้อมูล | 43 |
| 4.2 การออกแบบในส่วนที่ใช้ดึงข้อมูลและวิเคราะห์ข้อมูล | 43 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 4.2.1 การเก็บคีย์เวิร์ดจากเว็บเพจ 43
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|---------|--|----|
| 4.2.2 | การคำนวณค่าความสำคัญของคำที่ปรากฏในเว็บเพจ | 43 |
| 4.2.3 | การกำหนดค่า priority-key ของคำในตำแหน่งต่างๆ | 44 |
| 4.2.4 | วิธีการท่องเว็บไชท์ของโรบอท | 44 |
| 4.2.5 | การกำหนดจุดสิ้นสุดการทำงานของโรบอท | 46 |
| 4.2.6 | การกำหนดลักษณะการทำงานของโรบอท | 47 |
| 4.3 | การออกแบบในส่วนของการติดต่อระหว่างผู้ใช้กับระบบฐานข้อมูล | 54 |
| 4.4 | การออกแบบในส่วนของการจัดเก็บและค้นหาข้อมูล | 54 |
| 4.5 | การออกแบบในส่วนของหน้าจอที่ใช้ติดต่อกับผู้ใช้บริการ | 55 |
| 4.6 | การเลือกแพลตฟอร์มในการทำโครงการ | 55 |
| 4.7 | การเลือกเว็บเซิร์ฟเวอร์ | 55 |
| 4.8 | การเลือกภาษาในการทำโครงการ | 56 |
| บทที่ 5 | การทดลองและผลการทดลอง | 57 |
| 5.1 | การกำหนดยูอาร์แอลเริ่มต้น | 57 |
| 5.2 | การตรวจสอบยูอาร์แอลที่อยู่ในประเทศไทย | 57 |
| 5.3 | การวิเคราะห์คำและคำนวณค่าความสำคัญ | 57 |
| 5.4 | การค้นหาโดยเทียบเฉพาะคำที่ตรงกับคีย์เวิร์ด | 58 |
| 5.5 | การค้นหาโดยเทียบกับคำที่มีคีย์เวิร์ด | 58 |
| บทที่ 6 | บทวิจารณ์และสรุป | 63 |
| 6.1 | สรุปโครงสร้างการทำงานทั้งหมดของโครงการนี้ | 63 |
| 6.1.1 | ส่วนดึงข้อมูลและวิเคราะห์ข้อมูล | 63 |
| 6.1.2 | ส่วนเชื่อมต่อระหว่างผู้ใช้งานกับเซิร์ฟเวอร์ | 65 |
| 6.1.3 | ส่วนจัดเก็บข้อมูล | 66 |
| 6.1.4 | อุปสรรคในการทำงานของเซิร์ฟเวอนจิน | 68 |
| 6.2 | ประเมินผลโครงการ | 68 |
| 6.3 | วิเคราะห์ประสิทธิภาพของโครงการ | 68 |
| 6.4 | การพัฒนาโครงการในอนาคต | 68 |
| 6.5 | สรุปโครงการ | 69 |
| ภาคผนวก | | |
| | ภาคผนวก ก JDBC | 71 |
| | ภาคผนวก ข JAVA SERVLET | 78 |
| | ภาคผนวก ค คู่มือการใช้งานสำหรับผู้ใ้ | 82 |
| | ภาคผนวก ง รายละเอียดของโปรแกรม | 86 |
| | ภาคผนวก จ source code | 91 |

สารบัญตาราง

| | หน้า |
|--|------|
| ตารางที่ 2.1 สรุปความแตกต่างระหว่างเซิร์ฟเล็ตและซีจีไอ | 16 |
| ตารางที่ 2.2 ตัวอย่าง special character ที่นิยมใช้ | 32 |
| ตารางที่ 3.1 สรุปวิธีการไปยังเว็บไซต์ของเซิร์สเอนจิน | 37 |
| ตารางที่ 3.2 สิ่งที่น่ามาเป็นอินเด็กซ์ | 39 |
| ตารางที่ 3.3 วิธีการจัดอันดับความสัมพันธ์ | 39 |
| ตารางที่ 3.4 การคัด spam ออก | 40 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

| | หน้า |
|--|------|
| รูปที่ 2.1 ตัวอย่างฟอร์มเอชทีเอ็มแอลที่ผู้ใช้ป้อนข้อมูลเพื่อลงทะเบียนมายังเซิร์ฟเว็ท | 19 |
| รูปที่ 2.2 Composite Attribute | 22 |
| รูปที่ 2.3 ความสัมพันธ์ | 23 |
| รูปที่ 2.4 สัญลักษณ์ที่ใช้ในการสร้างอ็อบเจกต์โมเดลไดอะแกรม | 25 |
| รูปที่ 2.5 ตัวอย่างอ็อบเจกต์โมเดลไดอะแกรม | 26 |
| รูปที่ 4.1 การเก็บข้อมูลโดยวิธีการหาทางกว้างก่อน | 45 |
| รูปที่ 4.2 ความสัมพันธ์ระหว่างแต่ละส่วนของตัวค้นหาข้อมูล | 48 |
| รูปที่ 4.3 แผนภาพการทำงานของโรบอท | 49 |
| รูปที่ 4.4 การ update | 50 |
| รูปที่ 4.5 การ operate | 51 |
| รูปที่ 4.6 การวิเคราะห์เอกสาร HTML | 52 |
| รูปที่ 4.7 ส่วนติดต่อกับผู้ใช้ (user interface) | 53 |
| รูปที่ 5.1 แสดงการค้นหาโดยเทียบเฉพาะค่าที่ตรงกับคีย์เวิร์ด | 58 |
| รูปที่ 5.2 แสดงส่วนที่รับข้อมูลจากผู้ใช้โฮมเพจ | 59 |
| รูปที่ 5.3 แสดงผลจากการค้นหาแบบเทียบเฉพาะค่าที่ตรงกับคีย์เวิร์ด | 59 |
| รูปที่ 5.4 แสดงการค้นหาค่าโดยเทียบกับค่าที่มีคีย์เวิร์ด | 60 |
| รูปที่ 5.5 แสดงส่วนรับข้อมูลจากผู้ใช้ในโฮมเพจ | 60 |
| รูปที่ 5.6 แสดงผลจากการค้นหาโดยเทียบกับค่าที่มีคีย์เวิร์ด | 61 |
| รูปที่ 5.7 แสดงการขอลิสต์ของคีย์เวิร์ด | 61 |
| รูปที่ 5.8 แสดงลิสต์ของคีย์เวิร์ดที่หาแล้ว | 62 |
| รูปที่ 5.8 แสดงลิสต์ของคีย์เวิร์ดที่ยังไม่ถูกค้นหา | 62 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันนี้ การใช้งานระบบเครือข่ายอินเทอร์เน็ตได้รับความนิยมอย่างแพร่หลายในระยะเวลาอันรวดเร็ว เนื่องจากเครือข่ายอินเทอร์เน็ตเป็นแหล่งข้อมูลขนาดใหญ่สำหรับผู้ที่เข้ามาใช้บริการระบบเครือข่ายอินเทอร์เน็ต ทั้งนี้เพราะแต่ละไซต์จัดตั้งขึ้นเพื่อให้บริการข้อมูลตามจุดประสงค์ที่แตกต่างกัน โดยอาจเป็นแหล่งข้อมูลที่เกี่ยวข้องกับกีฬา วิชาการ การเมือง เศรษฐกิจ บันเทิง ไปจนถึงการโฆษณาขายสินค้า ดังนั้นในการที่จะได้มาซึ่งข้อมูลตามที่ต้องการนั้นทำได้ลำบากเพราะผู้ใช้บริการเองไม่สามารถทราบได้ว่าข้อมูลตามที่ต้องการนั้นอยู่ที่ใดและเป็นข้อมูลที่ถูกต้องเชื่อถือได้มากน้อยแค่ไหน (ข้อมูลข่าวสารจะมีประโยชน์ได้ก็ต่อเมื่อข้อมูลนั้นเป็นข้อมูลที่ทันต่อเหตุการณ์และมีความถูกต้อง)

ดังนั้นจึงได้มีการคิดค้นสิ่งที่จะสามารถช่วยให้ผู้ใช้บริการอินเทอร์เน็ตได้รับความสะดวกในการสืบค้นข้อมูล จึงแนวคิดในการที่จะทำการจัดหมวดหมู่ของข้อมูลที่มีอยู่อย่างมากมายนี้ให้เป็นระบบ แต่ถึงแม้ว่าจะได้มีการจัดหมวดหมู่ของข้อมูลไว้ก็ตาม การที่จะได้รับข้อมูลซึ่งตรงตามความต้องการก็เป็นไปได้ลำบาก เนื่องจากผลลัพธ์ในการค้นหานั้นมีมากทำให้ผู้ใช้บริการเลือกข้อมูลที่จะได้นำไปใช้ต่อไปได้ยาก

ซึ่งปัญหาความยากในการเลือกข้อมูลไปใช้ ก็สามารถพบในกรณีที่ผู้ใช้บริการต้องการค้นหาข้อมูลที่เกี่ยวข้องกับประเทศไทย ซึ่งเน้นค้นหาจากเว็บไซต์ (Web Site) ที่ตั้งอยู่ในประเทศไทย เช่นกัน ทางผู้จัดทำโครงการนี้ก็ให้เห็นความสำคัญของปัญหานี้จึงมีแนวคิดที่จะทำหุ่นยนต์ค้นหาข้อมูลบนเครือข่ายอินเทอร์เน็ตขึ้นมาเอง โดยมีขอบเขตของข้อมูลที่รวบรวมว่าเป็นข้อมูลของเว็บไซต์ที่ตั้งอยู่ในประเทศไทย เพื่ออำนวยความสะดวกต่อผู้ใช้บริการที่ต้องการเข้ามาหาข้อมูลที่เกี่ยวข้องกับประเทศไทยในด้านต่าง ๆ เช่น การท่องเที่ยวของประเทศ เทคโนโลยี วัฒนธรรม ศาสนา หรืออาจมีส่วนช่วยสร้างภาพพจน์ที่ดีให้กับประเทศไทยได้

นอกจากนี้โครงการนี้ยังมีทฤษฎีและการทำงานที่ตรงต่อความสนใจของผู้จัดทำโครงการ ดังนั้นผู้จัดทำโครงการจึงได้เริ่มต้นทำการศึกษา ทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับโครงการ เช่นการทำงานของอินเทอร์เน็ต คุณสมบัติและการทำงานของโรบอท การจัดเก็บข้อมูลบนระบบฐานข้อมูล เป็นต้น

1.1 วัตถุประสงค์ของโครงการ

1. สามารถทำการจัดทำฐานข้อมูลที่รวบรวมข้อมูลบนเครือข่ายอินเทอร์เน็ตของเว็บไซต์ที่ตั้งอยู่ในประเทศไทยไว้เป็นหมวดหมู่ได้
 2. สามารถคำนวณและจัดลำดับความสำคัญของคำที่เก็บในฐานข้อมูลได้
 3. สามารถจัดทำหุ่นยนต์ค้นหาข้อมูลบนอินเทอร์เน็ต ของเว็บไซต์ที่ตั้งอยู่ในประเทศไทยได้ตามเงื่อนไขที่กำหนด
 4. ทำการศึกษาทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับการทำงานของหุ่นยนต์ค้นหาข้อมูลและสามารถนำเอกสารมาประยุกต์ใช้งานได้จริง
- สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ขอบเขตการทำงานของโครงการ

โครงการนี้เป็นโครงการที่จัดทำกรขึ้นเพื่อทำการรวบรวมข้อมูลต่าง ๆ ที่อยู่บนเว็บไซต์ที่ตั้งอยู่ภายในประเทศไทย ทั้งนี้เพื่อเป็นการอำนวยความสะดวกต่อผู้ใช้บริการที่ต้องการ ทราบข้อมูลเกี่ยวกับประเทศไทย

ในส่วนของการทำงานของหุ่นยนต์ จะเป็นการให้หุ่นยนต์ได้วิ่งไปตามเว็บไซต์ต่าง ๆ ทำการตรวจสอบว่าอยู่ในขอบเขตที่กำหนดหรือไม่ (เฉพาะเว็บไซต์ที่อยู่ในประเทศไทย) เมื่อตรวจสอบพบว่าอยู่ในขอบเขตที่กำหนดไว้ก็จะทำการดึงข้อมูลกลับมา ข้อมูลที่ถูกดึงกลับมาจะถูกทำการเทียบค่าที่ตรงกับค่าที่ผู้ใช้บริการป้อน

หลังจากนั้นทำการสร้างส่วนที่ใช้ในการค้นหาข้อมูลบนฐานข้อมูลที่ใช้จัดเก็บข้อมูล โดยมีการกำหนดเงื่อนไขในการค้นหา เช่น รูปแบบตัวอักษรเหมือนต้นแบบ เป็นต้น เมื่อทำการค้นหาข้อมูลเรียบร้อยแล้ว ก็ทำการแสดงผลพร้อมออกมาทางหน้าจอ ในรูปแบบที่ผู้ใช้งานสามารถทำงานได้

การออกแบบการทำงานของส่วนต่าง ๆ ของโครงการ จะทำการออกแบบจากทฤษฎีที่เกี่ยวข้อง โดยจะทำการเลือกวิธีที่เหมาะสมต่อการทำงานในระดับการศึกษามากที่สุด ทั้งนี้เนื่องจาก การมีข้อจำกัดทางด้านงบประมาณ ทรัพยากรเครื่องคอมพิวเตอร์และอุปกรณ์เสริมต่าง ๆ ที่มีอยู่อย่างจำกัด ทฤษฎีที่เกี่ยวข้องกับโครงการนั้นสามารถแบ่งออกได้เป็นหลายส่วน

1. ทฤษฎีที่เกี่ยวข้องกับส่วนค้นหาและวิเคราะห์ข้อมูล อันได้แก่ ทฤษฎีเกี่ยวกับการทำงานของเอเจนต์, โรบอท, การเทียบค่า
2. ทฤษฎีที่เกี่ยวข้องกับส่วนที่ใช้ติดต่อระหว่างผู้ใช้บริการและระบบจัดเก็บข้อมูลอันได้แก่ทฤษฎีเกี่ยวกับการเขียนโปรแกรมเชื่อมต่อ (ServLet)
3. ทฤษฎีที่เกี่ยวข้องกับส่วนจัดเก็บและค้นหาข้อมูล ได้แก่ ทฤษฎีเกี่ยวกับระบบฐานข้อมูล การออกแบบฐานข้อมูล โปรแกรมที่ใช้สร้างระบบฐานข้อมูล รวมทั้งภาษาที่ใช้ในการดึงข้อมูล
4. ทฤษฎีที่เกี่ยวข้องกับการรับและแสดงผลพร้อมออกมาทางหน้าจอผู้ใช้งาน ได้แก่ การศึกษาทฤษฎีเกี่ยวกับภาษาเ็ชทีเอ็มแอล (HTML)

1.3 วิธีการดำเนินงาน

การดำเนินงานของโครงการนี้มีขั้นตอนดังนี้

1.3.1 ศึกษา ค้นคว้า รวบรวมความรู้ต่างเกี่ยวกับการทำงานของหุ่นยนต์ค้นหาข้อมูลบนอินเทอร์เน็ต การจัดเก็บข้อมูลลงในฐานข้อมูล และการดึงข้อมูลเพื่อนำเสนอต่อผู้ใช้ เป็นการศึกษาถึงโครงสร้าง รูปแบบ ความสามารถในการทำงาน และการนำไปประยุกต์ใช้ เพื่อให้เกิดความเข้าใจถึงหลักการทำงานอย่างแท้จริง

1.3.2 ทำการออกแบบวิธีการทำงานของหุ่นยนต์ ออกแบบระบบฐานข้อมูลที่ใช้จัดเก็บ โดยอาศัยหลักการและเหตุผลจากการค้นคว้าที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.3.3 พัฒนาและทดสอบโปรแกรมที่ได้ออกแบบมานั้น เพื่อนำผลการทดสอบมาวิเคราะห์ถึงการนำทฤษฎีและความรู้ที่ได้จากการศึกษามาปฏิบัติจริง

1.4 ประโยชน์ที่ได้รับ

1.4.1) ในแง่ของงานวิจัย

- 1.ศึกษาความเป็นมาและรายละเอียดต่างๆ ที่เกี่ยวข้องกับเทคโนโลยีหุ่นยนต์ค้นหาข้อมูล
- 2.สามารถระบุความจำเป็นรวมทั้งประโยชน์ที่ได้รับจากการนำเทคโนโลยีนี้มาใช้งาน
- 3.สามารถวิเคราะห์แนวโน้มของระบบค้นหาข้อมูลใหม่ๆ ที่กำลังจะเกิดขึ้นในอนาคตเพื่อให้สามารถรองรับความต้องการการใช้งานอินเทอร์เน็ตได้
- 4.การจัดทำโครงการนี้ สามารถแก้ปัญหาการค้นหาข้อมูลซึ่งเว็บไซต์ที่ตั้งอยู่ในประเทศไทยในการช่วยคลั่นกรองข้อมูลให้ตรงกับความต้องการของผู้ใช้ได้เป็นอย่างดี

1.4.2) ในแง่ของประโยชน์ทั่วไปที่ได้รับ

- 1.ได้รับความรู้จากการค้นคว้าเพิ่มขึ้น เนื่องจากเนื้อหาส่วนใหญ่จะไม่ค่อยมีอยู่ในตำราหรือหนังสือประกอบการเรียน ดังนั้นผู้จัดทำจะต้องค้นหาข้อมูลจากแหล่งต่าง ๆ
- 2.เข้าใจถึงรูปแบบการทำงานของหุ่นยนต์ค้นหาข้อมูล รวมถึงส่วนที่เกี่ยวข้องอื่น ๆ อย่างชัดเจน
- 3.เห็นถึงความจำเป็นที่ต้องมีระบบค้นหาข้อมูลบนอินเทอร์เน็ตเพื่อคลั่นกรองให้ได้ข้อมูลที่ตรงตามความต้องการของผู้ใช้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

ในการจัดทำโครงการนี้ทางผู้วิจัยได้ทำการศึกษาทฤษฎีต่าง ๆ ที่มีความเกี่ยวข้องกับโครงการงาน เช่น โรบอท (robot) , เซิร์ฟเล็ต (Servlet : CGI ภาษาจาวา) , ระบบฐานข้อมูล ในแง่ของหลักการ , การทำงาน เพื่อนำมาใช้อ้างอิงเปรียบเทียบข้อดีและข้อเสียในขั้นตอนของการออกแบบ นอกจากนี้ยังได้ทำการศึกษาภาษาจาวาในส่วนที่เกี่ยวข้องกับการทำโครงการงาน เพื่อที่จะนำมาใช้ในการเขียน โปรแกรม ตัวค้นหาข้อมูล (Search Engine)

ตัวค้นหาข้อมูล คือ ตัวค้นหาข้อมูลบนเครือข่ายอินเทอร์เน็ตที่ทำการรวบรวมข้อมูลจากเว็บไซต์ต่าง ๆ แล้วนำข้อมูลที่รวบรวมได้มาทำการวิเคราะห์เลือกข้อมูลที่มีความสำคัญเพื่อนำมาจัดเก็บในรูปแบบที่เหมาะสม พร้อมทั้งจะทำการค้นหาข้อมูลตามเงื่อนไขที่กำหนด

ประเภทของตัวค้นหาข้อมูล

- 1 เซิร์สเอนจิน (search engines) บางครั้งอาจเรียกว่า สไปเดอร์ (spider) หรือ คอรัลเลอร์ (crawler) เป็นเว็บโรบอทที่วิ่งเก็บเว็บไซต์ต่าง ๆ อย่างอัตโนมัติ เช่น ฮอทบอท (Hotbot) เป็นต้น
- 2 ไดเรกทอรี (Directories) เป็นการแบ่งกลุ่มของเว็บไซต์ที่ใช้มนุษย์ โดยเว็บไซต์ที่จะถูกจัดกลุ่มจะต้องถูกส่งไปให้ตัวค้นหาข้อมูลเพื่อทำการแบ่งแยกหมวดหมู่ต่อไป เช่น ยาฮู (Yahoo) เป็นต้น
- 3 ไฮบริดเซิร์สเอนจิน (Hybrid search engines) เป็นเซิร์สเอนจินที่มีการจัดการแบบไดเรกทอรี

การทำงานในส่วนต่าง ๆ ของตัวค้นหาข้อมูล ประกอบด้วย 3 ส่วนคือ

ส่วนที่ 1 : สไปเดอร์ หรือบางครั้งเรียก คอรัลเลอร์ โดยสไปเดอร์จะทำการวิ่งไปยังเว็บเพจจากนั้นก็ทำการอ่านเพจนั้นแล้วลิงค์ (link) ตามไปยังเพจอื่น ๆ ที่มีในไซท์นั้น สไปเดอร์จะกลับไปยังเพจที่ได้ไปมาแล้วทุก ๆ ช่วงเวลาหนึ่ง ๆ ตามแต่จะกำหนดเพื่อตรวจสอบว่ามีการเปลี่ยนแปลงภายในเพจนั้นหรือไม่

ส่วนที่ 2 : สิ่งที่ได้จากการที่สไปเดอร์เดินทางไปยังเว็บเพจต่าง ๆ จะถูกนำเข้าสู่ส่วนที่ 2 ของตัวค้นหาข้อมูล นั่นคือ ส่วนของดัชนี (index) หรือบางทีอาจเรียกว่าแคตตาล็อก (catalog) โดยดัชนีนี้เปรียบได้กับสมุดเล่มใหญ่ ๆ เล่มหนึ่งซึ่งเก็บรวบรวมสำเนาของทุก ๆ เว็บเพจที่สไปเดอร์เดินทางไป ถ้าเว็บเพจมีการเปลี่ยนแปลงเกิดขึ้นแล้วสมุดเล่มนี้ก็ต้องมีการแก้ไขเพื่อให้ได้ข้อมูลใหม่ที่ถูกต้อง

ส่วนที่ 3 : ซอฟต์แวร์ของตัวค้นหาข้อมูล (search engine software) เป็นโปรแกรมที่ใช้จัดการกับเพจต่าง ๆ กว่า ล้านเพจที่มีอยู่ในดัชนีเพื่อให้สอดคล้องสัมพันธ์กับลำดับความสำคัญตามที่ได้กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดลำดับความสัมพันธ์ของเว็บเพจ

เนื่องจากตัวค้นหาข้อมูลไม่ใช่บรรณารักษ์ ดังนั้นจึงไม่สามารถถามผู้ใช้บริการได้ว่าต้องการค้นหาอะไรเพื่อหาขอบเขตของการค้นหาได้อีกทั้งไม่สามารถตัดสินใจและใช้ประสบการณ์ที่ผ่านมาในการบอกว่าเพจใดมีความสัมพันธ์ตรงตามความต้องการของผู้ใช้บริการที่สุดตามลำดับได้ซึ่งเป็นวิธีการการทำงานของมนุษย์จึงกลายเป็นความต้องการที่จะทำให้เอเจนต์ที่ฉลาดทำการเลียนแบบการทำงานของมนุษย์

วิธีการที่จะให้ตัวค้นหาข้อมูลตรวจสอบความสัมพันธ์นี้จึงเป็นไปตามกฎชุดหนึ่งซึ่งมีความเกี่ยวข้องกับกับบริเวณ (location) และความบ่อย (frequency) ที่จะสามารถพบคีย์เวิร์ด (keyword) นี้ในเว็บเพจ โดยวิธีการนี้จะกำหนดไว้ว่าคีย์เวิร์ดที่ปรากฏอยู่ในไตเติ้ล (Title) ของเว็บเพจถือว่าตรงตามความต้องการ มากกว่าที่ปรากฏอยู่ในส่วนอื่น ๆ ของเพจ ตัวค้นหาข้อมูลจะตรวจสอบดูด้วยว่าถ้าพบคีย์เวิร์ดอยู่ในบริเวณใกล้ ๆ ส่วนบนของเว็บเพจ เช่น ปรากฏในเฮดไลน์ (headline) หรือปรากฏในย่อหน้าแรก ก็จะเข้าใจได้ว่า จากคีย์เวิร์ดนั้นเราสามารถค้นหาเพจที่สัมพันธ์กัน ได้ถูกทางแล้ว

ความบ่อยที่พบคีย์เวิร์ดนั้นในเพจก็เป็นอีกปัจจัยหนึ่งที่สำคัญในการค้นหาเพจที่มีความสัมพันธ์กับคีย์เวิร์ดนั้น เช่นกัน กล่าวคือ ยิ่งพบว่าถ้ามีคีย์เวิร์ดนั้นปรากฏในเว็บเพจบ่อยมาก ๆ ก็ยิ่งแสดงว่าเพจนั้นมีความสัมพันธ์กับคีย์เวิร์ดที่ให้มามากกว่าเพจอื่น ๆ ที่ปรากฏคีย์เวิร์ดนั้นน้อยกว่า

โครงการนี้ประกอบด้วยการทำงานใน 4 ส่วนหลักดังนี้

- 2.1 ส่วนดึงข้อมูลและวิเคราะห์ข้อมูล
- 2.2 ส่วนเชื่อมต่อระหว่างผู้ใช้บริการและส่วนจัดเก็บข้อมูล
- 2.3 ส่วนจัดเก็บและค้นหาข้อมูล
- 2.4 ส่วนหน้าจอรับข้อมูลและแสดงผล

2.1 ส่วนดึงข้อมูลและวิเคราะห์ข้อมูล

เป็นส่วนการทำงานของผู้ช่วยค้นหา (search agent) ทำหน้าที่ดังนี้ คือ

- 1 ทำการท่องไปบนเว็บเพจต่าง ๆ
- 2 ตรวจสอบว่าเว็บเพจนั้นอยู่ในขอบเขตที่กำหนดหรือไม่
- 3 ทำการดึงข้อมูลกลับมา
- 4 ทำการคำนวณและวิเคราะห์ข้อมูลที่ดึงกลับมา

โดยในส่วนของการทำงานขั้นตอนนี้จะเป็นการทำงานของโปรแกรมประยุกต์ที่เรียกว่า ไรบอท ดังนั้นจึงทำการศึกษาถึงทฤษฎีและหลักการของไรบอท (ซึ่งเป็นประเภทหนึ่งของเอเจนต์) ดังมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 ความรู้เกี่ยวกับเอเจนต์

สำหรับเทคโนโลยีทางคอมพิวเตอร์ซึ่งมีการพัฒนาอย่างรวดเร็วในทุกวันนี้ นักวิจัย รวมทั้งบรรดาบริษัทเอกชนต่าง ๆ ได้พยายามพัฒนาเทคโนโลยีทางคอมพิวเตอร์ขึ้นมา เพื่อช่วยให้มนุษย์สามารถทำงานได้สะดวก รวดเร็ว และลดงานที่ซ้ำซ้อน ไม่จำเป็น งานต่าง ๆ เหล่านี้แน่นอนว่าจะต้องเป็นงานที่ใช้คอมพิวเตอร์เป็นหลัก และเป็นงานที่โปรแกรมคอมพิวเตอร์สามารถทำงานได้ดีมีประสิทธิภาพเท่าเทียมหรือมากกว่ามนุษย์ โดยโปรแกรมคอมพิวเตอร์เหล่านี้จะถูกเรียกรวม ๆ ว่า “โปรแกรมตัวแทนผู้ช่วย” (Software Agents) จึงสามารถอธิบายได้ว่า เอเจนต์ เป็นโปรแกรมที่มนุษย์ให้ข้อมูลและเป้าหมายในการทำงานใด ๆ แล้ว เอเจนต์ นั้นก็นำเอาข้อมูลที่ได้และรายละเอียดที่เก็บเป็นเร็คคอร์ด ภายในตัวมันเองมารวมกันและประมวลผลเพื่อทำงานที่มนุษย์สั่งให้ทำ โดยเอเจนต์ นั้นจะมีสิทธิอำนาจตามแต่ที่ผู้เป็นเจ้าของหรือผู้สั่งงานมอบให้ และเอเจนต์ จะมีการทำงานภายใต้สภาพแวดล้อมของมัน

ดังนั้นเอเจนต์ คือโปรแกรมคอมพิวเตอร์ที่จำลองรูปแบบความสัมพันธ์ของมนุษย์ โดยเป็นการกระทำในลักษณะของการที่บุคคลอื่นสามารถทำสิ่งใดให้เราได้ นั่นเอง

เอเจนต์บนอินเทอร์เน็ตมีอยู่ 4 ประเภทด้วยกันคือ

1. Web robots , spiders , and wanderers
2. Web commerce agents
3. Worms and viruses
4. MUD agents and chatterbots

เว็บโรบอท (Web Robot) , สไปเดอร์ (Spider) , วันเดอร์อร์ (Wanderer)

เป็นโปรแกรมที่ท่องไปตามข้อมูลในเวิร์ลไวด์เว็บ (www) โปรแกรมเหล่านี้จะย้ายจากเว็บหนึ่งไปยังอีกเว็บหนึ่งโดยอาศัยลักษณะของ ไฮเปอร์ลิงค์ (Hyperlink) ซึ่งมีในเว็บเพจ เว็บโรบอทเหล่านี้จะอาศัย เอชทีทีพี โพรโทคอล (HTTP Protocol) ของเวิร์ลไวด์เว็บในการที่จะดึงเอกสารจากเซิร์ฟเวอร์ต่าง ๆ เพื่อค้นหาแหล่งข้อมูลใหม่ ๆ เพื่อนำมาจัดทำดัชนีในการค้นหาข้อมูล

เว็บคอมเมิร์ซ (Web Commerce Agent)

เป็นเอเจนต์ซึ่งจะทำหน้าที่ช่วยในการตัดสินใจเลือกบริการหรือซื้อสินค้าบนระบบออนไลน์ (on-line) โดยทำการเลือกสิ่งที่ดีที่สุด เหมาะสมที่สุดจะมีเอกสารและรายละเอียดสินค้าและบริการต่างๆ ของบริษัทและร้านค้าต่าง ๆ เอเจนต์จะทำหน้าที่ติดต่อเสมือนพ่อค้าคนกลางในการเลือกใช้บริการหรือซื้อสินค้าบนระบบออนไลน์นี้

เวิร์ม (Worm) , ไวรัส (Virus)

เป็นเอเจนต์ที่มีเจตนาไม่ดีและไม่เป็นที่ปรารถนา หนอนและไวรัสจะจำลองตัวเองในรูปแบบที่หลีกเลี่ยงจากการตรวจสอบจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง หรือ จากระบบเครือข่ายหนึ่งไปยังอีกเครือข่าย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้างหนึ่ง ในสมัยก่อนมักจะพบโปรแกรมเหล่านี้บนดิสก์แต่ทุกวันนี้แม้แต่ในอินเทอร์เน็ตก็ไม่อาจรอดพ้นจากโปรแกรมเหล่านี้ ซึ่งโปรแกรมเหล่านี้มักจะนำความเสียหายมาสู่ระบบและข้อมูลต่าง ๆ

เอ็มยูดีเอเจนท์ (MUD Ageng) , แชตเตอร์บอท (Chatter Bot)

เอ็มยูดีเอเจนท์ เป็นโปรแกรมที่มีประโยชน์ในการให้บริการต่าง ๆ สำหรับมนุษย์ เช่น ตอบสนองที่ผู้ใช้ต้องการหรือให้คำแนะนำ โดยผ่านทางยูสเซอร์อินเทอร์เฟซ (User Interface) ที่ใช้ภาษาเฉพาะ ส่วนแชตเตอร์บอท เป็นเอเจนท์ที่สามารถสนทนา โดยหน้าที่หลักคือ สื่อสารกับคนที่ใช้โปรแกรมอยู่โดย แชตเตอร์บอท ไม่เหมือนกับเอ็มยูดีเอเจนท์ตรงที่ แชตเตอร์บอท จะกว้างกว่าเอ็มยูดีเอเจนท์ที่เป็นเฉพาะทางมากกว่า

2.1.2 คุณสมบัติของเอเจนท์ (Agent Attributes)

โปรแกรมเอเจนท์แตกต่างจากโปรแกรมทั่วไป กล่าวได้ว่า เอเจนท์ทุกเอเจนท์เป็น โปรแกรม แต่แต่ละโปรแกรมไม่จำเป็นต้องเป็นเอเจนท์เสมอไป ซึ่ง โปรแกรมที่จะเป็นเอเจนท์ได้จะต้องมีคุณสมบัติดังนี้

- 1 เป็นอัตโนมัติ คือสามารถเริ่มต้น จบ และควบคุมการทำงานโดยตัวมันเองได้
- 2 มีความฉลาด เพราะเอเจนท์จะมีความทำงานแบบ AI ซึ่งทำให้มันสามารถตัดสินใจในการทำงานได้โดยจะต้องมีการเรียนรู้และคิดแปลงให้เหมาะสม
- 3 ความหนักแน่นและแน่นอนที่จะทำงานให้สำเร็จผล โดยไม่มีการเปลี่ยนแปลงการทำงานไปจากที่ได้รับมอบหมาย
- 4 เดินทางไปตามสถานที่หรือเครื่องอื่น ๆ เพื่อเรียกใช้ทรัพยากรได้
- 5 ยืดหยุ่นการทำงานได้ เช่นมีการจัดลำดับความสำคัญของงาน
- 6 ติดต่อสื่อสารกับเครื่องอื่น ๆ ที่เดินทางไปใช้ข้อมูล ติดต่อสื่อสารกับเอเจนท์ด้วยกันเองและติดต่อสื่อสารกับผู้ใช้งานมันเองได้
- 7 มีระบบรักษาความปลอดภัยและดูแลรับผิดชอบทรัพย์สินของผู้ใช้งานได้ : กรณีที่เป็นเอเจนท์ในเรื่องการเงิน
- 8 มีปฏิกริยาโต้ตอบ สามารถสังเกต รู้สึก และมีปฏิกริยาตอบสนองต่อการเปลี่ยนแปลงในสภาพแวดล้อมของมันและมีการตัดสินใจว่าจะทำอย่างไรกับการเปลี่ยนแปลงนั้น

บางเอเจนท์จะมีไม่ครบทุกคุณสมบัตินี้และบางเอเจนท์อาจจะมีความสมบัตินอกเหนือจากนี้ ซึ่งนั้นก็ขึ้นอยู่กับกรอบการออกแบบการใช้งานของเอเจนท์

เอเจนท์มีอยู่หลายประเภท แต่เอเจนท์ที่นำมาใช้ในโรงงาน คือ โรบอทประเภทเว็บโรบอท ดังนั้นทางผู้วิจัยจึงทำการศึกษาคุณสมบัติ การทำงาน รวมทั้งข้อกำหนดต่าง ๆ ที่เกี่ยวกับการทำงานของโรบอทประเภทเว็บโรบอท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 คุณสมบัติและการทำงานของเว็บโรบอท

การนำเว็บโรบอทไปใช้งาน

เว็บโรบอทในปัจจุบันได้มีการนำมาใช้งานกันอย่างแพร่หลายมาก โดยในระยะเริ่มแรกนั้น ได้มีการนำมาใช้เพื่อตรวจวัดอัตราการเจริญเติบโตของอินเทอร์เน็ต โดยใช้โรบอทที่สร้างขึ้นทำการนับจำนวนเว็บไซต์ ที่เพิ่มขึ้นมา ต่อมาได้มีการนำโรบอทมาใช้งานทางด้านอื่น ๆ มากขึ้น

เราสามารถแบ่งการใช้งานเว็บโรบอทออกได้เป็น 3 อย่าง

- 1 เว็บรีซอร์สดีคัฟเวอรี (Web Resource Discovery)
- 2 เว็บเมนเทนแนนซ์ (Web Maintenance)
- 3 เว็บไมร์เรอริง (Web Mirroring)

เว็บรีซอร์สดีคัฟเวอรี

เป็นตัวที่เข้ามาเกี่ยวข้องกับงานในการแก้ไขปัญหาในการทำการค้นหาข้อมูลอันเป็นประโยชน์ภายในเว็บที่มีอยู่เป็นจำนวนมาก เนื่องจากค่าใช้จ่าย การเป็นข้อมูลแบบกระจายการเปลี่ยนแปลงที่เกิดขึ้นบ่อย ๆ และลักษณะเฉพาะของเว็บทำให้ผู้ที่เข้ามาใช้งานมีความสนุกสนาน แต่ก็ต้องได้รับความลำบากอย่างมากในการที่จะต้องทำการค้นหาข้อมูลเนื่องจากจะทำการค้นหาได้ยาก

ดังนั้นจึงได้มีการคิดค้นการทำตัวค้นหาข้อมูลขึ้นมาเพื่อใช้ในการช่วยหาข้อมูลซึ่งตัวค้นหาข้อมูลนี้ก็จำเป็นที่จะต้องใช้โรบอทเข้ามาช่วย ซึ่งโดยทั่วไปแล้วมักจะเรียกว่าสไปเดอร์ หรือเว็บโรบอท ซึ่งเป็นโปรแกรมที่ทำการท่องไปตามเว็บต่าง ๆ อย่างอัตโนมัติ แล้วนำข้อมูลกลับมาทำเป็นดัชนีเพื่อใช้ในการค้นหา

เว็บเมนเทนแนนซ์

สิ่งหนึ่งซึ่งเป็นการยากในการที่จะดูแลข้อมูลบนเว็บ เนื่องจากต้องมีการทำการตรวจสอบ อยู่ตลอดเวลาว่าเว็บที่เราต้องการเชื่อมต่อไปนั้น ได้มีการเปลี่ยนแปลงข้อมูลหรือมีการลบทิ้งไปหรือไม่ ซึ่งหากมีการทำการเคลื่อนย้ายหรือทำการลบเว็บที่มีการเชื่อมต่อ ถ้าทิ้งไปก็อาจทำให้เกิดกรณีการขาดการเชื่อมต่อ

ในปัจจุบันนี้ยังไม่มีมีการทำการตรวจสอบเรื่องดังกล่าว โดยอาศัยการทำงานแบบอัตโนมัติดังนั้นบางเซิร์ฟเวอร์ จะหยุดการทำงาน เอ็ชทีทีพีรีเควส (HTTP Request) เนื่องจากการเกิดการขาดการเชื่อมต่อขึ้นอยู่กั ยูอาร์แอล (URL) ตัวแรกที่เราจะทำการติดต่อคือในขณะที่มีการคืนค่ากลับมาด้วย ดังนั้นข้อมูลที่อยู่ภายในล็อกไฟล์ (log file) ของเซิร์ฟเวอร์ ก็สามารถที่จะนำมาใช้ในการบอกว่ามีรายการของเว็บใดบ้างที่มีการขาดการเชื่อมต่ออยู่แล้ววิธีดังกล่าวนี้ก็ไม่ใช่วิธีที่ดี เนื่องจากเว็บต่าง ๆ ก่อนข้างที่จะมีความเป็นอิสระ จึงเป็นการยากที่จะทำการตรวจสอบการติดต่อระหว่าง เว็บไซด์ต่าง ๆ ได้ทั้งหมดจากการใช้โรบอทในการทำการตรวจสอบ

ดังนั้นการทำงานทางด้านนี้จึงมักเป็นที่นิยมในการที่จะใช้งาน โรบอทประเภทที่เรียกว่า เว็บเมนเทนแนนซ์ สไปเดอร์ ซึ่งจะเป็โปรแกรมที่ทำหน้าที่ช่วยผู้ที่เป็เจ้าของเว็บและเว็บมาสเตอร์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุผลแบบสงวนเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Web Master) ในการที่จะทำการดูแลรักษา โครงสร้างต่าง ๆ ของเว็บ โดยจะทำการท่องโดยอัตโนมัติไป ตามการเชื่อมต่อด่าง ๆ ที่เกี่ยวข้องแล้วทำการตรวจสอบหา การขาดการเชื่อมต่อด่างว่ามีปรากฏหรือไม่

เว็บไมร์เรอริง

ไมร์เรอริงเป็นเทคนิคทั่วไปที่ใช้การทำการติดตั้ง สำหรับโครงสร้างของข้อมูล ตัวอย่างเช่น ไมร์เรอริง เอฟทีพี ไซต์ (Mirroring FTP Site) ที่เกี่ยวข้องกับการทำการคัดลอกเอฟทีพีไฟล์ทั้งหมด โดยอาศัยการทำงานแบบการทำซ้ำ (Recursive) และทำการสร้างตัวมันเองขึ้นมาอีกในเครื่องอีกเครื่องหนึ่ง ภายในระบบเครือข่ายเอฟทีพีไซต์ ที่เป็นที่รู้จักกันโดยทั่วไปมักจะถูกทำการจำลอง (Mirror) ไปยังส่วนต่าง ๆ ของระบบเครือข่าย ซ้ำแล้วซ้ำเล่าเพื่อทำการร่วมใช้ข้อมูล ไปยังส่วนที่ต้องการ แต่ก็อาจเป็นการเสี่ยงที่จะทำให้เกิดการหยุดการทำงานได้ง่าย

โรบอทที่ทำหน้าที่ตามหน้าที่ของเว็บไมร์เรอริงนั้นในปัจจุบันมีอยู่แล้วซึ่งจะเป็น โปรแกรม ที่ทำการคัดลอกได้ดีกว่าที่มีอยู่แล้วในการที่จะทำการคัดลอกข้อมูลต่าง ๆ จากเอฟทีพีไซต์ แต่อย่างไรก็ตามโรบอทที่ทำหน้าที่เป็นเว็บแมนเทนแนนซ์ ก็ไม่สามารถที่จะทำการตรวจสอบการเปลี่ยนแปลงข้อมูลต่าง ๆ บนเว็บไซต์ได้

2.1.4 กฎของโรบอท

ในการที่จะสร้างโรบอทขึ้นมาใช้งานสักตัวหนึ่ง เราควรที่จะทำตามข้อกำหนดต่าง ๆ ต่อไปนี้ ซึ่งได้รวบรวมข้อควรระวังและข้อควรปฏิบัติสำหรับโรบอท โดยทำการรวบรวมข้อมูลโดยเว็บมาสเตอร์ที่มีความชำนาญและผู้เชี่ยวชาญทางด้านโรบอท ซึ่งกฎทั้ง 4 ข้อ มีดังต่อไปนี้

- 1 เว็บโรบอทต้องประกาศตัวอย่างชัดเจน
- 2 เว็บโรบอทต้องปฏิบัติตามข้อกำหนดมาตรฐาน
- 3 เว็บโรบอทต้องไม่ยึดครองแหล่งข้อมูลแบบถาวร
- 4 เว็บโรบอทต้องแสดงข้อผิดพลาดเมื่อเกิดความผิดพลาด

เว็บโรบอทต้องประกาศตัวอย่างชัดเจน

เว็บมาสเตอร์ผู้ที่ทำการดูแลเซิร์ฟเวอร์ต้องการที่จะรู้ว่าโรบอทที่เข้ามาติดต่อด้วยนี้ มีหน้าที่ในการทำการเข้าถึงข้อมูลแบบใด และใครเป็นผู้เขียนเพื่อในกรณีที่เกิดปัญหาขึ้นจะ ได้สามารถติดต่อกลับไปได้ และนอกจากเว็บมาสเตอร์มักจะอยากรู้ว่าโรบอทเหล่านี้รู้จักไซต์ของพวกเขาได้อย่างไร ดังนั้นเว็บโรบอท สามารถที่จะบอกแก่เว็บมาสเตอร์ได้โดยการ กำหนดค่าต่าง ๆ ในฟิลด์ต่อไปนี้ ยูสเซอร์เอเจนท์ ฟิสต์ , ฟอรัมฟิลด์และเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การแสดงตัวของเว็บโรบอท

เว็บไคลเอนต์ที่สามารถที่จะกำหนดในการแสดงค่าเกี่ยวกับตัวเองได้โดยใช้ยูสเซอร์เอเจนต์ฟิลด์ที่สนับสนุนในเอชทีทีพีรีควส อย่างเช่นภายในเน็ตสเคพ (Netscape) สามารถที่จะเรียกใช้โมซิลลา (Mozilla) ที่มีอยู่ของมันเองได้ดังนี้

User-agent : Mozilla / 1.1 N

ดังนั้นเว็บโรบอทสามารถที่จะใช้ฟิลด์ ยูสเซอร์ – เอเจนต์ ในการที่จะแสดงชื่อและบอกเกี่ยวกับเวอร์ชันต่าง ๆ ดังตัวอย่าง

User-agent : Terminator / 1.0

ซึ่งยูสเซอร์ – เอเจนต์ ฟิลด์ นี้จะช่วยให้เว็บมาสเตอร์ ในการกำหนดให้สามารถถูกใช้งานได้โดยคนที่ใช้งานเว็บเบราว์เซอร์

- การแสดงคำสั่งการทำงานของโรบอท

เอชทีทีพี จะมีรูปแบบฟิลด์อยู่ภายในส่วนของรีควสเฮดเดอร์ เพื่อจะใช้ในการกำหนดหน้าที่ต่าง ๆ ที่ได้รับมา ซึ่งโดยทั่วไปมักจะใช้อีเมลแอดเดรส (e-mail address) ในการกำหนดค่าที่ฟิลด์นี้ ดังตัวอย่างต่อไปนี้

Form : joe.robomas@roboiland.com

ซึ่งค่าดังกล่าวนี้จะช่วยให้เว็บมาสเตอร์สามารถติดต่อกับผู้ที่ใช้งาน โรบอทตัวดังกล่าวอยู่ได้ในกรณีที่เกิดปัญหาขึ้น นั่นทำให้ผู้ที่สร้าง โรบอทหรือใช้งาน โรบอทอยู่สามารถที่จะติดต่อกลับไปยังเว็บมาสเตอร์ได้

- การอ้างอิงเว็บเพจ

เว็บมาสเตอร์ มักจะแปลกใจว่าคนทั่วไปสามารถเข้ามาเรียนรู้เกี่ยวกับเว็บไซต์ของพวกเขาได้อย่างไร เมื่อโรบอทมีการเข้าถึงข้อมูลในเว็บเพจต่าง ๆ มันจะเป็นการดีมากสำหรับเว็บโรบอทที่จะทำการบอกแก่เว็บเซิร์ฟเวอร์ ซึ่งเป็นผู้ดูแลเอกสารดังกล่าวอยู่ ซึ่งเอกสารนี้เราเรียกว่า เว็บเพจรีเฟอเรน เอชทีทีพี จะมีฟิลด์ สำหรับใช้ในการกำหนดเอกสาร ซึ่งข้อมูลเหล่านี้จะแจ้งแก่เว็บมาสเตอร์ให้รู้ว่าเว็บเพจที่กำลังถูกดึงข้อมูลอยู่นี้ได้รับการรีเฟอเรนมาจากตัวอย่าง

referer : [HTTP://WWW.referRus.com/launchpad.html](http://WWW.referRus.com/launchpad.html)

เว็บโรบอทต้องปฏิบัติตามข้อตกลง

มาตรฐานสำหรับโรบอท เอ็กซ์คลูชันนี้ถูกกำหนดขึ้นโดย มาร์ติน คอสเตอร์ ในเรื่องที่เกี่ยวข้องกับการติดต่อระหว่างเว็บเซิร์ฟเวอร์กับเว็บโรบอท ซึ่งมีข้อกำหนดว่าโรบอทตัวใดจะมีสิทธิบ้างและตัวใดมีสิทธิในการเข้าถึงข้อมูลอย่างน้อยเพียงใด ดังตัวอย่างข้างต้นมีการจำกัดไม่ให้โรบอท เข้าไปถึงข้อมูลในเว็บต้องห้าม ซึ่งเป็นสิ่งที่เว็บโรบอทจะต้องปฏิบัติตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บโรบอทต้องไม่ยึดครองแหล่งข้อมูลแบบถาวร

เว็บโรบอทเป็นตัวที่ช่วยในการหาแหล่งข้อมูลโดยจะทำให้อินเทอร์เน็ตขนาดใหญ่ดูเล็กลง ดังนั้นการที่เว็บโรบอทจะทำงานได้ดีควรมีการกำหนดสิ่งที่มันต้องการไว้อย่างชัดเจนดังต่อไปนี้

- เอ็ชทีทีพี จะมีเฮด ริเควส เมตรอดที่จะดึงข้อมูลเฉพาะเฮดเดอร์ ในรูปแบบแอกชั่นจากเว็บ เอกสาร โดยไม่นำส่วนของตัวเอ็ชทีเอ็มแอล (HTML) กลับมาด้วย ซึ่งการใช้สิ่งนี้ทำให้เราสามารถลด โอเวอร์เฮด (Overhead) ที่เกิดขึ้นได้แตกต่างจากได้รับริเควสที่จะทำการดึงข้อมูลทั้งเอกสาร การใช้สิ่งนี้ จะเป็นประโยชน์ต่อโรบอท ในการที่จะทำงานกับเฉพาะที่มีอยู่โดยไม่ต้องทำไปตามการเชื่อมต่อทั้งหมด ที่มีอยู่ในไฮเปอร์ลิงค์ คอนเทนท (Hyperlink content)

- เอ็ชทีทีพี จะมีฟิลด์ที่กำหนดในริเควส เฮดเดอร์ สำหรับเว็บโรบอทในการที่จะกำหนดค่าให้ แก่เซิร์ฟเวอร์ว่ามีข้อมูลชนิดใดบ้างที่มันสามารถเข้าถึงสำหรับ โรบอทที่ถูกออกแบบมาเพื่อที่จะทำการ วิเคราะห์ในรูปแบบแอกชั่นเท่านั้นจะกำหนดได้ดังนี้

Accept : x – text

การกำหนดสิ่งที่ต้องการจะช่วยให้ลดช่วงกว้างของระบบเครือข่ายที่จะต้องพิจารณาเนื่องจากเว็บ เซิร์ฟเวอร์ จะไม่จัดส่งข้อมูลในประเภทที่โรบอทไม่สามารถเข้าถึงได้

- ยูอาร์แอล ที่อยู่ในตอนท้ายจะเป็นตัวที่ช่วยบอกว่ามีข้อมูลชนิดใดที่อยู่สุดท้ายของการเชื่อมต่อ ถ้าไฟล์ที่มีส่วนนามสกุล ที่เป็น ps , zip , z , gif ซึ่งจะเป็นไฟล์ที่ไม่ได้รับความสนใจเนื่องจาก โรบอท สามารถที่จะติดต่อกับเฉพาะข้อมูลที่เป็นเท็กซ์

เว็บโรบอท มักจะเสี่ยงกับการที่จะต้องเข้าไปในส่วนต้องห้าม ดังนั้น เว็บโรบอท จะต้องมีการกำหนดรายการ ที่ควรหลีกเลี่ยงดังตัวอย่าง ยูอาร์แอลที่ขึ้นต้นด้วย news และ WAIS ควรจะถูกตัด ออกจากส่วนที่เว็บโรบอทจะผ่านเข้าไป นอกจากนี้ยังต้องระวังในเรื่องของซบเพจรีเฟรชเพื่อเป็นการ หลีกเลี่ยงไม่ให้มีการดึงข้อมูลจากเอกสารเดียวกันมากกว่าหนึ่งครั้ง

- บางระบบจะมีช่วงเวลาที่เหมาะสมในการทำการดึงข้อมูลเพราะจะเป็นช่วงที่มีสิ่งข้อมูลน้อย ดังนั้นโรบอทจึงควรที่จะต้องคำนึงถึงสิ่งนี้ด้วย

- โรบอท ไม่ควรที่จะพิจารณาว่าเอ็ชทีเอ็มแอล เอกสารทั้งหมดไม่มีข้อผิดพลาดโดยเฉพาะใน ขณะที่มีมีการตรวจหา ยูอาร์แอลจะต้องทำการระวังสิ่งต่าง ๆ อย่างเช่นดังต่อไปนี้

A HREF = “ [HTTP://somehost.somedom/doc](http://somehost.somedom/doc) ”

ซึ่งเป็นการอ้างถึงสองครั้ง แต่ก็มีเว็บไซต์จำนวนมากที่ไม่ใช้เครื่องหมายกั้นใน ยูอาร์แอล สำหรับไคเร็ก ทอรี ซึ่งหมายความว่า วิธีในการหาชื่อของยูอาร์แอลใช้การอ้างอิงจากชื่อ

- เป็นการเสี่ยงต่ออันตรายอย่างมาก หากเกิดกรณีที่มีการวนรูปแบบไม่รู้จบขึ้นบนเว็บ โดยไม่ เคยได้คาดเดาไว้ก่อนว่าจะเกิดอะไรขึ้น และไม่สามารถเดาได้ว่าเกิดอะไรขึ้น ดังนั้นเพื่อป้องกันปัญหา ดังกล่าวนี้ เราจำเป็นต้องมีการเก็บค่าเอาไว้ว่า เว็บใดที่โรบอทได้ผ่านไปแล้ว นอกจากนี้ยังต้องมีการ ตรวจสอบถึงกรณีที่อาจจะเกิดการมีชื่อของเซิร์ฟเวอร์มากกว่า 1 เซิร์ฟเวอร์ อยู่บนเครื่องเดียวกันนั่นคือมี เอกสารไอพี แอดเดรสเดียวกันสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถึงแม้ว่าเว็บโรบอทจะสามารถที่จะเข้าถึงข้อมูลได้คราวละหลายร้อยเอกสารต่อนาทีแต่ปัญหาในการใช้งานอย่างหนัก และมีการเข้าถึงหลาย ๆ อย่างพร้อม ๆ กันทำให้เซิร์ฟเวอร์ดึงข้อมูลมาก ทำให้โรบอทต้องใช้การทำการค้นตามหลักการราวด์ – โรบิน (round – robin) และอาจจะต้องหยุดการทำงานในช่วงเวลาสั้น ๆ ในบางครั้ง ดังนั้นการที่เราจะให้โรบอทสามารถที่จะทำการดึงข้อมูลเพียง 1 เอกสารต่อ 1 นาที ในแต่ละเซิร์ฟเวอร์จึงเป็นวิธีการที่น่าสนใจมากกว่าการทำการเข้าถึงทำละหลาย ๆ เอกสาร
- บางเว็บสามารถที่จะทำการค้นหาได้โดยใช้ไอเอสอินเด็กซ์ (isindex) ที่มีอยู่ในเอชทีเอ็มแอล ในขณะที่บางเว็บไม่สามารถค้นหาได้ รวมทั้งยังมีการเปลี่ยนแปลงอยู่บ่อย ๆ ดังนั้น จึงไม่เป็นการง่ายและมีข้อมูลมากเพียงพอ ที่จะทำการวิ่งตามการเชื่อมต่อและทำการค้นหาว่าจะอยู่ที่ใด การวิเคราะห์ไฟล์บนเอชทีเอ็มแอลของเว็บเอกสารจะถูกกระทำโดยโรบอทที่จะช่วยกำหนดว่าจะหาข้อมูลที่ใด

เว็บโรบอทต้องมีการแสดงข้อผิดพลาด

เมื่อโรบอทมีการวิ่งไปตามเว็บต่าง ๆ อาจจะเกิดกรณีที่มีการวิ่งไปตามการขาดการเชื่อมต่อซึ่งชี้ไปยังเว็บที่ไม่สามารถที่เข้าไปได้ ซึ่งสาเหตุเหล่านี้ อาจเกิดจากการที่เว็บมาสเตอร์ทำการเคลื่อนย้ายเว็บเพจไปยังที่ใหม่ โดยการย้ายไปยังเครื่องใหม่หรือทำการย้าย / เปลี่ยน ใดเรียกทอร์ไป นอกจากนี้ยังมีสาเหตุมาจากการที่มีการเปลี่ยนชื่อของเอกสารรวมทั้งการเปลี่ยนโดเมนเนม (Domain Name) ของเซิร์ฟเวอร์

เมื่อเป็นเช่นนี้โรบอทควรที่ทำการส่งข่าวสารแสดงถึงข้อผิดพลาดไปยังแอดเดรสที่ระบุไว้ตรง เมล์บู (Mailto) เพื่อบอกแก่เว็บมาสเตอร์ผู้ดูแลไซต์ดังกล่าว

2.1.5 สิ่งที่ต้องปฏิบัติในการสร้างโรบอท

1 เพื่อเป็นการดีในการทำการติดต่อสื่อสาร ดังนั้นจึงเป็นการเหมาะสมที่จะมีการประกาศโรบอทก่อนที่จะมีการส่งโรบอทวิ่งออกไปใช้งานจริง

- ถ้าหากว่าเราได้ทำการประกาศโรบอทออกไปก่อนที่จะมีการใช้งานจริงจะทำให้เว็บมาสเตอร์ต่าง ๆ รู้ว่าจะมีโรบอทเข้ามาก็ไม่ทำการสังเกตหรือแปลกประหลาดใจต่อการทำงานของโรบอท ซึ่งจะเป็นการทำให้โรบอทเป็นที่ยอมรับและยินดีให้เข้าให้บริการข้อมูลต่าง ๆ ของเซิร์ฟเวอร์มากกว่าที่ไม่ได้มีการประกาศ ซึ่งการทำการประกาศใช้งานโรบอทที่เราสามารถ ทำได้โดยการส่งข่าวสารไปยังยูสเน็ตนิวกู๊ป (USENET newgroup)

comp.infosystem.www.providers

หรืออาจจะทำได้โดยการส่งอีเมลล์ ไปยังแอดเดรสข้างล่างนี้

robots@nexor.co.uk

ทางที่ดีควรจะมีการรวมข้อมูลเกี่ยวกับปัญหาต่าง ๆ ที่อาจจะเกิดขึ้นได้จากโรบอทส่งไปด้วย

- พิจารณาไซต์เป้าหมายหากว่าโรบอทมีการทำงานกับไซต์เพียงเล็กน้อยเราควรที่จะทำการส่งข่าวสารไปยังไซต์ปลายทาง เหล่านั้นเพื่อเป็นการติดต่อกับเว็บมาสเตอร์โดยตรง

- ทำการรายงานการทำงานของโรบอทที่จะเกิดขึ้นหรือบริการต่าง ๆ ที่โรบอทต้องการใช้แก่ผู้ดูแลเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนระบบที่ไซต์ที่เราติดต่อด้วยรับรู้ โดยเน้นเกี่ยวกับเรื่องของการเพิ่มการจราจรบนระบบเครือข่าย หรือความไม่várกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำเป็นในการที่จะต้องใช้เนื้อที่ (space) ในการทำงานเมื่อมีการทำงานของโรบอทเกิดขึ้น ซึ่งถ้าหากมีข้อผิดพลาดเกิดขึ้นผู้ดูแลระบบจะทำการเตือนและทำการแก้ปัญหาต่าง ๆ

2 ในการทดสอบการทำงานของโรบอทเราควรที่จะทำการตรวจสอบการทำงาน จากเว็บเซิร์ฟเวอร์ในระดับโลคอล (Locally) ก่อน ไม่ควรที่จะทำการทดลองกับเซิร์ฟเวอร์ที่ไกล ก่อนที่จะมีการทดสอบจนกระทั่งไม่มีข้อผิดพลาดอยู่ ดังนั้นเมื่อมีการทำการทดสอบในครั้งแรก เราก็ควรที่จะทำการทดสอบกับเซิร์ฟเวอร์ใกล้ ๆ โดยอาศัยโลคอล ยูอาร์แอล

เมื่อมีการทดสอบในระดับใกล้ ๆ แล้วควรที่จะทำการวิเคราะห์ถึงประสิทธิภาพ ในการทำงาน ของโรบอท รวมทั้งผลลัพธ์ที่ได้มาว่ามีความถูกต้องมากเพียงใด เพื่อนำสิ่งที่ได้จากการวิเคราะห์นี้ไปใช้เป็นแนวทางในการแก้ไขปรับปรุงสำหรับการแก้ไขให้เรียบร้อย

3 ขึ้นว่าจะทำงานอะไร อย่างไร รวมทั้งสามารถที่จะสามารถทำการควบคุมได้ ซึ่งสามารถทำได้ จากแนวทางต่อไปนี้

- มีการจัดทำล็อก (log) เพื่อทำการดูแล้ว ขณะนี้โรบอทอยู่ที่ใดรวมทั้งดูผลการทำงานต่าง ๆ ของโรบอทว่าทำงานคืบหน้าไปถึงไหน รวมทั้งเพื่อดูแลให้สามารถทำการควบคุมการทำงานของโรบอทได้ รวมทั้งโรบอทก็ต้องมีการควบคุมไม่ให้เกิดการทำงานแบบวนรอบและต้องดูแลการใช้เนื้อที่ต่าง ๆ (disk space requirement) ในแต่ละช่วงเวลาต่าง ๆ เพื่อเป็นประโยชน์ในการดูแลไม่ให้เกิดดิสก์ สเปซ ครันช์ (disk space crunch)

- ต้องออกแบบโรบอทให้สามารถทำงานได้ง่าย รวมทั้งมีการทำคำสั่งที่สามารถกำหนดเส้นทางของโรบอท , การยกเลิกการทำงานของโรบอท , การข้ามเซิร์ฟเวอร์ที่ไม่ต้องการและต้องมีการติดตามว่าการทำงานของโรบอทว่าหยุดการทำงานหรือผิดพลาดหรือไม่

เมื่อทำการรันการทำงานของโรบอท เราควรจะแน่ใจว่า เว็บมาสเตอร์ของไซต์ที่เราติดต่อกับ สามารถที่จะส่งข่าวสารหรือพูดคุยกับเราได้ ในกรณีที่จำเป็นเนื่องจากหากมีความผิดพลาดต่าง ๆ เกิดขึ้น จากการทำงานของโรบอท จะมีผู้เขียนโรบอทเพียงคนเดียวที่สามารถที่จะทำการแก้ไขได้รวดเร็วที่สุด ถ้าเป็นไปได้จึงควรที่จะทำการล็อกออน (log on) เครื่องที่ใช้ในการทำการรัน โรบอทเอาไว้ตลอดในช่วงที่ทำการรันโรบอทควรที่จะทำการรันโรบอทเฉพาะในช่วงที่เราอยู่เท่านั้น ไม่ควรทำการรันในกรณีที่เราจะไม่อยู่ในเวลานาน เช่น ไม่อยู่เป็นเวลา หนึ่งอาทิตย์

4 ระหว่างที่มีการทำงานของโรบอทอยู่โรบอทอาจจะผ่านไปยังเว็บไซต์จำนวนมาก อาจจะไปมีผลทำให้เว็บมาสเตอร์ มีความไม่พอใจในการทำงานของโรบอท ดังนั้นผู้ที่ทำการรันโรบอทจะต้องเตรียมพร้อมในการตอบสนองและพยายามอธิบายว่า โรบอทกำลังทำอะไรอยู่ เพื่อเป็นการป้องกันไม่ให้โรบอทมีการทำความไม่พอใจให้แก่เว็บมาสเตอร์ เราควรที่จะทำการกำหนดให้โรบอททำงานเฉพาะโฮมเพจของไซต์ดังกล่าว

5 การทำงานควรจะมีการรวบรวม เว็บไซต์ให้มากที่สุดเท่าที่จะมากได้ โดยการนำโรบอทไปทำงานบนอินเทอร์เน็ตที่มีขนาดใหญ่ แล้วนำข้อมูลต่อไปนี้ไปใช้ให้เกิดประโยชน์ในการใช้ เป็นสิ่งที่ช่วยในการค้นหา ซึ่งสามารถทำได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ร่วมใช้ข้อมูลที่ได้รับมาประกอบด้วยเว็บเพจ ที่ดึงกลับมาไม่ว่าจะมาจาก เอฟทีพีเวิร์ลไวด์เว็บ ซึ่งข้อมูลเหล่านี้จะถูกเก็บและถูกใช้โดยผู้ใช้อื่นเพื่อที่จะไม่ต้องมีการทำงานซ้ำอีกครั้งในการหาข้อมูล จากโรบอทตัวเดียวกัน

- เว็บโรบอทถูกนำมาใช้งานในงานเฉพาะทาง อาจจะถูกนำมาใช้ในการช่วยทำฐานข้อมูลหรือทำการรวบรวมสิ่งต่าง ๆ เพื่อนำมาใช้ประโยชน์ซึ่งจะช่วยให้โรบอทได้รับการยอมรับมากขึ้น มากกว่าการเป็นเพียงสิ่งที่ทำงานแล้วทำให้เซิร์ฟเวอร์มีการดึงข้อมูลมากขึ้น

2.1.6 ภาษาที่ใช้ในการเขียนโรบอท

จากหลักการเบื้องต้น เราจะนำไปใช้อ้างอิงในการออกแบบโรบอทที่จะนำมาใช้งานในโครงการนี้ โดยเลือกใช้ภาษาจาวา

เนื่องจากการพัฒนาของเทคโนโลยีด้านภาพ ความสามารถและความเร็วบนเว็บเบราว์เซอร์ได้เพิ่มขึ้นอย่างรวดเร็ว เราสามารถโหลดเว็บเพจได้ในเวลารวดเร็ว , ภาพจะถูกโหลดขึ้นมาพร้อมกันหลาย ๆ ภาพ นอกจากนี้ภาษาเฉพาะของเว็บที่เรียกว่า เอชทีเอ็มแอล ก็ได้ถูกเพิ่มเติมและเปลี่ยนเวอร์ชันเป็นเวอร์ชัน 3 ทำให้เว็บเพจที่ดูผ่านเบราว์เซอร์นี้ น่าตื่นตาตื่นใจและสวยงามกว่าเดิมมาก ไม่ว่าจะเป็นเรื่องของการควบคุมการติดต่อกับผู้ใช้ , ตัวอักษรและสีสรรที่สวยงาม

มีการพัฒนาภาษาโปรแกรมแบบใหม่เพื่อใช้สำหรับเว็บที่มีชื่อว่า จาวา โดย Sun Microsystem และทาง Sun ยังได้ออกเบราว์เซอร์ตัวใหม่ที่มีชื่อ HotJava ซึ่งใช้มาตรฐาน เอชทีเอ็มแอล 3.0 และสนับสนุนจาวาอย่างเต็มที่ ภาษาจาวาเป็นภาษาที่พัฒนามาจากภาษา C++ ซึ่งเป็นภาษาแบบออปเจ็ค จึงกล่าวได้ว่าภาษาจาวา มีคุณสมบัติดังนี้

- เป็นภาษาที่ง่าย (Simple)
- เป็นแบบออปเจ็ค (Object - orient)
- ประมวลผลแบบกระจาย (Distribute)
- ทำตามลำดับขั้น (Interprete)
- แข็งแกร่ง (Robust)
- มีความปลอดภัย (Secure)
- มีสถาปัตยกรรมที่เป็นกลาง (Architecture nurtral)
- เคลื่อนย้ายได้ (Portable)
- ประสิทธิภาพสูง (High Performance)
- มัลติเธรด (Multitred)
- ไม่อยู่นิ่ง (Dynamic)

2.1.7 คุณสมบัติในเรื่องพอร์ตเทเบิลและความปลอดภัยของจาวา

พอร์ตเทเบิลของจาวาจะช่วยลดเวลาในการดูแลแก้ไขและการทำตลาดในเรื่องของแอปพลิเคชัน เอกสารที่หลายแพลตฟอร์มได้ โดยเพียงแต่เขียนแอปพลิเคชันเพียงครั้งเดียว หลังจากคอมไพล์แล้วแอปพลิเคชัน ไม่ว่าจะรันใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถจะรันได้ทุกแพลตฟอร์มที่มีจาวาบราวเซอร์ทำงานอยู่ ซึ่งหมายความว่าแอปพลิเคชันจะเป็นอิสระจากตัวรันใหม่ของแต่ละแพลตฟอร์มและคลาสไลบรารีที่ใช้ เช่น ภาษานี้จะไม่สนใจว่าผู้ใช้มีหน้าจอที่เขียนอยู่ทำงานอยู่บนวินโดวส์หรือ MacOS ผู้เขียนเพียงแต่เขียนคำสั่งหรือเขียนการแสดงผลที่ต้องการบนหน้าจอโดยใช้เอททริบิวต์ปกติเท่านั้น แม้การเขียนออปเจกต์แบบกลาง ๆ นี้จะทำให้ผู้เขียนไม่สามารถใช้คุณสมบัติที่มีในแต่ละแพลตฟอร์มได้ แต่วิธีนี้ช่วยเพิ่มคุณสมบัติพอร์ตเทเบิลให้กับแอปพลิเคชันได้ เมื่อมีแพลตฟอร์มใหม่ ๆ เกิดขึ้นเพียงแต่ย้ายตัวรันใหม่จาวาไปบนแพลตฟอร์มใหม่นี้เท่านั้น แอปพลิเคชันก็จะสามารถทำงานบนแพลตฟอร์มใหม่นี้ได้

จาวาเป็นภาษาแบบอินเทอร์พรีเตอร์ที่แตกต่างจากตัวอื่น ๆ เพราะมันสามารถทำงานจากซอร์สโค้ดที่มีอยู่ได้ ภาษาจะถูกคอมไพล์ให้เป็นภาษากลางที่เรียกว่า architecture – neutral bytecode ซึ่งมีขนาดเล็กแต่ง่ายต่อการแปลงเพื่อนำไปใช้งานมากกว่าซอร์สโค้ด หลังจากคอมไพล์เพียงครั้งเดียว โปรแกรมนี้จะถูกเก็บในเว็บเซิร์ฟเวอร์จนกว่าจะมีบราวเซอร์ที่สนับสนุนจาวาติดต่อเข้ามา และดาวน์โหลดมันไปแล้วตัวรันใหม่ในบราวเซอร์นั้นจะรันภาษากลางนี้เอง

เรื่องความปลอดภัยของจาวานั้นมีคุณสมบัติในการจำกัดและควบคุมการใช้ทรัพยากรที่ภาษาจะทำได้ เช่น แอปพลิเคชันที่เขียนขึ้นได้รับการอนุญาตให้สร้างคอนเนกชันกับโฮสต์ของตนบนเน็ตเวิร์กนั้นและไม่สามารถสร้างโปรเซสเซอร์ใหม่เองได้ ขอบเขตในการเข้าถึงไฟล์ในฝั่งโลกออนไลน์น้อยมาก

บราวเซอร์ที่สนับสนุนจาวาจะส่งโค้ดภาษานี้ไปยังตัวตรวจสอบไบนารีโค้ดของจาวา เพื่อตรวจสอบว่าโค้ดดังกล่าวไม่ผิดต่อกฎในเรื่องความปลอดภัย ต่อมาจึงรันตัวโหลดคลาสจาวาเพื่อตรวจสอบว่าโค้ดดังกล่าวไม่เรียกฟังก์ชันที่ไม่มีในคลาสที่มีอยู่

2.2 ส่วนเชื่อมต่อระหว่างผู้ใช้บริการและส่วนจัดเก็บข้อมูล

เป็นส่วนที่เชื่อมต่อการทำงานของโฮมเพจของตัวเองค้นหาข้อมูลกับผู้ใช้บริการ โดยจะทำหน้าที่ดังนี้

- ทำการรับตัวแปรและเงื่อนไขการทำงานของผู้ใช้บริการจากโฮมเพจ
- ส่งค่าตัวแปรและเงื่อนไขไปยังเซิร์ฟเวอร์ที่ให้บริการ
- รับค่าผลลัพธ์จากเซิร์ฟเวอร์ที่ให้บริการ แล้วส่งออกมาทางหน้าจอให้ผู้ใช้บริการ

การทำงานของส่วนเชื่อมต่อนี้จะอาศัยการทำงานของโปรแกรมที่เรียกว่า เซิร์ฟเล็ต ซึ่งมีหลักการและลักษณะการทำงานดังนี้

2.2.1 เซิร์ฟเล็ต (Servlet)

เซิร์ฟเล็ต คือ โปรแกรมที่จะถูกเรียกให้ปฏิบัติงานเฉพาะกิจกรรมหนึ่ง ๆ โดยร่วมกับเว็บเซิร์ฟเวอร์เมื่อมีการร้องขอมาจากเว็บไคลเอนท์ (เช่น การค้นหาข้อมูลจากระบบฐานข้อมูล) เซิร์ฟเล็ต มีแพลตฟอร์ม (Platform) เป็นภาษาจาวาอย่างสมบูรณ์ที่ใช้ทางฝั่งเซิร์ฟเวอร์เพื่อให้สามารถใช้งานร่วมกับจาวาเว็บเซิร์ฟเวอร์ (Java Web Server) ได้อย่างลงตัว โดยเซิร์ฟเล็ตจะเพิ่มความสามารถให้กับเซิร์ฟเวอร์ในด้านการลดโอเวอร์เฮด (minimal overhead), ในด้านการดูแลรักษา (maintenance), ในด้านการสนับสนุน (support) เนื่องจากเซิร์ฟเล็ตเป็นจาวาไบนารีโค้ด (Java bytecode) ทำให้สามารถดาวน์โหลด

โหลด (download) ผ่านเครือข่ายได้ เป็นไปตามคำกล่าวที่ว่า “ เขียนครั้งเดียว รันได้ทุกที่ ” (Write Once , Run Anywhere)

เซิร์ฟเล็ตจึงแตกต่างจากซีจีไอสคริปต์ (CGI scripts) ตรงที่เซิร์ฟเล็ตสามารถรันบนแพลตฟอร์มใดก็ได้ เพราะเป็นคอมโพเนนต์ (component) หนึ่งของจาวาที่สามารถดาวน์โหลดมาใช้ได้เมื่อต้องการ ดูรายละเอียดจากตาราง

| เซิร์ฟเล็ต | ซีจีไอ |
|--|---|
| <ol style="list-style-type: none"> 1. ต้องเขียนด้วยภาษาจาวา 2. โปรแกรมแบบเซิร์ฟเล็ตต้องทำงานเป็น เซรด หนึ่งของเว็บเซิร์ฟเวอร์ 3. มีระบบรักษาความปลอดภัยสูง โดยมีการซีเคียวริตี้แมนเนเจอร์คอยตรวจสอบการใช้ทรัพยากร หากพบการเข้าถึงทรัพยากรที่มีได้ รับอนุญาตระบบจะหยุดทำงานเซิร์ฟเล็ตนั้น ๆ ได้ทันทีอย่างอัตโนมัติ | <ol style="list-style-type: none"> 1. สามารถเขียนด้วยภาษาใดก็ได้ 2. โปรแกรม ซีจีไอ ถูกเรียกใช้ เป็นอีกโปรแกรมหนึ่ง แยกออกจากตัวเว็บเซิร์ฟเวอร์ 3. ต้องระมัดระวังในการเขียนโปรแกรมที่ใช้ ทรัพยากรของระบบ เพราะ โปรแกรมสามารถ เข้าถึงทรัพยากรของระบบได้โดย (ค่อนข้าง) อิสระ ความผิดพลาดของโปรแกรมสามารถ ทำให้ระบบโดยรวมทำงานผิดพลาดได้ |

ตาราง 2.1 สรุปความแตกต่างระหว่างเซิร์ฟเล็ตและซีจีไอ

โปรแกรมเซิร์ฟเล็ตก็คล้ายกับ โปรแกรมซีจีไอที่สามารถเขียนและใช้งานได้ง่าย รวมทั้งไม่ยุ่งเกี่ยวกับบราวเซอร์แต่อย่างใด (เพราะทำงานอยู่ด้านเซิร์ฟเวอร์) เซิร์ฟเล็ตทำงานหลายอย่างที่แอปเพล็ตทำไม่สามารถทำได้ ไม่ว่าจะเป็นการอ่านเขียนไฟล์ หรือการติดต่อไปยังศูนย์เว็บอื่น และเนื่องจากเซิร์ฟเล็ตจะทำงานเป็นเซรดในเว็บเซิร์ฟเวอร์ เซิร์ฟเล็ตจึงทำงานได้เร็วกว่าซีจีไอมาก

2.2.2 การติดต่อกับฐานข้อมูลของเซิร์ฟเล็ตกับฐานข้อมูล

ในการติดต่อกับกับฐานข้อมูลนั้นเซิร์ฟเล็ตซึ่งเขียนด้วยภาษาจาวาจะทำงานร่วมกับ JDBC (Java DataBase Connectivity : ซึ่งเป็น API ของบริษัท Sun โดยที่ JDBC เป็นการสร้างระดับการเชื่อมต่อเพื่อสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับ ODBC (Open Database Connection ของบริษัท ไมโครซอฟท์) ซึ่งในปัจจุบันได้ถือว่าเป็นมาตรฐานการเชื่อมต่อกับฐานข้อมูลสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) และระบบเครือข่ายท้องถิ่น (LAN))

JDBC ช่วยให้โปรแกรมภาษาจาวาสามารถเชื่อมต่อกับดาต้าเบสเอนจิน (database engine) ได้ ปลอดภัยและเป็นไปในแนวเดียวกัน เนื่องจาก JDBC ทำงานร่วมกับโปรแกรมภาษาจาวาใด ๆ ดังนั้น แอปเพล็ต (applet) หรือแอปพลิเคชันฝั่งไคลเอ็นท์ จึงสามารถใช้ ไดรเวอร์ของ JDBC ในการอินเทอร์ แร็ค (interact) กับฐานข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การใช้งาน Java servlet API โดย web application

โดยส่วนใหญ่แล้วเซิร์ฟเว็ลต์เว็บแอปพลิเคชัน (servlet web application) จะทำการอิมพลิเมนต์ (implement) HttpServlet interface โดยที่ http requests ที่เข้ามาจะถูกเราท์ (routed) ไปยัง เมธอด doGet() และ doPost() ซึ่งจะทำการจัดการให้ข้อมูล (data) ที่เข้ามาแสดงข้อมูลเอ็ชทีเอ็มแอล ออกทางหน้าจอ ทั้งสองเมธอดนี้จะรับเอาออปเจ็ค HttpServletRequest และ ออปเจ็ค HttpServletResponse เป็นพารามิเตอร์ โดยที่ ออปเจ็ค HttpServletRequest ประกอบด้วยหลายเมธอดด้วยกันเพื่อช่วยให้การเข้าถึงข้อมูล (information) ใด ๆ ซึ่งรับมาจาก http request ดังเช่น ฟอรั่มพารามิเตอร์ (form parameter) และ ตัวแปร ซีจีไอ (cgi variable) ได้ง่าย ส่วน HttpServletResponse ก็มีหลายเมธอดเช่นกันเพื่อให้จัดการกับฟังก์ชันของเอาท์พุทได้สะดวกยิ่งขึ้น เช่นการจัดการกับคอนเท็กซ์ (context) ให้แสดงออกมาในรูปแบบเอ็ชทีเอ็มแอล

เซิร์ฟเว็ลต์สามารถกำหนดให้เมธอด init() และ เมธอด destroy() ซึ่งจะรันเมื่อเว็บเซิร์ฟเวอร์มีการโหลด (load) หรือ อันโหลด (unload) เซิร์ฟเว็ลต์ และเนื่องจากมีการโหลดเซิร์ฟเว็ลต์เพียงครั้งเดียวเท่านั้นและรีเควสแต่ละรีเควสจะถูกทำเป็นเรซิดในอินสแตนซ์ (instant) เดียวกัน เมธอด init() จะทำการกำหนดค่าตัวแปรเริ่มต้น ในขณะที่เมธอด destroy() เป็นการใช้เพื่อสิ้นสุดการทำงานของยูสเซอร์รีเควส (user request) ที่ยังอยู่เมื่อมีการอันโหลดเซิร์ฟเว็ลต์ ยิ่งไปกว่านั้นคือ เราสามารถใช้ประโยชน์จากคลาส HttpSession ในการยอมให้เว็บแอปพลิเคชันเก็บข้อมูลเกี่ยวกับผู้ใช้ที่เจาะจงได้

2.2.4 การเขียนเซิร์ฟเว็ลต์

ก่อนเขียนเซิร์ฟเว็ลต์ ต้องทราบถึงวิธีการใช้ฟอรั่ม เอ็ชทีเอ็มแอล เพื่อให้ผู้ใช้ปีก่อน เราสามารถกำหนด URL ในแอ็กชันเพื่อใช้ส่งข้อมูลในฟอรั่มนั้นออกไปเหมือนกับการส่งข้อมูลใน CGI จากตัวอย่างโค้ด เห็นได้ว่าเรากำหนดให้

- action = "/servlet/RegServlet" " ซึ่ง RegServlet คือโปรแกรมเซิร์ฟเว็ลต์ที่เราต้องการให้ส่งข้อมูลในฟอรั่มไปให้
- method = " post " เป็นการกำหนดวิธีการรับส่งข้อมูลจากฟอรั่มไปยังเซิร์ฟเวอร์ นิยมใช้กับสองวิธีคือ

1 การส่งข้อมูลจากฟอรั่มไปยังเซิร์ฟเวอร์ โดยการนำข้อมูลต่อท้ายไปกับ URL และเรียกใช้โปรแกรมเซิร์ฟเว็ลต์ที่เซิร์ฟเวอร์ด้วยการเรียกผ่านคำสั่ง doGet() ของเซิร์ฟเว็ลต์ที่กำหนดในแอ็กชัน เช่น สมมติว่ามีการส่งข้อมูลในฟอรั่มไปยัง /servlet/RegServlet ด้วยการ GET เว็บเบราว์เซอร์จะส่งข้อมูลไปยังเซิร์ฟเวอร์ผ่านทาง URL โดยที่

- ข้อมูลจากฟอรั่มจะส่งไปยังเซิร์ฟเวอร์ผ่านทาง URL โดยการเพิ่มข้อมูลจากฟอรั่ม HTML หลังเครื่องหมาย ?

- ข้อมูลแต่ละฟิลด์ถูกแยกออกด้วยเครื่องหมาย &

ในกรณีที่มีช่องว่างจะถูกแปลงเป็นเครื่องหมาย +
ไม่ว่ากรณีใด ๆ ในกรณีที่มีตัวอักษรที่เป็นไบนารีจะถูกแปลงเป็น %xx เช่น %3D คือเครื่องหมาย = เป็นต้น

นอกจากนี้เว็บเซิร์ฟเวอร์ยังสามารถแคช URL ที่ใช้คำสั่ง GET ได้ทำให้โปรแกรมซีจีไอ หรือ เซิร์ฟเล็ตไม่จำเป็นต้องทำงานทุกครั้งที่มีการเรียก URL

2 POST ส่งข้อมูลจากฟอร์มผ่านให้เซิร์ฟเวอร์หลังคำสั่ง POST ในโพรโตคอล HTTP ซึ่งไม่ได้แสดงในรูปของ URL เหมือนกับคำสั่ง GET ถึงแม้ว่าการใช้คำสั่ง POST ไม่ได้สะดวกและรวดเร็วเหมือนกับคำสั่ง GET แต่ข้อดีของ POST คือความยืดหยุ่นในการใช้งาน เช่น ข้อมูลที่ส่งมีขนาดได้ไม่จำกัดเพราะไม่ได้อยู่ในรูปของ URL และทุกครั้งที่มีการ POST โปรแกรมที่เซิร์ฟเวอร์ต้องทำงานเสมอเพราะเว็บเซิร์ฟเวอร์ไม่สามารถแคช URL ได้

ในแพ็คเกจของจาวาเว็บเซิร์ฟเวอร์จะมีคลาสให้นำมาเป็นต้นแบบในการสร้างเซิร์ฟเล็ตได้ คือ คลาส `javax.servlet.http.HttpServlet` โดยปกติ `HttpServlet` อนุญาตให้มีเธรดหลาย ๆ ตัวทำงานพร้อมกันได้ หากเราต้องการให้เซิร์ฟเล็ตของเราทำงานแบบซิงเกิลเธรดหลาย ๆ ตัวทำงานพร้อมกันได้ หากเราต้องการให้เซิร์ฟเล็ตของเราทำงานแบบซิงเกิลเธรด เราเพียงเพิ่มเติมให้เซิร์ฟเล็ตของเรา implements `javax.servlet.SingleThreadModel` เท่านั้น โดยเขียนโค้ดเพิ่มเติมอีกดังตัวอย่าง

```
public class SurveyServlet extends HttpServlet implements SingleThreadModel
{
    /* typicall servlet code , with no threading concerns
    * in the service method. No extra code for the
    * Single ThreadModel interface
    */
}
```

```
1 <html>
2 <head>
3 <title> Java Web Server Registration
Form </title>
4 </head>
5 <body>
6 <center><h2> Java Web Server Registration
Form </h2></center>
7<form name = "testform"
action="/servlet/RegServlet"
method="post">
8 <table>
9<tr>
<td><b>UserID</b></td><td><input type ="text"
name = "userID" size = 8 maxlength = 8></td>
<td><b>First Name</b></td><td><input type="text" name="firstName" size=40
maxlength40></td>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ 10 ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ที่ส่งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

11<tr><td><b>Last Name</b></td><td>
<input type="text" name="lastName" size=40
maxlength=40></td>
12<tr><td><b>E-mail Address</b></td><td>
<input type="text" name="email" size=40
maxlength=40></td>
13 </table>
14 <br>
15 <center>
16 <input type = "submit" name = "submitClick"
value = "Register">
17 <input type = "reset" name = "resetClick"
value = "Reset">
18 </center>
19 </form>
20 </body>
21 <</html>

```

รูปที่ 2.1 แสดงตัวอย่างฟอร์ม เอชทีเอ็มแอล ที่ให้ผู้ใช้ป้อนข้อมูลเพื่อลงทะเบียนมายังเซิร์ฟเว็ต

2.3 ส่วนจัดเก็บและค้นหาข้อมูล

การทำงานของเซิร์ฟเอนจินจำเป็นจะต้องมีการสร้างส่วนที่ใช้จัดเก็บข้อมูลที่ได้จากโรบอท โดยข้อมูลที่จะนำมาเก็บนั้นจะต้องเป็นข้อมูลที่ได้รับบริการวิเคราะห์แล้วและมีความสำคัญมากพอที่จะนำมาเก็บไว้

การจัดเก็บข้อมูลสามารถทำได้หลายวิธี ไม่ว่าจะเป็นการจัดสร้างโครงสร้างข้อมูลเพื่อใช้ในการจัดเก็บขึ้นมาเอง หรือ จะใช้งานระบบฐานข้อมูลอยู่แล้วก็ได้ ทั้งนี้ขึ้นกับความเหมาะสมของการใช้งาน สำหรับโครงการนี้เลือกการจัดเก็บข้อมูลบนระบบฐานข้อมูล ดังจะอธิบายถึงทฤษฎี หลักการ ต่อไป

2.3.1 ระบบฐานข้อมูล (Database System)

ความหมายและที่มา

ฐานข้อมูล คือแหล่งเก็บข้อมูลและความสัมพันธ์ระหว่างฐานข้อมูลเหล่านี้

ในอดีตนั้นการจัดเก็บและค้นหาข้อมูลยังเก็บในรูปแบบของไฟล์ ซึ่งก็ได้งานในระบบงานขนาดเล็ก มีผู้ใช้งานอยู่เพียงหนึ่งคนหรือไม่ก็คน แต่ในระบบขนาดใหญ่ขึ้นมีผู้ใ้ใช้มากขึ้น การค้นหาและจัดเก็บข้อมูลจึงต้องการประสิทธิภาพที่สูงขึ้น สาเหตุที่ทำให้มีการนำระบบฐานข้อมูลมาใช้ระบบไฟล์แบบเก่านี้ไม่ได้อีกต่อไป ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ลดความซ้ำซ้อนของข้อมูล
การที่แผนกต่าง ๆ ต้องการข้อมูลชนิดเดียวกัน แต่ต่างก็แยกกันเก็บทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูลของหน่วยความจำ
- หลีกเลี่ยงการขัดแย้งกันเองของข้อมูล
เมื่อมีการเปลี่ยนแปลงข้อมูลที่จัดเก็บในหลายๆ ที่หลายๆ ไฟล์จะต้องทำการเปลี่ยนแปลงทุกไฟล์ที่จัดเก็บถ้าทำ ไม่ครบทุกไฟล์ก็จะทำให้ข้อมูลมีความผิดพลาดไปจากความจริงและขัดแย้งกัน
- แก้ปัญหาการรักษาความปลอดภัยของข้อมูล
เนื่องจากการเก็บข้อมูลกระจัดกระจาย ทำให้ยากต่อการรักษาความปลอดภัยต่อการรั่วไหลและคัดแปลงข้อมูล
- ทำให้ข้อมูลมีคุณภาพมากที่สุด หรือ การควบคุมความถูกต้องของข้อมูล
- ทำให้สามารถควบคุมการใช้งาน โดยส่วนกลางได้
- สามารถใช้ข้อมูลร่วมกันได้
- มีความคล่องตัวและความยืดหยุ่นในการใช้งาน
- มีการควบคุมมาตรฐานร่วมกัน

2.3.2 ระบบจัดการฐานข้อมูล (Database Management System)

ระบบจัดการฐานข้อมูล คือ ซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และ โปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล หรือ

ระบบจัดการฐานข้อมูล คือ ซอฟต์แวร์ที่ทำหน้าที่จัดการการเข้าถึงข้อมูลอย่างดูฐานข้อมูล

เนื่องจากการใช้และควบคุมดูแลฐานข้อมูลเป็นเรื่องซับซ้อนยุ่งยาก ระบบจัดการฐานข้อมูลจึงเป็นเครื่องมือสำคัญในการที่จะลดภาระของผู้ใช้ไปอย่างมาก ทำให้การใช้งานระบบฐานข้อมูลมีประสิทธิภาพสูงขึ้น

2.3.3 โมเดลข้อมูล

โมเดลของข้อมูล ได้แก่ประเภทของระบบการจัดการฐานข้อมูล แบ่งออกโดยจำแนกตามการแสดงความสัมพันธ์ระหว่างข้อมูลได้ 3 โมเดลดังนี้

1 แบบโครงข่าย (Network Model)

เป็นการแสดงความสัมพันธ์ด้วยลิงค์ลิสต์เป็นโครงข่ายโดยมีตัวชี้ระหว่างข้อมูลเชื่อมกันเป็นชุด มีความสัมพันธ์แบบหนึ่งต่อหลาย (1 to many)

2 แบบแผนภูมิต้นไม้ (Hierarchical Model)

ทำการแทนความสัมพันธ์ระหว่างข้อมูลในลักษณะของแผนภูมิต้นไม้โดยมีความสัมพันธ์ในลักษณะหลายต่อหลาย (many to many) ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 แบบสัมพันธ์ (Relational Model)

เป็นการเก็บข้อมูลที่มีลักษณะการเก็บในรูปตาราง 2 มิติธรรมดา คือ มีแถวและคอลัมน์

2.3.4 ข้อดีของการจัดเก็บข้อมูลแบบฐานข้อมูล

การจัดเก็บข้อมูลโดยใช้ระบบฐานข้อมูลเข้ามาช่วยนั้นเป็นวิธีการที่ได้รับความนิยมอย่างแพร่หลาย เนื่องจากมีประสิทธิภาพและการใช้งานที่เหมาะสมดังมีข้อดีต่อไปนี้

1 สามารถลดความซ้ำซ้อนของข้อมูล

การจัดเก็บข้อมูลโดยวิธีอื่น เช่น การจัดเก็บข้อมูลเป็นไฟล์ อาจทำให้เกิดปัญหาการจัดเก็บข้อมูลซ้ำซ้อนกัน เช่น การที่ผู้ใช้งานทุกคนต่างก็ทำการคัดลอกไฟล์ดังกล่าวเก็บไว้เพื่อใช้งานซึ่งการซ้ำซ้อนในการเก็บข้อมูลนี้ จะทำให้เกิดการเสียเนื้อที่ในการจัดเก็บข้อมูล โดยไม่จำเป็นซึ่งหากใช้การเก็บข้อมูลแบบฐานข้อมูล สามารถที่จะช่วยลดปัญหาดังกล่าวได้ เนื่องจากการจัดเก็บข้อมูลโดยใช้ฐานข้อมูลจะมีการออกแบบและควบคุมข้อมูลที่จัดเก็บเพื่อลดความซ้ำซ้อนและนอกจากนี้จุดประสงค์หลักของการใช้งานข้อมูลก็คือ การลดความซ้ำซ้อนของข้อมูลให้มากที่สุด

2 สามารถควบคุมความถูกต้องของข้อมูลได้

การจัดเก็บข้อมูลโดยวิธีทั่วไป อาจเกิดปัญหาในการทำการควบคุมความถูกต้องของข้อมูลอันอาจจะมีผลเนื่องจากการเก็บข้อมูลซ้ำซ้อนของผู้ใช้งาน แต่ละคนสามารถทำการเปลี่ยนแปลงข้อมูลได้ ซึ่งทำให้ข้อมูลแต่ละชุดที่คัดลอกไปอาจจะไม่เหมือนกัน จนไม่อาจแยกแยะได้ว่าข้อมูลใดเป็นข้อมูลที่ถูกต้อง แต่หากมีการใช้การเก็บข้อมูลโดยใช้ฐานข้อมูลจะมีการควบคุมการเปลี่ยนแปลงข้อมูล โดยจะอนุญาตให้มีเพียงผู้ใช้งานเพียงคนเดียวในการทำการเปลี่ยนแปลงแต่ละครั้ง ทำให้สามารถควบคุมความถูกต้องของข้อมูลได้

3 สามารถใช้ข้อมูลร่วมกันได้

การที่ทำการจัดเก็บข้อมูลโดยใช้ฐานข้อมูลทำให้ข้อมูลที่จัดเก็บสามารถที่จะใช้ร่วมกันได้ระหว่างผู้ใช้หลายคน โดยที่สามารควบคุมความถูกต้องของข้อมูลได้ ในขณะที่หากเราใช้การเก็บข้อมูลแบบอื่น อาจมีปัญหาในการใช้ข้อมูลร่วมกันในเรื่องความถูกต้องของข้อมูล

4 สามารถตรวจสอบความขัดแย้งกันของข้อมูลที่จัดเก็บได้

5 การรักษาความปลอดภัยของข้อมูลสามารถทำได้อย่างมีประสิทธิภาพ

ในการที่จะสร้างฐานข้อมูลขึ้นมาจะต้องทำการรวบรวมความต้องการของผู้ใช้งานแล้วจึงทำการออกแบบฐานข้อมูล แต่การออกแบบจำเป็นต้องมีหลักในการออกแบบซึ่งหลักในการออกแบบนี้มีอยู่วิธีไม่ว่าจะเป็น ไนแอม (Niam) , นอร์มอลไลซ์ (Normalize) , อีอาร์โมเดล (ER - model) ซึ่งทางผู้วิจัยใช้วิธีอีอาร์โมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

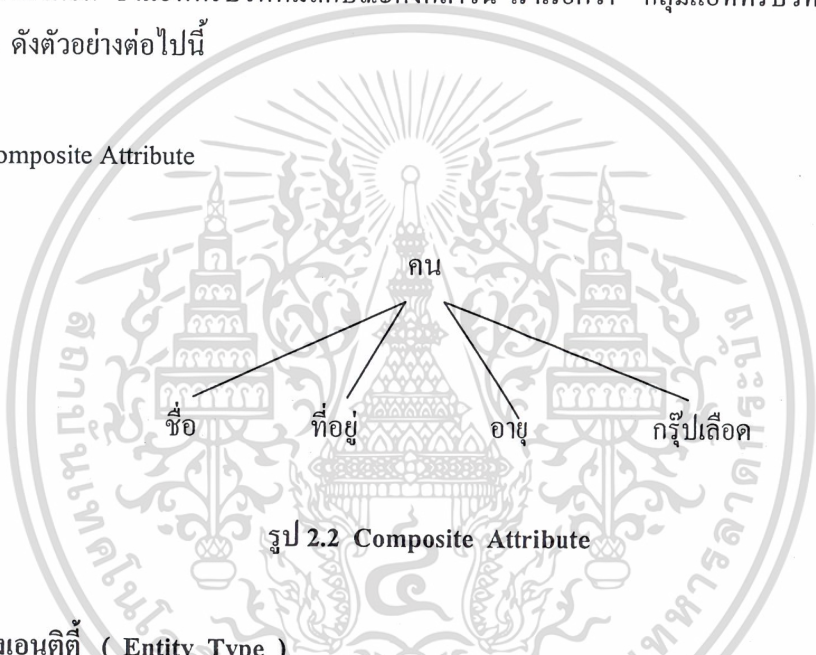
2.3.5 หลักการของอีอาร์โมเดล

ในการออกแบบตามหลักของอีอาร์โมเดล จะต้องเข้าใจส่วนประกอบต่าง ๆ ของอีอาร์โมเดล ไดอะแกรมว่ามีความหมายว่าอย่างไร

เอนติตี้ (Entity) และ แอททริบิวท์ (Attribute)

ภายในการออกแบบโดยอาศัยอีอาร์โมเดล จะมีการแสดงสิ่งต่าง ๆ ที่มีอยู่บนโลก บนไดอะแกรม โดยเรียกว่าเอนติตี้ เช่น คน บ้าน รถ หรืออาจจะเป็นสิ่งที่ถูกสร้างขึ้น เช่น บริษัท มหาวิทยาลัย งาน แต่ละเอนติตี้จะมีคุณลักษณะของตัวเอง โดยสร้างสิ่งที่ใช้แสดงคุณสมบัติดังกล่าวนี้ เรียกว่าแอททริบิวท์ เช่นเอนติตี้ คน อาจจะมีแอททริบิวท์ของ ชื่อ , ที่อยู่ , อายุ เป็นต้น บางแอททริบิวท์ยังสามารถแบ่งต่อออกไปได้อีก ซึ่งแอททริบิวท์ที่มีลักษณะดังกล่าวนี้ เราเรียกว่า กลุ่มแอททริบิวท์ (Composite Attribute) ดังตัวอย่างต่อไปนี้

ตัวอย่าง Composite Attribute



รูป 2.2 Composite Attribute

ประเภทของเอนติตี้ (Entity Type)

ในระบบฐานข้อมูลมักประกอบด้วยกลุ่มของแอททริบิวท์ ที่คล้าย ๆ กัน เช่นภายในบริษัทอาจจะมีพนักงานจำนวนมาก โดยแต่ละคนจะมีข้อมูลคล้าย ๆ กัน นั่นคือพนักงานเหล่านี้เป็นเอนติตี้ประเภทเดียวกัน มีแอททริบิวท์เหมือนกันจะต่างก็เพียงค่าของแอททริบิวท์ของพนักงานแต่ละคน ดังนั้นเราจึงมีการกำหนดให้กลุ่มของแอททริบิวท์ที่คล้ายกันรวมกันอยู่ในเอนติตี้ไทป์เดียวกัน เช่น

| | |
|-------------------------|-------------------------------------|
| พนักงาน | บริษัท |
| ชื่อ , อายุ , เงินเดือน | ชื่อ , ที่อยู่ , ประธาน |
| e1 | e1 |
| (นาย ก , 21 , 3000) | (ปูนไทย , บางพลี , นายสมชาย) |
| (นาย ข , 31 , 56000) | (กระเบื้องไทย , กรุงเทพ , นายปาน) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

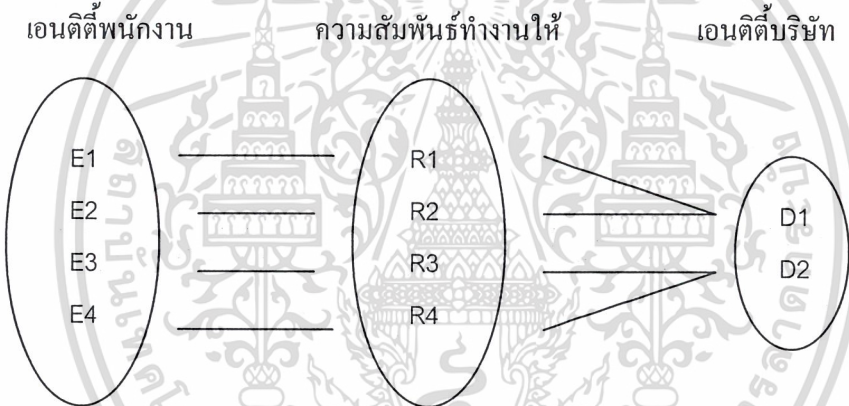
คีย์แอททริบิวท์ (Key Attribute)

คีย์แอททริบิวท์เป็นส่วนที่สำคัญมากสำหรับเอนติตี้เพราะ คีย์แอททริบิวท์เป็นตัวที่สามารถแยกแยะเอนติตี้ แต่ละตัวภายในเอนติตี้ไทป์เดียวกันได้ เช่น เอนติตี้ไทป์ บริษัทอาจจะมีชื่อเป็นคีย์แอททริบิวท์

ความสัมพันธ์ (Relationship)

ในระบบฐานข้อมูลจะประกอบด้วยหลายเอนติตี้ไทป์ ซึ่งแต่ละเอนติตี้ในระบบฐานข้อมูลก็จะมี ความสัมพันธ์กับเอนติตี้ตัวอื่น เช่น เอนติตี้พนักงาน มีความสัมพันธ์กับเอนติตี้บริษัทโดยจะต้องมีการ กำหนดความสัมพันธ์ระหว่างเอนติตี้พนักงานและเอนติตี้บริษัทจะอยู่ในรูปของความสัมพันธ์ที่เรียกว่า ทำงานให้ (work_for) ซึ่งความสัมพันธ์นี้อาจจะเป็นแบบหนึ่งต่อหนึ่ง , หนึ่งต่อหลาย , หรือ หลายต่อหลาย ก็ได้แล้วแต่ความสัมพันธ์

ตัวอย่าง ความสัมพันธ์



รูป 2.3 ความสัมพันธ์

นอกจากนี้ยังมีการกำหนด ลักษณะพิเศษต่าง ๆ เพิ่มอีก เช่น จากตัวอย่าง ถ้าบริษัทมีการกำหนดว่า พนักงานทุกคนต้องทำงานให้แก่แผนก ดังนั้นเอนติตี้พนักงานจะอยู่ได้ก็ต่อเมื่อมีความสัมพันธ์ทำงานให้เกิดขึ้น เราเรียกความสัมพันธ์ของเอนติตี้พนักงานกับความสัมพันธ์ทำงานให้ว่าเป็นแบบโททอล นั่นหมายความว่าทุกเอนติตี้พนักงานจะต้องมีความสัมพันธ์กับเอนติตี้แผนก ผ่านทางความสัมพันธ์ทำงานให้

บางเอนติตี้ก็ไม่มีคีย์แอททริบิวท์ของตนเอง นั่นแสดงว่า เราไม่สามารถที่จะทำการแยกความแตกต่างระหว่างเอนติตี้ได้เพราะว่าแม้ว่าเราจะใช้เป็นการคอมบิเนชัน (Combination key) ก็ไม่สามารถแยกแยะได้เพราะต่างก็มีคีย์เหมือนกันเราเรียกกรณีนี้ว่า วิกเอนติตี้ (Weak Entities)

เมื่อทำการออกแบบฐานข้อมูลเรียบร้อยแล้ว ก็ทำการเลือกระบบฐานข้อมูลที่จะนำมาใช้ในการเก็บข้อมูลโดยในโครงการนี้ได้เลือกที่จะใช้ ออราเคิล (Oracle) เป็นฐานข้อมูลด้วยเหตุผลต่อไปนี้

- 1 มีประสิทธิภาพในการจัดการฐานข้อมูลสูง
 - 2 การติดต่อระหว่างโปรแกรมต่าง ๆ ภายในโครงการทำได้ง่าย
- เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเขียนเพื่อใช้ในการศึกษา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเนื้อหาที่ละเอียดและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 ออราเคิล (Oracle)

ออราเคิลคือ Relational Database Management System (RDBMS) ที่สามารถทำงานได้หลายลักษณะไม่ว่าจะเป็นการทำงานในลักษณะที่มีผู้ใช้งานเพียงคนเดียวหรือผู้ใช้งานหลายคน ออราเคิลสามารถรันได้บนแพลตฟอร์มดังต่อไปนี้ วินโดวส์ เอ็นที , โนเวล เน็ตแวร์ , ยูนิกซ์ , ลินุกซ์

2.3.7 ภาษาเอสคิวแอล

Structured Query Language หรือที่เรียกกันว่าภาษาเอสคิวแอล เป็น Standard Relational Language ที่ใช้ในการจัดการข้อมูลที่เกี่ยวข้องอยู่บนฐานข้อมูลแบบรีเลชันนอล คาด้าเบส ภาษาเอสคิวแอลเป็นภาษาที่ไม่ case sensitive คือไม่ว่าจะพิมพ์เป็นตัวเล็กหรือตัวใหญ่ ก็จะได้ตัวใหญ่ออกมาเสมอ เราสามารถแบ่งประเภทการทำงานของภาษาเอสคิวแอลได้ 2 โหมดการทำงานคือ

1 Interactive Mode คือ โหมดที่ให้ผู้ใช้งานใส่คำสั่ง SQL ทางคีย์บอร์ด เมื่อรับคำสั่งแล้วก็จะมีการแสดงผลพร้อมออกมาตรง ๆ ซึ่งการทำงานในโหมดนี้จะมีประโยชน์มาก ในกรณีที่เราจะใช้ในการทดสอบคำสั่ง การตอบคำถามเฉพาะกิจในงานลักษณะของ MIS

2 Embedded Mode คือ การใช้คำสั่ง SQL ร่วมกับภาษา host เช่น หากอยากได้น้ำจอสวยก็ต้องใช้ภาษาอื่นเข้าช่วย

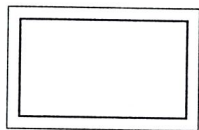
รูปแบบของภาษาเอสคิวแอล

คำสั่งของภาษาเอสคิวแอลจะเป็นรูปแบบที่แน่นอน ดังจะมีรูปแบบโดยรวมดังนี้

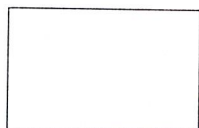
```
SELECT < colname > | * |
FROM < table name >
WHERE < row condition >
GROUP BY < colname >
HAVING < group condition >
ORDER BY < column list >
```

ต่อไปก็จะทำการอธิบายหลักการโดยรวมของภาษาเอสคิวแอลโดยสองคำสั่งแรกคือ คำสั่ง SELECT, FROM จะต้องมีเสมอทุกครั้งที่มีการใช้คำสั่งเอสคิวแอล ส่วนคำสั่ง WHERE เป็นคำสั่งที่ช่วยให้เราสามารถระบุได้ว่าแถวใดบ้างที่เราต้องการเป็นคำตอบและแถวประเภทใดที่ต้องแยกออกส่วนคำสั่ง GROUP BY, HAVING จะเป็นคำสั่งที่ใช้ในการจัดกลุ่ม แต่ต่างกันที่การใช้งานและเงื่อนไขที่ใช้ สำหรับกรณีที่เราต้องการที่จะจัดลำดับการทำงานของผลลัพธ์ เราสามารถทำได้โดยการใช้คำสั่ง ORDER BY

อีอาร์โมเดลไดอะแกรม



ENTITY TYPE



WEAK ENTITY TYPE



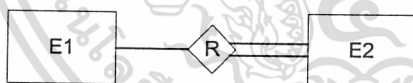
ATTRIBUTE



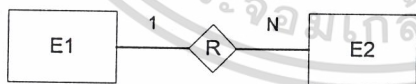
KEY ATTRIBUTE



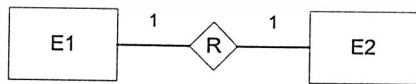
COMPOSITE ATTRIBUTE



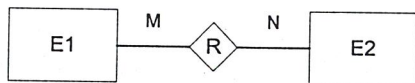
TOTAL PARTICIPAION



CADINARY RATIO 1 : N

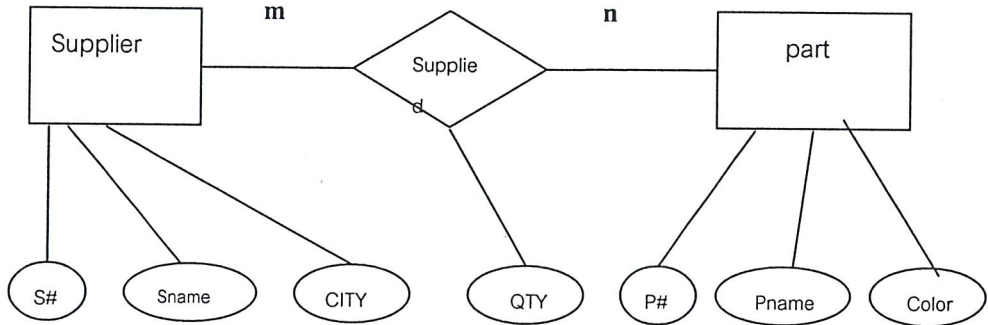


CADINARY RATIO 1 : 1



CADINARY RATIO M : N

เอกสารนี้เป็นเอกสารที่สงวนรูปที่ 2.4 สัญลักษณ์ที่ใช้ในการสร้างอีอาร์โมเดลไดอะแกรมไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.5 ตัวอย่างอีอาร์โมเดลไดอะแกรม

2.4 ส่วนหน้าจอร์ับข้อมูลและแสดงผล

ส่วนหน้าจอร์ับข้อมูลและแสดงผลนั้นบนอินเทอร์เน็ตนี้หมายถึง โฮมเพจ ซึ่งเป็นเอกสารบนเว็บ เราจะต้องทำการจัดสร้างโฮมเพจของเซิร์ฟเวอร์ขึ้นมาเพื่อที่จะใช้ในการติดต่อรับข้อมูลจากผู้ใช้บริการ โดยข้อมูลที่รับก็คือ คำที่ต้องการค้นหา (คีย์เวิร์ด) นอกจากนี้ส่วนที่ใช้ในการแสดงผลัพท์ที่ได้จากการค้นหาข้อมูลในระบบฐานข้อมูลโดยจะทำการรับส่งค่าต่าง ๆ ผ่านโปรแกรมเซิร์ฟเล็ต

การทำการสร้างโฮมเพจนั้นเราจะใช้ภาษาที่เรียกว่า ภาษา เอ็ชทีเอ็มแอล ดังนั้นผู้จัดทำโครงการจึงศึกษาวิธีการเขียนโฮมเพจด้วยภาษา เอ็ชทีเอ็มแอล ดังนี้

ภาษา เอ็ชทีเอ็มแอล

HTML (HyperText Markup Language) ถูกออกแบบมาเพื่อใช้กำหนดรูปแบบของเอกสารในระดับโลจิคอล ซึ่งแตกต่างจากเวิร์ดโปรเซสเซอร์โดยทั่วไป เนื่องจากเอกสารเดียวกันนี้สามารถดูได้จากบราวเซอร์ต่างชนิดกัน

การจัดโครงสร้างแฟ้มเอกสาร

ในความง่ายของภาษาเอ็ชทีเอ็มแอล นั้นเป็นเพราะภาษานี้ไม่มีโครงสร้างใด ๆ มากำหนดนอกจากโครงสร้างพื้นฐานเท่านั้น หรือแม้แต่จะไม่มีโครงสร้างพื้นฐานอยู่ โปรแกรมที่เขียนขึ้นมาก็สามารถทำงานได้เสมือนมีโครงสร้าง ทั้งนี้เป็นเพราะว่าตัวโปรแกรมเว็บบราวเซอร์จะมองเห็นทุกสิ่งทุกอย่างในโปรแกรมเอ็ชทีเอ็มแอลเป็นส่วนเนื้อหาทั้งสิ้น ยกเว้นในส่วนหัวที่ต้องมีการกำหนดแยกออกไปให้เห็นชัดเท่านั้น จะเขียนคำสั่งหรือข้อความที่ต้องการให้แสดงอย่างไรก็ได้ เป็นเสมือนกับการพิมพ์งานเอกสารธรรมดาทั่วไป เพียงแต่ที่ตำแหน่งใดมีการกำหนดรหัสพิเศษขึ้นมา เว็บบราวเซอร์ก็จะแสดงผลออกมาตามที่ถูกกำหนด โดยใช้คำสั่งให้ตรงกับรหัสที่กำหนดเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลที่เว็บเบราว์เซอร์

หลังจากพิมพ์โปรแกรมเสร็จเรียบร้อยแล้วให้บันทึกไฟล์ที่มีนามสกุลเป็น .html จากนั้นเรียกโปรแกรมเว็บเบราว์เซอร์ที่มีอยู่ขึ้นมาทำการทดสอบโปรแกรม เพียงแต่ผู้ใช้โหลดโปรแกรมที่เขียนขึ้นมาจากโปรแกรมเว็บเบราว์เซอร์เท่านั้น ข้อมูลที่เราสร้างก็จะถูกนำแสดงออกมาที่จอภาพ

อีลีเมน (element) ในเอกสาร เอ็มทีเอ็มแอล

คำสั่งที่สามารถทำงานได้โดยเอ็ชทีเอ็มแอล ถูกเรียกว่า HTML element หรือเรียกย่อ ๆ ว่าแท็ก (Tag) ใช้ในการกำหนดรูปแบบของข้อมูล ซึ่งมีรูปแบบพื้นฐาน ดังนี้

`<element_name>`

เป็นTagที่อยู่ในตำแหน่งเริ่มต้นของบล็อกของข้อมูลที่ต้องการกำหนดรูปแบบ โดยมีชื่อของอีลีเมนที่อยู่ระหว่าง Angle brackets

`</element_name>`

เป็นTagที่อยู่ในตำแหน่งสิ้นสุดของบล็อกของข้อมูล

เอ็มทีอีลีเมนท์ (Empty Elements)

บางอีลีเมนท์สามารถเป็นอีลีเมนท์ว่างได้ นั่นคือไม่ส่งผลกระทบต่อบล็อกของข้อมูล อีลีเมนท์ชนิดนี้ไม่จำเป็นต้องใช้แท็กปิดท้ายบล็อกของข้อมูล

อัปเปอร์และโลเวอร์เคสอีลีเมนท์ (UppercaseLowerCase Element)

การเขียนอีลีเมนท์เนม สามารถใช้ได้ทั้งตัวใหญ่และตัวเล็กซึ่งให้ผลเหมือนกัน เช่นการขีดเส้นในแนวนอนสามารถเขียนได้ดังนี้

`` , `` หรือ ``

อีลีเมนท์หลายชนิดสามารถมีอาร์กิวเมนต์เพื่อใช้ในการผ่านค่าพารามิเตอร์ต่าง ๆ ไปยังอีลีเมนท์ได้ อาร์กิวเมนต์เหล่านี้ถูกเรียกว่า แอททริบิวท์ของอีลีเมนท์ (Attributes of the element)

ตัวอย่าง

` marked text `

จากตัวอย่าง a เป็น element name ซึ่งปิดท้ายด้วย Tag

HREF เป็นแอททริบิวท์ซึ่งใช้กำหนดค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 โครงสร้างของเอกสารแบบเอชทีเอ็มแอล

เอกสารแบบเอชทีเอ็มแอล ประกอบด้วย 2 ส่วน คือเฮดและบอดี

- เฮด ประกอบด้วยข้อมูลเกี่ยวกับเอกสาร
- บอดี ประกอบด้วยส่วนเนื้อข้อมูลที่แสดง

โครงสร้างของเอกสาร

```

<HTML>
<HEAD>
.
<element_name>.....</element_name>
.
</HEAD>
<BODY>
</BODY>
.
<element_name>.....</element_name>
</BODY>
</HTML>

```

2.4.2 เฮด (HEAD)

ในส่วนเฮดนี้จะประกอบไปด้วยข้อมูลเกี่ยวกับเอกสารแต่ไม่มีการแสดงผลเหมือนกับในส่วนบอดี Mark-up อีลีเมนต์ที่สามารถใช้ในส่วนเฮดนี้ได้แก่

- TITLE
- ISINDEX
- NEXTID
- LINK
- BASE

2.4.2.1 TITLE

ส่วน Title ของเอกสารถูกกำหนดโดย TITLE อีลีเมนต์ในส่วนเฮดซึ่งมีได้เพียง Title เดียวในแต่ละเอกสาร มักใช้เป็น Label ของวินโดว์ในการแสดงผล การกำหนด Title ของเอกสารควรจะมีค่านัยแต่ไม่ยาวมากนัก (ควรจะน้อยกว่า 64 ตัวอักษร)
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.2 ISINDEX

อีลีเมนต์นี้จะบอกตัวแสดงผลว่าเอกสารนี้เป็นเอกสารดัชนี (index document) นั่นคือ ในการแสดงหรืออ่านเอกสารสามารถทำการค้นหาข้อมูลตามคีย์เวิร์ดที่กำหนดให้ โดยการเติมเครื่องหมาย “?” ตามหลังยูอาร์แอลนั้นและถ้าหากคีย์เวิร์ดมีหลายคำ ให้ใช้เครื่องหมาย “+” ในการแยกคำ เช่น ยูอาร์แอลที่สามารถค้นหาข้อมูลได้

<http://www.here.ca/cgi-bin/srch>

การค้นหาคำว่า ‘instructional’ ทำได้โดยเปลี่ยนแปลงยูอาร์แอลเป็น

<http://www.here.ca/cgi-bin/srch?instructional>

โดยปกติ ISINDEX อีลีเมนต์นี้ถูกสร้างโดย server-side script โดยอัตโนมัติ สำหรับเซิร์ฟเวอร์ที่สามารถทำการค้นหาได้เท่านั้น ดังนั้นจึงไม่ควรใส่อีลีเมนต์นี้เข้าไปเองในเอกสาร

2.4.2.3 NEXTID

แท็กนี้เป็นแอททริบิวต์เดี่ยวอยู่ในส่วนเศของเอกสารถูกออกแบบมาเพื่อใช้สำหรับ Automatic HyperText Editors

Old anchor ids จะไม่ถูกนำกลับมาใช้ใหม่เมื่อมีการเปลี่ยนแปลงเอกสารขณะที่เอกสารอื่นอ้างอิงถึงเอกสารเหล่านี้

2.4.2.4 LINK

เป็นอีลีเมนต์ใช้สำหรับบอกความสัมพันธ์ระหว่างเอกสารกับเอกสาร หรือ ออปเจกอื่น ในปัจจุบันไม่ค่อยใช้กันแล้ว

LINK อีลีเมนต์ที่เป็นอีลีเมนต์ว่างมีแอททริบิวต์ต่าง ๆ ดังนี้

- HREF กำหนดเอกสารหรือส่วนหนึ่งของเอกสารที่ต้องการติดต่อ
- NAME สำหรับตั้งชื่อ LINK เพื่อเป็นปลายทางสำหรับเอกสารอื่น
- REL อธิบายความสัมพันธ์ที่ถูกกำหนดโดยการเชื่อมต่อนี้
- REV คล้าย ๆ กับ REL แต่เป็นความสัมพันธ์ตรงข้ามกัน
- URN กำหนดชื่อของ Uniform Resourc Name
- TITLE ใช้เป็นไตเติลของ HREF แอททริบิวต์ของ LINK อีลีเมนต์
- METHOD อธิบาย HTTP เมธอดที่ออปเจกอ้างอิงใน HREF ของ LINK อีลีเมนต์

2.4.2.5 BASE

อีลีเมนต์นี้ใช้สำหรับเก็บออร์จินอลยูอาร์แอลของเอกสาร ซึ่งจะช่วยให้สามารถย้ายเอกสารไปยัง

ใดเรททอร์หรือไซท์อื่น มีแอททริบิวต์เดียว คือ HREF ใช้สำหรับเก็บ เบสยูอาร์แอลของเอกสาร

เอกสารนี้เป็นเอกสารที่วางไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรเอาไปใช้ประโยชน์ด้านการค้า เพื่อใช้กับ Partial ยูอาร์แอลภายในเอกสาร โดยใช้ Base ยูอาร์แอลเป็นจุดเริ่มต้น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.6 META

อ็ลลิเมนต์ที่ใช้ในการกำหนดเมตาดาต้าอินฟอร์เมชัน ภายในเอกสาร โดยมีแอททริบิวต์ต่าง ๆ ดังนี้

- HTTP – EQUIV แอททริบิวต์นี้ใช้ติดต่อรหว่าง META อ็ลลิเมนต์ เพื่อตอบสนองต่อโปรโตคอลเฉพาะ ซึ่งถูกสร้างโดย HTTP เซิร์ฟเวอร์ที่เก็บเอกสาร
- NAME กำหนดชื่อของอินฟอร์เมชันในเอกสารไม่เหมือนกับTITLEเนื่องจาก “ meta name ” เป็นการแบ่งประเภทของอินฟอร์เมชัน
- CONTENT “ meta name ” สำหรับ content ที่เกี่ยวข้องกัชื่อที่กำหนดโดย NAME แอททริบิวต์หรือการตอบสนองที่ถูกกำหนดโดย HTTP – EQUIV

2.4.3 บอดี (Body)

บอดีอ็ลลิเมนต์ประกอบด้วยอินฟอร์เมชันเกี่ยวกับเอกสารและ mark-up อ็ลลิเมนต์ต่าง ๆ ที่ใช้ในการกำหนดรูปแบบของตัวอักษร , รูปภาพ และอื่น ๆ

2.4.3.1 Section Headings (H1 , H2 ,...,H6)

HTML ขอมให้มีเฮดคิงได้ 6 ระดับ โดยใช้แท็ก H1, H2, ... , H6 ในการใช้ไม่จำเป็นต้องเรียงตามลำดับ แต่เพื่อความเข้าใจที่ตรงกันจึงควรที่จะใช้ <H1> เป็นเฮดคิงที่สำคัญที่สุด และใช้แท็กอื่นกับเฮดคิงที่สำคัญรองลงมา

2.4.3.2 Marking Paragraphs with HTML

อ็ลลิเมนต์ <P> ใช้สำหรับแบ่งพารากราฟและใช้เมื่อต้องการแบ่ง Text 2 Blocks ออกเป็น 2 พารากราฟ ซึ่งในการใช้ต้องทำเครื่องหมายทุกอ็ลลิเมนต์และข้อความทุกอย่างที่ต้องการให้อยู่ในพารากราฟเดียวกัน

2.4.3.3 Line Breaks

อ็ลลิเมนต์
 ใช้สำหรับกำหนดให้ข้อความที่ตามหลังแท็ก ขึ้นบรรทัดใหม่

2.4.3.4 IMG (In-Line Images) Element

อ็ลลิเมนต์นี้ใช้สำหรับให้ Graphical Browser แสดงรูปภาพในตำแหน่งที่แท็กปรากฏมีชนิดเป็นอ็ลลิเมนต์ว่างจึงไม่จำเป็นต้องปิดท้ายด้วย ประกอบด้วยแอททริบิวต์ต่าง ๆ ดังนี้

- SRC="image_url" ใช้กำหนดยูอาร์แอลของ image
- ALIGN=BOTTOM (MIDDLE or TOP) ใช้บอกตำแหน่งของรูปภาพที่จะแสดง
- ALT="atlternative text" ใช้กำหนดข้อความที่แสดงผลสำหรับบราวเซอร์ที่ไม่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ISMAP ใช้สำหรับกำหนดให้รูปภาพเป็น active image map นั่นคือสามารถให้ผู้ใช้คลิกเมาส์ไปในตำแหน่งที่แตกต่างกันสามารถให้ผลตอบสนองที่แตกต่างกัน

2.4.3.5 Hypertext Anchors

Anchor คือการกำหนดให้ข้อความหรือออปเจกต์ชนิดต่าง ๆ ที่อยู่ระหว่าง <A> และ ใช้สำหรับ Hypertext link

Anchor Attributes เหมือนกับ LINK Attributes

- HREF (link to object) *
- NAME (link from object) *
- REL (relationship between objects)
- REV (relationship between objects)
- URN (URN for the document)
- TITLE (TITLE of document)
- METHOD (how to link)

2.4.3.6 Lists

HTML สนับสนุนการสร้างลิสต์โดยมีอิลีเมนต์ต่าง ๆ ให้เลือกใช้ซึ่งแบ่งออกเป็น 2 ประเภท คือ Glossary Lists : <DL> , Regular lists : , , <MENU> and <DIR>

2.4.3.7 Horizontal Ruled Line

HR อิลีเมนต์ใช้สำหรับวาดเส้นในแนวนอน ใช้ในการแบ่งบล็อกข้อมูล จัดเป็นประเภทอิลีเมนต์ที่ว่างไม่จำเป็นต้องมี <HR> ปิดท้าย

2.4.3.8 Special Characters in HTML

สัญลักษณ์พิเศษบางตัวถูกใช้โดยเอชทีเอ็มแอลไปตามวัตถุประสงค์ที่ต่างกัน ดังนั้นเมื่อผู้ใช้ต้องการให้แสดงสัญลักษณ์พิเศษบนบราวเซอร์จึงมีเอนติตี้ของตัวอักษรพิเศษเพื่อใช้ในการแสดงสัญลักษณ์พิเศษเหล่านั้น ซึ่งเอนติตี้จะประกอบด้วย 3 ส่วน คือ

- ขึ้นต้นด้วย Ampersand “ & “
- ตัวเลขหรือข้อความของตัวอักษรที่ต้องการแสดง
- ลงท้ายด้วย Semicolon “ ; “

| Entity | Special Character | Description |
|--------|-------------------|--------------------------|
| < | < | Less than sign |
| > | > | Greater than sign |
| & | & | The amperand sign itself |
| " | “ | Double quote |
| | | A non-breaking space |

ตาราง 2.2 ตัวอย่าง Special Character ที่นิยมใช้

2.4.3.9 Character Emphasis Modes

HTML ขอมให้มีกำหนดรูปแบบของตัวอักษรได้ เช่น ตัวหนา , ตัวเอียง เป็นต้น ซึ่งแบ่งเป็น 2 วิธี คือ

Logical Styles

| | |
|--------|--|
| EM | ตัวเอียง |
| STRONG | ตัวหนา |
| CODE | ขนาดตายตัว |
| SAMP | ลำดับของตัวอักษร |
| KBD | Text ที่ถูก mark จะเหมือนกับอินพุตจากคีย์บอร์ด |
| VAR | ชื่อตัวแปร |
| DFN | กำหนดค่า |
| CITE | คำที่ใช้อ้าง มักจะเป็นตัวเอียง |

Physical styles

| | |
|----|---|
| TT | ใช้รูปแบบตัวอักษรเหมือนพิมพ์ดีด ขนาดคงที่ |
| B | ตัวหนา |
| I | ตัวเอียง |
| U | ขีดเส้นใต้ |

2.4.3.10 Special Text Formatting Modes

HTML มีรูปแบบการแสดงผลข้อความในระดับ logical อยู่ 3 แบบ แบ่งตามลักษณะการแสดงผลของบราวเซอร์ คือ

PRE (Preformatted Text)

แสดงข้อความที่อยู่ระหว่างแท็ก <PRE> และ </PRE> ให้มีรูปแบบแน่นอนตามที่กำหนด มี

เอกสาร ออปชั่นแอททริบิวต์ คือ WIDTH (default = 80) ใช้กำหนดจำนวนตัวอักษรมากที่สุดที่สามารถแสดงได้
 ไม่ไว้ในบรรทัดอื่น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCKQUOTE

ใช้สำหรับกำหนดให้บรรทัดข้อความเรียงข้อมูลที่อยู่ระหว่างแท็ก <BLOCKQUOTE> และ </BLOCKQUOTE> ตามหน้าจอแสดงผล

ADDRESS

ใช้สำหรับข้อมูลที่เกี่ยวข้องกับที่อยู่, signatures, authorship เป็นต้น

2.4.3.11 TABLES

การสร้างตารางสามารถทำได้โดยใช้ TABLE อีลีเมนต์ซึ่งมีแอททริบิวต์ต่าง ๆ ดังนี้

- ALIGN ใช้ในการกำหนดตำแหน่งของตาราง ซึ่งมีค่าที่เป็นไปได้ดังนี้ BLEEDLEFT, LEFT, CENTER, RIGHT, BLEEDRIGHT, JUSTIFY
- BORDER ใช้กำหนดขนาดของเส้นขอบของตาราง
- WIDTH ใช้กำหนดความกว้างของตารางสามารถกำหนดเป็น % ของความกว้างของส่วนที่ใช้ในการแสดงผลได้
- COLSPEC ใช้กำหนดตำแหน่งของไอเทมในคอลัมน์
- CAPTION ใช้สำหรับตั้งชื่อตาราง มีแอททริบิวต์ คือ Align ใช้สำหรับกำหนดตำแหน่งของชื่อตารางซึ่งมีค่าต่างๆ
- TH (Table Header) และ TD (Table Data) ใช้กำหนดชนิดของข้อมูลที่อยู่ในตาราง โดย TH จะเป็นตัวหนา ส่วน TD จะเป็นตัวอักษรธรรมดาทั้ง 2 อีลีเมนต์นี้มีแอททริบิวต์เหมือนกัน

2.4.3.12 FORMS Element for fill-out Forms

FORMS เป็นรูปแบบที่ทำให้ผู้ใช้สามารถติดต่อกับ HTTP เซิร์ฟเวอร์ได้มากกว่าการใช้ ISINDEX โดย FORM อีลีเมนต์จะสร้าง fill-out ฟอรัม ที่ผู้ใช้สามารถกรอกข้อมูลลงไปในช่วงที่กำหนดมาเพื่อส่งต่อไปให้ซีจีไอบนสคริปต์บน HTTP เซิร์ฟเวอร์

อีลีเมนต์ที่ใช้ในฟอรัมประกอบด้วย

FORM

ใช้สำหรับกำหนดวิธีการรับข้อมูลและวิธีการจัดการกับอินพุตที่ได้มาเพื่อให้ได้ผลลัพธ์ที่ต้องการ มีแอททริบิวต์ดังนี้

- ACTION ใช้กำหนดยูอาร์แอลของโปรแกรมหรือสคริปต์ที่จะใช้ในการกระทำงานถ้าไม่กำหนดส่วนนี้ Base ยูอาร์แอลจะถูกเรียกใช้
- METHOD เป็นการกำหนด HTTP เมธอดที่ใช้ในการส่งข้อมูลจากฟอร์มไปยังเซิร์ฟเวอร์ซึ่งมีได้ 2 แบบ คือ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. GET (default) ข้อมูลจากฟอร์มจะถูกต่อท้ายยูอาร์แอล (หลังเครื่องหมายคำถาม “?”) เหมือนกับ ISINDEX querys
2. POST ข้อมูลจากฟอร์มจะถูกส่งไปยังเซิร์ฟเวอร์ในส่วนบอดี้ของข่าวสาร การที่จะใช้วิธีนี้ต้องแน่ใจว่า HTTP เซิร์ฟเวอร์ที่ใช้สนับสนุนวิธีนี้ด้วย

INPUT

ใช้สำหรับรวบรวมข้อมูลจากผู้ใช้มีแอททริบิวต์ต่าง ๆ ดังนี้

- ALIGN ใช้กับ IMAGE TYPE ในการกำหนดตำแหน่งของ image ที่สัมพันธ์กับ text (TOP , MIDDLE หรือ BOTTOM)

- CHECKED ใช้ร่วมกับ CHECKBOX และ RADIO button ในการเลือกในตอนเริ่มต้นถ้าหากไม่กำหนดจะไม่มีตัวใดถูกเลือก

- MAXLENGTH กำหนดจำนวนตัวเลขและตัวอักษรมากที่สุดที่ผู้ใช้สามารถเติมได้ โดยปกติจะไม่ถูกจำกัด

- NAME ชื่อของตัวแปรที่ใช้ในการส่งข้อมูล

- SIZE กำหนดความกว้างของช่องที่ใช้ป้อนข้อมูล

- SRC ใช้กำหนดยูอาร์แอลของ image ไฟล์เมื่อไทป์ถูกกำหนดเป็นแบบ image

TYPE กำหนดรูปแบบของฟิลด์ที่ให้ผู้ป้อนข้อมูล

- CHECKBOX สามารถให้ค่าได้หลายค่า

- HIDDEN สำหรับค่าที่ถูกกำหนดโดยฟอร์มโดยไม่ต้องรับอินพุตจากผู้ใช้

- IMAGE ใช้รูปภาพเป็นตัวรับอินพุตจากผู้ใช้โดยส่งค่า x, y ที่ถูกคลิกโดยเมาส์

PASSWORD เป็นฟิลด์สำหรับป้อนข้อมูลแบบเท็กซ์แต่ไม่แสดงผล

- RADIO ใช้สำหรับเก็บข้อมูลที่สามารถเลือกได้เพียงครั้งเดียวในแต่ละครั้ง

- RESET ใช้เคลียร์ค่าให้กลับเป็นค่ามาตรฐาน

- SUBMIT เป็นปุ่มที่ใช้สำหรับการส่งข้อมูลในฟอร์มใช้ VALUE ในการกำหนดข้อความที่แสดงบนปุ่ม

- TEXT ใช้สำหรับ single – line of text

- TEXT เป็นช่องให้เติมข้อมูลแบบเท็กซ์บรรทัดเดียว

- VALUE ใช้กำหนดค่าเริ่มต้นของฟิลด์หรือกำหนดค่าของฟิลด์เมื่อถูกเลือก

- SELECT อีลีเมนต์ที่จะแสดงทางเลือกสำหรับให้ผู้ใช้เลือกค่าใดค่าหนึ่งจากหลาย ๆ ค่าโดยใช้
ออปชันเป็นตัวกำหนดค่าต่าง ๆ เหล่านั้นมีแอททริบิวต์ต่าง ๆ ดังนี้

- NAME ชื่อที่ใช้ในการส่งข้อมูลที่ผู้ใช้เลือก

- MULTIPLE โดยปกติผู้ใช้สามารถเลือกใช้ได้เพียงอย่างเดียวแต่ MULTIPLE แอททริบิวต์นี้จะทำให้ผู้ใช้สามารถเลือกได้มากกว่าหนึ่งอย่าง

- SIZE กำหนดจำนวนของไอเทมที่ผู้ใช้สามารถมองเห็นได้ ถ้าไอเทมมีมากกว่าที่กำหนดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

แสดงผลจะอยู่ในรูปของลิสต์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPTION

อ็ลลลเมนนท์น้จะปรากฎภายใน SELECT อ็ลลลเมนนท์และใช้การแสดงตัวเลอ้กต่าง ๆ ภายใน SELECT อ็ลลลเมนนท์แเอททรรลลวท้ต่าง ๆ ดั่งน้

- SELECTED กำหนดให้ออปชั่นถูกลลลลในทอนเรมด้น
- VALUE ใช้กำหนดค่าที่จะถูกส่งกลับไปย้งเซิร์ฟเวอร์ของแต่ลลออปชั่น ถ้าไม่มีแเอททรรลลวท้น้ ค่าที่จะถูกส่งกลับไปคือค่าถูกกำหนด โดยออปชั่นแเอททรรลลวท้

TEXT AREA

อ็ลลลเมนนท์น้จะใช้ในการรวบรวมข้อมูลชนลลลทลลลท้มีหลายบรรท้ดจากผู้ใช้น้ Scroll Bar ทำให้อสามารถลลลลนไปย้งตำแหน่งต่าง ๆ ได้ TEXTAREA มีแเอททรรลลวท้ต่าง ๆ ดั่งน้

- NAME ชลลลที่ จะใช้ในการส่งค่ากลับ
- ROWS จำนวนของแถวที่แสดงผล
- COLS จำนวนของคอลลลลนที่แสดงผล



เอกสารน้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งลลลน อลลลท้ทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ความรู้เกี่ยวกับเซิร์สเอนจิน

3.1 ประเภทของเซิร์สเอนจิน

- 1 เซิร์สเอนจิน (search engines) บางครั้งอาจเรียกว่า สไปเดอร์ (spider) หรือ คอรัลเลอร์ (crawler) เป็นเว็บโรบอทที่วิ่งเก็บเว็บไซต์ต่าง ๆ อย่างอัตโนมัติ เช่น ฮอทบอท (Hotbot) เป็นต้น
- 2 ไดเร็คทอรี (Directories) เป็นการแบ่งกลุ่มของเว็บไซต์โดยใช้นุญช์ โดยเว็บไซต์ที่จะถูกจัดกลุ่มจะต้องถูกส่งไปให้เซิร์สเอนจินเพื่อทำการแบ่งแยกหมวดหมู่ต่อไป เช่น ยาฮู (Yahoo) เป็นต้น
- 3 ไฮบริดเซิร์สเอนจิน (Hybrid search engines) เป็นเซิร์สเอนจินที่มีการจัดการแบบไดเร็คทอรี

3.2 การออกแบบเซิร์สเอนจิน

ในการทำการค้นหาข้อมูลของเซิร์สเอนจินนั้นมันจะมีผลลัพธ์ที่สัมพันธ์กับเว็บเพจต่าง ๆ กว่าร้อยถึงพันเพจ ซึ่งมีเพียง 10 เพจที่สัมพันธ์กับความต้องการที่สุดเท่านั้นที่ถูกนำออกแสดงหน้าจอก่อน ดังนั้นในการที่จะทำให้เว็บไซต์ของคุณสามารถติดอันดับ 1 ใน 10 ได้ จะต้องกระทำดังนี้

- 1 เลือกคำที่ใช้เป็นคีย์เวิร์ด : นั่นคือคีย์เวิร์ดควรจะประกอบด้วยคำตั้งแต่ 2 คำขึ้นไป
- 2 ตำแหน่งของคีย์เวิร์ด : ต้องอยู่ในตำแหน่งที่สำคัญภายในเว็บเพจ เช่น ตำแหน่งไตเติล หรือจัดให้คีย์เวิร์ดอยู่ในส่วนบนของเพจ หรืออยู่ในย่อหน้าแรก ระวังไว้ว่าตารางจะทำให้ส่วนข้อความ (text) นั้นตกไปอยู่ด้านล่างของเพจ ซึ่งจะทำให้คีย์เวิร์ดที่มีอยู่ในส่วนนี้มีความสำคัญน้อยลง เนื่องจากตารางจะดึงความสนใจของเซิร์สเอนจินไปที่ตารางเอง ทำให้ส่วนข้อความนี้อยู่ในลำดับความสนใจที่น้อยลง เช่นเดียวกับส่วนจาวาสคริปต์ก็จะดึงความสนใจของเซิร์สเอนจินไปเช่นเดียวกับการมีตาราง เนื่องจากเซิร์สเอนจินจะอ่านข้อมูลในสคริปต์ก่อนซึ่งทำให้ข้อความเอชทีเอ็มแอลทั่วไปปรากฏอยู่ในส่วนล่างของเพจ
- 3 มีข้อมูลตรงกัน : การเปลี่ยนแปลงไตเติลและเพิ่มเมต้าแท็ก (Meta Tag) ก็ไม่ได้ช่วยให้เพจนั้นมีคีย์เวิร์ดที่ดีได้ คีย์เวิร์ดนั้นจำเป็นที่จะต้องสามารถสะท้อนถึงข้อมูลของเพจได้ นั่นหมายถึงเราต้องมีข้อความเอชทีเอ็มแอลในเพจ เนื่องจากในบางครั้งไซต์ต่าง ๆ ก็มีการนำเสนอส่วนต่าง ๆ ด้วยกราฟิก ซึ่งคู่มือเซิร์สเอนจินไม่สามารถอ่านกราฟิกเหล่านี้ได้ซึ่งการไม่มีเนื้อข้อมูลเป็นการทำให้ไซต์นั้นดูเหมือนว่าจะไม่มีข้อมูลที่ตรงตามคีย์เวิร์ด สำหรับเซิร์สเอนจินบางตัวจะมีการทำอินเด็กซ์ (index) ไปยัง ALT text และมีการอธิบายกำกับข้อมูลไว้ในเมต้าแท็กด้วย
- 4 หลีกเลี่ยงเพจที่มีข้อมูลไม่เหมาะสม : เซิร์สเอนจินบางตัวจะมีวิธีการไปดูเว็บด้วยวิธีที่ใช้กับบราวเซอร์รุ่นเก่า ซึ่งไม่สามารถอ่าน image maps , Frames ได้ ดังนั้นเราจึงต้องคำนึงถึงปัญหานี้ด้วย หรือเซิร์สเอนจินก็ไม่ควรทำอินเด็กซ์กับเว็บเพจนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การมี **HTML link** : บ่อยครั้งที่นักออกแบบมักจะทำเฉพาะ image map links จากโฮมเพจไปยังเพจที่อยู่ภายใน ซึ่งเซิร์ฟเวอร์เงินไม่สามารถตามเข้าไปอ่านได้ และโดยมากเพจที่ตรงกับความต้องการมักจะอยู่ภายในเพจมากกว่าอยู่ที่โฮมเพจ การแก้ปัญหาคือเพิ่ม HTML hyperlinks เข้าไปในโฮมเพจด้วยเพื่อให้เซิร์ฟเวอร์เงินสามารถตามเข้าไปอ่านข้อมูลในเพจได้

Dynamic Doorblock : การเงินเนอเรตเพจผ่านซีจีไอจะทำให้เซิร์ฟเวอร์เงินบางตัวไม่สามารถทำอินเด็กซ์ได้ ดังนั้นการสร้างสเตติกเพจแล้วให้ฐานข้อมูลทำการอัปเดตเพจเหล่านั้น รวมถึงการเลี่ยงการใช้สัญลักษณ์ต่าง ๆ ในยูอาร์แอลโดยเฉพาะเครื่องหมาย ? จะทำให้เซิร์ฟเวอร์เงินสามารถทำงานได้

- 5 **การใช้เมต้าแท็ก** : ช่วยแก้ปัญหาที่เกิดจากการมีตาราง เฟรม ได้รวมถึงช่วยในการควบคุม description ของไซท์ในเอนจินที่สนับสนุนอยู่ได้
- 6 **ปฏิเสธ Search Engine Spamming** : เซิร์ฟเวอร์เงินไม่ทำงานกับ spamming โดยเซิร์ฟเวอร์เงินจะค้นหา spamming แล้วทำการกำจัดเพจนั้นออกจากลิสต์
- 7 **ลงทะเบียนเพจที่สำคัญๆ** : คุณควรที่จะลงทะเบียนเพจที่สำคัญอันดับแรกๆ ของ เว็บไซต์ของคุณนอกจากคอยให้เซิร์ฟเวอร์เงิน ค้นหาเพจเหล่านั้นเอง เพราะบางตัวอาจหาไม่พบก็ได้
- 8 **คอยทดสอบเพจของคุณในลิสต์** : เมื่อคุณลงทะเบียนแล้วควรจะทดลองค้นหาดูว่าเพจของคุณปรากฏอยู่ในลิสต์ของเซิร์ฟเวอร์เงิน และควรตรวจสอบทุกๆ หนึ่งถึงสองสัปดาห์ และถ้าหากเกิดปัญหาควรจะรีบแก้ไขทันที
- 9 **การเลือกคำ** : ควรเลือกคำอธิบายเพิ่มเติมเพราะคำที่ดีจะทำให้เซิร์ฟเวอร์เงินเลือกเพจของเราขึ้นก่อนหน้าเพจอื่น
- 10 **การที่คนเลือกที่จะเข้าเพจได้ไม่ได้ผ่านเพียงเซิร์ฟเวอร์เงินเท่านั้น** ยังสามารถผ่านทาง สื่อท้องถิ่น, newsgroup, หรือลิงค์จากเว็บอื่นๆ

ต่อไปนี้เป็นตารางสรุปตัวแปรต่าง ๆ ที่มีผลต่อการทำอินเด็กซ์และจัดลำดับ ของแต่ละเซิร์ฟเวอร์เงิน

| Crawling | Yes | No | Notes |
|------------------|-------------------------------------|---|--|
| Deep Crawl | Alta Vista , Inktomi , NLight | Excite , Infoseek Lycos , WebCrawler | WebCrawler only lists Home pages |
| Instant Indexing | Alta Vista , Infoseek MSN Search | Excite , Inktomi , Lycos Nlight , Google | Pages will appear within a day or two after submission |
| Frames Support | Alta Vista , Nlight | Excite , Inktomi , Infoseek , Lycos | Lycos provides limited support |
| Image Maps | Alta Vista , Infoseek , NLight | Excite , Inktomi , Lycos | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Crawling | Yes | No | Notes |
|----------------------------------|--------------------------|------------------------------------|-------|
| Password Protected Sites | Alta Vista , Inktomi | Excite , Infoseek , Lycos , Nlight | |
| Robot.txt | All | n/a | |
| Meta Robots Tag | All | n/a | |
| Link Popularity Helps Deep Crawl | Excite , Inktomi , Lycos | Alta Vista , Infoseek , Nlight | |
| Learns Frequency | Alta Vista , Infoseek | Excite , Inktomi , Lycos Nlight | |

ตารางที่ 3.1 สรุปวิธีการไปยังเว็บไซต์ของเซิร์ฟเวอร์

Crawling เป็นตัวแปรที่บอกถึงวิธีการไปยังเว็บไซต์ต่าง ๆ ของเซิร์ฟเวอร์แต่ละตัว ประกอบด้วย

- **Deep Crawl** การไปยังเว็บไซต์ต่าง ๆ ของแต่ละเซิร์ฟเวอร์ด้วยการนำเพจหลาย ๆ เพจในเว็บไซต์มาทำเป็นลิ้งค์ถึงแม้ว่าเพจนั้นจะยังไม่ถูกซัชมิต (submit) ก็ตาม โดยทั่วไปแล้วยังเซิร์ฟเวอร์มีอินเด็กซ์ใหญ่มากเท่าใดจำนวนเพจต่อไซต์ก็ยังมีมากเท่านั้น
- **Instant Indexing** เซิร์ฟเวอร์ที่ใช้วิธีนี้จะแสดงเพจที่ได้ซัชมิตแล้วประมาณ 1-2 วันหลังจากซัชมิต
- **Frames Support** วิธีนี้ทำเซิร์ฟเวอร์สามารถเข้าไปดูข้อมูลตามเฟรมลิงค์ได้
- **Image Maps** เซิร์ฟเวอร์สามารถตามเข้าไปดูข้อมูลใน image maps ฟังก์ชันโคลเอ็นท์ได้
- **Password Protected Sites** เซิร์ฟเวอร์บางตัวสามารถเข้า password ของ password protected site ได้ เนื่องจากเว็บมาสเตอร์ได้ทำ user name และ password ไว้ให้ เพื่อต้องการให้รู้ว่าไซต์นี้มีข้อมูลอยู่
- **robot.txt** เป็นวิธีที่เว็บมาสเตอร์ใช้สำหรับกันไม่ให้เซิร์ฟเวอร์เข้ามาดูข้อมูลที่ไซต์ของตน
- **Meta Robots Tag** ใช้สำหรับเว็บมาสเตอร์ที่ไม่ได้ทำ ไฟล์ robot.txt
- **Link Popularity Helps Deep Crawl** เซิร์ฟเวอร์สามารถตรวจสอบว่าเพจใดเป็นที่นิยมมาก ๆ ได้จากการวิเคราะห์หาจำนวนลิงค์ที่ลิงค์มายังเพจ ๆ นั้น ซึ่งเซิร์ฟเวอร์บางตัวจะใช้วิธีนี้ในการตรวจหาว่าเพจใดที่ควรนำมารวมไว้ในอินเด็กซ์
- **Learns Frequency** เซิร์ฟเวอร์หลายตัวที่สามารถศึกษาความบ่อยในการเปลี่ยนแปลงของเพจแต่ละเพจ เพจที่เปลี่ยนแปลงบ่อย จะต้องเข้าไปดูข้อมูลบ่อยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Indexing | Yes | No | Notes |
|------------------|--|------------------------------|------------------------------------|
| Full Body Text | All | n/a | Some stop words may not be indexed |
| Stop Words | Alta Vista , Excite , Inktomi , Lycos , Google | Infoseek , NLight | |
| Meta Description | All but.... | Lycos , NLight | |
| Meta Keywords | All but | Excite , Lycos , NLight | |
| ALT text | Alta Vista , Infoseek , Lycos | Excite , Inktomi , NLight | |
| Comments | Inktomi | Others | |

ตารางที่ 3.2 สิ่งที่น่ามาเป็นอินเด็กซ์

Indexing เป็นการกล่าวถึงสิ่งที่นำกลับมาเป็นอินเด็กซ์ หลังจากที่เซิร์ฟเอนจินไปที่เพจนั้นแล้ว

- **Full Body Text** เซิร์ฟเอนจินหลัก ๆ ทุกตัวทำอินเด็กซ์ของส่วน body text ทั้งหมดของเพจ โดยบางตัวจะไม่รวมเอา stop words ไว้ หรือ คัดเอา spam ออก
- **Stop Words** เซิร์ฟเอนจินบางตัวจะทิ้งคำไว้เมื่อได้ทำอินเด็กซ์ของเพจนั้นแล้ว หรืออาจจะไม่ค้นหาคำเหล่านี้ในระหว่างที่ทำคิวรี (query) ซึ่ง stop words เหล่านี้จะถูกคัดออกเพื่อช่วยประหยัดเนื้อที่หรือช่วยในด้านความเร็ว
- **Meta Description & Meta Keywords** ใช้เพื่อแสดงว่าเซิร์ฟเอนจินใดสนับสนุน meta description และ meta keywords tags แต่ไม่ได้หมายความว่าแท็กเหล่านี้จะช่วยให้เพจนั้นมีอันดับในระดับสูง
- **ALT Text / Comments** ใช้เพื่อแสดงว่าเซิร์ฟเอนจินใดที่ทำอินเด็กซ์ของ ALT text ซึ่งสัมพันธ์กับ images หรือ text ใน comment tags

| Ranking | Yes | No | Notes |
|-------------------------------|--|--|------------------------------|
| Meta Tags Boost Ranking | Infoseek , Inktomi | Alta Vista , Excite , Lycos , Nlight | |
| Reviewed Status Boost Ranking | Excite , Infoseek | Alta Vista , Inktomi , Lycos , Nlight | Very important with Infoseek |
| Link Popularity Boost Ranking | Alta Vista , Excite , Google , Infoseek | Inktomi , Lycos , Nlight | Most important at Google |
| Direct Hit Boost Ranking | HotBot | Others | |

ตารางที่ 3.3 วิธีการจัดอันดับความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษารายละเอียดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการศึกษา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ranking โดยส่วนมากแล้วเซิร์สเอนจินจะใช้ บริเวณ และ ความบ่อย ที่พบคีย์เวิร์ดบนเว็บเพจ เป็นพื้นฐานในการจัดลำดับเพื่อสนองต่อคิวิที่เข้ามา โดยวิธีการนั้นก็แตกต่างกันไปตามแต่เซิร์สเอนจินนั้น ๆ

- **Meta Tags Boost Ranking** เซิร์สเอนจินบางตัวที่สนับสนุน meta description and keywords tag จะให้สิทธิพิเศษแก่เพจนั้น ๆ ในกรณีที่มี search terms อยู่ในบริเวณนั้นนั้น ไม่ได้หมายความว่าเซิร์สเอนจินทุกตัวที่สนับสนุนแท็กนี้ก็จะได้รับการจัดลำดับไว้ในตำแหน่งที่สูง
- **Reviewed Status Boosts Rankings** เซิร์สเอนจินบางตัวมีการรีวิว (review) หรือ นำไซท์นั้นมาเก็บในลิสต์ซึ่งสัมพันธ์กับ ไคเร็กทอรี ซึ่งอาจจะมีการให้สิทธิพิเศษแก่ไซท์ที่ถูกเก็บในลิสต์ด้วยวิธีการนี้ก็ได้
- **Link Popularity Boosts Rankings** เป็นจัดอันดับตามจำนวนลิงค์ที่มีมายังเพจนั้น
- **Direct Hit Boosts Rankings** จัดอันดับจากจำนวนการคลิกหลังจากที่ได้แสดงผลลัพธ์หลังจากการค้นหา

| Spam | Yes | No | Notes |
|----------------|---------------------------------|------------------------------------|-------|
| Meta Refresh | Alta Vista , Infoseek | Excite , Inktomi , Lycos Nlight | |
| Invisible Text | Others | Excite | |
| Tiny Text | Alta Vista , Inktomi , Lycos | Excite , Infoseek , NLight | |

ตารางที่ 3.4 การคัด spam ออก

Spam เซิร์สเอนจินหลัก ๆ ทุกตัวจะมีการกันไซท์ที่พยายามจะอวดอ้างตนเองเพื่อให้อันดับในลำดับสูงออกไป

- **Meta Refresh** เจ้าของไซท์บางที่ได้สร้างเพจเป้าหมายซึ่งจะนำผู้ที่เข้าชมไซท์นั้นไปยังเพจที่ต่างไปโดยอัตโนมัติซึ่งอยู่ในเว็บไซท์นั้นเอง ซึ่ง meta refresh tag ก็เป็นวิธีหนึ่งที่จะทำอย่างที่ว่ามา ซึ่งเซิร์สเอนจินบางตัวจะไม่ทำอินเด็กซ์เพจที่มีอัตร meta refresh สูง
- **Invisible Text** เป็นวิธีการพราง text ในเพจด้วยการทำให้ text นั้นมีสีเหมือนกันสีของแบล็กกราวนด์ ซึ่งมนุษย์ไม่สามารถมองเห็นได้ เซิร์สเอนจินหลายตัวที่ปฏิเสธไม่ทำอินเด็กซ์ของ text ประเภทนี้ หรือบางตัวไม่ทำอินเด็กซ์เพจใด ๆ ที่มี invisible text
- **Tiny Text** คือการนำให้ text นั้นมีขนาดฟอนท์ (font) ที่เล็กมาก เพจที่มี text ที่มีขนาดเล็กมาก ๆ อาจจะไม่ได้ถือว่าเป็น spam หรืออาจจะไม่ทำอินเด็กซ์ของ tiny text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ตัวอย่างของเซิร์สเอนจินที่ได้รับความนิยม

3.3.1 Alta Vista เป็นเซิร์สเอนจินที่เร็วและมีความสามารถสูงในการค้นหาข้อมูลที่ซับซ้อน

ชนิดของการค้นหา : ค้นหาด้วยคีย์เวิร์ด

อุปสรรคในการค้นหา : ธรรมดา หรือ ค้นหาล่วงหน้า , ค้นหาละเอียด

ค้นหาจาก : เว็บ , ยูสเน็ต

การค้นหาอย่างละเอียด : ใช้บูลีน “AND”, “OR”, “NOT”, “NEAR” สามารถค้นหาแบบย้อนกลับได้ (เช่น สามารถพบเว็บไซต์ที่ลิงก์ไปเพจอื่นได้) เราสามารถตัดสินใจได้ว่า search terms ควรมีน้ำหนักเท่าใดและจะหาได้จากส่วนใดในเอกสาร ในการค้นหาแบบละเอียดนี้ยังให้เครื่องมือช่วย (tools) ที่มีประสิทธิภาพรวมกับความละเอียดจะทำให้ได้ผลลัพธ์ที่ดี

การจัดลำดับของความสัมพันธ์ : ขึ้นอยู่กับจำนวนของ search terms ในเพจ , อยู่ในส่วนของเอกสาร , ความใกล้เคียงกับ search terms อื่น ๆ

การนำเสนอผลลัพธ์ด้วย : บรรทัดแรก ๆ ของเอกสาร

การติดต่อกับผู้ใช้ : มีการตอบสนองที่ดี ผู้ใช้ที่เพิ่งเคยใช้จะรู้สึกว่ายาก การคลิกล่วงหน้าจะทำให้สามารถค้นหาได้ละเอียดขึ้นจะมีอยู่ตอนท้ายของแต่ละ results page

ไฟล์ช่วยเหลือ : สมบูรณ์ แต่จะเข้าใจยาก

ข้อดี : ค้นหาได้เร็ว , รู้จักอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่รวมถึงคำเฉพาะ , มีฐานข้อมูลขนาดใหญ่ , สามารถค้นพบในสิ่งที่เซิร์สเอนจินอื่นทำไม่ได้ , Alta Vista จะค้นหาทั้งที่เว็บและที่ยูสเน็ต และค้นหาทั้งที่เป็นคำและเป็นวลี รวมทั้งชื่อและไต่เคิ้ล เราสามารถทราบได้ด้วยว่ามีกี่คนที่ทำลิงค์มาที่ไซต์ของเรา ในหน้าผลลัพธ์จากการค้นหาสามารถแปลได้เป็นหลายภาษา

ข้อด้อย : เพจหลายเพจจากไซต์เดียวกันจะแสดงออกมาบ่อยเกินไป , บางครั้งการจัดอันดับจะแปลก ๆ โดยเฉพาะจากการค้นหาแบบธรรมดา

โดยรวม : A

3.3.2 HotBot

ชนิดของการค้นหา : ค้นหาด้วยคีย์เวิร์ด

อุปสรรคในการค้นหา : แบบธรรมดา , แบบปรับปรุง , แบบผู้เชี่ยวชาญ

ค้นหาจาก : เว็บ

การค้นหาอย่างละเอียด : มีหลายชนิด , รวมทั้งด้วยวลี , บุคคลและบูลีน เช่นตัวเลือกจาก pull-down boxes สำหรับการค้นหาแบบผู้เชี่ยวชาญเราสามารถค้นหาด้วยวันที่หรือแม้กระทั่งโดย media types ต่าง ๆ เช่น Java , Javascript , shockwave , VRML

การจัดลำดับของความสัมพันธ์ : search terms ในไต่เคิ้ล จะถูกจัดอันดับไว้สูงสูงกว่าที่อยู่ในส่วน text มีการนับความบ่อยและการจัดลำดับในลำดับที่สูงกว่าถ้า search terms มีบ่อยในเอกสารที่สั้นกว่าการนำเสนอผลลัพธ์ด้วย : คะแนนและยูอาร์แอลที่สัมพันธ์กัน

การติดต่อกับผู้ใช้ : เจ้งและใช้ง่าย ผู้ใช้บางคนอาจไม่ชอบใจกับแบล็กกราวนด์สีเขียวสว่าง

ไฟล์ช่วยเหลือ : FAQ สำหรับตอบคำถามจากผู้ใ้

ข้อดี : อ้างว่าค้นหาได้เร็วเพราะทำงานแบบขนานเป็นการกระจายโหลดของคิวรี

ข้อด้อย : มีข้อจำกัดบางประการเกี่ยวกับ บูลีน และไฟล์ช่วยเหลือยังไม่ดีเท่าที่ควร

โดยรวม : B

3.3.3 Yahoo แม้จะไม่ใช่เซิร์สเอนจินไซท์ที่ถูกต้องแม่นยำแต่ก็เป็นเว็บรีซอร์สที่สำคัญ เนื่องจากการทำงานเป็นระดับขั้น

ชนิดของการค้นหา : ค้นหาด้วยคีย์เวิร์ด

อุปชันในการค้นหา : แบบธรรมดา , แบบล่องหน้า

ค้นหาจาก : อินเด็กซ์ของยาฮูเอง, ยูสเน็ต, อี-แมล์ แอคเดรส, ยาฮูเซิร์สไต่เต็ล, ยูอาร์แอลและคำจำกัดความสั้นจากเว็บไซท์ยาฮูอินเด็กซ์

การค้นหาโดยละเอียด : การจัดลำดับยังไม่ชัดเจน

การนำเสนอผลลัพธ์ด้วย : บอกหมวดหมู่ที่พบข้อมูลและให้คำอธิบายประมาณ 2 บรรทัด

การติดต่อผู้ใช้ : ดีมาก , ใช้ง่าย

ไฟล์ช่วยเหลือ : ยังไม่สมบูรณ์ แต่เนื่องจากมีอุปชันน้อยจึงไม่ค่อยจำเป็นต้องใช้

ข้อดี : แบ่งหมวดหมู่ไว้ให้ใช้ง่าย หากรู้ว่าการค้นหาอะไร ยาฮูจะเป็นผู้ช่วยค้นหาได้ดีทีเดียว

ข้อด้อย : การแยกหมวดหมู่โดยยาฮูทำได้กับส่วนเล็ก ๆ ในเว็บ

โดยรวม : A

บทที่ 4

การออกแบบและการสร้าง

เมื่อทำการศึกษากฎเกณฑ์ต่าง ๆ แล้ว ขั้นตอนต่อไปของการทำโครงการนี้คือ การออกแบบและการสร้าง โดยการออกแบบนี้จะแบ่งเป็นฟังก์ชันต่าง ๆ ในการทำงาน

4.1 การกำหนดฟังก์ชันการทำงานของตัวค้นหาข้อมูล

การทำงานของตัวค้นหาข้อมูลประกอบด้วยส่วนต่าง ๆ ดังนี้

1. ฟังก์ชันในการดึงข้อมูลและวิเคราะห์ข้อมูล
 - ฟังก์ชันในการตรวจสอบยูอาร์แอล
 - ฟังก์ชันในการดึงข้อมูล
 - ฟังก์ชันการเทียบคำ
 - ฟังก์ชันในการคำนวณค่าความสำคัญ
 - ฟังก์ชันในการเก็บค่าลงในฐานข้อมูล
2. ฟังก์ชันในการติดต่อระหว่างผู้ใช้บริการและเซิร์ฟเวอร์โดยผ่าน โปรแกรมเซิร์ฟเล็ต
 - ฟังก์ชันในการเรียกใช้บริการจากเซิร์ฟเวอร์
 - ฟังก์ชันในการรับข้อมูลจากผู้ใช้บริการมายังเซิร์ฟเวอร์
 - ฟังก์ชันในการส่งข้อมูลจากเซิร์ฟเวอร์กลับไปยังผู้ใช้บริการ
3. ฟังก์ชันในการค้นหาข้อมูลบนฐานข้อมูล

4.2 การออกแบบในส่วนที่ใช้ดึงข้อมูลและวิเคราะห์ข้อมูล

เป็นการกำหนดอัลกอริทึม (Algorithm) ต่าง ๆ ที่เกี่ยวข้องกับการทำงานของโรบอทในส่วนที่เกี่ยวข้องกับการดึงข้อมูลและวิเคราะห์ข้อมูล

4.2.1 การเก็บคีย์เวิร์ดจากเว็บเพจ

หลังจากที่โรบอทได้ดึงเว็บเพจกลับมาแล้ว ในขั้นตอนต่อมาคือการวิเคราะห์ข้อมูลนั่นคือการนำคีย์เวิร์ดในลิสต์มาเทียบกับคำในเพจที่ดึงมา ถ้าในเพจนั้นมีคำตรงกับคีย์เวิร์ดในลิสต์ก็จะเข้าสู่ขั้นตอนการคำนวณหาค่าความสำคัญแล้วจึงเก็บลงในฐานข้อมูลเป็นลำดับต่อไป

4.2.2 การคำนวณค่าความสำคัญของคำที่ปรากฏในเว็บเพจ

ภายหลังจากการเทียบคีย์เวิร์ดในลิสต์กับคำในเพจที่ดึงมา คีย์เวิร์ดที่มีในเพจจะต้องนำมาคำนวณค่าความสำคัญเพื่อทำการจัดเก็บลงในฐานข้อมูลเพื่อใช้เป็นคีย์เวิร์ดในการค้นหาต่อไป ซึ่งวิธีการคำนวณหาค่าความสำคัญของคำต่าง ๆ สามารถหาได้โดยใช้สูตร ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

$$\text{Priority} = S(\text{Priority_Key}_i)$$
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ Priority คือ ค่าความสำคัญของคำ n ใด ๆ

Priority_Key คือ ค่าความสำคัญของคำที่อยู่ ณ ตำแหน่ง i

จากสูตร การคำนวณหาค่าความสำคัญของคำ ทำได้โดยนำค่า Priority_Key ของตำแหน่งที่ปรากฏ คำ ๆ หนึ่งทั้งหมดในเว็บเพจมารวมกันเพื่อให้ได้ค่า Priority ของคำ ๆ นั้น

ตัวอย่าง

คำว่า " มหาวิทยาลัย " ปรากฏอยู่ในเว็บเพจ 3 ที่

ตำแหน่งที่ 1 มีค่า Priority_Key₁ = 100

ตำแหน่งที่ 2 มีค่า Priority_Key₂ = 50

ตำแหน่งที่ 3 มีค่า Priority_Key₃ = 15

เพราะฉะนั้น Priority = 100 + 50 + 15
= 165

จากค่าความสำคัญที่ได้นี้ยังมีค่ามากเท่าใดก็หมายถึงเพจนั้นมีแนวโน้มจะเป็นเพจที่ตรงตามความต้องการของผู้ใช้มากเท่านั้น

4.2.3 การกำหนดค่า Priority_Key ของคำในตำแหน่งต่าง ๆ

ค่า Priority_Key เป็นค่าที่ใช้กำหนดความสำคัญของคำ ณ ตำแหน่งใด ๆ ในเว็บเพจ สิ่งที่จะใช้กำหนดค่า Priority_Key นี้คือ แท็ก นั่นคือ ค่าของ Priority_Key จะแตกต่างกันไปมากน้อยขึ้นอยู่กับว่า ณ ตำแหน่งนั้นอยู่ระหว่างแท็กใด เช่น ค่า Priority_Key ณ ตำแหน่งที่อยู่ระหว่าง <TITLE> กับ </TITLE> จะมีค่า Priority_Key มากกว่าค่า Priority_Key ณ ตำแหน่งที่อยู่ระหว่าง <H1> กับ </H1> เป็นต้น และแท็กบางแท็กจะไม่มีผลต่อค่า Priority_Key เลย

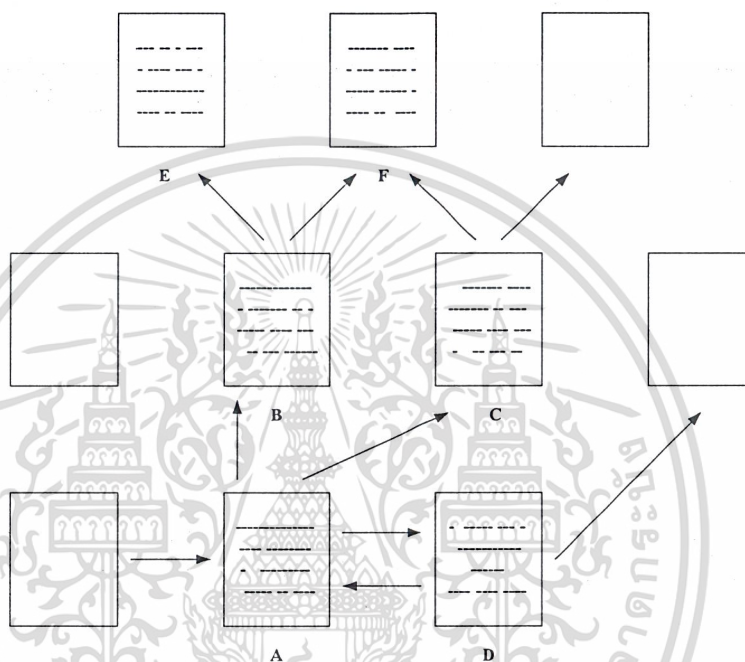
4.2.4 วิธีการท่องเว็บไซต์ของโรบอท

ในครั้งแรกของการทำงานของโรบอท โรบอทจะต้องติดต่อไปยังเว็บไซต์ต่าง ๆ ในอินเทอร์เน็ตที่ถูกกำหนดเอาไว้ในลิสต์และทำการขอข้อมูลจากเว็บไซต์นั้นให้ส่งกลับมาที่เซิร์ฟเวอร์ จากนั้นจึงส่งข้อมูลหรือเว็บเพจที่ได้ไปทำการเทียบหาคีย์เวิร์ดกับคำที่มีในเพจนั้นเก็บไว้เป็นฐานข้อมูลเพื่อใช้ในการค้นหา นอกจากนี้ยังต้องเลือก ยูอาร์แอล ที่ปรากฏในเว็บเพจนั้นเอาไปใส่ในลิสต์

ในการค้นหาข้อมูลในครั้งต่อ ๆ มาหากยังมีคีย์เวิร์ดที่ผู้ใช้ต้องการค้นหาซึ่งไม่มีในฐานข้อมูล (เก็บอยู่ใน un_search list) ยังมีไม่ครบจำนวนที่ได้กำหนดไว้ โรบอทจะทำการติดต่อไปยังเว็บไซต์ที่ยังไม่ได้ติดต่อไปต่อไปเรื่อย ๆ เมื่อใดที่จำนวนคีย์เวิร์ดใน un_search list มีจำนวนตามที่กำหนดไว้ โรบอทจะทำการย้อนกลับไปเริ่มต้นเว็บที่เว็บไซต์แรกเพื่อทำการเทียบคีย์เวิร์ดที่มีอยู่ใน un_search list กับคำในเพจนั้น ๆ อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการค้นหาข้อมูลจากเว็บไซต์ต่างๆของ ีบอทเก็บข้อมูลนี้จะใช้วิธีการค้นหาในทางกว้างก่อน (Breadth-First Search) นั่นคือ ีบอทจะทำการดึงข้อมูลจากเว็บไซต์เริ่มต้นที่ถูกกำหนดไว้ในลิสต์ของยูอาร์แอลที่จะต้องทำการเก็บข้อมูล เมื่อได้ข้อมูลที่ต้องการแล้วจึงส่งข้อมูลที่ได้ไปให้ส่วนวิเคราะห์ทำหน้าที่เทียบหาค่าที่ตรงคีย์เวิร์ดจากผู้ใช้เพื่อจะได้เก็บไว้เป็นฐานข้อมูลเพื่อใช้ในการค้นหาต่อไป ส่วนยูอาร์แอลที่ปรากฏอยู่ในเว็บเพจที่ติดต่อยังยูอาร์แอลอื่นจะถูกนำไปตรวจสอบ (อยู่ในประเทศไทย หรืออยู่ในลิสต์ของยูอาร์แอลที่เราสนใจ) และถ้ายูอาร์แอลนั้นผ่านการตรวจสอบ ก็จะถูกนำไปต่อท้ายไว้ในลิสต์เพื่อให้ ีบอทเก็บ ข้อมูลใช้ในการติดต่อยังข้อมูลต่อไป



รูปที่ 4.1 การเก็บข้อมูลโดยวิธีการหาทางกว้างก่อน

ตัวอย่าง

แสดงลิสต์ของยูอาร์แอลที่ ีบอทจะต้องติดต่เพื่อขอข้อมูล เมื่อ ีบอทต้องทำการเก็บข้อมูลจากยูอาร์แอลตามรูปข้างบน โดยสมมติให้ ยูอาร์แอลทั้งหมดที่ปรากฏในรูปเป็น ยูอาร์แอลที่อยู่ในประเทศไทย

| | | | | | | | | | |
|---|------|--|--|--|--|--|--|--|--|
| A | NULL | | | | | | | | |
|---|------|--|--|--|--|--|--|--|--|

ลิสต์ 1. ลิสต์รายการเริ่มต้นเป็นลิสต์ที่กำหนดไว้ตอนเริ่มต้นเพื่อให้ ีบอทใช้เป็นยูอาร์แอลเริ่มต้นในการติดต่อยังข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | | | | | | |
|---|---|---|------|--|--|--|--|--|--|
| B | C | D | NULL | | | | | | |
|---|---|---|------|--|--|--|--|--|--|

ลิสต์ 2. หลังจากได้ข้อมูลจาก “A” แล้ว นำเอา “A” ออกจากลิสต์และนำยูอาร์แอลที่ “A” ติดต่อกันไปตรวจสอบเมื่อผ่านการตรวจสอบแล้ว นำยูอาร์แอลที่ผ่านการตรวจสอบทั้งหมดไปใส่ไว้ในลิสต์จากรูป “A” มีไฮเปอร์ลิงก์ไปยัง “B” , “C” และ “D” เมื่อนำยูอาร์แอลทั้ง 3 ไปตรวจสอบและผ่านการตรวจสอบแล้วจึงนำยูอาร์แอลทั้งหมดไปใส่เข้าไปในลิสต์

| | | | | | | | | | |
|---|---|---|---|------|--|--|--|--|--|
| C | D | E | F | NULL | | | | | |
|---|---|---|---|------|--|--|--|--|--|

ลิสต์ 3. หลังจากเก็บข้อมูลจาก “B” แล้วปรากฏว่ามีไฮเปอร์ลิงก์ไปยัง “E” และ “F” เมื่อนำไปตรวจสอบแล้วจึงนำยูอาร์แอลทั้งหมดที่ผ่านการตรวจสอบไปใส่ไว้ในลิสต์

| | | | | | | | | | |
|---|---|---|------|--|--|--|--|--|--|
| D | E | F | NULL | | | | | | |
|---|---|---|------|--|--|--|--|--|--|

ลิสต์ 4. หลังจากเก็บข้อมูลจาก “C” แล้วปรากฏว่ามีไฮเปอร์ลิงก์ไปยัง “F” ซึ่งเมื่อตรวจสอบดูแล้วปรากฏว่า “F” อยู่ในลิสต์ของยูอาร์แอลที่จะต้องทำการเก็บข้อมูลอยู่ ดังนั้นจึงไม่ต้องเพิ่มเข้าไปในลิสต์อีก

| | | | | | | | | | |
|---|---|------|--|--|--|--|--|--|--|
| E | F | NULL | | | | | | | |
|---|---|------|--|--|--|--|--|--|--|

ลิสต์ 5. หลังจากเก็บข้อมูลจาก “D” แล้วปรากฏว่าไม่มีไฮเปอร์ลิงก์ไปที่ใดจึงไม่มีการเพิ่มยูอาร์แอลเข้าไปในลิสต์อีก

| | | | | | | | | | |
|---|------|--|--|--|--|--|--|--|--|
| F | NULL | | | | | | | | |
|---|------|--|--|--|--|--|--|--|--|

ลิสต์ 6. หลังจากเก็บข้อมูลจาก “D” แล้วปรากฏว่าลิสต์ว่าง เนื่องจากเว็บเพจทั้งหมดที่อยู่ในลิสต์ถูกเก็บข้อมูลหมดแล้ว ดังนั้นการทำงานของโรบอทจึงถูกสั่งให้หยุด ณ จุดนี้

4.2.5 การกำหนดจุดสิ้นสุดการทำงานของโรบอท

การกำหนดจุดสิ้นสุดของการทำงานของโรบอทนับว่าเป็นส่วนสำคัญส่วนหนึ่งของการทำงานเนื่องจากเครือข่ายอินเทอร์เน็ตเป็นเครือข่ายที่มีขนาดใหญ่มาก ถ้าต้องการให้โรบอททำการเก็บข้อมูลจากทุกเว็บเพจที่ปรากฏอยู่ในอินเทอร์เน็ตจะต้องใช้เวลาอย่างมาก ดังนั้นจึงมีวิธีการในการกำหนดจุดสิ้นสุดในการเก็บข้อมูล เพื่อเป็นทางเลือกในการเก็บข้อมูลและอัปเดต (Update) ข้อมูล ทำให้การเก็บข้อมูลใช้เวลารวดเร็วขึ้นและตรงตามความต้องการของผู้ใช้มากขึ้นดังนี้คือ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. หยุดการทำงานม้อลิสต์ว่าง

วิธีนี้เป็นกรเก็บข้อมูลทีสมบูรณ์และครบถ้วนทีสุด เนื่องจกโรบอทจะทำกรค้นหา และทำกรเก็บข้อมูลจกทุกเว็บทีสามารถติดต่อไปถึงได้ วิธีนี้เป็นวิธีทีมนใจได้ว่เว็บเพจใด ๆ ก็ตามทีมีเส้นทางกรลิงค์จกเว็บเพจของยูอาร์แอลเรมต้นในลิสต์ไปถึงแล้ว จะต้อง ถูกเก็บข้อมูลไว้เป็นฐนข้อมูลเพื่อใช้ในการค้นหาแน่นอน

2. หยุดการทำงานเมื่อโรบอททำงานจนกรบจนวนกรลิงค์ตามทีกำหนดไว้

วิธีนี้เป็นกรกำหนดจนวนกรลิงค์ของกรเก็บข้อมูลและให้โรบอทเก็บข้อมูลสิ้นสุดกรทำงานและทำให้อลิสต์ว่าง เมื่อจนวนกรลิงค์ของกรเก็บข้อมูลเทกกับค้ทีกำหนดไว้ โดยจนวนกรลิงค์ของกรเก็บข้อมูล ค้อ จนวนเพจทีโรบอทได้ทำกรลิงค์ไปแล้ว

4.2.6 กรกำหนดลักษณะกรทำงานของโรบอท

เนื่องจกอินเทอร์เน็ตเป็นกรอ่ยทีมีขนาดใหญ ในการเก็บข้อมูลแต่ละครั้งต้องใช้เวลาค่อนข้างมก กรทีจะให้โรบอททีทำกรเก็บข้อมูลทุกครั้งจกทุกเพจทีมีอยู่ในอินเทอร์เน็ตจึงเป็นกรกระทำทีเสียเวลามกและอจเป็นกรเสียเวลาโดยเปล้ประโยชน์ เนื่องจกข้อมูลทีถูกส่งมเป็นข้อมูลทีมีอยู่ในฐนข้อมูลอยู่แล้วและส่วนใหญ่เว็บเพจจะไม่มีการเปล้ยนเปล้งบ่อย ๆ ค้งนั้นจึงกำหนดลักษณะกรทำงานของโรบอทค้งนี้

1. ต้องมีคีย์เวิร์ดอยู่ในข้อมูลอยู่ก่อนแล้ว

แรกสุดก่อนทีจะให้โรบอทออกไปติดต่อกับเว็บไซท์ต่าง ๆ เรดต้องทำกรกำหนดคีย์เวิร์ดเบือ่งต้นชุดหนึ่งขึ้นมเพื่อให้โรบอททำกรนำมไปเทียบหาค้ทีสะกตตรงตามคีย์เวิร์ดต่าง ๆ ทีได้กำหนดไว้ ข้อมูลจกเว็บเพจซึ่งมีคีย์เวิร์ดตามทีค้นหานั้นจะถูกนำมาเก็บในฐนข้อมูล เพื่อเป็นฐนข้อมูลเบือ่งต้นของกรค้นหาจกผู้ใช้

2. ต้องทำกรเทียบหาค้ทีตรงกับคีย์เวิร์ด

เนื่องจกเป็นการค้นหาข้อมูลภษไทยทำให้อบปัญหากรตัดค้ ค้งนั้นเพื่อให้สามารถค้นหาข้อมูลภษไทยได้จึงใช้กรเทียบค้แทนกรตัดค้ เพื่อลดปัญหาควมยุ่งยก ค้งนั้นเพจทีมีค้ตรงกับคีย์เวิร์ดจึงถูกนำมาวิเคราะห์หาค้ข้อมูลทีเกี่ยวข้องต่าง ๆ (เช่น ยูอาร์แอล , ขนาดของเพจ , หมายเลขประจำเพจ เป็นต้น) มกเก็บลงในฐนข้อมูลทุกเพจแม้จะมีค้ควมสำคัญต่ำมกก็ตาม เนื่องจกรจะถือว่กรเทียบหาค้ในเพจให้ตรงกับคีย์เวิร์ดจกผู้ใช้ นั้นค่อนข้างหยาก จึงต้องเก็บทุกเพจทีมี โดยมีกรกำหนดอัตรควมเป็นไปได้ทีจะถูกต้องตรงตามควมต้องการของผู้ใช้เป็นระดับ ค้อ

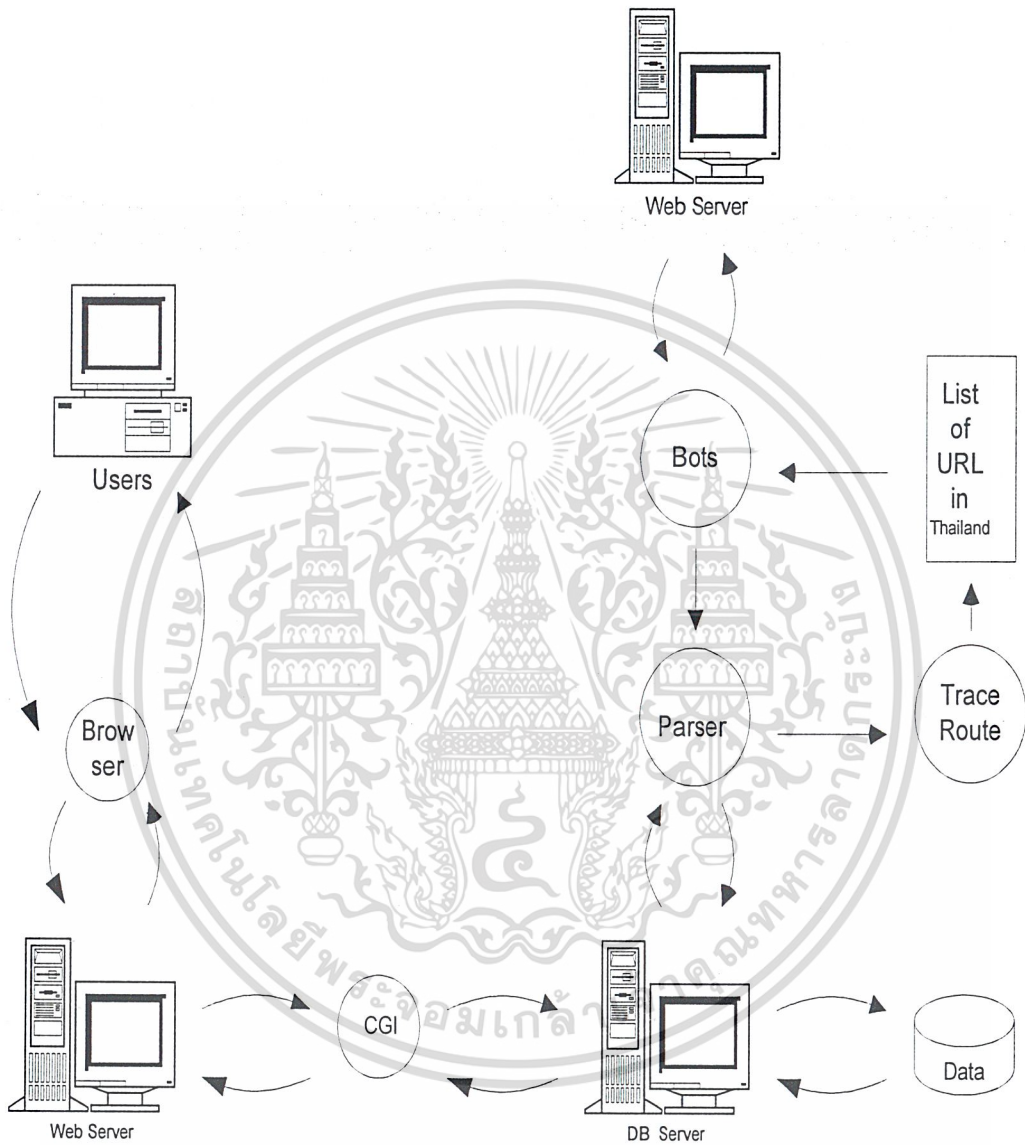
ระดับที่ 1 : ***** (ดีมาก)

ระดับที่ 2 : **** (ดี)

ระดับที่ 3 : *** (พอใช้)

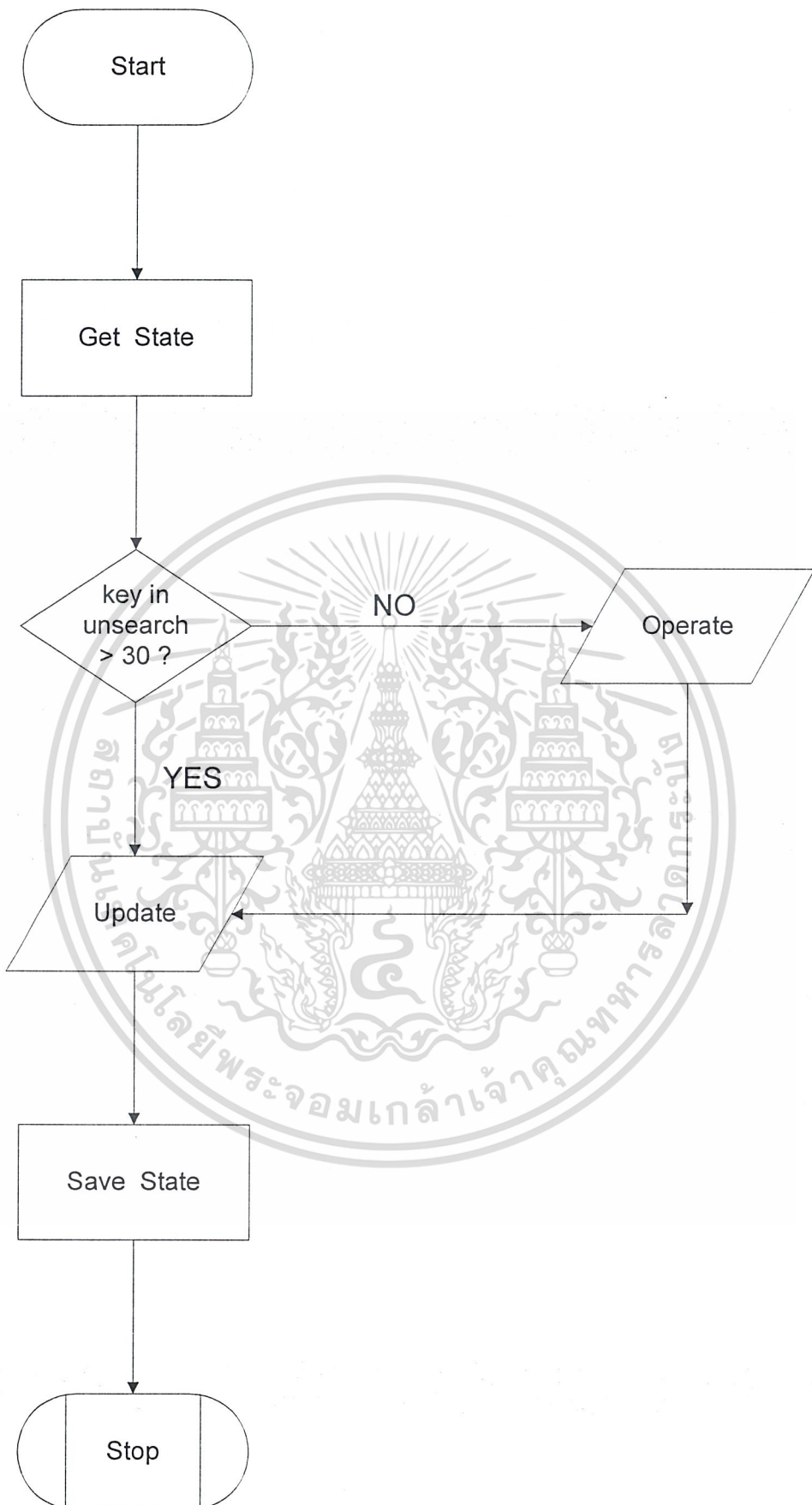
เอกสารนี้เป็นเอกสารทีสงวนไว้ส้หรับกรใช้งานเพื่อกรศึกษาเท่านั้น ไม่อนุญตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่กรณใด ๆ ทั้งสิ้น อีกรท้งห้มมิให้ตัดเปล้งเนื้อหาและต้องอ้งอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำมไปใช้

ระดับที่ 4 : ** (แย่)
 ระดับที่ 5 : * (แย่มาก)



รูปที่ 4.2 ความสัมพันธ์ระหว่างการทำงานแต่ละส่วนของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

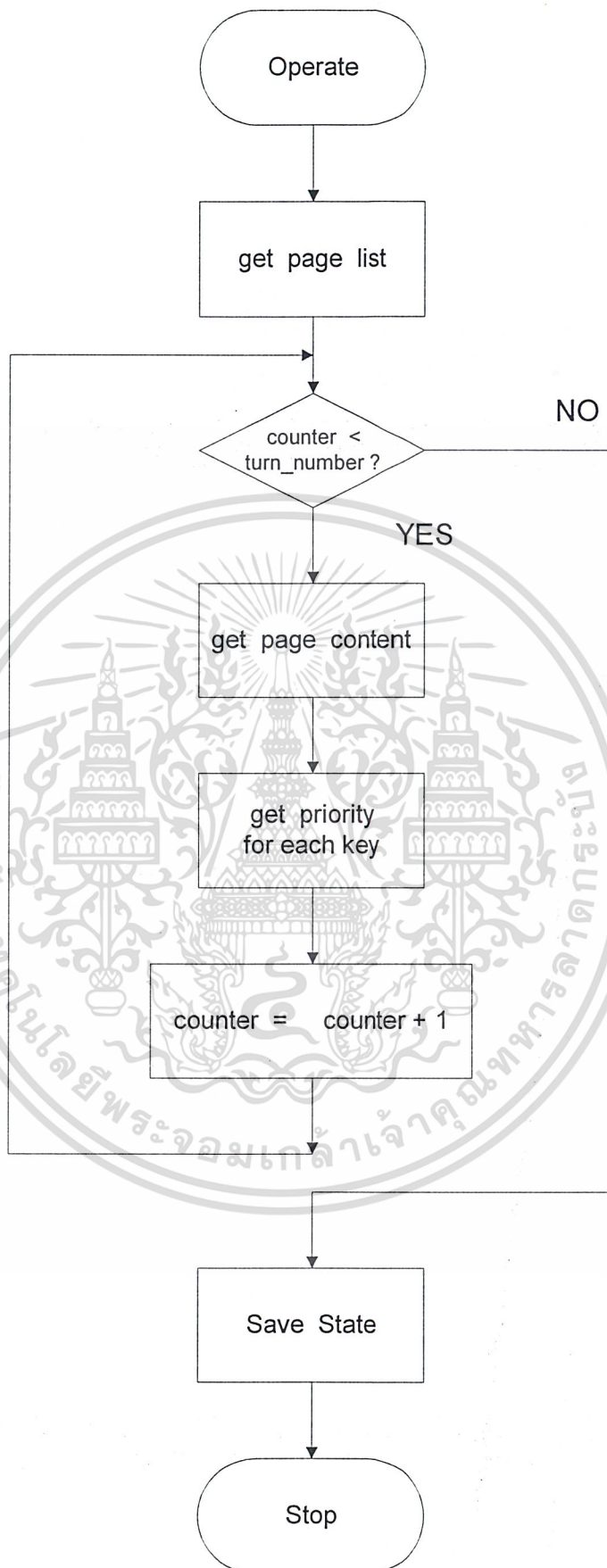


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่โดยไม่ได้รับอนุญาตของเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

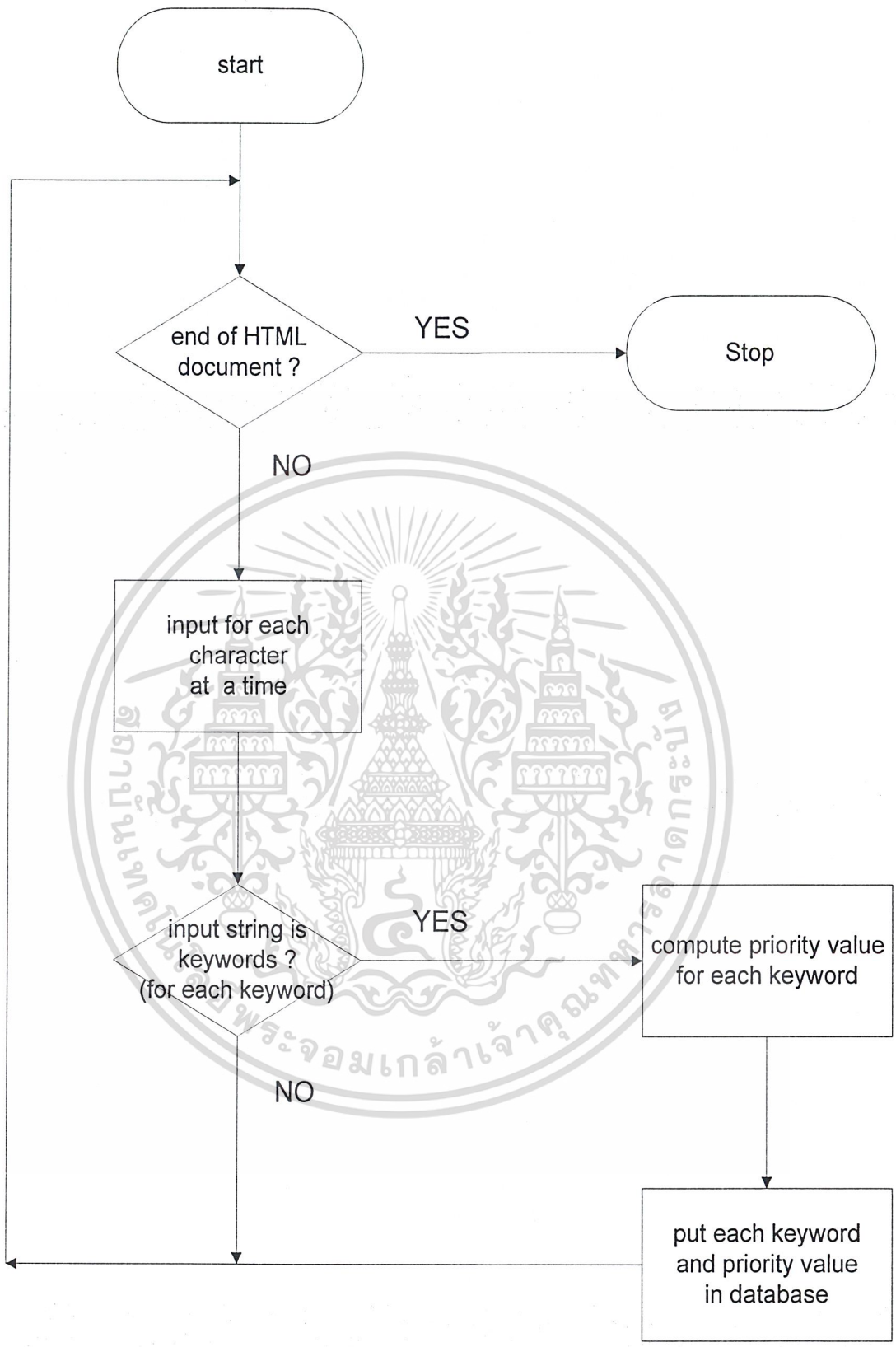
รูปที่ 4.3 แผนภาพการทำงานของโรบอท



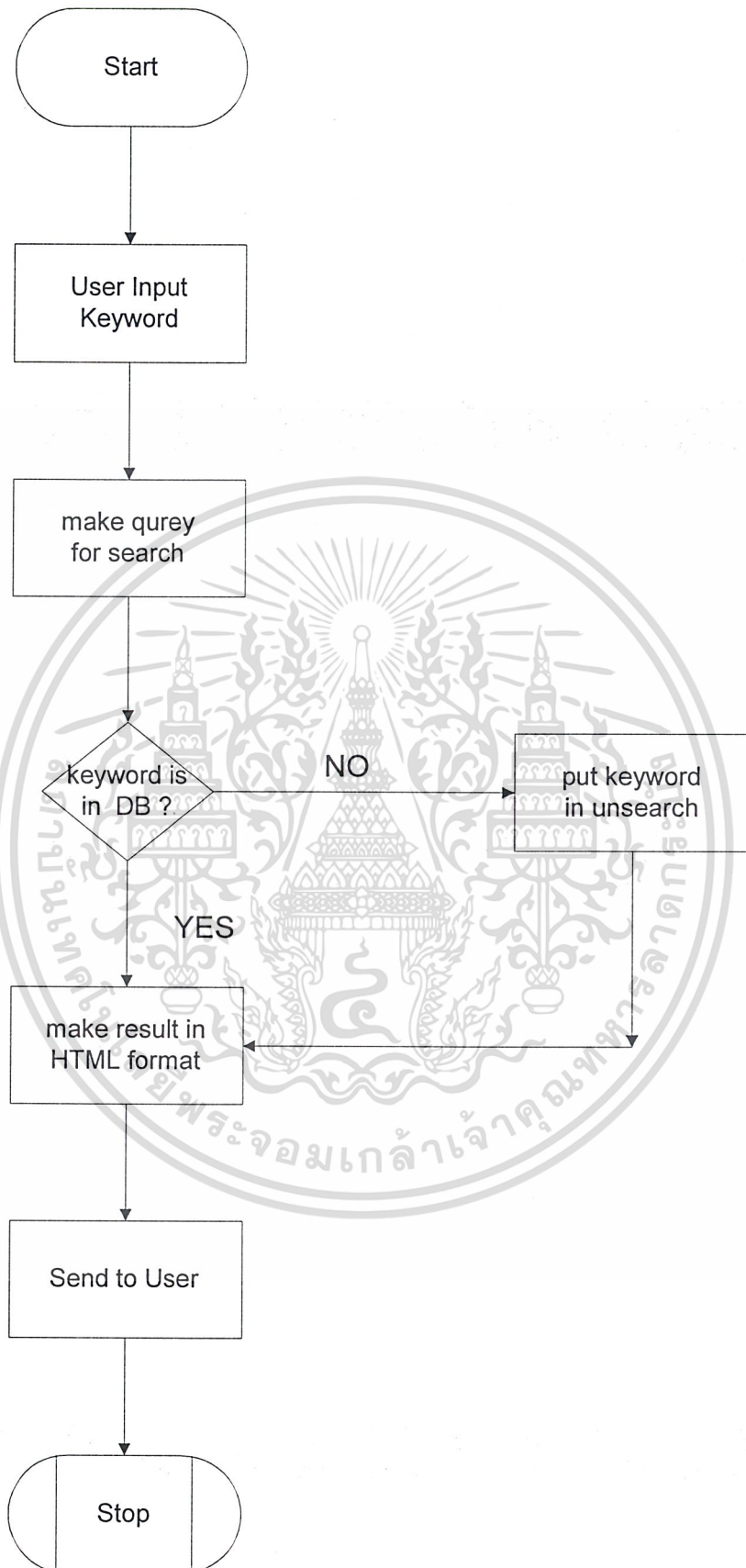
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 4.4 การ update ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง **รูปที่ 4.5 การ operate** ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.6 การวิเคราะห์เอกสาร HTML
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และเผยแพร่ไปยังผู้อื่นของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 ส่วนติดต่อกับผู้ใช้ (user interface)

เมื่อทำการออกแบบหน้าที่และการทำงานของส่วนแรกแล้ว ต่อไปคือการออกแบบส่วนซีจีไอ (ในโครงการนี้ใช้เซิร์ฟเล็ตทำ)

4.3 การออกแบบในส่วนของการติดต่อระหว่างผู้ใช้บริการกับฐานข้อมูล

โปรแกรมเซิร์ฟเล็ตจะทำหน้าที่ในการส่งค่าข้อมูลที่ได้รับจากผู้ใช้บริการให้แก่เซิร์ฟเวอร์และทำการรับค่าผลลัพธ์ออกมาส่งกลับให้ผู้ใช้บริการโดยผ่านทางบราวเซอร์ โดยข้อมูลที่ส่งผ่านโปรแกรมเซิร์ฟเล็ตในที่นี้ คือ

1. สตริงที่เป็นคำที่ผู้ใช้บริการต้องการค้นหา
2. ค่าออฟชันในการเลือกแบบในการค้นหา ซึ่งมีอยู่ 2 ค่า
 - ค่าของวิธีการเทียบหาคีย์เวิร์ดนั้นในฐานข้อมูล
 - ค่าของการเก็บคีย์เวิร์ดนั้นไว้ค้นหาในครั้งต่อไป

นอกจากโปรแกรมเซิร์ฟเล็ตจะทำหน้าที่ในการส่งข้อมูลให้แก่เซิร์ฟเวอร์แล้ว ก็ยังทำหน้าที่ในการส่งข้อมูลกลับให้ผู้ใช้บริการด้วย โดยค่าที่ส่งกลับมามีดังนี้

1. HTTP header
2. คำที่ผู้ใช้ต้องการค้นหา
3. ยูอาร์แอลที่พบคำที่ต้องการ
4. Title ของยูอาร์แอลที่พบคำที่ต้องการ
5. Description ของยูอาร์แอลที่พบคำที่ต้องการ
6. ขนาดของเว็บเพจที่มีคำที่ต้องการ

4.4 การออกแบบในส่วนของการจัดเก็บและค้นหาข้อมูล

ส่วนจัดเก็บข้อมูล

ในการออกแบบการจัดเก็บข้อมูลตามหลักการ อีอาร์โมเดล โดยข้อมูลที่ต้องทำการเก็บรวบรวมไว้ประกอบด้วย

- ข้อมูลเกี่ยวกับคำที่ใช้สำหรับการอ้างอิงคำค้น ประกอบด้วย
 1. คำที่ใช้สำหรับค้นหา
 2. หมายเลขประจำยูอาร์แอล
 3. ค่าความสำคัญ
- ข้อมูลเกี่ยวกับยูอาร์แอลที่มีคำที่เก็บไว้อยู่
 1. หมายเลขประจำยูอาร์แอล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ขนาดของเพจที่มีค่าน้อย
4. Title ของยูอาร์แอล
5. Description ของยูอาร์แอล

ส่วนค้นหาข้อมูล

มีหน้าที่ในการค้นหาข้อมูลจากฐานข้อมูล โดยจะทำการค้นหาตามข้อมูลที่ได้รับ คือ คำที่ต้องการหา เมื่อเซิร์ฟเวอร์ได้รับรีเควสบริการจากผู้ใช้ก็ทำการเรียกโปรแกรมจาวาขึ้นมาทำงานกับระบบฐานข้อมูลเพื่อใช้ในการค้นหาข้อมูล โดยในการค้นหาข้อมูลจะอัลกอริธึมดังต่อไปนี้

4.5 การออกแบบในส่วนของหน้าจอที่ใช้ติดต่อกับผู้ใช้บริการ

ส่วนของหน้าจอที่ใช้ติดต่อกับผู้ใช้บริการเป็นส่วนที่มีความสำคัญมาก เนื่องจากเป็นส่วนที่ใช้ในการรับข้อมูลจากผู้ใช้บริการและเป็นส่วนที่ใช้แสดงผลลัพธ์ด้วย นอกจากนี้ยังอาจมีส่วนช่วยดึงความสนใจของผู้ใช้บริการให้มาใช้บริการ ดังนั้นการออกแบบหน้าจอ จึงมีการออกแบบตามหลักการต่อไปนี้

1. หน้าจอต้องมีความสวยงามและเหมาะกับลักษณะงาน
2. การใช้งานผ่านหน้าจอต้องทำได้ง่าย

4.6 การเลือกแพลตฟอร์มในการทำโครงการงาน

ในโครงการนี้เลือกใช้ระบบปฏิบัติการวินโดวส์เอ็นที เป็นแพลตฟอร์มในการทำงานด้วยสาเหตุดังต่อไปนี้

1. วินโดวส์เอ็นที เป็นแพลตฟอร์มที่สามารถทำงานแบบ Multitasking ได้อย่างมีประสิทธิภาพ
2. สามารถใช้งานโปรแกรมประยุกต์ได้หลากหลายชนิด
3. ผู้จัดทำมีความคุ้นเคยต่อวินโดวส์เอ็นที พอสมควร ทำให้สะดวกในการทำงาน
4. มีความปลอดภัยสูง
5. เป็นแพลตฟอร์มที่เป็นที่นิยม ดังนั้นจึงเป็นการสะดวกต่อผู้ที่ต้องการนำโครงการนี้ไปทำการรันเพื่อการใช้งาน หรือศึกษาในภายหลัง

4.7 การเลือกเว็บเซิร์ฟเวอร์

เนื่องจากในปัจจุบันนี้มีเว็บเซิร์ฟเวอร์ให้เลือกใช้มากมายโดยแต่ละตัวก็มีความสามารถในการทำงานต่าง ๆ ที่แตกต่างกันออกไป หลังจากที่ได้ศึกษาถึงแนวทางความต้องการในการใช้งานเว็บเซิร์ฟเวอร์ทางผู้จัดทำจึงตกลงใจเลือกใช้จาวาเว็บเซิร์ฟเวอร์ (Java Web Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาเหตุที่เลือกใช้ภาษาเว็บเซิร์ฟเวอร์

1. เป็นเว็บเซิร์ฟเวอร์ที่มีขนาดเล็กทำให้ไม่เปลืองเนื้อที่ในการจัดเก็บข้อมูลปัญหาในเรื่องหน่วยความจำไม่พอ
2. สามารถทำงานได้ตามที่ผู้จัดทำต้องการ
3. สามารถดาวน์โหลดมาใช้ได้
4. สนับสนุนการทำงานของเซิร์ฟเล็ต
5. สามารถทำงานได้บนวินโดวส์เอ็นที

4.8 การเลือกใช้ภาษาในการทำโครงการงาน

ในการทำงานของโครงการนี้ทั้ง 3 ส่วน อันได้แก่ ส่วนดึงข้อมูลและวิเคราะห์ข้อมูล , ส่วนติดต่อระหว่างผู้ใช้บริการและส่วนจัดเก็บข้อมูลที่เซิร์ฟเวอร์ , ส่วนจัดเก็บข้อมูล ทั้งหมดนี้ถูกเขียนด้วยภาษาจาวา

สาเหตุที่เลือกใช้ภาษาจาวา

1. มีประสิทธิภาพและง่ายต่อการทำการศึกษา
2. สามารถทำ user interface ได้ง่ายและสวยงาม
3. สามารถนำมาใช้เขียนโรบอท , เซิร์ฟเล็ต รวมทั้งในตัวจาวาเองก็มีส่วนที่สามารถติดต่อกับฐานข้อมูลได้จึงเป็นการง่ายต่อการติดต่อการทำงานกันระหว่างแต่ละส่วนการทำงาน
4. ผู้จัดทำมีความรู้เกี่ยวกับภาษาจาวาอยู่บ้างจึงเป็นการประหยัดเวลาในการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

เมื่อทำการสร้างโรบอทเสร็จแล้วก็ทำการทดลองทำงานตามที่ได้ออกแบบไว้ ดังต่อไปนี้

5.1 การกำหนดยูอาร์แอลเริ่มต้น

การกำหนดยูอาร์แอลเริ่มต้นนี้ ทำเพื่อใช้ในการกำหนดจุดเริ่มต้นในการหาข้อมูลของโรบอท โดยในการกำหนดยูอาร์แอลเริ่มต้นนี้ จะกำหนดยูอาร์แอลที่มีหลายลิงค์เพื่อให้โรบอทสามารถวิ่งหาข้อมูลได้ทั่วถึงและครบถ้วนเพราะได้ออกแบบให้โรบอทวิ่งค้นหาทางกว้างก่อน ยูอาร์แอลเริ่มต้นที่กำหนดไว้มีดังนี้

<http://www.kmitl.ac.th/>

<http://www.nectec.or.th/>

<http://www.ku.ac.th/>

<http://www.ksc.net.th/>

<http://www.tu.ac.th/>

<http://www.au.ac.th/>

<http://www.chula.ac.th/>

<http://www.itv.co.th/>

<http://www.loxinfo.co.th/>

<http://www.inet.co.th/>

5.2 การตรวจสอบยูอาร์แอลที่อยู่ในประเทศไทย

ใช้วิธีการตรวจสอบจากโฮสต์ กล่าวคือ ถ้าโฮสต์ลงท้ายด้วย .th แสดงว่าเป็นเว็บไซต์ที่อยู่ในประเทศไทยแน่นอน แต่ก็ยังมีบางเว็บไซต์ที่ไม่ได้เป็น .th ซึ่งในการทำโครงการนี้จะไม่สนใจในจุดนี้เพราะเว็บไซต์ที่เป็น .th ก็มีมากพออยู่แล้ว

5.3 การวิเคราะห์คำและกำหนดค่าความสำคัญ

การกำหนดค่าความสำคัญของคำต่าง ๆ ที่ได้จากการวิเคราะห์ข้อมูลทำให้สามารถจัดลำดับความเป็นไปได้ที่จะได้ข้อมูลที่ตรงตามความต้องการของผู้ใช้มากที่สุด การใช้วิธีนี้จะได้ข้อมูลที่เหมาะสมหรือไม่ขึ้นอยู่กับลักษณะการเขียนโฮมเพจของแต่ละคน นั่นคือ ถ้าคำที่ต้องการหาค่าความสำคัญนั้นอยู่ในตำแหน่งที่สำคัญของโฮมเพจ เช่น อยู่ระหว่าง <META> หรืออยู่ในส่วน <TITLE> โอกาสที่เพจนั้นจะเป็นเพจที่ตรงตามความต้องการของผู้ใช้ก็มีมากกว่าเพจที่มีค่าน้อยในส่วนอื่น ๆ

ในส่วนที่จะกล่าวต่อไปคือการทดลองค้นหาข้อมูลจากฐานข้อมูลตามเงื่อนไขที่ผู้ใช้บริการกำหนด ซึ่งผู้ใช้สามารถกำหนดเงื่อนไขต่าง ๆ ได้ 2 ลักษณะ คือ

1. วิธีการค้นหา
2. การค้นหาในครั้งต่อไป

โดยในการกำหนดเงื่อนไข ผู้ใช้สามารถใช้เงื่อนไขนี้ร่วมกันเพื่อให้การค้นหาข้อมูลตรงตามความต้องการของผู้ใช้มากที่สุด ซึ่งสามารถแยกเงื่อนไขการค้นหาด้วยวิธีต่าง ๆ ดังนี้

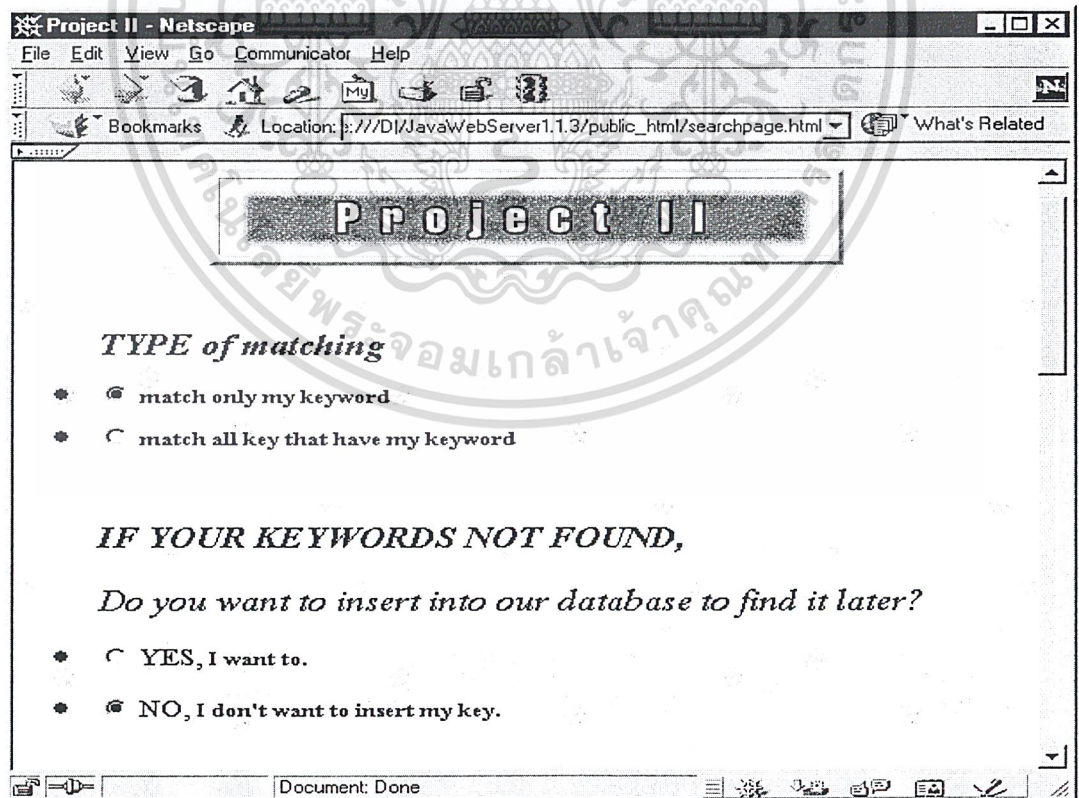
5.4 การค้นหาโดยเทียบเฉพาะค่าที่ตรงกับคีย์เวิร์ด

เป็นการใช้ลักษณะตัวอักษรเป็นเกณฑ์ในการกำหนดเงื่อนไข นั่นคือจากคีย์เวิร์ดที่ผู้ใช้ได้ป้อนลงมา เราจะค้นหาคีย์เวิร์ดในฐานข้อมูล ถ้าพบค่าที่ตรงกับคีย์เวิร์ดนั้นก็ดึงข้อมูลนั้นแสดงต่อผู้ใช้ แต่ถ้าหาแล้วไม่พบก็จะแสดงข้อความว่าหาไม่เจอต่อผู้ใช้แทน

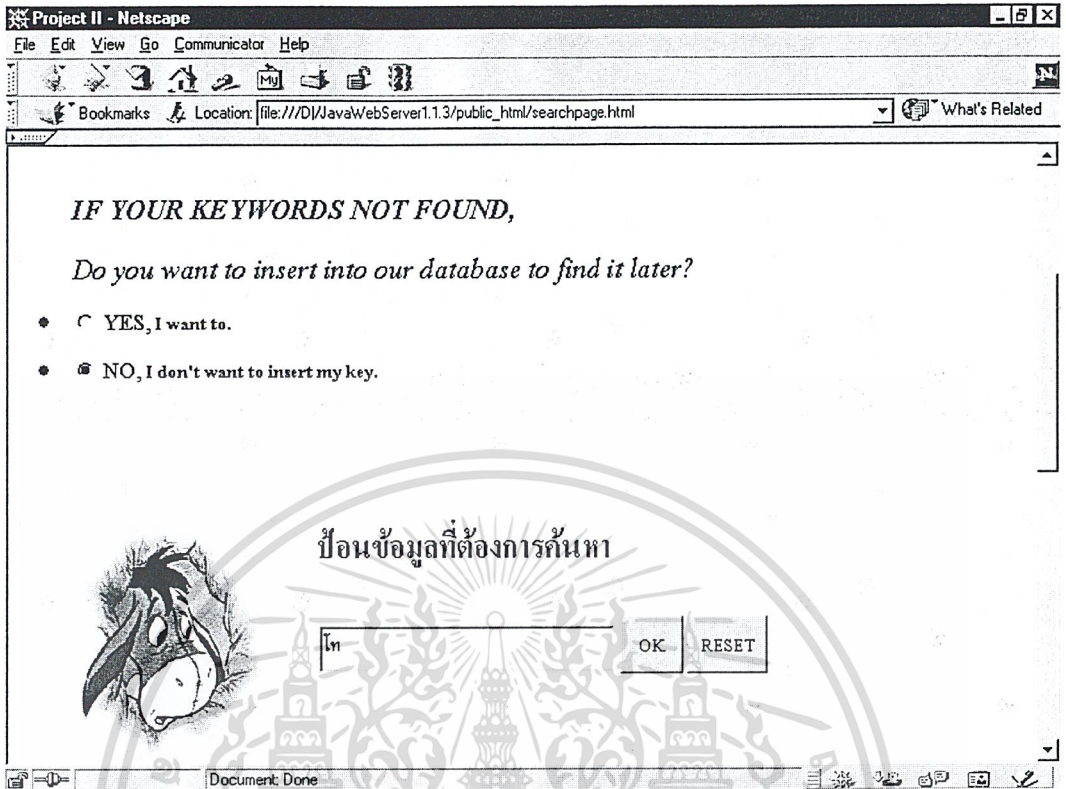
5.5 การค้นหาโดยเทียบกับค่าที่มีคีย์เวิร์ด

จะคล้ายกับวิธีการเทียบค่าที่ตรงกับคีย์เวิร์ดทุกประการ แต่จะต่างกันตรงที่ถ้าภายในค่า ๆ นั้นมีคีย์เวิร์ดที่ต้องการค้นหาอยู่ ก็จะแสดงข้อมูลของคีย์เวิร์ดนั้นด้วย เช่น ต้องการค้นหาคำว่า "โท" ถ้าในฐานข้อมูลมีข้อมูลของคำว่า ปริญญาโท , โทรศัพท์ , โทรคมนาคม ข้อมูลของคำทั้ง 3 นี้จะถูกนำไปแสดงต่อผู้ใช้ด้วย

ตัวอย่างหน้าจอในกรณีต้องการค้นหาโดยเทียบเฉพาะค่าที่ตรงกับคีย์เวิร์ด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 5.1 แสดงการค้นหาโดยเทียบเฉพาะค่าที่ตรงกับคีย์เวิร์ด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



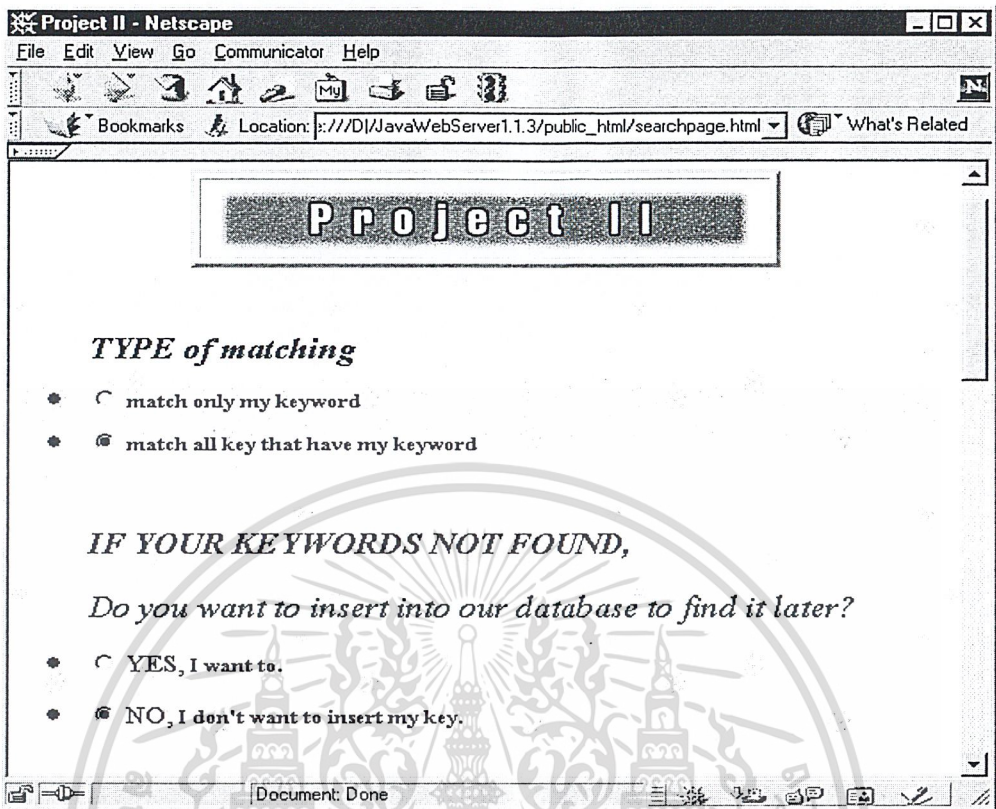
รูปที่ 5.2 แสดงส่วนที่รับข้อมูลจากผู้ใช้ในโฮมเพจ



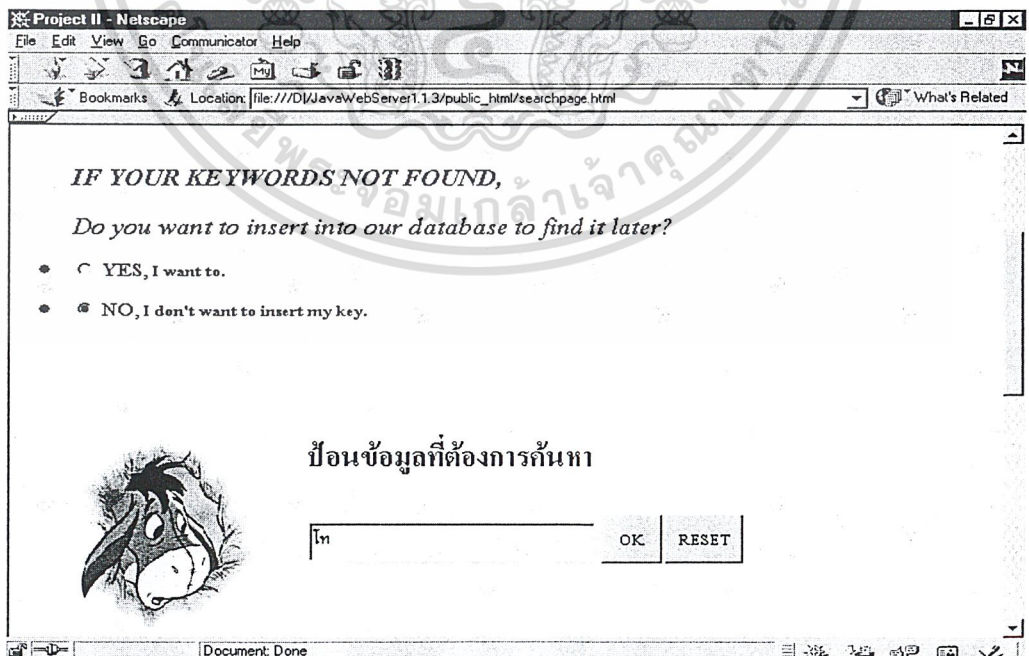
รูปที่ 5.3 แสดงผลจากการค้นหาแบบเทียบเฉพาะคำที่ตรงกับคีย์เวิร์ด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเชิงพาณิชย์เพื่อการศึกษาเท่านั้น เมื่อผู้ใช้ได้เห็นเว็บไซต์ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

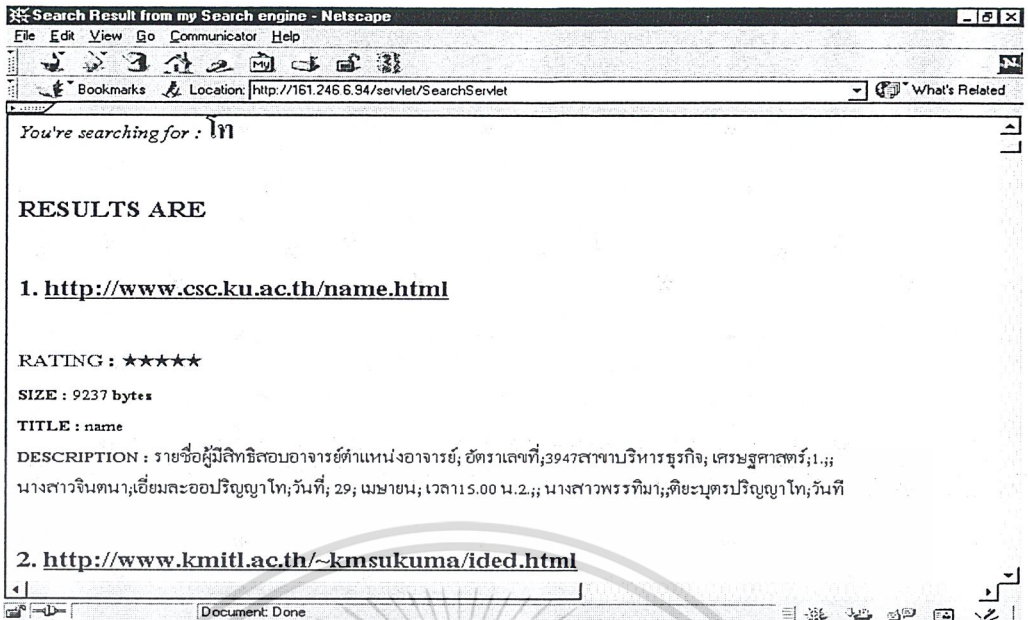
ตัวอย่างหน้าจอในกรณีต้องการค้นหาโดยเทียบกับคำที่มีคีย์เวิร์ด



รูปที่ 5.4 แสดงการค้นหาคำโดยเทียบกับคำที่มีคีย์เวิร์ด



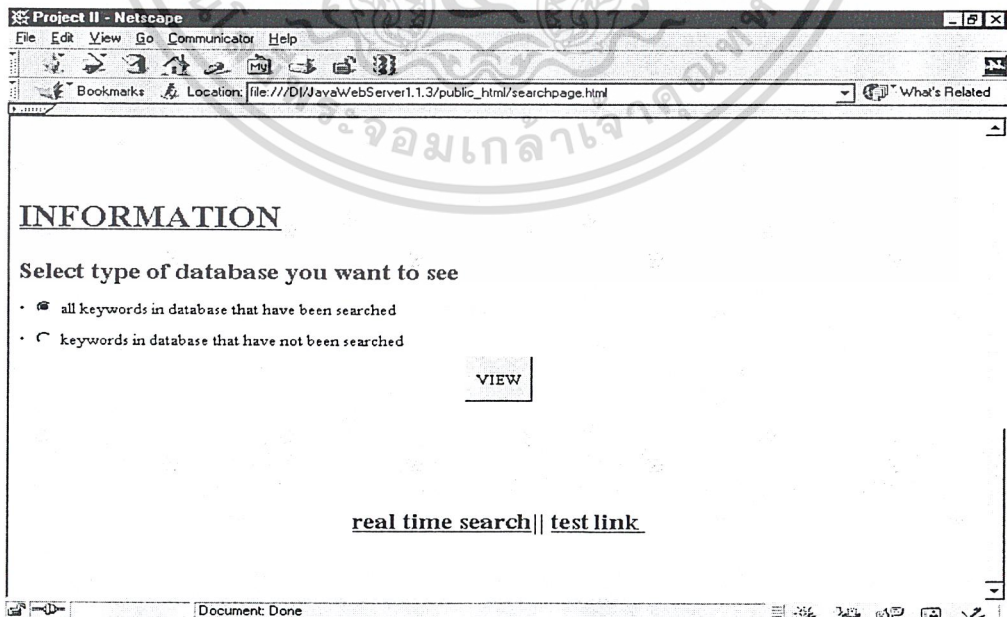
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่สามารถให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 5.5 แสดงส่วนรับข้อมูลจากผู้ใช้ในโฮมเพจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



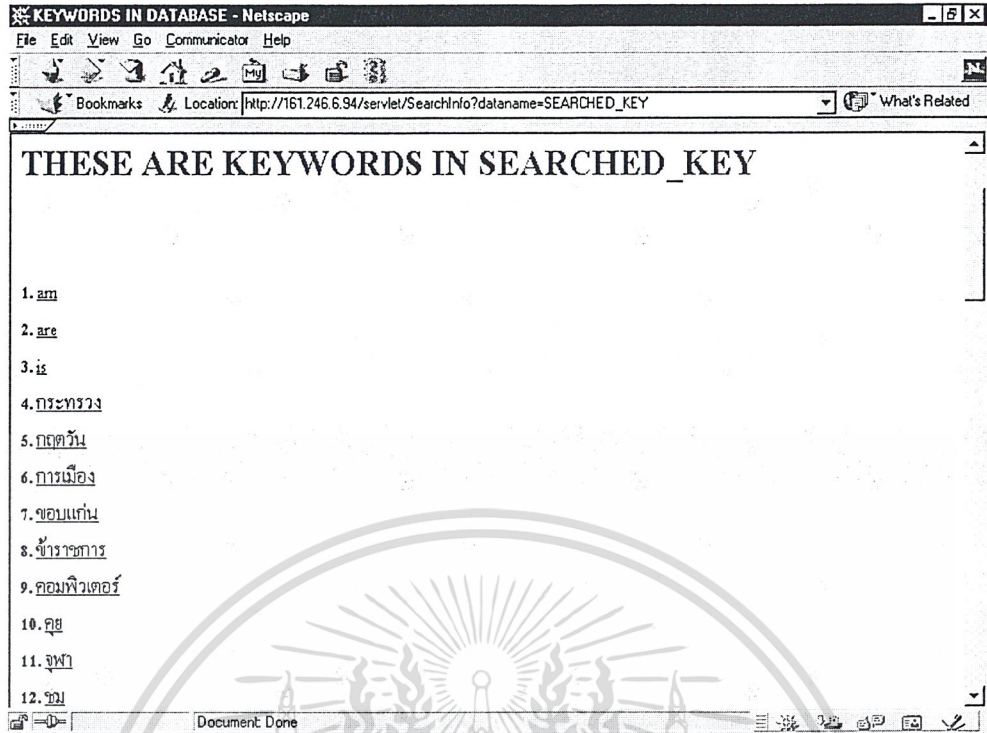
รูปที่ 5.6 แสดงผลจากการค้นหาโดยเทียบกับคำที่มีคีย์เวิร์ด

สำหรับเงื่อนไขอีกอย่างหนึ่งที่ต้องเติมก่อนที่จะทำการค้นหาข้อมูลคือ การถามถึงความต้องการของผู้ใช้บริการในกรณีที่ในฐานะข้อมูลไม่มีค่าที่ตรงกับคีย์เวิร์ดที่ผู้ใช้บริการป้อนมา ว่าต้องการให้เก็บคีย์เวิร์ดนั้นไว้ค้นหาให้ในโอกาสต่อไปหรือไม่

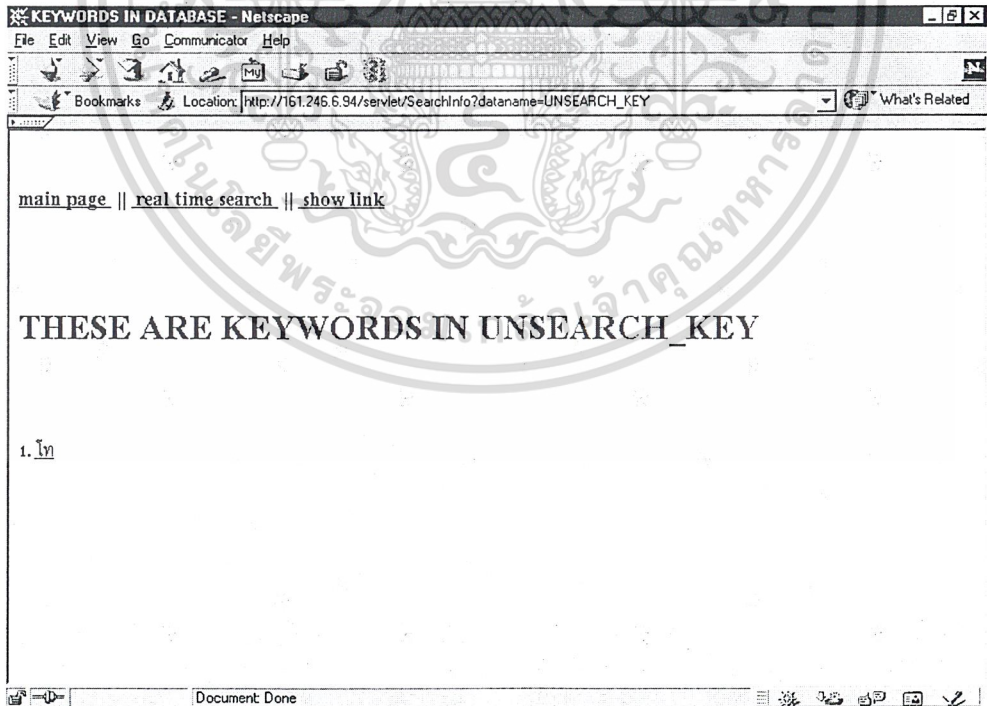
โดยเราสามารถขอลิสต์ของคีย์เวิร์ดที่ได้ทำการค้นหาแล้ว และลิสต์ของคีย์เวิร์ดที่จะทำการค้นหาในโอกาสต่อไปได้ ซึ่งการกระทำดังกล่าวจะอยู่ตอนท้ายของโฮมเพจ ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้รูปที่ 5.7 แสดงการขอลิสต์ของคีย์เวิร์ดเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 แสดงลิสต์ของกีย์เวิร์ดที่หาแล้ว



รูปที่ 5.9 แสดงลิสต์ของกีย์เวิร์ดที่ยังไม่ถูกค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทวิจารณ์และสรุป

สรุปการทำงานของโครงการเป็นดังต่อไปนี้

6.1 สรุปโครงสร้างการทำงานทั้งหมดของโครงการ WEB APPLICATION DEVELOPMENT

6.1.1 ส่วนดึงข้อมูลและวิเคราะห์ข้อมูล

เป็นส่วนการทำงานของผู้ช่วยค้นหา (search agent) ทำหน้าที่ดังนี้ คือ

- 1 ทำการท่องไปบนเว็บเพจต่าง ๆ
- 2 ตรวจสอบว่าเว็บเพจนั้นอยู่ในขอบเขตที่กำหนดหรือไม่
- 3 ทำการดึงข้อมูลกลับมา
- 4 ทำการคำนวณและวิเคราะห์ข้อมูลที่ดึงกลับมา

โดยในส่วนของการทำงานขั้นตอนนี้จะเป็นการทำงานของโปรแกรมประยุกต์ที่เรียกว่า ไรบอท กฎของไรบอท

ในการที่จะสร้างไรบอทขึ้นมาใช้งานสักตัวหนึ่ง เราควรที่จะทำตามข้อกำหนดต่าง ๆ ต่อไปนี้ ซึ่งได้รวบรวมข้อควรระวังและข้อควรปฏิบัติสำหรับไรบอท โดยทำการรวบรวมข้อมูลโดยเว็บมาสเตอร์ที่มีความชำนาญและผู้เชี่ยวชาญทางด้านไรบอท ซึ่งกฎทั้ง 4 ข้อ มีดังต่อไปนี้

- 1 เว็บไรบอทต้องประกาศตัวอย่างชัดเจน
- 2 เว็บไรบอทต้องปฏิบัติตามข้อกำหนดมาตรฐาน
- 3 เว็บไรบอทต้องไม่ยึดครองแหล่งข้อมูลแบบถาวร
- 4 เว็บไรบอทต้องแสดงข้อผิดพลาดเมื่อเกิดความผิดพลาด

วิธีการท่องเว็บไซต์ของไรบอท

การรวบรวมข้อมูลจากเว็บไซต์ต่าง ๆ ของไรบอทในโครงการนี้จะทำการค้นหาทางกว้างก่อน (Breadth-First-Search) นั่นคือไรบอทจะทำการดึงข้อมูลจากเว็บไซต์เริ่มต้นที่กำหนดเอาไว้ในลิสต์ของยูอาร์แอล แล้วค่อยขยับไปเก็บข้อมูลยังเว็บไซต์ต่อไปตามอัลกอริทึมในการค้นหาแบบทางกว้าง ลักษณะการเก็บคีย์เวิร์ดจากเว็บเพจ

จากเว็บเพจที่ไรบอทได้ดึงกลับมาเพื่อทำการวิเคราะห์ข้อมูลนั้นคีย์เวิร์ดที่ผู้ใช้ต้องการค้นหาจะอยู่ในลิสต์ซึ่งเราจะนำคีย์เวิร์ดนี้มาเทียบกับคำต่างๆ ในเพจเพื่อหาคำความสำคัญและจัดเก็บลงในฐานข้อมูลต่อไป

การคำนวณค่าความสำคัญของคำที่ปรากฏในเว็บเพจ

เมื่อได้คำที่เป็นคีย์เวิร์ดแล้วต้องนำมาคำนวณค่าความสำคัญเพื่อทำการจัดเก็บลงฐานข้อมูลเพื่อใช้เป็นคีย์เวิร์ดในการค้นหาต่อไป ซึ่งวิธีการคำนวณหาค่าความสำคัญของคำต่าง ๆ สามารถหาได้โดยใช้สูตรดังต่อไปนี้

$$\text{Priority} = S(\text{Priority_Key}_i)$$

เมื่อ Priority คือ ค่าความสำคัญของคำ n ใด ๆ

Priority_Key คือ ค่าความสำคัญของคำที่อยู่ ณ ตำแหน่ง i

จากสูตร การคำนวณหาค่าความสำคัญของคำ ทำได้โดยนำค่า Priority_Key ของตำแหน่งที่ปรากฏ คำ ๆ หนึ่งทั้งหมดในเว็บเพจมารวมกันเพื่อให้ได้ค่า Priority ของคำ ๆ นั้น

ตัวอย่าง

คำว่า " มหาวิทยาลัย " ปรากฏอยู่ในเว็บเพจ 3 ที่

ตำแหน่งที่ 1 มีค่า Priority_Key₁ = 100

ตำแหน่งที่ 2 มีค่า Priority_Key₂ = 50

ตำแหน่งที่ 3 มีค่า Priority_Key₃ = 15

เพราะฉะนั้น Priority = 100 + 50 + 15

= 165

จากค่าความสำคัญที่ได้นี้ยังมีค่ามากเท่าใดก็หมายถึงเพจนั้นมีแนวโน้มจะเป็นเพจที่ตรงตามความต้องการของผู้ใช้มากเท่านั้น

การกำหนดค่า Priority_Key ของคำในตำแหน่งต่าง ๆ

ค่า Priority_Key เป็นค่าที่ใช้กำหนดความสำคัญของคำ ณ ตำแหน่งใด ๆ ในเว็บเพจ สิ่งที่จะใช้กำหนดค่า Priority_Key นี้คือ แท็ก นั่นคือ ค่าของ Priority_Key จะแตกต่างกันไปมากน้อยขึ้นอยู่กับว่า ณ ตำแหน่งนั้นอยู่ระหว่างแท็กใด เช่น ค่า Priority_Key ณ ตำแหน่งที่อยู่ระหว่าง <TITLE> กับ </TITLE> จะมีค่า Priority_Key มากกว่าค่า Priority_Key ณ ตำแหน่งที่อยู่ระหว่าง <H1> กับ </H1> เป็นต้น และแท็กบางแท็กจะไม่มีผลต่อค่า Priority_Key เลย

การกำหนดจุดสิ้นสุดการทำงานของโรบอท

การกำหนดจุดสิ้นสุดของการทำงานของ โรบอทนับว่าเป็นส่วน สำคัญส่วนหนึ่งของการทำงาน เนื่องจากเครือข่ายอินเทอร์เน็ตเป็นเครือข่ายที่มีขนาดใหญ่มาก ถ้าต้องการให้โรบอททำการเก็บข้อมูลจากทุกเว็บเพจที่ปรากฏอยู่ในอินเทอร์เน็ตจะต้องใช้เวลาอย่างมาก ดังนั้นจึงมีวิธีการในการกำหนดจุดสิ้นสุดในการเก็บข้อมูล เพื่อเป็นทางเลือกในการเก็บข้อมูลและอัปเดต (Update) ข้อมูล ทำให้การเก็บข้อมูลใช้เวลารวดเร็วขึ้นและตรงตามความต้องการของผู้ใช้มากขึ้นดังนี้คือ

1. หยุดการทำงานเมื่อลิสต์ว่าง

วิธีนี้เป็นารเก็บข้อมูลที่สมบูรณ์และครบถ้วนที่สุด เนื่องจากโรบอทจะทำการค้นหา และทำการเก็บข้อมูลจากทุกเว็บที่สามารถติดต่อไปถึงได้ วิธีนี้เป็นวิธีที่มั่นใจได้ว่าเว็บเพจใด ๆ ก็ตามที่มีเส้นทาง การลิงค์จากเว็บเพจของยูอาร์แอลเริ่มต้นในลิสต์ไปถึงแล้ว จะต้อง ถูกเก็บข้อมูลไว้เป็นฐานข้อมูลเพื่อใช้ในการค้นหาแน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หยุดการทำงานเมื่อไรบอททำงานจนครบจำนวนการลิงค์ตามที่กำหนดไว้

วิธีนี้เป็นการกำหนดจำนวนการลิงค์ของการเก็บข้อมูลและให้ไรบอทเก็บข้อมูลสิ้นสุดการทำงานและทำให้ลิสต์ว่าง เมื่อจำนวนการลิงค์ของการเก็บข้อมูลเท่ากับค่าที่กำหนดไว้ โดยจำนวนการลิงค์ของการเก็บข้อมูล คือ จำนวนเพจที่ไรบอทได้ทำการลิงค์ไปแล้ว

การกำหนดลักษณะการทำงานของไรบอท

เนื่องจากอินเทอร์เน็ตเป็นเครือข่ายที่มีขนาดใหญ่ ในการเก็บข้อมูลแต่ละครั้งต้องใช้เวลาค่อนข้างมาก การที่จะให้ไรบอทที่ทำการเก็บข้อมูลทุกครั้งจากทุกเพจที่มีอยู่ในอินเทอร์เน็ตจึงเป็นการกระทำที่เสียเวลามากและอาจเป็นการเสียเวลาโดยเปล่าประโยชน์ เนื่องจากข้อมูลที่ถูกส่งมาเป็นข้อมูลที่มีอยู่ในฐานข้อมูลอยู่แล้วและส่วนใหญ่เว็บเพจจะไม่มีเปลี่ยนแปลงบ่อย ๆ ดังนั้นจึงกำหนดลักษณะการทำงานของไรบอทดังนี้

1. ต้องมีคีย์เวิร์ดอยู่ในข้อมูลอยู่แล้ว

แรกสุดก่อนที่จะให้ไรบอทออกไปติดต่อกับเว็บไซต์ต่าง ๆ เราต้องทำการกำหนดคีย์เวิร์ดเบื้องต้นชุดหนึ่งขึ้นมาเพื่อให้ไรบอททำการนำไปเทียบหาค่าที่สะกดตรงตามคีย์เวิร์ดต่าง ๆ ที่ได้กำหนดไว้ ข้อมูลจากเว็บเพจซึ่งมีคีย์เวิร์ดตามที่คั่นหน้านั้นจะถูกนำมาเก็บในฐานข้อมูล เพื่อเป็นฐานข้อมูลเบื้องต้นของการค้นหาจากผู้ใช้

2. ต้องทำการเทียบหาค่าที่ตรงกับคีย์เวิร์ด

เนื่องจากการค้นหาข้อมูลภาษาไทยทำให้พบปัญหาการตัดคำ ดังนั้นเพื่อให้สามารถค้นหาข้อมูลภาษาไทยได้จึงใช้การเทียบคำแทนการตัดคำ เพื่อลดปัญหาความยุ่งยาก ดังนั้นเพจที่มีคำตรงกับคีย์เวิร์ดจึงถูกนำมาวิเคราะห์หาข้อมูลที่เกี่ยวข้องต่าง ๆ (เช่น ยูอาร์แอล , ขนาดของเพจ , หมายเลขประจำเพจ เป็นต้น) มาเก็บลงในฐานข้อมูลทุกเพจแม้จะมีค่าความสำคัญต่ำมากก็ตาม เนื่องจากเราจะถือว่าการเทียบหาค่าในเพจให้ตรงกับคีย์เวิร์ดจากผู้ใช้ นั้นค่อนข้างหายาก จึงต้องเก็บทุกเพจที่มี โดยมีข้อกำหนดอัตราความเป็นไปได้ที่จะถูกต้องตรงตามความต้องการของผู้ใช้เป็นระดับ คือ

- ระดับที่ 1 : ***** (ดีมาก)
- ระดับที่ 2 : **** (ดี)
- ระดับที่ 3 : *** (พอใช้)
- ระดับที่ 4 : ** (แย)
- ระดับที่ 5 : * (แย่มาก)

6.1.2 ส่วนเชื่อมต่อระหว่างผู้ใช้งานกับเซิร์ฟเวอร์

เป็นส่วนที่เชื่อมต่อการทำงานของโฮมเพจของตัวค้นหาข้อมูลกับผู้ใช้บริการ โดยจะทำหน้าที่ดังนี้

- 1 ทำการรับตัวแปรและเงื่อนไขการทำงานของผู้ใช้บริการจากโฮมเพจ
- 2 ส่งค่าตัวแปรและเงื่อนไขไปยังเซิร์ฟเวอร์ที่ให้บริการ
- 3 รับค่าผลลัพธ์จากเซิร์ฟเวอร์ที่ให้บริการ แล้วส่งออกมาทางหน้าจอให้ผู้ใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดยบริษัทเอกชนที่ผลิตขึ้นโดยผู้ให้บริการรายนี้ ขอสงวนสิทธิ์ในด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของส่วนเชื่อมต่อนี้จะอาศัยการทำงานของโปรแกรมที่เรียกว่า เซิร์ฟเล็ต ซึ่งมีหลักการและลักษณะการทำงานดังนี้

เซิร์ฟเล็ต คือโปรแกรมที่จะถูกเรียกให้ปฏิบัติงานเฉพาะกิจกรรมหนึ่ง ๆ โดยร่วมกับเว็บเซิร์ฟเวอร์เมื่อมีการร้องขอมาจากเว็บไคลเอนท์ (เช่น การค้นหาข้อมูลจากระบบฐานข้อมูล) เซิร์ฟเล็ต มีแพลตฟอร์ม (Platform) เป็นภาษาจาวาอย่างสมบูรณ์ที่ใช้ทางฝั่งเซิร์ฟเวอร์เพื่อให้สามารถใช้งานร่วมกับจาวาเว็บเซิร์ฟเวอร์ (Java Web Server) ได้อย่างลงตัว โดยเซิร์ฟเล็ตจะเพิ่มความสามารถให้กับเซิร์ฟเวอร์ ในด้านการลดโอเวอร์เฮด (minimal overhead), ในด้านการดูแลรักษา (maintenance), ในด้านการสนับสนุน (support) เนื่องจากเซิร์ฟเล็ตเป็นจาวาไบต์โค้ด (Java bytecode) ทำให้สามารถดาวน์โหลด (download) ผ่านเครือข่ายได้ เป็นไปตามคำกล่าวที่ว่า “ เขียนครั้งเดียว รันได้ทุกที่ ” (Write Once , Run Anywhere)

6.1.3 ส่วนจัดเก็บข้อมูล

ในโครงการนี้มีการจัดเก็บข้อมูลไว้ในระบบฐานข้อมูลเนื่องจากข้อดีคือ

- ลดความซ้ำซ้อนของข้อมูล

การที่แผนกต่าง ๆ ต้องการข้อมูลชนิดเดียวกัน แต่ต่างก็แยกกันเก็บทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูลของหน่วยความจำ
- หลีกเลี่ยงการขัดแย้งกันเองของข้อมูล

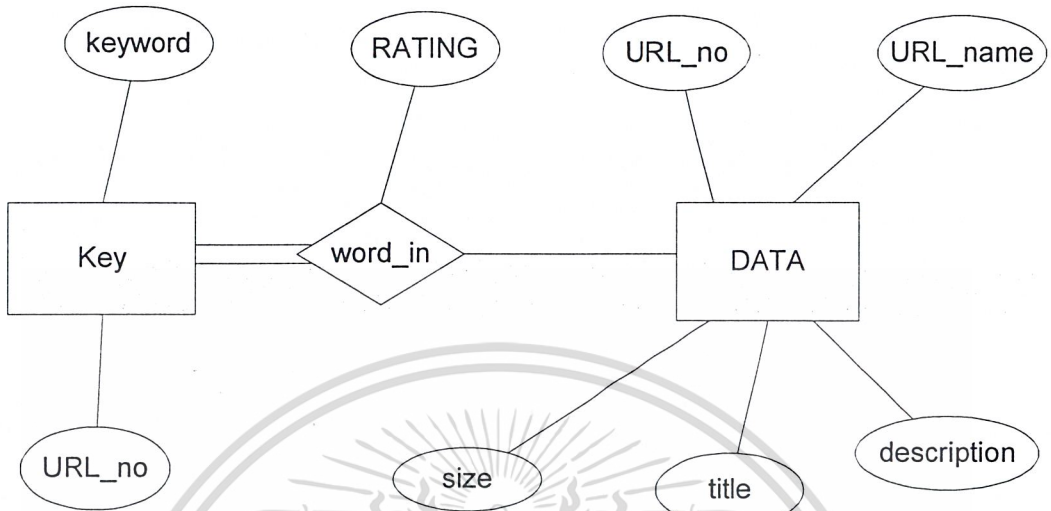
เมื่อมีการเปลี่ยนแปลงข้อมูลที่จัดเก็บในหลายๆ ที่หลายๆ ไฟล์จะต้องทำการเปลี่ยนแปลงทุกไฟล์ที่จัดเก็บถ้าทำ ไม่ครบทุกไฟล์ก็จะทำให้ข้อมูลมีความผิดพลาดไปจากความจริงและขัดแย้งกัน
- แก้ปัญหาการรักษาความปลอดภัยของข้อมูล

เนื่องจากการเก็บข้อมูลกระจัดกระจาย ทำให้ยากต่อการรักษาความปลอดภัยต่อการรั่วไหลและคัดแปลงข้อมูล
- ทำให้ข้อมูลมีคุณภาพมากที่สุด หรือ การควบคุมความถูกต้องของข้อมูล
- ทำให้สามารถควบคุมการใช้งานโดยส่วนกลางได้
- สามารถใช้ข้อมูลร่วมกันได้
- มีความคล่องตัวและความยืดหยุ่นในการใช้งาน
- มีการควบคุมมาตรฐานร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

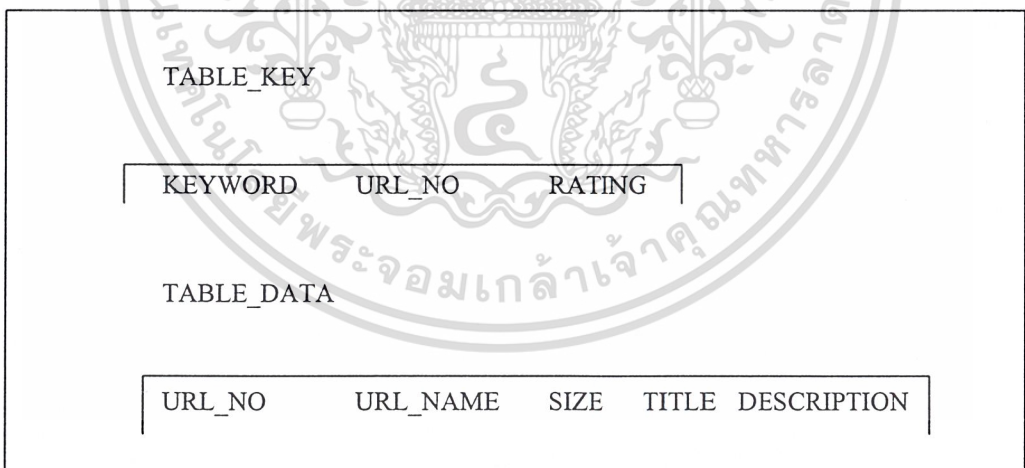
การออกแบบฐานข้อมูลตามหลักการของอีอาร์โมเดล

ในการออกแบบระบบฐานข้อมูลที่ใช้โครงงานนี้ เราจะทำการออกแบบตามหลักการอีอาร์โมเดล



รูปที่ 6.1 แสดงอีอาร์โมเดล

เมื่อทำการออกแบบระบบฐานข้อมูลเรียบร้อยแล้วก็ทำการ map อีอาร์โมเดลที่ได้ให้เป็นตารางที่จะนำมาใช้งานจริงต่อไป



รูปที่ 6.2 แสดงตารางที่ได้จากการ map อีอาร์โมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.4 ออปชันการทำงานของตัวค้นหาข้อมูล

ในการทำงานของตัวค้นหาข้อมูล เราทำการกำหนดส่วนของการค้นหาข้อมูลตามเงื่อนไขให้ผู้ใช้สามารถเลือกได้ดังต่อไปนี้

1. การค้นหาข้อมูลโดยเทียบเฉพาะคำที่ตรงกับคีย์เวิร์ด
2. การค้นหาข้อมูลโดยเทียบคำกับที่มีคีย์เวิร์ด

6.2 ประเมินผลโครงการงาน

สามารถประเมินผลการทำงานได้จาก

1. ไรบอทที่สร้างขึ้นสามารถตรวจสอบเว็บไซต์ที่ผ่านไปถึงได้ว่าอยู่ในขอบเขตที่กำหนดไว้หรือไม่
2. ไรบอทสามารถดึงข้อมูลจากเว็บไซต์ที่ต้องการได้
3. ไรบอทสามารถวิเคราะห์ข้อมูลที่ดึงมาได้
4. ไรบอทสามารถจัดเก็บคีย์เวิร์ดและค่าความสำคัญลงในฐานข้อมูลได้
5. ฐานข้อมูลที่สร้างไว้สามารถเก็บข้อมูลได้ครบถ้วน
6. สามารถทำการค้นหาข้อมูลในฐานข้อมูลได้ตามเงื่อนไขที่กำหนดจากผู้ใช้บริการ
7. สามารถแสดงผลพร้อมกลับมายังหน้าจอของผู้ใช้บริการได้และเป็นไปตามที่กำหนด

6.3 วิเคราะห์ประสิทธิภาพของโครงการงาน

1. การตรวจสอบเว็บไซต์ของไรบอทยังมีปัญหาในการตรวจ เนื่องจากยังไม่ครอบคลุมข้อมูลทั้งหมด เช่น กรณีที่เว็บไซต์ของไทยแต่ไปตั้งอยู่ในต่างประเทศ
2. ไรบอทสามารถดึงข้อมูลจากเว็บไซต์ที่ต้องการกลับมาได้ ยกเว้นในกรณีที่ระบบเครือข่ายมีความขัดข้องเกิดขึ้น
3. ไรบอทสามารถทำการเทียบคำแล้วหาค่าความสำคัญของคีย์เวิร์ดในเว็บเพจนั้นได้
4. ไรบอทสามารถนำคีย์เวิร์ดที่ได้ค้นหาและพบแล้วนำไปเก็บลงฐานข้อมูลได้รวมถึงข้อมูลต่างๆ ที่เกี่ยวข้องกับคีย์เวิร์ดนั้น
5. ฐานข้อมูลที่ได้ออกแบบไว้สามารถจัดเก็บข้อมูลได้ตามที่ต้องการ
6. การค้นหาข้อมูลในฐานข้อมูลสามารถทำได้ถูกต้อง
7. การแสดงผลพร้อมของข้อมูลที่ออกมาทางหน้าจอของผู้บริการยังไม่สมบูรณ์

6.4 การพัฒนาโครงการงานในอนาคต

โครงการนี้สามารถที่จะพัฒนาขีดความสามารถได้อีกเรื่องต่อไปนี้

1. พัฒนาหลักการในการเทียบคำและคำนวณค่าความสำคัญให้เหมาะสมยิ่งขึ้น

2. พัฒนาในส่วนการ update ข้อมูลให้มีประสิทธิภาพยิ่งขึ้น

3. พัฒนารูปแบบการจัดเก็บข้อมูลในฐานข้อมูลเพื่อให้ได้ประโยชน์สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่แบบสงวนเนื้อหา และต้องอ้างอิงถึงเจ้าของสิทธิ์ทุกครั้งที่มีการนำไปใช้

6.5 สรุปโครงการ

โครงการนี้สามารถทำงานอันเป็นประโยชน์ในด้านการช่วยค้นหาข้อมูลเกี่ยวกับประเทศไทย ด้วยภาษาไทยบนเครือข่ายอินเทอร์เน็ตได้ตามขอบเขตและจุดประสงค์ที่ตั้งไว้ แต่ก็พบปัญหาที่เกิดขึ้นในการทำงานในด้านประสิทธิภาพของการทำงาน เช่น ความเร็ว , การเทียบคำที่ผู้ใช้บริการจะต้องป้อนคีย์เวิร์ดให้สะกดตรงตามข้อมูลที่จัดเก็บอยู่ในฐานข้อมูล เป็นต้น อันควรได้รับการพัฒนาต่อไปในอนาคตตามแนวทางที่ได้ศึกษาไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

JDBC

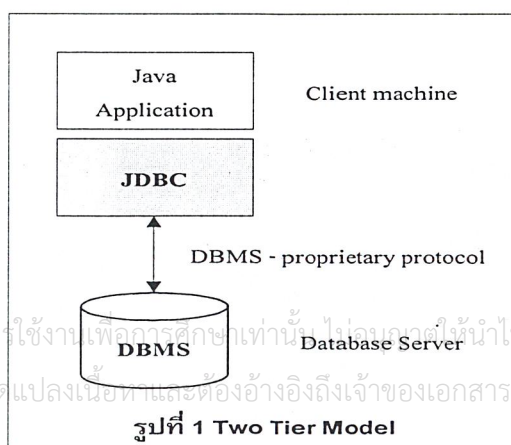
JDBC คืออะไร

JDBC (Java Database Connectivity) คือฟังก์ชันมาตรฐาน (API : Application Programming Interface) สำหรับการเชื่อมต่อกับระบบฐานข้อมูล JDBC ประกอบด้วยคลาส (Class) และส่วนเชื่อมโยง (Interface) ต่าง ๆ ซึ่งถูกเขียนขึ้นมาด้วยภาษาจาวา (ยกเว้น JDBC – ODBC driver และ Native - API – Partly – JAVA Drivers ซึ่งมีบางส่วนเขียนด้วยภาษา C++) ทำให้เราสามารถเขียนโปรแกรมซึ่งเป็นภาษาจาวาล้วน ๆ โดยใช้ฟังก์ชัน API ของ JDBC เพื่อทำการติดต่อกับระบบฐานข้อมูล

JDBC สร้างระดับการเชื่อมต่อเพื่อสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับ ODBC (Open Database Connection ของบริษัทไมโครซอฟท์) ซึ่งในปัจจุบันได้ถือว่เป็นมาตรฐานการเชื่อมต่อกับฐานข้อมูลสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) และระบบเครือข่ายท้องถิ่น (LAN) ไปเสียแล้ว อย่างไรก็ตามหลักการทำงานของทั้ง JDBC และ ODBC ตั้งอยู่บนมาตรฐานเดียวกันคือ X/Open SQL Call Level Interface ของระบบ X – Windows ซึ่งนี่ก็คือเหตุผลว่าทำไม JDBC ถึงมีการพัฒนาอย่างรวดเร็วในเวลาเพียงไม่กี่เดือน

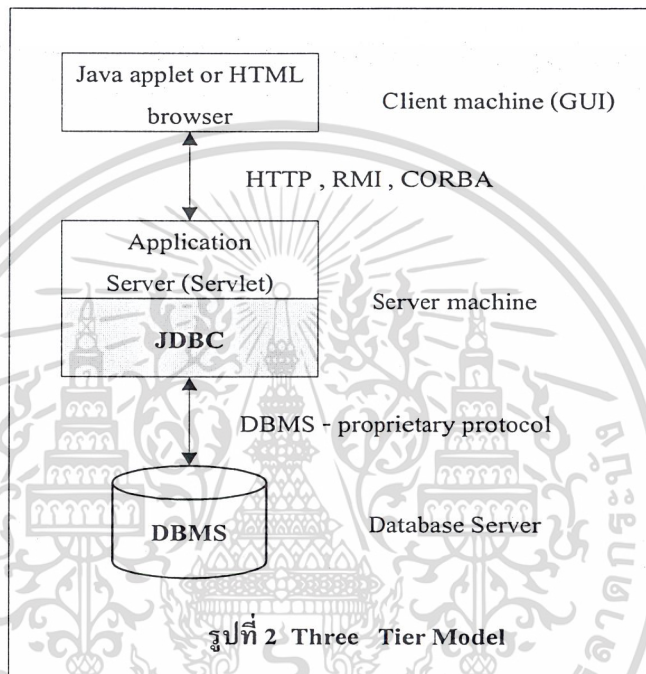
JDBC API สนับสนุนรูปแบบการเชื่อมต่อฐานข้อมูลทั้งแบบ Two – Tier model และ Three – Tier model

Two – Tier Model : จาวาแอปเพล็ต (Applet) หรือจาวาแอปพลิเคชัน (Application) จะติดต่อกับฐานข้อมูลโดยตรง จึงมีความจำเป็นที่โปรแกรมจาวาต้องการ JDBC ไดรฟ์เวอร์พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ รูปแบบการเชื่อมต่อแสดงดังรูปที่ 1 ถ้าตั้งในการเรียกค้นข้อมูลในรูปของภาษา SQL (Structured Query Language) จะถูกส่งจากผู้ไปสู่ฐานข้อมูล หลังจากนั้นผลจากการประมวลผลของระบบการจัดการฐานข้อมูล ก็จะถูกส่งกลับมาสู่ผู้ใช้งานข้อมูลดังกล่าวนี้ ส่วนมากจะติดตั้งอยู่ต่างเครื่องคอมพิวเตอร์ โดยผู้ใช้งานสามารถเชื่อมต่อผ่านระบบเครือข่าย (Network) รูปแบบ Two – Tier นี้ใช้หลักการเดียวกันกับรูปแบบไคลเอนท์ / เซิร์ฟเวอร์ (Client / Server) ที่เรารู้จักกันดี โดยที่เครื่องคอมพิวเตอร์ของผู้ใช้คือไคลเอนท์และเครื่องคอมพิวเตอร์ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครื่องข่ายคอมพิวเตอร์ที่นิยมใช้รูปแบบ Two – Tier มักเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

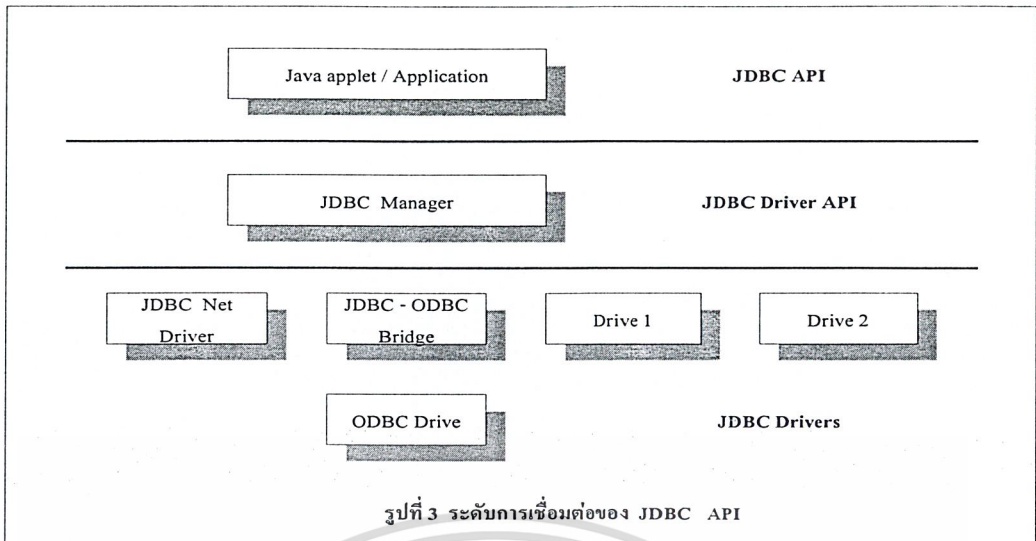
Three Tier Model : คำสั่งเรียกค้นต่าง ๆ จากผู้ใช้จะถูกส่งไปให้กับ Middle Tier หรือส่วนกลางของการบริการเสียก่อน หลังจากนั้น Middle Tier จะแปลงคำสั่งเหล่านี้ให้เป็นภาษา SQL เพื่อส่งไปให้กับระบบจัดการฐานข้อมูลเพื่อทำการประมวลผล ข้อมูลผลลัพธ์ที่ได้ก็จะถูกส่งกลับคืนไปให้กับ Middle Tier และส่งต่อไปให้ผู้ใช้ในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเทอร์เน็ตซึ่งประกอบด้วยเครื่องคอมพิวเตอร์หลากหลายชนิด และเว็บเซิร์ฟเวอร์ (Web – Server) ซึ่งทำหน้าที่เป็น Middle Tier ก็จะเป็นตัวกลางในการจัดการให้คอมพิวเตอร์ทั้งไคลเอนท์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนแปลงข้อมูลตัวใหม่จะไม่มีผลกระทบซึ่งกันและกันเกิดขึ้น



โครงสร้างของ JDBC

จากรูปที่ 3 โครงสร้างการเชื่อมต่อภายใน JDBC ประกอบด้วย 3 ระดับหลักก็คือ JDBC API , JDBC Driver และ JDBC Driver ระดับบนสุด JDBC API เป็นระดับของฟังก์ชัน API ที่อำนวยความสะดวกให้แก่โปรแกรมประยุกต์ระดับกลาง JDBC Driver API ทำหน้าที่จัดการส่งถ่ายคำสั่งจากระดับบน JDBC API ไปให้กับไดร์ฟเวอร์ในระดับล่าง JDBC Driver (มีไดร์ฟเวอร์ที่ต่างกันอยู่ 4 ชนิด) ที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแบบของ JDBC ไดรฟ์เวอร์

JDBC ไดรฟ์เวอร์หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่าง ๆ ของ JDBC ถูกจัดแบ่งออกเป็นทั้งหมด 4 ชนิดคือ

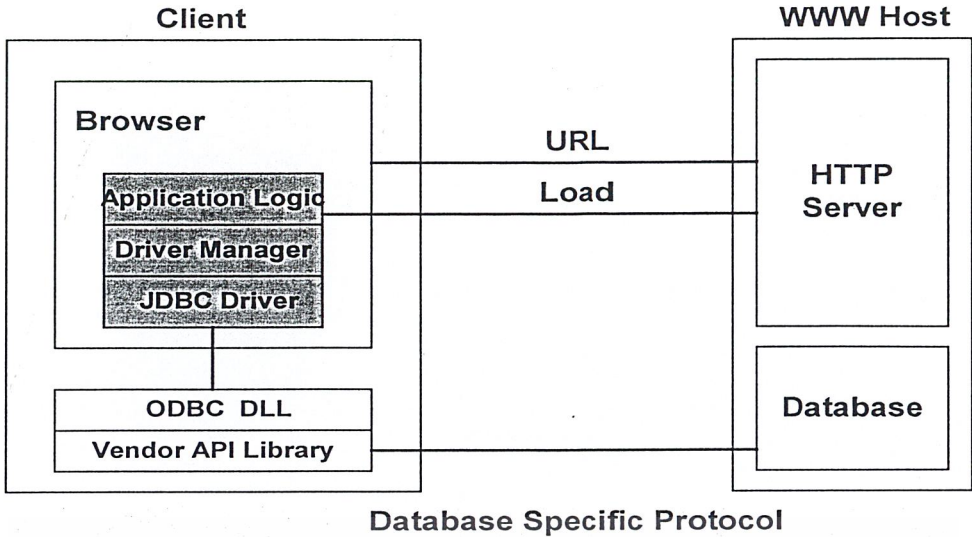
1. JDBC - ODBC Bridge
2. Native - API (partly Java) Drivers
3. Net - Protocol (All Java) Drivers
4. Native - Protocol (All Java) Drivers

1 JDBC - ODBC Bridge : ทำหน้าที่เป็นตัวกลางในการเข้าถึงข้อมูลโดยแปลงฟังก์ชันของ JDBC ให้อยู่ในรูปแบบฟังก์ชันของ ODBC ทำให้ JDBC สามารถติดต่อกับฐานข้อมูลได้โดยผ่านการทำงานของ ODBC รูปแบบการเชื่อมต่อกับฐานข้อมูลของ JDBC - ODBC Bridge แสดงดังในรูปที่ 4

ข้อดี ของไดร์ฟเวอร์ชนิดนี้คือ ไม่ต้องติดตั้งและปรับเปลี่ยนไดร์ฟเวอร์ใหม่ทั้งหมดเพราะ ODBC ได้ถูกติดตั้งอยู่บนเครื่องคอมพิวเตอร์ส่วนมากที่ใช้งานในปัจจุบันอยู่แล้ว (Window 95 / NT , Macintosh และ UNIX บางระบบ) ทำเพียงแค่นำ JDBC - ODBC Bridge มาต่อเชื่อมเพิ่มประสิทธิภาพให้กับ ODBC เท่านั้น

ข้อเสีย คือ JDBC - ODBC Bridge ไดรฟ์เวอร์ถูกเขียนขึ้นจากภาษา C/C++ ในบางส่วนจึงทำให้ไม่สามารถดาวน์โหลด (Download) ข้ามเครือข่ายอินเทอร์เน็ตและแปล (Interpret) บนเว็บเบราว์เซอร์ (WebBrowser) ได้เพราะภาษา C/C++ ต้องรัน (run) แต่ละระบบเท่านั้น ดังนั้น ODBC ไดรฟ์เวอร์ในรูปแบบของไลบรารีไฟล์หรือ DLL (Dynamic Link Library) และ JDBC - ODBC Bridge ไดรฟ์เวอร์ต้องได้รับการติดตั้งอยู่ที่ Middle Tier หรือเครื่องคอมพิวเตอร์ตัวกลาง เช่น Web - Server เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

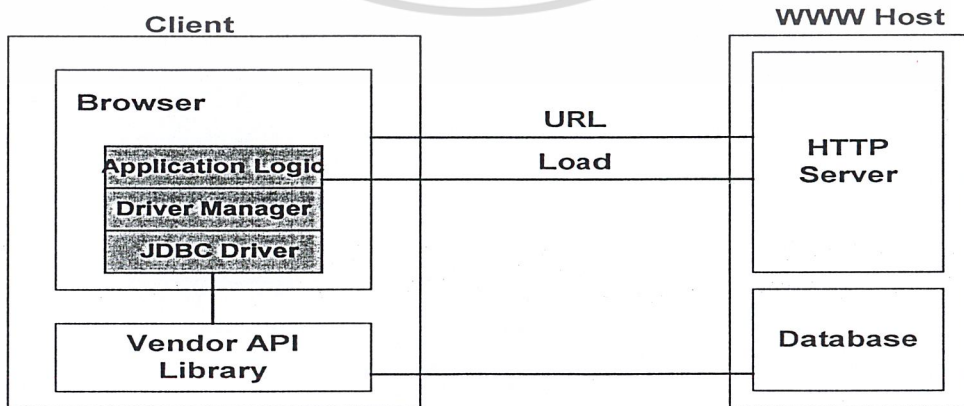


รูปที่ 4 การติดต่อ JDBC โดยรูปแบบที่ 1 JDBC-ODBC Bridge

2 Native-API (Partly-Java) drivers : ทำงานโดยการแปลงฟังก์ชันของ JDBC ให้เป็นฟังก์ชันของ JDBC ให้เป็นฟังก์ชันของ API เฉพาะของแต่ละฐานข้อมูลเป้าหมาย Native-API Drivers มีหลักการทำงานคล้ายกับ ODBC ที่ต้องมีไดรฟ์เวอร์ (DLL) ติดตั้งอยู่บนไคลเอนท์แต่ละเครื่อง ยกตัวอย่างเช่น การเชื่อมต่อกับฐานข้อมูลอราเคิล (Oracle) (ดังรูปที่ 5) ก็ต้องใช้ JDBC ไดรฟ์เวอร์และ DLL เฉพาะสำหรับติดต่อผ่านโปรโตคอล SQL-NET ของอราเคิล เท่านั้น ฉะนั้น Native-API Drivers จึงเหมาะกับการทำงานแบบ Two-Tier คือไคลเอนท์ติดต่อกับฐานข้อมูลบนเซิร์ฟเวอร์โดยตรง

ข้อดี ของไดรฟ์เวอร์ชนิดนี้คือ สามารถใช้ Native-API Driver ติดต่อกับฐานข้อมูลนั้น ๆ โดยตรงโดยผ่านโปรโตคอลเดิมที่ใช้อยู่ก่อนแล้วทำให้เหมาะสมกับการเชื่อมต่อกับฐานข้อมูลแบบ Two-Tier Client / Server

ข้อเสีย คล้ายคลึงกับ JDBC-ODBC Bridge คือต้องติดตั้งไดรฟ์เวอร์ (DLL) และโปรโตคอลไดรฟ์เวอร์ (เช่น SQL-NET , INET ,Open Client , ฯลฯ) บนเครื่องคอมพิวเตอร์นั้น ๆ ที่สำคัญ ส่วนหนึ่งของ Native-API Driver ถูกเขียนขึ้นจากภาษา C++ (Partly-Java) จึงไม่สามารถดาวน์โหลดผ่านเครือข่ายอินเทอร์เน็ตได้และไม่สามารถเชื่อมต่อนฐานข้อมูลข้ามชนิดกันได้



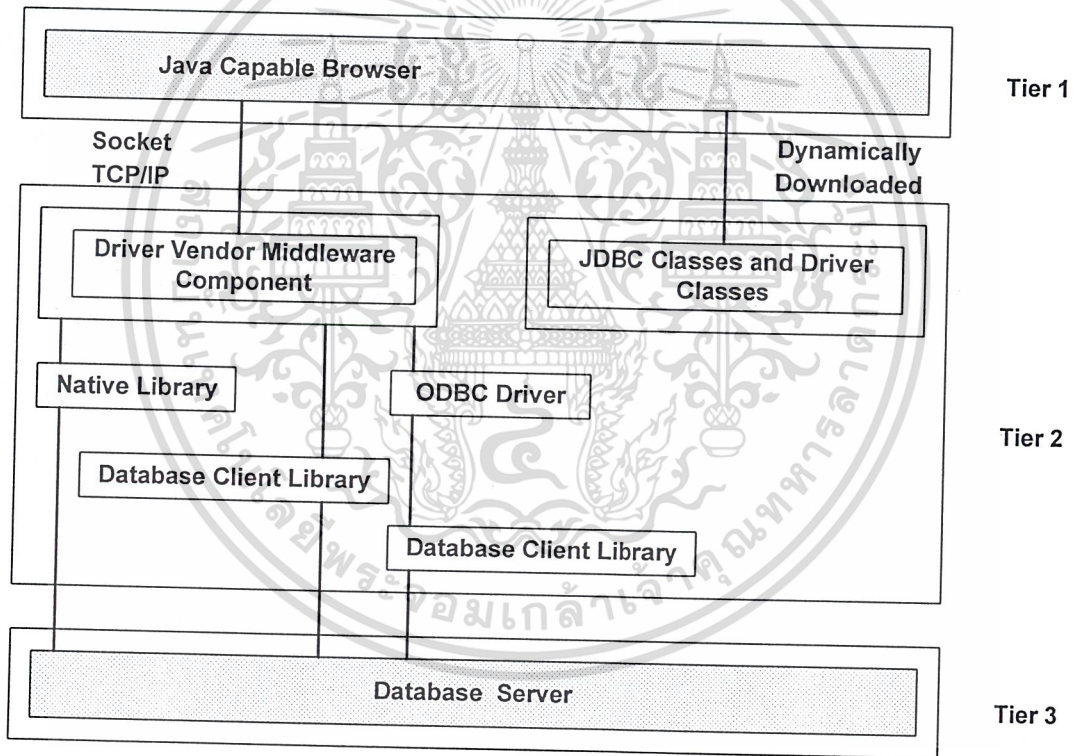
รูปที่ 5 การติดต่อ JDBC โดยรูปแบบที่ 2 Native-API ,partly Java driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุที่แบบสงวนเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 Net-Protocol (All-Java) drivers ทำหน้าที่แปลงฟังก์ชันของ JDBC ในอยู่ในรูปแบบของเครือข่ายโปรโตคอลร่วม (DBMS-independent network Protocol) ซึ่งหลังจากนั้นจะถูกแปลงให้อยู่ในรูปแบบเฉพาะของแต่ละฐานข้อมูลบนเซิร์ฟเวอร์นั้น ๆ รูปแบบการเชื่อมต่อแสดงดังรูปที่ 6 สำหรับ JDBC ไดรฟ์เวอร์ชนิดนี้เป็นไดร์ฟเวอร์ที่ใช้งานได้กว้างขวางเพราะเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งทำให้สามารถเชื่อมต่อกับฐานข้อมูลหลาย ๆ ตัวที่ต่างกัน

ข้อดี คือเป็นไดร์ฟเวอร์ที่ยืดหยุ่นคล่องตัวที่สุด เพราะว่าเขียนขึ้นจากภาษาจาวาทั้งหมด ทำให้ไม่จำเป็นต้องมีไดร์ฟเวอร์ร่วมที่เขียนขึ้นจากภาษาอื่นซึ่งต้องติดตั้งเฉพาะบนไคลเอนท์นั้น และสามารถรันบนระบบใดก็ได้ที่สนับสนุนสภาพแวดล้อมเสมือนของจาวาหรือ JVM (Java Virtual Machine) การปฏิบัติงานทุกรูปแบบทั้งดาวน์โหลดข้ามเครือข่ายหรือรันบนเครื่องเดียว ๆ สามารถกระทำได้หมด

ข้อเสีย คือในเรื่องความปลอดภัยเพราะทางเลือกในการใช้งานต่าง ๆ จะขึ้นอยู่กับผู้ผลิตไดร์ฟเวอร์ว่าจะอนุญาตให้ใช้งานส่วนใดได้บ้าง



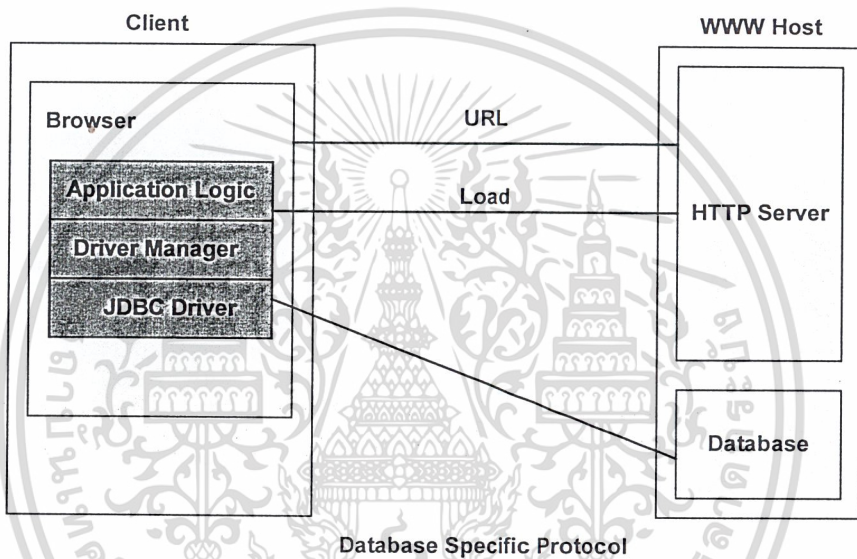
รูปที่ 6 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol , all-Java driver ยกตัวอย่างของ dbAnywhere , SequelLink

4 Native-Protocol (All-Java) Drivers : ทำหน้าที่แปลงคำสั่งของ JDBC ให้อยู่ในรูปของเครือข่ายโปรโตคอลเฉพาะของฐานข้อมูลนั้นโดยตรง การร้องขอข้อมูลจากไคลเอนท์จึงกระทำโดยตรงต่อเซิร์ฟเวอร์ดังแสดงในรูปที่ 7 ไดรฟ์เวอร์ชนิดนี้ถูกเขียนขึ้นจากภาษาจาวาใหม่ทั้งหมดจากผู้ผลิตระบบไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลนั้น ๆ และไดร์ฟเวอร์ใหม่นี้จะทำงานแทนที่เครือข่ายโปรโตคอลเดิม (Oracle SQL – NET , Informix I – Connection , Sybases OpenClient , ฯลฯ)

ข้อดี ของไดร์ฟเวอร์ชนิดนี้คือไม่ต้องมีการปรับเปลี่ยนระบบฐานข้อมูลเดิมที่ใช้งานอยู่แล้วในแต่ละองค์กร และไม่มีควมจำเป็นที่ต้องติดตั้งไดร์ฟเวอร์ตัวกลาง (เช่น ODBC) บนไคลเอนท์แต่ละเครื่องจึงเหมาะสำหรับการพัฒนาการเชื่อมต่อฐานข้อมูลแบบ Two – Tier สำหรับเครือข่ายอินเทอร์เน็ตภายในองค์กร

ข้อเสีย คือไม่เหมาะที่จะนำไปใช้บนเครือข่ายอินเทอร์เน็ตเพราะปัญหาเรื่องความปลอดภัย การเชื่อมต่อข้อมูลไม่สามารถกระทำข้ามระบบที่แตกต่างกันได้ ต้องใช้ไดร์ฟเวอร์เฉพาะสำหรับฐานข้อมูลตัวนั้น ๆ



รูปที่ 7 การติดต่อ JDBC โดยรูปแบบที่ 4 Native-protocol , all-Java driver

JDBC ไดรฟ์เวอร์ชนิดที่ 3 และ 4 คือไดร์ฟเวอร์ที่คาดว่าจะเป็นสิ่งที่ต้องการสำหรับการเชื่อมต่อกับฐานข้อมูลในอนาคตเพราะถูกเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งมีความปลอดภัยและความคล่องตัวมากกว่า (เช่นมีการติดตั้งไดร์ฟเวอร์โดยอัตโนมัติเมื่อมีการดาวน์โหลดจาวาแอปเพลต) สำหรับไดร์ฟเวอร์ชนิดที่ 1 และ 2 คือไดร์ฟเวอร์ “ ขัดตาทัพ ” ซึ่งถูกพัฒนาขึ้นมาเพื่อใช้งานร่วมกับไดร์ฟเวอร์ดั้งเดิม (ODBC หรือเน็ตเวิร์คโปรโตคอลเฉพาะฐานข้อมูล) ที่มีใช้อยู่แล้วในปัจจุบันไปพลาง ๆ ก่อนซึ่งแน่นอนว่าไดร์ฟเวอร์ทั้ง 2 ชนิดนี้มีข้อจำกัดในการใช้งานมากกว่า 2 ชนิดแรกโดยเฉพาะเรื่องความปลอดภัยและความไม่เป็นอิสระจากระบบคอมพิวเตอร์

ขั้นตอนพื้นฐานสำหรับการเชื่อมต่อกับฐานข้อมูล

2.4.1 เปิดการเชื่อมต่อกับฐานข้อมูล

2.4.2 ส่งคำสั่ง SQL ไปให้แกฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับความใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 จัดการกับผลลัพธ์ที่ได้

2.4.4 ปิดการเชื่อมต่อ

และเพื่อให้เข้าใจการทำงานของ JDBC ยิ่งขึ้น ตัวอย่างโปรแกรมบางส่วนต่อไปนี้จะแสดงให้เห็นถึงขั้นตอนทั้ง 4 ในการติดต่อฐานข้อมูลผ่าน ODBC ไดรฟ์เวอร์โดยใช้ JDBC-ODBC Bridge (ฟังก์ชัน API ต่าง ๆ ของ JDBC ที่มาพร้อมกับ JDK เวอร์ชัน 1.1 ขึ้นไปถูกเก็บรวบรวมไว้ในแพ็คเกจ java.sql.*)

```

// (1) เปิดการเชื่อมต่อกับฐานข้อมูล
Class.forName( " sun.jdbc.odbc.JdbcOdbcDriver " );
Connection con = DriverManager.getConnection( "jdbc:odbc:myDB" , login , password );
// (2) ส่งคำสั่ง SQL ไปให้แกฐานข้อมูล
Statement stmt = con.createStatement() ;
ResultSet rs = stmt.executeQuery( " SELECT a , b , c FROM Table1 " ) ;
// (3) จัดการกับผลลัพธ์ที่ได้
while( rs.next() ) {
    int I = rs.getInt( "a" ) ;
    String s = rs.getString( "b" ) ;
    float f = rs.getFloat( "c" ) ;
}
// (4) ปิดการเชื่อมต่อ
stmt.close() ;
con.close() ;

```

จากตัวอย่างสิ่งแรกที่ทำคือเลือกชนิดของ JDBC ไดรฟ์เวอร์ (JDBC-ODBC Bridge) ระบุชื่อฐานข้อมูล (DSN : Data Source Name) รวมถึงชื่อผู้ใช้ (Login) และรหัสผ่าน (Password) จากนั้นเป็นการสร้างอ็อบเจกต์ stmt (คลาส Statement) สำหรับการเรียกค้นข้อมูลและจัดส่งคำสั่ง SQL ไปในรูปแบบสตริง (String) ผ่านเม็ททอด executeQuery() ผลลัพธ์ที่ได้จากเม็ททอดนี้คือผลลัพธ์จากกประมวลผลของฐานข้อมูลซึ่งจะเก็บอยู่ในอ็อบเจกต์ rs (คลาส ResultSet) จากนั้นข้อมูลที่ได้รับนี้จะถูกถ่ายทอด (Mapping) ให้อยู่ในชนิดข้อมูลที่จำวารู้จัก (เช่น int , String , float , ฯลฯ) เพื่อนำไปใช้ประโยชน์อื่นต่อไป สุดท้ายคือการปิดการเชื่อมต่อกับฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

จาวาเซิร์ฟเล็ต (JAVA SERVLET)

จาวาเซิร์ฟเล็ต (JAVA SERVLET) คือ โปรแกรมภาษาจาวา ซึ่งสามารถนำไปใช้เพื่อขยายฟังก์ชันของเว็บเซิร์ฟเวอร์ในทางต่าง ๆ ได้ โดยโปรแกรมจะมีขนาดเล็ก ไม่ขึ้นกับแพลตฟอร์ม และสามารถถูกโหลดแบบไดนามิกได้ เปรียบเทียบได้เป็นจาวาแอปเพล็ต (java applet) ที่ทำงานในฝั่งเซิร์ฟเวอร์นั่นเอง

ข้อแตกต่างของจาวาเซิร์ฟเล็ตกับจาวาแอปเพล็ต คือ

1. เซิร์ฟเล็ตทำงานทางฝั่งเซิร์ฟเวอร์ ส่วนแอปเพล็ตทำงานทางฝั่งไคลเอนท์
2. เซิร์ฟเล็ตไม่ต้องการยูสเซอร์อินเตอร์เฟซแบบกราฟฟิค (Graphical User Interface, GUI) เพราะเซิร์ฟเล็ตทำงานในตัวเซิร์ฟเวอร์เอง
3. เซิร์ฟเล็ตทำงานโดยใช้หลักของรีควีส-เรสปอนด์ (request-response paradigm) ซึ่งเป็นโมเดลที่โปรโตคอลเอชทีทีพี (HTTP) ใช้อยู่
4. เซิร์ฟเล็ตไม่มีเมทอด (method) stop ซึ่งสามารถแสดงได้ดังตาราง

| เหตุการณ์ | เมทอดของแอปเพล็ต | เมทอดของเซิร์ฟเล็ต |
|---------------|------------------|---|
| สร้างออบเจกต์ | Init | Init(ServletConfig sc) |
| เริ่มทำงาน | Start | Service(ServletRequest sr, ServletResponse res) |
| หยุดทำงาน | Stop | - |
| ทำลายออบเจกต์ | Destroy | Destroy |

ตัวอย่างการใช้งานเซิร์ฟเล็ต

1. ประมวลผลข้อมูลที่ถูกส่งมาในฟอร์มของเอกสารเอชทีเอ็มแอล (HTML form) เช่น ใบบัตรซื้อสินค้า หรือ ข้อมูลสินค้า ได้
2. สามารถนำไปใช้ซิงโครไนซ์ (synchronous) คำร้องขอเพื่อสนับสนุนระบบต่าง ๆ เช่น การประชุมออนไลน์ (on-line conferencing) ได้ เพราะเซิร์ฟเล็ตสามารถจัดการคำร้องขอหลายอย่างในเวลาเดียวกันได้

สถาปัตยกรรมของเซิร์ฟเล็ต

ศูนย์กลางที่อยู่ใน JSDK (Java Servlet Development Kit) คือ อินเตอร์เฟซ Servlet ทุก ๆ เซิร์ฟเล็ตจะต้องอิมพลิเมนต์ (implement) อินเตอร์เฟซนี้ ไม่ว่าจะโดยตรง หรือขยายจากคลาส (class) อื่นที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิมพลีเมนต์อินเตอร์เฟซนี้ เช่น คลาส HttpServlet เป็นต้น อินเตอร์เฟซ Servlet จะมีเมธอดที่ไว้จัดการเซิร์ฟเล็ต และการติดต่อระหว่างเซิร์ฟเล็ตกับไคลเอนท์

เมื่อเซิร์ฟเล็ตรับการเรียกจากไคลเอนท์ มันจะได้รับสองอาร์กิวเมนต์ คือ ServletRequest และ ServletResponse ซึ่งจะมีหน้าที่ดังนี้

- ServletRequest จะประกอบไปด้วยข้อมูลการติดต่อจากไคลเอนท์ไปให้เซิร์ฟเวอร์ เช่น พารามิเตอร์ที่ไคลเอนท์ส่งให้เซิร์ฟเวอร์ โพรโตคอลที่ใช้ เป็นต้น
- ServletResponse จะประกอบไปด้วยข้อมูลการติดต่อจากเซิร์ฟเวอร์ไปให้ไคลเอนท์ รวมทั้งยังมีเมธอดที่ใช้ในการตอบไปสู่อไคลเอนท์ หรือค่าง่าชนิดของข้อมูล (MIME type) เป็นต้น

วงจรชีวิตของเซิร์ฟเล็ต

เซอร์วิส (service) จะเป็นตัวโหลดและรันเซิร์ฟเล็ต โดยอาจจะรับคำร้องขอจากไคลเอนท์มาประมวลผล และส่งข้อมูลคืนสู่ไคลเอนท์ รวมถึงยังเป็นผู้ทำลายเซิร์ฟเล็ตนั้นด้วย โดยมีวงจรชีวิตดังนี้

1. เริ่มแรกเซอร์วิสโหลดเซิร์ฟเล็ต มันจะรันเมธอด init ของเซิร์ฟเล็ตนั้น ซึ่งในเมธอดนี้ จะไม่มีการทำพร้อมกัน (concurrency) แม้ว่าเซิร์ฟเล็ตจะรันอยู่ในมัลติเทรดเซอร์วิส (multi-thread service) ก็ตาม โดยเมธอดนี้จะถูกเรียกเพียงครั้งเดียว และจะรันใหม่เมื่อเซอร์วิสโหลดเซิร์ฟเล็ตนั้นขึ้นมาใหม่ เซอร์วิสจะไม่สามารถโหลดใหม่ได้จนกว่ามันจะกำจัดเซิร์ฟเล็ตนั้น โดยการเรียกใช้เมธอด destroy ของเซิร์ฟเล็ตนั้นเสียก่อนเท่านั้น
2. เมื่อเซอร์วิสโหลดและเริ่มต้นเซิร์ฟเล็ตแล้ว เซิร์ฟเล็ตนั้นจะพร้อมที่จะรับคำร้องขอจากไคลเอนท์ โดยมันจะทำอยู่ในเมธอด service ของมัน แต่ละคำร้องขอของไคลเอนท์จะถูกรันในแต่ละเทรดของเซิร์ฟเล็ตนั้น เพื่อรับคำร้องขอและส่งผลลัพธ์ไปสู่อไคลเอนท์
3. เซิร์ฟเล็ตจะถูกรันจนกระทั่งมันถูกทำลายออกจากเซอร์วิส ซึ่งเซอร์วิสจะรันเมธอด destroy ของเซิร์ฟเล็ตนั้น

การพัฒนาเซิร์ฟเล็ตเพื่อรับคำร้องจากไคลเอนท์

แม้ว่าเซิร์ฟเล็ตจะต้องอิมพลีเมนต์อินเตอร์เฟซ javax.servlet.Servlet ก็ตาม แต่ในการพัฒนาเซิร์ฟเล็ตที่ใช้งานบนเว็บ ซึ่งใช้โพรโตคอลเอชทีทีพี จะนิยมเขียนโดยการขยายคลาส

javax.servlet.http.HttpServlet มากกว่า และจะเขียนทับ (override) เมธอดที่ใช้จัดการกับวิธีการร้องขอของเอชทีทีพี (HTTP request) ซึ่งได้แก่

- doGet สำหรับจัดการการร้องขอแบบ GET
- doPost สำหรับจัดการการร้องขอแบบ POST
- doPut สำหรับจัดการการร้องขอแบบ PUT
- doDelete สำหรับจัดการการร้องขอแบบ DELETE

ไม่ว่าเราจะเขียนทับเมธอดใดก็ตาม มันจะต้องรับสองอาร์กิวเมนต์ คือ HttpServletRequest และ HttpServletResponse โดยออบเจ็กต์ HttpServletRequest จะเสนอการเข้าถึงข้อมูลเฮดเดอร์ของเอช

ที่ที่พี (HTTP header data) เช่น คุกกี้ (cookies), วิธีการร้องขอของเอชทีทีพี ซึ่งเราสามารถรับข้อมูลจากไคลเอนต์โดยวิธีต่าง ๆ ได้ดังนี้

- สำหรับทุก ๆ วิธีการร้องขอของเอชทีทีพี เราสามารถใช้เมทอด `getParameterValues` ซึ่งจะคืนค่าของพารามิเตอร์นั้น หรือเราสามารถที่จะรับคำร้องขอเองได้
- สำหรับเมทอด GET ของเอชทีทีพี เราสามารถใช้เมทอด `getQueryString` เพื่อหาค่าสตริงที่ถูกส่งมา

สำหรับการส่งข้อมูลไปให้ไคลเอนต์ ออบเจ็กต์ `HttpServletRequest` จะมีเมทอด `getWriter` สำหรับการแสดงผลแบบข้อความ หรือเมทอด `getOutputStream` สำหรับการแสดงผลแบบไบนารีต่อไปคือ ตัวอย่างเซิร์ฟเล็ต ชื่อว่า "SimpleServlet" ซึ่งใช้จัดการคำร้องขอแบบ GET เพื่อแสดงข้อความบนหน้าจอบราวเซอร์ของไคลเอนต์

```
/**
```

- This is a simple example of an HTTP Servlet. It responds to the GET methods of the HTTP protocol.

```
*/
```

```
public class SimpleServlet extends HttpServlet
{ public void doGet(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException
{ // set header field first
  res.setContentType("text/html");

  // then get the writer and writer the response data
  PrintWriter out = res.getWriter();
  out.println("<HEAD><TITLE> SimpleServlet Output</TITLE></HEAD><BODY>");
  out.println("<h1>SimpleServlet Output </h1>");
  out.println("<P>This is output from SimpleServlet.");
  out.println("</BODY>");
  out.close();
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยมีการทำงานดังนี้

1. คลาสนี้ จะชื่อ SimpleServlet ซึ่งขยายคลาส HttpServlet จึงอิมพลิเมนต์อินเตอร์เฟซ Servlet แล้ว
2. มีเมทอดไว้สำหรับจัดการกับคำร้องขอแบบ GET คือ เมทอด doGet ซึ่งจะส่งข้อมูลคืนสู่ไคลเอนท์ในรูปแบบของเอกสารเอชทีเอ็มแอล โดยเมทอด setContentType() นั่นเอง

เครื่องมือในการพัฒนาเซิร์ฟเล็ต

เครื่องมือที่ใช้ในการพัฒนา เราสามารถใช้เอดิเตอร์ใด ๆ ก็ได้ โดยต้องบันทึกลงท้ายด้วย .java เช่น SimpleServlet.java เราจะนำไฟล์นี้ไปสร้างเป็นคลาส โดยใช้เจดีเค (Java Development Kit) ของซัน โดยถ้าเป็นเวอร์ชัน 1.2 เซิร์ฟเล็ตจะเป็นสแตนดาร์ดเอพีไอ (Standard API) ส่วนเจดีเคเวอร์ชันต่ำกว่า 1.2 เซิร์ฟเล็ตจะเป็นเพียงส่วนขยายของสแตนดาร์ดเอพีไอ นั่นคือ เราต้องมีการนำแพคเกจ (package) ที่เกี่ยวกับเซิร์ฟเล็ตมาลงเพิ่มเติมในเจดีเคนั้นด้วย จึงจะสามารถคอมไพล์เซิร์ฟเล็ตเป็นไบต์โค้ดได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คู่มือการใช้งานสำหรับผู้ใช้

โครงการนี้ประกอบด้วยการใช้งานใน 3 เรื่อง คือ

1. โหมดการค้นหาข้อมูลโดยค้นหาจากฐานข้อมูล
2. โหมดการค้นหาข้อมูลแบบ ณ เวลานั้น โดยกำหนดจำนวนเพจที่ต้องการค้นหาต่อครั้ง
3. โหมดการตรวจสอบดูว่าแต่ละบูร์แอลมีลิงค์ไปที่ใดบ้าง

โดยจะอธิบายการใช้งานไปตามลำดับดังนี้

โหมดการค้นหาข้อมูลโดยค้นหาจากฐานข้อมูล

Project II - Netscape

File Edit View Go Communicator Help

Location: s:///D:/JavaWebServer1.1.3/public_html/searchpage.html

Project II

TYPE of matching

- match only my keyword
- match all key that have my keyword

IF YOUR KEYWORDS NOT FOUND,
Do you want to insert into our database to find it later?

- YES, I want to.
- NO, I don't want to insert my key.

ป้อนข้อมูลที่ต้องการค้นหา

OK RESET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INFORMATION

Select type of database you want to see

- all keywords in database that have been searched
- keywords in database that have not been searched

[real time search](#) || [test link](#)

Document: Done

นั่นคือ

รูปแสดงโฮมเพจการค้นหาข้อมูลจากฐานข้อมูล

ในการค้นหาด้วยวิธีนี้ ก่อนที่จะป้อนคีย์เวิร์ดผู้ใช้ต้องป้อนเงื่อนไขในการค้นหา 2 เงื่อนไขก่อน

TYPE of matching

- match only my keyword
- match all key that have my keyword

เป็นถามถึงวิธีการ ในการค้นหาว่าต้องการให้ค้นหาโดย

- เทียบคีย์เวิร์ดกับคำในฐานข้อมูลให้ตรงกัน
- เทียบคีย์เวิร์ดกับคำในฐานข้อมูลโดยเทียบว่าจะต้องมีคีย์เวิร์ดนั้นอยู่ในคำในฐานข้อมูล

IF YOUR KEYWORDS NOT FOUND,

Do you want to insert into our database to find it later?

- YES, I want to.
- NO, I don't want to insert my key.

ในกรณีที่ไม่มีคีย์เวิร์ดที่ต้องการค้นหาอยู่ในฐานข้อมูล ผู้ใช้ต้องบอกถึงความต้องการของตนเองที่จะให้ เซิร์ฟเวอร์เก็บคีย์เวิร์ดนั้นไว้เพื่อทำการค้นหาให้ใหม่เมื่อต้องการอัปเดตข้อมูลหรือไม่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ป้อนข้อมูลที่ต้องการค้นหา

โท

INFORMATION

Select type of database you want to see

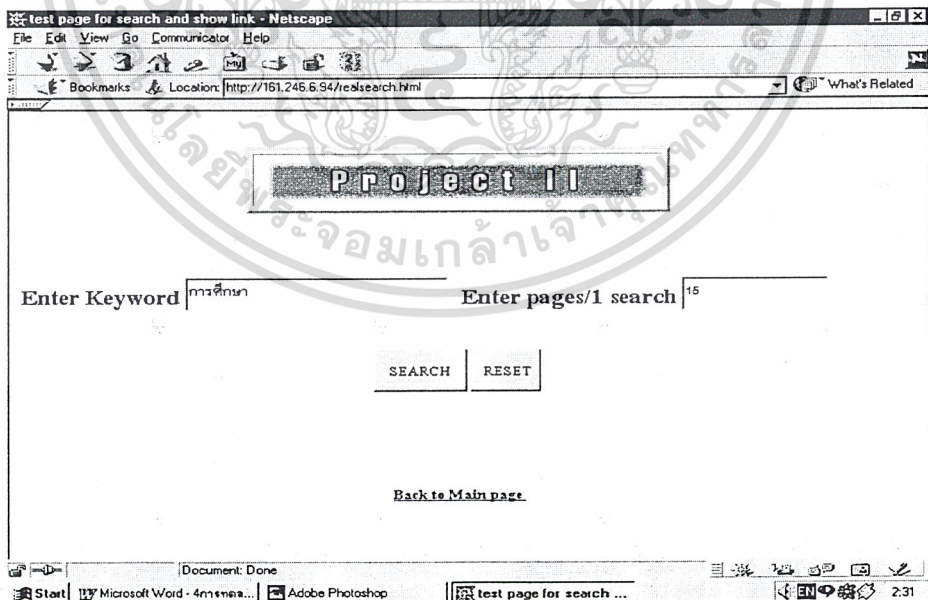
- all keywords in database that have been searched
- keywords in database that have not been searched

เป็นส่วนที่ใช้สำหรับป้อนคีย์เวิร์ดที่ต้องการค้นหา

ในตอนท้ายของโฮมเพจนี้ หากผู้ใช้ต้องการทราบว่ามีคำใดบ้างที่มีในฐานข้อมูล และคำใดบ้างที่จะเป็นคำที่จะต้องค้นหาต่อไป ก็สามารถตรวจสอบได้จากส่วนนี้

และท้ายสุดของโฮมเพจนี้ ผู้ใช้สามารถลิงค์ไปยังการใช้งาน 2 ส่วนที่เหลือได้

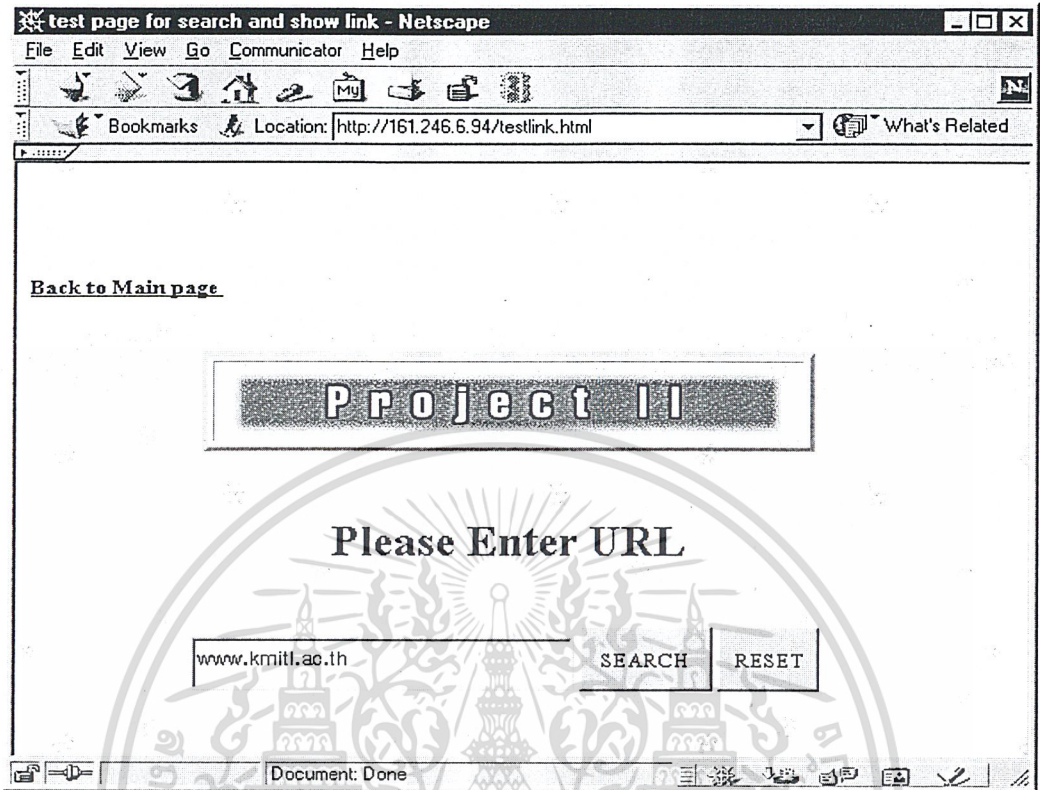
โฮมเพจการค้นหาข้อมูลแบบ ณ เวลานั้น โดยกำหนดจำนวนเพจที่ต้องการค้นหาต่อครั้ง



ในช่อง Enter Keyword ให้ใส่คีย์เวิร์ดที่ต้องการให้ค้นหา

ในช่อง Enter pages / 1 search ให้ใส่ตัวเลขจำนวนเพจที่ต้องการให้ค้นหาให้ต่อการค้นหา 1 ครั้ง
 ไม่ว่าการมีได้ และในตอนท้ายของโฮมเพจ จะมีลิงค์กลับไปยังหน้าโฮมเพจปกติ

โฮมเพจการตรวจสอบดูว่าแต่ละยูอาร์แอลมีลิงค์ไปที่ใดบ้าง



สำหรับโฮมเพจหน้านี้ใช้สำหรับตรวจสอบว่า ยูอาร์แอลนั้นมีลิงค์ไปที่ใดบ้าง โดยผู้ใช้ต้องเติมยูอาร์แอลที่ต้องการตรวจสอบลิงค์ลงในช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

รายละเอียดของโปรแกรม

ในโครงการนี้ เราสามารถแยกการทำงานออกเป็นสองส่วนใหญ่ คือ

1 ส่วนการทำงานของจาวาแอปพลิเคชัน (java application) จะมีไฟล์ที่สำคัญอยู่ 5 ไฟล์ คือ

1.1 ไฟล์ MyProject.java

จะเป็นไฟล์การทำงานหลัก ใช้เป็นตัวควบคุมโปรแกรมในส่วนที่เป็นจาวาแอปพลิเคชันทั้งหมด

1.2 ไฟล์ MyPageFile.java

จะเป็นไฟล์ที่ใช้ในการดึงเนื้อหาของเพจที่ได้มาจากยูอาร์แอลที่กำหนด มาสู่ไฟล์ชั่วคราว (temp file) โดยจะมีเมธอดที่ใช้ตรวจสอบว่าเนื้อหาที่ได้นั้นสามารถใช้ประโยชน์ได้หรือไม่

1.3 ไฟล์ MyLink.java

เป็นไฟล์ที่ใช้ในการหาไฮเปอร์ลิงก์ที่มีคุณสมบัติตรงตามที่ต้องการในเพจนั้นทั้งหมด โดยจะนำไฮเปอร์ลิงก์นั้นไปเขียนเก็บไว้ในไฟล์ชั่วคราวอีกไฟล์หนึ่ง เพื่อเตรียมที่จะจัดเก็บลงฐานข้อมูลต่อไป

1.4 ไฟล์ MyOperate.java

เป็นไฟล์ที่ใช้ในการทำงานเกี่ยวกับการหาข้อมูลที่ต้องการในเพจนั้น ได้แก่ การหาหัวเรื่อง (title) การหาคำอธิบาย (description) การหาค่าไพเออร์รีตีของคีย์เวิร์ดแต่ละคำ การหาขนาดของเพจ เป็นต้น

1.5 ไฟล์ MyDatabase.java

เป็นไฟล์ที่ใช้จัดการเกี่ยวกับฐานข้อมูลทั้งหมด ได้แก่ การเปิดการติดต่อฐานข้อมูล การเก็บลงฐานข้อมูล การดึงข้อมูลที่ต้องการจากฐานข้อมูล เป็นต้น

2 ส่วนการทำงานของจาวาเซิร์ฟเล็ต (java servlets) จะมีไฟล์ที่สำคัญอยู่ 4 ไฟล์ คือ

2.1 ไฟล์ SearchServlet.java

เป็นไฟล์ที่ทำหน้าที่ในการแสดงผลการค้นหาข้อมูล ไม่ว่าจะเป็นข้อมูลที่ถูกร้องขอด้วยวิธีโพสต์ (post) หรือ เก็ต (get) ก็ตาม รวมทั้งยังสามารถเก็บคีย์เวิร์ดที่ต้องการลงฐานข้อมูลด้วย

2.2 ไฟล์ SearchInfo.java

เป็นไฟล์ที่ทำหน้าที่ในการแสดงคีย์เวิร์ดที่มีอยู่ในฐานข้อมูลทั้งหมด

2.3 ไฟล์ SearchLink.java

เป็นไฟล์ที่ทำหน้าที่ในการแสดงวิธีการหาไฮเปอร์ลิงก์ทั้งหมดของยูอาร์แอลที่กำหนดให้

2.4 ไฟล์ RealSearch.java

เป็นไฟล์ที่ทำหน้าที่คล้ายกับส่วนของจาวาแอปพลิเคชันทั้งหมด เพื่อแสดงการค้นหาข้อมูลแบบเรียลไทม์ (real time)

โดยจะมีรายละเอียดและขั้นตอนในการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานจะเริ่มจากเรียกไฟล์ MyProject.class ขึ้นมาใช้งาน โดยใช้คำสั่งที่ต่อสพร้อมว่า

```
java MyProject
```

จากนั้น โปรแกรมจะถามถึงจำนวนเพลงที่จะให้รับในหนึ่งครั้งของการทำงาน เมื่อผู้ใช้ใส่จำนวนเพลงแล้ว โปรแกรมจะเรียกเมทอดของไฟล์ MyDatabase.java เพื่อที่จะดึงค่าต่าง ๆ ที่ถูกเก็บไว้ในฐานข้อมูลเพื่อนำมาใช้งานต่อไป แล้วโปรแกรมจะรับยูอาร์แอลและเรียก MyPageFile เพื่อให้ตัวโปรแกรมไปดึงเนื้อหาของยูอาร์แอลที่กำหนดนั้นมา และเก็บไว้ในไฟล์ชั่วคราว แล้วจึงนำไฟล์นั้นไปให้ไฟล์ MyLink.java ทำการหาไฮเปอร์ลิงค์ทั้งหมด และให้ไฟล์ MyOperate ทำการค้นหาข้อมูลเกี่ยวกับเพลงนั้นทั้งหมด แล้วจึงเรียกไฟล์ MyDatabase เพื่อเก็บลงฐานข้อมูลตามต้องการ

รายละเอียดของแต่ละไฟล์ และเมทอดโดยย่อ

public class MyProject

```
public MyProject(int round)
```

คอนสตรัคเตอร์ สำหรับสร้างออบเจกต์ของคลาสนี้ โดยรับอาร์กิวเมนต์เป็นจำนวนของเพลงที่จะให้ทำในหนึ่งครั้ง

```
public void getStart()
```

เมทอดที่ใช้สำหรับเริ่มต้นค่าต่าง ๆ รวมทั้งยังไว้หาชนิดของการกระทำในครั้งนั้นด้วย

```
private void getState()
```

เมทอดที่ใช้สำหรับดึงค่าเริ่มต้นที่เก็บไว้ในฐานข้อมูล เช่น จำนวนไฮเปอร์ลิงค์ทั้งหมดในฐานข้อมูล เป็นต้น

```
private void saveState()
```

```
private void saveData()
```

เมทอดที่ใช้สำหรับเก็บค่าต่าง ๆ ลงสู่ฐานข้อมูล ไม่ว่าจะเป็นสเตทหรือข้อมูลต่าง ๆ

```
private int getType()
```

สำหรับหาชนิดของการกระทำในครั้งนั้น ๆ

```
private void deleteFileContent()
```

เมทอดที่ไว้สำหรับลบเนื้อหาในไฟล์และตั้งค่าใหม่ตั้งแต่ต้น

```
private void continueSearch()
```

หนึ่งในชนิดของการกระทำ

```
private void continueUpdate()
```

หนึ่งในชนิดของการกระทำ

```
//===== end class MyProject =====
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public class MyPageFile

```
public MyPageFile(String url)
```

คอนสตรัคเตอร์ที่รับอาร์กิวเมนต์เข้ามาเป็นสตริง ที่เป็นชื่อของยูอาร์แอลนั้น

```
public String getHostName()
```

เมทอดที่คืนค่าสตริงที่เป็นชื่อโฮสต์ของยูอาร์แอลนั้น

```
public String getFileName()
```

เป็นเมทอดที่คืนค่าสตริงที่เป็นชื่อของไฟล์ที่เราใช้เก็บเนื้อหาของเพจชั่วคราวจากยูอาร์แอลที่กำหนด

```
public File getURLFile()
```

เมทอดที่คืนค่าเป็นไฟล์ที่เราไว้เก็บเนื้อหาของเพจจากยูอาร์แอลที่กำหนด

```
public boolean isUsable()
```

เมทอดที่ใช้ตรวจสอบว่าไฟล์ที่รับมานั้นใช้ได้หรือไม่

```
public void getToFile()
```

เป็นเมทอดที่ใช้สำหรับดึงเนื้อหาจากยูอาร์แอลที่กำหนดมาไว้ในไฟล์ชั่วคราวที่เรากำหนด

```
private void clearFile()
```

เป็นเมทอดที่ไว้สำหรับลบเนื้อหาในไฟล์ชั่วคราวของเรา

```
//===== end class MyPageFile =====
```

public class MyOperate

```
public MyOperate(File file)
```

```
public MyOperate(String tofile)
```

คอนสตรัคเตอร์ที่รับอาร์กิวเมนต์มาเป็นไฟล์ หรือสตริงเพื่อมาแปลงเป็นไฟล์ที่เราเก็บเนื้อหาของเพจชั่วคราว เพื่อนำไปหาข้อมูลต่าง ๆ ต่อไป

```
private int getTagValue(int nowvalue,String tag)
```

เมื่อเราอ่านจนพบแท็กในเอชทีเอ็มแอล เราก็จะกำหนดค่าให้กับแท็กนั้น เช่น แท็กไอดีล์มีค่าเป็น 30 เป็นต้น

```
public int getValue(String keyword)
```

รับอาร์กิวเมนต์มาเป็นคีย์เวิร์ดที่ต้องการหาค่าไพโรอริตี้ เพื่อคำนวณหาค่าไพโรอริตี้ของแต่ละคีย์เวิร์ดนั้น

แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public int getSize()
```

เมธอดที่ใช้สำหรับหาค่าของขนาดของเพจนั่น ๆ

```
public String getTitle()
```

เมธอดที่ใช้สำหรับหาไตเติ้ล (title) ของแต่ละเพจ

```
public String getDescription()
```

เมธอดที่ใช้สำหรับหาคำอธิบาย (description) ของแต่ละเพจ

```
private String getDoubleQuote(String sttest)
```

เนื่องจากในการเก็บข้อมูลลงฐานข้อมูล ตัวโปรแกรมจะใช้เครื่องหมายคำพูดเป็นตัวกำหนด (") ดังนั้น ถ้าสตริงไหน มีเครื่องหมายคำพูดอยู่ เราต้องแปลงให้มีเครื่องหมายนี้ซ้อนกันเสียก่อน ("") โดยใช้เมธอดนี้

```
//===== END CLASS MyOperate =====
```

```
public class MyLink
```

```
public MyLink(String unname,File myfile)
```

คอนสตรัคเตอร์ที่รับอาร์กิวเมนต์มาเป็นสตริงที่เป็นชื่อของยูอาร์แอล และ ไฟล์ที่ใช้เก็บเนื้อหาชั่วคราวของมัน

```
private String getDeciseBase(String oldbase)
```

ใช้สำหรับค้นหายูอาร์แอลเบสที่สมบูรณ์เพื่อนำไปหาไฮเปอร์ลิงค์ต่อไป

```
public void getLink()
```

ใช้สำหรับค้นหาไฮเปอร์ลิงค์ในเพจนั่น

```
private void writeFile(String llist)
```

เขียนไฮเปอร์ลิงค์ที่ค้นหามาได้ลงไฟล์ เพื่อเตรียมที่จะนำไปเก็บไว้ในฐานข้อมูลต่อไป

```
private String getBase()
```

ใช้สำหรับค้นหายูอาร์แอลเบสที่สมบูรณ์เพื่อนำไปหาไฮเปอร์ลิงค์ต่อไป

```
//===== END CLASS MYLINK =====
```

```
public class MyDatabase
```

```
public MyDatabase()
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนสตรัคเตอร์ของคลาสที่ใช้สำหรับติดต่อกับฐานข้อมูลทั้งหมด ในส่วนนี้จะรวมถึง การเปิดการติดต่อกับฐานข้อมูลด้วย

```
public void getClose() throws SQLException
```

ใช้สำหรับปิดการติดต่อกับฐานข้อมูล

```
public void insertToKeyTable(File f1) throws SQLException
```

สำหรับนำค่าไพรอริตี้ คีย์เวิร์ด และเพจที่เกี่ยวข้อง ไปเก็บไว้ในฐานข้อมูล

```
public void insertToDataTable(File f1) throws SQLException
```

สำหรับนำค่าต่าง ๆ ที่ค้นได้จากเพจหนึ่ง ๆ ไปเก็บไว้ในฐานข้อมูล

```
public int insertToLinkTable(File f1,int linkorder) throws SQLException
```

สำหรับนำไฮเปอร์ลิงค์ต่าง ๆ ที่ค้นหาได้ ไปเก็บไว้ในฐานข้อมูล

```
private boolean checkRepeat(String test) throws SQLException
```

สำหรับตรวจสอบว่าไฮเปอร์ลิงค์ที่จะนำไปเก็บซ้ำซ้อนกับที่มีอยู่แล้วหรือไม่ ถ้าซ้ำซ้อน ก็จะไม่เก็บไว้ในฐานข้อมูล

```
public ResultSet selectLink() throws SQLException
```

```
public ResultSet getUpdateList(int up) throws SQLException
```

สำหรับเลือกยูอาร์แอลที่บอทจะทำการค้นหาต่อไป

```
public ResultSet getOperateKey() throws SQLException
```

```
public ResultSet getUpdateKey() throws SQLException
```

สำหรับเลือกคีย์เวิร์ดที่จะนำไปหาไพรอริตี้ของแต่ละยูอาร์แอลที่บอทนั้น วิ่งไปหามาได้

```
public int getPageOrder() throws SQLException
```

```
public int getLinkOrder() throws SQLException
```

```
public int getUpdateNumber() throws SQLException
```

ใช้สำหรับหาค่าสเตตต่าง ๆ ที่เก็บไว้ เพื่อนำไปใช้งาน

```
public void deleteFromLinkTable() throws SQLException
```

```
public void deleteUnSearchKey() throws SQLException
```

ใช้สำหรับลบไฮเปอร์ลิงค์หรือคีย์เวิร์ดที่ไม่ต้องการออกจากฐานข้อมูล

```
public int countUnSearchKey() throws SQLException
```

ใช้สำหรับนับคีย์เวิร์ด เพื่อค้นหาว่าในครั้งนั้นบอทจะกระทำการกระทำชนิดใด

```
//===== end class MyDatabase =====
```

นอกจากนั้น สามารถดูรายละเอียดจากโค้ดของโปรแกรมที่ได้ให้มาด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

Source Code

1. Myproject.java

```

package SEARCH;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;

public class MyProject
{
    private int pagenumber; // number of page have get
    private int linknumber;
    private int updatenumber;
    private int operatetype;
    private int turn_number;
    File linklist = new File("linklist.lst");
    File key = new File("key.ins");
    File data = new File("data.ins");

//-----
    public MyProject(int round)
    { turn_number = round;
      deleteFileContent();
      getState();
      operatetype = getType();
      System.out.println("TYPE OF OPERATION IS "+operatetype);
    }

//-----
    public void getStart()
    { switch(operatetype)
      { case 1:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        { continueUpdate();
          break;
        }
      case 2:
        { moveKey();
          continueUpdate();
          break;
        }
      case 3:
        { continueSearch();
          break;
        }
      }
    }
}
//-----
private void getState() // for get status of search
{ try
  { MyDatabase mystate = new MyDatabase();
    pagenumber = mystate.getPageOrder();
    System.out.println("pagenumber: "+pagenumber);
    linknumber = mystate.getLinkOrder();
    System.out.println("linknumber: "+linknumber);
    updatenumber = mystate.getUpdateNumber();
    System.out.println("updatenumber: "+updatenumber);
    mystate.getClose();
  } catch(SQLException ioe)
  { System.out.println("get state error: "+ioe);
  }
  System.out.println("get state complete");
}
//-----
private void saveState() // for set status of search
{ try

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        save.keepState(pagenumber,linknumber);
        save.getClose();
    } catch(SQLException sse)
    { System.out.println("save state error: "+sse);
    }
    System.out.println("save state complete");
}
//-----
private void moveKey()
{ try
    { MyDatabase mymove = new MyDatabase();
      ResultSet move = mymove.getUnSearchKey();
      while( move.next() )
      { String un_key = move.getString("KEYWORDS");
        un_key = un_key.trim();
        try
        { MyDatabase settodel = new MyDatabase();
          settodel.setKeyToDelete(un_key);
          settodel.insertToSearchedKey(un_key,1);
          settodel.getClose();
        } catch(SQLException sx)
        { System.out.println("error while move key");
        }
      }
      mymove.deleteUnSearchKey();
      mymove.getClose();
      System.out.println();
      System.out.print("      ");
      System.out.println("MOVE KEY COMPLETE");
    } catch(SQLException s)
    { System.out.println("error when move key and is: "+s);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void saveData()
{ try
    { System.out.println("Start to save data to database");
      MyDatabase mydatabase4 = new MyDatabase();
      mydatabase4.insertToKeyTable(key);
      mydatabase4.insertToDataTable(data);
      mydatabase4.getClose();
    } catch(SQLException eee)
    { System.out.println("error while keep key and data: "+eee);
    }
  try
    { MyDatabase mydatabase5 = new MyDatabase();
      linknumber = mydatabase5.insertToLinkTable(linklist,linknumber);
      mydatabase5.getClose();
    } catch(SQLException ss)
    { System.out.println("error while insert to database: "+ss);
    }
  try
    { MyDatabase mydatabase6 = new MyDatabase();
      mydatabase6.deleteFromLinkTable();
      mydatabase6.getClose();
      System.out.println("save data complete");
    } catch(SQLException sos)
    { System.out.println("error while delete from link table: "+sos);
    }

  deleteFileContent();
  saveState();
}
//-----

```

```
private void saveUpdateData()
```

```

{ try
    { System.out.println("Start to save update data to database");
      MyDatabase mydatabase7 = new MyDatabase();
      mydatabase7.insertToKeyTable(key);

```

```

mydatabase7.changeUpdateNumber(updatenumber);
mydatabase7.getClose();
} catch(SQLException eee)
{ System.out.println("error while keep key and data: "+eee);
}
deleteFileContent();

}

//-----
private int getType()
{ int o_type = 3;
  if( updatenumber!=0 )
  { o_type = 1;
  }
  else
  { int countkey=0;
    try
    { MyDatabase keytype = new MyDatabase();
      countkey = keytype.countUnSearchKey();
      keytype.getClose();
      if( countkey>30 )
      { o_type = 2;
      }
      else
      { o_type = 3;
      }
    } catch(SQLException sq)
    { System.out.println("get Type error: "+sq);
    }
  }
  return o_type;
}

//-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 private void deleteFileContent()
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

if( pagelist.next() )
{
    pagename = pagelist.getString("urlname").trim();
    System.out.println("get page name: "+pagename);
    System.out.println("counter: "+counter);
    System.out.println("Page number: "+pagenumber);
    MyPageFile mypage = new MyPageFile(pagename);
    mypage.getToFile();
    if( mypage.isUsable() )
    {
        MyLink mylink = new MyLink(pagename,mypage.getURLFile());
        mylink.getLink();
        MyOperate myoperate = new MyOperate(mypage.getURLFile());
        System.out.println("Start to get key");
        int ii = 1;
        try
        {
            FileWriter fw1 = new FileWriter("key.ins",true);
            BufferedWriter bw1 = new BufferedWriter(fw1);

            FileWriter fw2 = new FileWriter("data.ins",true);
            BufferedWriter bw2 = new BufferedWriter(fw2);

            try
            {
                String keyword = "";
                MyDatabase mydatabase2 = new MyDatabase();
                ResultSet keylist = mydatabase2.getOperateKey();
                while( keylist.next() )
                {
                    keyword=keylist.getString("KEYWORDS");
                    keyword = keyword.trim();
                    System.out.println("get key: "+ii);
                    int priority = myoperate.getValue(keyword);
                    if( priority>0 )
                    {
                        bw1.write(keyword+ "|" +pagenumber+ "|" +priority);
                        bw1.newLine();
                    }
                    bw1.flush();
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        System.out.println("page number "+pagenumber+
        ". get key complete while counter = "+counter);
        ii++;
    }        // END WHILE HAVE KEYWORD
    mydatabase2.getClose();
} catch(SQLException ex)
{ System.out.println("keyword error: "+ex);
}
bw1.close();
bw2.write(" " +pagenumber+ "|" +pagename+ "|" +myoperate.getSize()+
        "|" +myoperate.getDescription()+ "|" +myoperate.getTitle() );
bw2.newLine();
bw2.flush();
bw2.close();
System.out.println("operate page: "+ pagename+ ". COMPLETE");
System.out.println();
pagenumber++;
} catch(IOException ioe1)
{ System.out.println("error while save data: "+ioe1);
}
}        // end if mypage is usable
else
{ System.out.print(" ");
System.out.println("THIS PAGE IS NOT USABLE");
if( mypage.getConnectionErrorFlag() )
{ System.out.println(" because of CE PROXY !!!!!!!");
try
{ FileWriter fw3 = new FileWriter("linklist.lst",true);
BufferedWriter bw3 = new BufferedWriter(fw3);
bw3.write(pagename);
bw3.newLine();
bw3.flush();
bw3.close();
} catch(IOException io)

```

```

        { System.out.println("error while write connection failed link");
        }
        counterror++;
    }
}
try
{ MyDatabase mydatabase3 = new MyDatabase();
  mydatabase3.setLinkToDelete(pagename);
  mydatabase3.getClose();
} catch(SQLException sqlexcept)
{ System.out.println("set to delete error: "+sqlexcept);
}
} // end if pagelist.next();
else
{ System.out.println("Don't have page list in database");
}
mydatabase1.getClose();
} catch(SQLException s_exception)
{ System.out.println("list error : "+s_exception);
}
if( (counter!=0)&&((counter%5)==0) )
{ saveData();
}
counter++;
if( counterror==10 )
{ break OPER;
}
} // END WHILE COUNTER <= TURN_NUMBER
if( counterror==10 )
{ System.out.println("TERMINATE BECAUSE NETWORK PROBLEM");
}
else
{ System.out.println("OPERATION COMPLETE");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

saveData();

getState();

}

//-----

private void continueUpdate()
{ System.out.println("      START TO CONTINUE UPDATE...");

int counter = 1;

int counterror=0;

try

{ MyDatabase myupdate = new MyDatabase();

ResultSet updatelist = myupdate.getUpdateList(updatenumber);

ENDING:

while( updatelist.next() )

{ int order = updatelist.getInt("URLNUMBER");

String pagename = updatelist.getString("urlname").trim();

System.out.println("get page name: "+pagename);

System.out.println("counter: "+counter);

System.out.println("Page number: "+order);

MyPageFile mypage = new MyPageFile(pagename);

mypage.getToFile();

if( mypage.isUsable() )

{ File pagefile = mypage.getURLFile();

MyOperate myoperate = new MyOperate(pagefile);

System.out.println("Start to get key for update");

int up_count = 1;

try

{ FileWriter fw1 = new FileWriter("key.ins",true);

BufferedWriter bw1 = new BufferedWriter(fw1);

try

{ String keyword ="";

MyDatabase mydatabase2 = new MyDatabase();

ResultSet updatekeylist = mydatabase2.getUpdateKey();

while( updatekeylist.next() )

{ keyword= updatekeylist.getString("KEYWORDS");

```

```

keyword = keyword.trim();
System.out.println("get key: "+up_count);
int priority = myoperate.getValue(keyword);
if( priority>0 )
{ bw1.write(keyword+ "|" +order+ "|" +priority);
  bw1.newLine();
  bw1.flush();
}
System.out.println("page number "+order+
  ". get key complete while counter = "+counter);
up_count++;
} // END WHILE HAVE KEYWORD
mydatabase2.getClose();
} catch(SQLException ex)
{ System.out.println("keyword while update error: "+ex);
}
bw1.close();
System.out.println("operate page: "+ order+ " . COMPLETE");
System.out.println();
updatenumber = order ;
if( counter == turn_number )
{ break ENDING;
}
} catch(IOException ioe1)
{ System.out.println("error while save data: "+ioe1);
}
} // end if mypage is usable
else
{ System.out.print(" ");
System.out.println("THIS PAGE IS NOT USABLE");
if( mypage.getConnectionErrorFlag() )
{ System.out.println(" NETWORK PROBLEM");
counterror++;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if( countererror==5 )
    { break ENDING;
    }
    counter++;
}          // end while updatelist.next();
myupdate.getClose();

} catch(SQLException s_exception)
{ System.out.println("list error : "+s_exception);
}
saveUpdateData();
if( ( updatenumber+1 ) == pagenumber )
{ setUpdateComplete();
}
if( countererror==5 )
{ System.out.println("UPDATE TERMINATE BECAUSE OF NETWORK PROBLEM");
}
else
{ System.out.println("OPERATION UPDATE COMPLETE");
}
getState();
}
//-----
private void setUpdateComplete()
{ try
{ MyDatabase updatecomplete = new MyDatabase();
updatecomplete.changeUpdateNumber(0);
updatecomplete.changeToCommonKey();
updatecomplete.getClose();
} catch(SQLException sw)
{ System.out.println("error while set update from completeness: "+sw);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----
public static void main(String[] args)
{ int each = 0;
  try
  { InputStreamReader isr = new InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(isr);
    GETTURN:
    while(true)
    { System.out.print("Enter number of operation: ");
      String temp = br.readLine().trim();
      try
      { each = Integer.parseInt(temp);
        break GETTURN;
      } catch(Exception ee)
      { System.out.println("error while get turn: "+ee);
      }
      System.out.println();
    }
    br.close();
  } catch(Exception e)
  { System.out.println("error while get turn2: "+e);
  }
  MyProject myproject = new MyProject(each);
  myproject.getStart();
}
}
//===== end class project =====
```

2. MyPageFile.java

```
package SEARCH;
```

```
import java.io.*;
```

```
import java.net.*;
```

```
import java.util.*;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public class MyPageFile
{ URL myurl;
  String urlname;
  File myfile = new File("tempurl");
  boolean errorflag;
//-----
  public MyPageFile(String url)
  { errorflag = false;
    try
    { urlname= url;
      myurl = new URL(urlname);
    } catch(MalformedURLException mal)
    { System.out.println("myurl error: constructor: "+mal);
    }
  }
//-----
  public String getHostName()
  { return myurl.getHost();
  }
//-----
  public String getFileName()
  { return myurl.getFile();
  }
//-----
  public File getURLFile()
  { return myfile;
  }
//-----
  public boolean isUsable()
  { boolean doit=true;
    boolean page = false;
    boolean content = false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 String temp = "";
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

String lower = "";
try
{
    FileReader fr = new FileReader(myfile);
    BufferedReader br = new BufferedReader(fr);
    while(doit)
    {
        temp=br.readLine();
        if( temp==null )
        {
            doit = false;
        }
        else
        {
            lower = temp.toLowerCase().trim();
            if( lower.startsWith("<html>") )
            {
                doit = false;
            }
            if( lower.startsWith("http/1.0 2") )
            {
                page = true;
            }
            if( (lower.equals("content-type: text/html")) ||
                (lower.equals("content-type: text/plain")) )
            {
                content = true;
            }
        }
    }
    br.close();
} catch(IOException ioe)
{
    System.out.println("error: "+ioe);
}
System.out.println("check page complete");
return (page&&content);
}

```

```
//-----
```

```
public void getToFile()
```

```
{
    InputStreamReader in;
    BufferedReader br;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FileWriter fw;
BufferedWriter bw;
try
{ Socket socket = new Socket("proxy.ce.kmitl.ac.th",3128);
  socket.setSoTimeout(120000);
  OutputStream rawOut = socket.getOutputStream();
  InputStream rawIn = socket.getInputStream();

  DataOutputStream dos = new DataOutputStream(rawOut);
  dos.writeBytes("Get "+urlname+ " HTTP/1.0\r\n\r\n");
  in = new InputStreamReader(rawIn);
  br = new BufferedReader(in);
  fw = new FileWriter(myfile);
  bw = new BufferedWriter(fw);
  String temp="";
  while( (temp=br.readLine())!=null )
  { bw.write(temp);
    bw.newLine();
    bw.flush();
  }
  bw.close();
  System.out.println("page: "+ urlname);
  System.out.println("Retrive Successful!");
} catch(IOException ioe)
{ System.out.println("getToFile error: "+ioe);
  errorflag=true;
  clearFile();
}
}
//-----
public boolean getConnectionErrorFlag()
{ return errorflag;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void clearFile()
{ try
  { FileWriter fw1 = new FileWriter(myfile);
    BufferedWriter bw1 = new BufferedWriter(fw1);
    bw1.write("");
    bw1.flush();
    bw1.close();
  } catch(IOException ioe)
  { System.out.println("clear file error: "+ioe);
  }
}
}
//===== end class MyPageFile =====

```

3. MyLink.java

```

package SEARCH;

import java.io.*;
import java.util.*;
import java.net.*;

public class MyLink
{ File pagefile;
  String urlname;
  String urlbase;
  String urlhost;

//-----

// constructor accept 2 parameters are URL and File for store temp content
public MyLink(String uname,File myfile)
{ pagefile = myfile;
  urlname =uname;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----
private String getDeciseBase(String oldbase)
{ String newbase = "";
  if( oldbase.endsWith("/") )
  { newbase = oldbase;
  }
  else
  { int from = oldbase.lastIndexOf("/");
    int comma = oldbase.indexOf(".",from+1);
    if(from >10)
    { if( comma!= -1 )
      { newbase = oldbase.substring(0,from+1);
      }
      else
      { newbase = oldbase+"/";
      }
    }
    else newbase = oldbase+"/";
  }
  return newbase;
}
}

```

```
//-----
public void getLink()
{ boolean islink = false;
  boolean have = true;

  int i=0;

  char input;

  String link="";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 String refer = " ";
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

urlbase = getBase();
if( urlbase==null ) urlbase = urlname;

urlbase = getDeciseBase(urlbase);

int slash = urlbase.indexOf("/",10) ;
if( slash!= -1 ) urlhost = urlbase.substring(0,slash);
else urlhost = urlbase;

```

```

System.out.println("H: "+urlhost+":");
System.out.println("B: "+urlbase+":");

```

```

try
{
FileReader fr = new FileReader(pagefile);
BufferedReader br = new BufferedReader(fr);
readlink:
while(have)
{
i = br.read();
if(i==32) i=br.read();
if(i!==-1)
{
input = (char) i;
compare = compare.substring(1,2)+input;

if(compare.equalsIgnoreCase("<a")
{
islink = true;
}
//end if(compare...

```

```

while(islink)
{
i = br.read();
if(i==32) i=br.read();
if(i===-1) break readlink;

input = (char) i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 refer = refer.substring(1,5)+input;
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(refer.equalsIgnoreCase("href="))
{ link="";
  int countfun=0;
  boolean end = true;
  i=br.read();
  if(i==1) break readlink;
  while((i==32)||i==34||i==13)
  { if(i==32) i=br.read();
    if(i==13)
    { i=br.read();
      i=br.read();
    }
    if(i==34)
    { countfun++;
      i=br.read();
      if(countfun==2) end=false;
    }
    if(i==1) break readlink;
  }
  input = (char) i;
  while(end)
  { link = link+input;
    i = br.read();
    if(i==1) break readlink;
    input = (char) i;
    if( (i==32)||i==34||(input=='>' ) end=false;
  } // end while(input...
  boolean check = link.equals("");
  if(!check)
  { writeFile(link);
  }
  islink=false;
} // end if(refer...
} // end while(islink)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }                //end if
else
    { have = false;
    }                // end else if
}                // end while(have)
br.close();
}                // end try
catch(IOException ioe)
{ System.out.println(ioe);
}
System.out.println("get link complete");
}                // end main
//-----
private void writeFile(String llist)
{ llist = llist.toLowerCase();
String temp = "";
boolean yes = true;
try
{ FileWriter fw = new FileWriter("linklist.lst",true);
BufferedWriter bw = new BufferedWriter(fw);

if( llist.startsWith("http://"))
{ try
{ URL myu = new URL(llist);
String temphost = myu.getHost().toLowerCase();
if( temphost.endsWith(".th") )
{ temp = llist;
}
else yes = false;
} catch(MalformedURLException mfe)
{ System.out.println("this link is malformed: "+llist);
yes=false;
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( yes )
{
    if( (llist.endsWith(".exe")) || (llist.endsWith(".gif")) ||
        (llist.endsWith(".zip")) || (llist.endsWith(".jpg")) )
    {
        yes = false;
    }
}
}

else
{
    if( llist.indexOf(".") == -1 )
    {
        if( llist.startsWith("/") )
        {
            temp = urlhost+llist;
        }
        else
        {
            if( llist.startsWith("../") )
            {
                int j = urlbase.lastIndexOf("/", urlbase.length()-2);
                temp = urlbase.substring(0,j)+llist.substring(2);
            }
            else temp = urlbase+llist;
        }
    }
    else
    {
        yes = false;
    }
}

if( (llist.indexOf(".asp?") != -1 )
{
    yes = false;
}

if( yes )
{
    if( temp.indexOf("#") != -1 )
    {
        temp = temp.substring(0, temp.indexOf("#"));
    }

    temp = temp.trim();

    if( !(temp.endsWith("/") ) )
    {
        int k = temp.lastIndexOf("/");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if( k<10 )
    { temp = temp+"/";
    }
    else
    { int from = temp.indexOf(".",k);
      if( from == -1 )
      { temp = temp+"/";
      }
    }
  }
  bw.write(temp);
  bw.newLine();
  bw.flush();
  bw.close();
}
}
catch(IOException ioe)
{ System.out.println(ioe);
}
}
//-----
private String getBase()
{ int i=0;

  char input = ' ';

  String tag ="";
  String base="";

  boolean doit= true;
  boolean nothavebase = true;
  boolean istag = false;
  boolean isbase = false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

try
{
    FileReader fr = new FileReader(pagefile);
    BufferedReader br = new BufferedReader(fr);

    while( (doit)&&(nothavebase) )
    {
        i = br.read();
        input = (char) i;
        if( i== -1 )
        {
            doit = false;
        }
        if( input=='<' )
        {
            istag = true;
        }
        if( istag )
        {
            tag = tag+input;
        }
        if( isbase )
        {
            base = base +input;
        }
        if( tag.equalsIgnoreCase("<base") )
        {
            isbase = true;
        }
        if( tag.equalsIgnoreCase("</head>") )
        {
            doit = false;
        }
        if( input=='>' )
        {
            istag = false;
            tag = "";
        }
        if( (isbase)&&(input=='>') )
        {
            nothavebase = false;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ System.out.println("error: "+ioe);
}

if( base.length(>0 )
{ base = base.trim();
String temp = base.toLowerCase();
int k= temp.indexOf("href=");
if( k!=-1 )
{
base = base.substring(k+5);
base = base.trim();

base = base.replace('\r',' ');
base = base.replace('\n',' ');

StringTokenizer st = new StringTokenizer(base);
if( st.hasMoreTokens() )
{ base = st.nextToken();
}

while( base.startsWith("\'") )
{ base = base.substring(1);
base = base.trim();
}

while( (base.endsWith("\'") || (base.endsWith(">"))) )
{ base = base.substring(0,base.length()-1);
base = base.trim();
}
}

else base = null;

if( (base!=null) && (!(base.startsWith("http://"))) &&
(!(base.startsWith("HTTP://"))) )
{ base = null;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else base = null;

    return base;
}
//-----
}
//===== END CLASS MYLINK =====

```

4. MyOperate.java

```
package SEARCH;
```

```

import java.util.*;
import java.net.*;
import java.io.*;

public class MyOperate
{ File htmlfile;
  String[] opentag = {"<comment>","<title>","<big>","<i>","<b>","<u>","<strong>","<em>","<h1>","
<h2>","<h3>","<h4>","<h5>","<h6>"};
  String[] closetag = {"</comment>","</title>","</big>","</i>","</b>","</u>","</strong>","</em>","
</h1>","</h2>","</h3>","</h4>","</h5>","</h6>"};
  int[] value = {0,30,7,7,7,7,7,20,17,14,11,9,8};
//-----

  public MyOperate(File file)
  { htmlfile = file;
  }
//-----

  public MyOperate(String tofile)
  { htmlfile = new File(tofile);
  }
//-----

```

```
private int getTagValue(int nowvalue,String tag)
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนสิทธิ์ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
for(int k=0; k<opentag.length; k++)
{ if(tag.equalsIgnoreCase(opentag[k]))
{ return nowvalue+value[k];
}
if(tag.equalsIgnoreCase(closetag[k]))
{ if(nowvalue>value[k])
{ nowvalue= nowvalue-value[k];
}
return nowvalue;
}
}
return nowvalue;
}
}
//-----
public int getValue(String keyword)
{ int keyvalue=0;
int tagvalue=2;
int i =0;

char input=' ';

String matchkey ="";
String matchtag="";
String matchmeta="";

boolean ismeta=false;
boolean istag = false;

for(int j=0; j<keyword.length(); j++)
{ matchkey = matchkey+" ";
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 try
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ FileReader fr = new FileReader(htmlfile);
  BufferedReader br = new BufferedReader(fr);
  while(i!=-1)
  { i = br.read();
    input = (char) i;
    matchkey = matchkey.substring(1)+input;

    if(input=='<') istag=true;

    if(matchkey.equals(keyword))
    { keyvalue+=tagvalue;
    }

    if(istag)
    { matchtag = matchtag+input;
      if(matchtag.equalsIgnoreCase("<meta")) // meta tag
      { ismeta = true; //
        tagvalue+=50;
      }
      if(ismeta) matchmeta=matchtag;
      if(input=='>')
      { if(ismeta)
        { ismeta=false; //check about meta tag
          tagvalue-=50;
        }

        istag = false;
        tagvalue = getTagValue(tagvalue,matchtag);
        matchtag=""; // CLEAR MATCHTAG
      }

    } // END IF(ISTAG)
  } // END WHILE (I!=-1)
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 } catch(IOException ioe)
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { System.out.println("error: "+ioe);
    }
    return keyvalue;
}
//-----

public int getSize()
{ return (int) htmlfile.length();
}
//-----

public String getTitle()
{ char input=' ';
  int i = 0;

  String title = "";
  String tag = "";

  boolean doing = true;
  boolean istag = false;
  boolean istitle = false;

  try
  { FileReader fr = new FileReader(htmlfile);
    BufferedReader br = new BufferedReader(fr);

    while( (doing)&&(i!--1) )
    { i = br.read();
      input = (char) i;

      if(input=='<')
      { istag=true;
        tag = "";
      }

      if(istag)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if(istitle)
    { title = title+input;
    }

    if(tag.equalsIgnoreCase("<title>"))
    { istitle = true;
    }

    if(tag.equalsIgnoreCase("</title>"))
    { istitle = false;
      doing = false;
    }

    if(tag.equalsIgnoreCase("</head>"))
    { doing = false;
    }

    if(input ==>')
    { istag = false;
    }

  }
  br.close();
} catch(IOException ioe)
{ System.out.println("error: "+ioe);
}

if (title.length()>=8)
{ title = title.substring(0,title.length()-8);

  if ( title.indexOf("\"") >=0 )
  { title = getDoubleQuote(title);
  }

  if(title.length()>100) title = title.substring(0,99);
  while( title.endsWith("\"") )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อแบบลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    title = title.replace('|','');
    title = title.replace('\r',' ');
    title = title.replace('\n',' ');
}

if(title.length()==0) title="NONE";

title = title.trim();

return title;

}

//-----
public String getDescription()
{ char input=' ';
  int i = 0;
  int count = 0;

  String description = "";
  String tag = "";

  boolean doing = true;
  boolean istag = false;
  boolean isdescription = false;
  boolean haveddes=true;
  boolean allow = true;

  try
  { FileReader fr = new FileReader(htmlfile);
    BufferedReader br = new BufferedReader(fr);

    while( (doing)&&(i!=-1) )
    { i = br.read();

      input = (char) i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(input=='<')
{ istag=true;
  tag = "";
}
if(istag)
{ tag = tag+input;
}
if(isdescription)
{ description = description+input;
}

if(tag.equalsIgnoreCase("<meta name=\"description\" content=\""))
{ isdescription = true;
  haveddes = true;
}
if( (isdescription) && (input=='>') )
{ isdescription = false;
  doing = false;
}
if (!(haveddes) )
{ if( (tag.startsWith("<script") || (tag.startsWith("<SCRIPT"))) )
  { allow = false;
  }
  if( tag.equalsIgnoreCase("</script>") )
  { allow = true;
  }
}

if( !(istag) )
{ if( (allow)&&(input!='\r')&&(input!='\n')&&(input!='\t') )
  { description = description+input;
    count++;
  }
}
if( (description.endsWith(" ")) || (description.endsWith("&nbsp;"))
|| (description.endsWith("&NBSP;")) )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        { description = description.substring(0,description.length()-5);
          count = count -5;
        }
    }
    if (count==200)
    { doing = false;
    }
}
}
if( (tag.equalsIgnoreCase("</head>"))&&(description.length()==0) )
{ haveddes = false;
}

if(input == '>')
{ istag = false;
}

}
br.close();
} catch(IOException ioe)
{ System.out.println("error: "+ioe);
}

if (description.length()==0)
{ description ="NONE";
}
else
{ if( description.indexOf("\") >=0 )
  { description = getDoubleQuote(description);
  }

  if(description.length(>200) description = description.substring(0,199);

  while( (description.endsWith("\"")) || (description.endsWith("\'")) ||
  (description.endsWith(">")) )

```

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { description = description.substring(0,description.length()-1);
    }
    while ( description.startsWith("\\"") )
    { description = description.substring(1);
    }
    description = description.replace('|',' ');
    description = description.replace('\r',' ');
    description = description.replace('\n',' ');
    }
    description = description.trim();
    return description;
}

//-----

private String getDoubleQuote(String sttest)
{ int from =0;
  int to = 0;
  String temp="";
  while((to=sttest.indexOf('"',from))>=0)
  { temp += sttest.substring(from,to) +"";
    from = to+1;
  }
  temp+=sttest.substring(from);
  return temp;
}

//-----

//===== END CLASS MyOperate =====

```

5. MyDatabase.java

```
package SEARCH;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.io.*;
import java.net.*;
import java.sql.*;
import java.util.*;

public class MyDatabase
{
    Statement stmt;
    Connection con;
//-----
    public MyDatabase()
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            String url = "jdbc:odbc:search";
            String user = "purekik";
            String password = "purekik";
            con = DriverManager.getConnection(url,
                user, password);
            stmt = con.createStatement();

        } catch(SQLException sqlx)
        {
            System.out.println("error: "+sqlx);
        }
        catch(java.lang.Exception ex)
        {
            System.out.println("exception error: "+ex);
        }
    }
//-----

    public void getClose() throws SQLException
    {
        stmt.close();
        con.close();
        System.out.println("Close database complete");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุตบแต่งสงวนเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ String command = "INSERT INTO KEY VALUES (";
  System.out.println("Start to insert to key");
  try
  { FileReader fr = new FileReader(f1);
    BufferedReader br = new BufferedReader(fr);
    String temp="";
    boolean have = true;
    while(have)
    { temp=br.readLine();
      while( temp!=null&&temp.equals("") )
      { temp=br.readLine();
      }
      if (temp==null) have=false;
      else
      { StringTokenizer st = new StringTokenizer(temp,"");
        String keyword="";
        String urlnumber="";
        String priority="";
        int count=1;
        while(st.hasMoreTokens())
        { if(count<4)
          { switch(count)
            { case 1:
              { keyword=st.nextToken().trim();
                count++;
                break;
              }
            case 2:
              { urlnumber=st.nextToken().trim();
                count++;
                break;
              }
            case 3:
              {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        { priority=st.nextToken().trim();
          count++;
          break;
        }
      } // end switch
    } // end if count
  } // end while hasmore

  int unum = Integer.parseInt(urlnumber);
  int priori = Integer.parseInt(priority);

  String command1 = command+ ""
    +keyword+ ","
    +unum+ "," +priori+ ";";
  stmt.executeUpdate(command1);
} // else
} // end while temp
br.close();
System.out.println("INSERT TO KEY TABLE SUCCESSFUL");
} catch(IOException ioe)
{ System.out.println("io error: "+ioe);
}
}
//-----
public void insertToDataTable(File f1) throws SQLException

{ String command = "INSERT INTO DATA VALUES (";
  System.out.println("Start to insert to data");
  try
  { FileReader fr = new FileReader(f1);
    BufferedReader br = new BufferedReader(fr);
    String temp="";
    boolean have = true;
    while(have)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ temp=br.readLine();
while( temp!=null&&temp.equals("") )
{ temp=br.readLine();
}
if (temp==null) have=false;
else
{ StringTokenizer st = new StringTokenizer(temp,"|");
String urlnumber="";
String urlname="";
String pagesize="";
String description="";
String title="";

int count=1;
while(st.hasMoreTokens())
{ if(count<6)
{ switch(count)
{ case 1:
{ urlnumber=st.nextToken().trim();
count++;
break;
}
case 2:
{ urlname=st.nextToken().trim();
count++;
break;
}
case 3:
{ pagesize=st.nextToken().trim();
count++;
break;
}
case 4:
{ description=st.nextToken().trim();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        count++;
        break;
    }
    case 5:
    { title= st.nextToken().trim();
      count++;
      break;
    }
} // end switch
} // ent if count

} // end while hasmore

int unum = Integer.parseInt(urlnumber);
int size = Integer.parseInt(pagesize);

String command2 = command+urlnumber+ "," +urlname
+ "," +pagesize+ "," +description + "," +title+ "));
stmt.executeUpdate(command2);

} // else
} // end while temp
br.close();
System.out.println("INSERT TO DATA TABLE SUCCESSFUL!!!");
} catch(IOException ioe)
{ System.out.println("io error: "+ioe);
}
}

//-----

public int insertToLinkTable(File fl,int linkorder) throws SQLException

{ String command = "INSERT INTO LINKLIST VALUES (";
  System.out.println("Start to insert to link");
  int order = linkorder;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

try
{
    FileReader fr = new FileReader(f1);
    BufferedReader br = new BufferedReader(fr);
    String temp="";
    boolean have = true;
    while(have)
    {
        temp=br.readLine();
        while( temp!=null&&temp.equals("") )
        {
            temp=br.readLine();
        }
        if (temp==null) have=false;
        else
        {
            temp = temp.trim();
            boolean redund = true;
            redund = checkRepeat(temp);
            if( (temp.indexOf("")== -1)&&!(redund))
            {
                String command3 = command+temp+ "," +order+ " ";
                stmt.executeUpdate(command3);
                order++;
            }
        }
    }
    br.close();
    System.out.println("INSERT TO LINK SUCCESS");
} catch(IOException ioe)
{
    System.out.println("insert to linklist error: "+ioe);
}

return order;
}
//-----
private boolean checkRepeat(String test) throws SQLException
{
    String linktable = "SELECT * FROM LINKLIST WHERE URLNAME =" +test+ """;
    String datatable = "SELECT * FROM DATA WHERE URLNAME =" +test+ """;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ResultSet rmlink = stmt.executeQuery(linktable);
```

```
ResultSet rsdata = stmt.executeQuery(datatable);
```

```
boolean isrepeat = false;
```

```
if( (rmlink.next()) || (rsdata.next()) )
```

```
{ isrepeat = true;
```

```
}
```

```
else isrepeat = false;
```

```
return isrepeat;
```

```
}
```

```
//-----
```

```
public ResultSet selectLink() throws SQLException
```

```
{ String command = "SELECT DISTINCT * FROM LINKLIST "+
```

```
"WHERE LINKNUMBER > 0 "+
```

```
"ORDER BY LINKNUMBER ASC";
```

```
ResultSet linkset = stmt.executeQuery(command);
```

```
System.out.println("select link complete");
```

```
return linkset;
```

```
}
```

```
//-----
```

```
public ResultSet getOperateKey() throws SQLException
```

```
{ String command = "SELECT DISTINCT KEYWORDS "+
```

```
"FROM SEARCHED_KEY "+
```

```
"WHERE KEYTYPE=0";
```

```
ResultSet linkset = stmt.executeQuery(command);
```

```
System.out.println("select operate key complete");
```

```
return linkset;
```

```
}
```

```
//-----
```

```
public ResultSet getUpdateKey() throws SQLException
```

```
{ String command = "SELECT DISTINCT KEYWORDS "+
```

```
"FROM SEARCHED_KEY "+
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        "WHERE KEYTYPE=1";

        ResultSet keyset = stmt.executeQuery(command);
        System.out.println("select update key complete");
        return keyset;
    }
}
//-----

public ResultSet getUpdateList(int up) throws SQLException
{
    String command = "SELECT DISTINCT * "+
        "FROM DATA "+
        "WHERE URLNUMBER > "+ up;

    ResultSet linkset = stmt.executeQuery(command);
    System.out.println("select update page list complete");
    return linkset;
}
//-----

public int getPageOrder() throws SQLException
{
    String command = "SELECT STATEVALUE "+
        "FROM STATE "+
        "WHERE STATETYPE='PAGEORDER'";

    ResultSet stateset = stmt.executeQuery(command);
    int pageorder = 0;
    if (stateset.next())
    {
        pageorder = stateset.getInt("STATEVALUE");
    }

    System.out.println("select page order complete");
    return pageorder;
}
//-----

public int getLinkOrder() throws SQLException
{
    String command = "SELECT STATEVALUE "+
        "FROM STATE "+
        "WHERE STATETYPE='LINKORDER'";

    ResultSet stateset = stmt.executeQuery(command);
    int linkorder = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( stateset.next() )
{ linkorder = stateset.getInt("STATEVALUE");
}
System.out.println("select link order complete");
return linkorder;
}
//-----
public int getUpdateNumber() throws SQLException
{ String command = "SELECT STATEVALUE "+
    "FROM STATE "+
    "WHERE STATETYPE='PAGEUPDATE'";
ResultSet stateset = stmt.executeQuery(command);
int update = 0;
if( stateset.next() )
{ update = stateset.getInt("STATEVALUE");
}
System.out.println("select update complete");
return update;
}
//-----
public void keepState(int page,int link) throws SQLException
{ String command1 = "UPDATE STATE "+
    "SET STATEVALUE = " + page +
    "WHERE STATETYPE = 'PAGEORDER'";
String command2 = "UPDATE STATE "+
    "SET STATEVALUE = " + link +
    "WHERE STATETYPE = 'LINKORDER'";
stmt.executeUpdate(command1);
stmt.executeUpdate(command2);
System.out.println("insert state to database complete");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
public void changeUpdateNumber(int update) throws SQLException
 ไม่ว่าจะผิดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ String command3 = "UPDATE STATE "+
    "SET STATEVALUE = " + update +
    "WHERE STATETYPE = 'PAGEUPDATE'";
stmt.executeUpdate(command3);
System.out.println("change update state to database complete");
}

```

//-----

```

public void setLinkToDelete(String linkname) throws SQLException
{ String command = "UPDATE LINKLIST "+
    "SET LINKNUMBER = 0 "+
    "WHERE URLNAME = '" +linkname+ "'";
stmt.executeUpdate(command);
System.out.println("set link to delete complete");
}

```

//-----

```

public void setKeyToDelete(String keyname) throws SQLException
{ String command = "UPDATE UNSEARCH_KEY "+
    "SET KEYTYPE = 0 "+
    "WHERE KEYWORDS = '" +keyname+ "'";
stmt.executeUpdate(command);
System.out.println("set key in unsearch to delete complete");
}

```

//-----

```

public void changeToCommonKey() throws SQLException
{ String command = "UPDATE SEARCHED_KEY "+
    "SET KEYTYPE = 0 "+
    "WHERE KEYTYPE = 1";
stmt.executeUpdate(command);
System.out.println("change key to common in searched key complete");
}

```

//-----

```

public void deleteFromLinkTable() throws SQLException
{ String command = "DELETE FROM LINKLIST WHERE LINKNUMBER=0";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        System.out.println("delete searched link complete");
    }
//-----

public void deleteUnSearchKey() throws SQLException
{ String command = "DELETE FROM UNSEARCH_KEY WHERE KEYTYPE=0";
  stmt.executeUpdate(command);
  System.out.println("delete unsearch key complete");
}
//-----

public int countUnSearchKey() throws SQLException
{ String command = "SELECT DISTINCT * FROM UNSEARCH_KEY";
  ResultSet rs = stmt.executeQuery(command);
  int count=0;
  while( rs.next() )
  { count++;
  }
  System.out.println("count unsearch key complete");
  return count;
}
//-----

public ResultSet getUnSearchKey() throws SQLException
{ String command = "SELECT DISTINCT * FROM UNSEARCH_KEY";
  ResultSet rs = stmt.executeQuery(command);
  System.out.println("get unsearch key for move complete");
  return rs;
}
//-----

//===== FOR TEST AND INITIAL DATABASE =====

public void insertToUnSearchKey(File f1) throws SQLException

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

System.out.println("Start to insert to unsearch_key table");
try
{
    FileReader fr = new FileReader(f1);
    BufferedReader br = new BufferedReader(fr);
    String temp="";
    boolean have = true;
    while(have)
    {
        temp=br.readLine();
        while( temp!=null&&temp.equals(""))
        {
            temp=br.readLine();
        }
        if (temp==null) have=false;
        else
        {
            temp = temp.trim();
            if( (temp.indexOf("")== -1) )
            {
                String command3 = command+temp+"",1";
                stmt.executeUpdate(command3);
            }
        }
    }
    br.close();
    System.out.println("INSERT TO UNSEARCH_KEY SUCCESS");
} catch(IOException ioe)
{
    System.out.println("insert to unsearch_key table error: "+ioe);
}
}

```

//-----

```
public void insertToSearchedKey(File f1) throws SQLException
```

```
{ String command = "INSERT INTO SEARCHED_KEY VALUES (";
```

```
System.out.println("Start to insert to searched key table");
```

```
try
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ FileReader fr = new FileReader(f1);
  BufferedReader br = new BufferedReader(fr);
  String temp="";
  boolean have = true;
  while(have)
  { temp=br.readLine();
    while( temp!=null&&temp.equals(""))
    { temp=br.readLine();
    }
    if (temp==null) have=false;
    else
    { temp = temp.trim();
      if( (temp.indexOf(" ")== -1)
        { String command3 = command+temp+ " ,0)";
          stmt.executeUpdate(command3);
        }
      }
    }
  br.close();
  System.out.println("INSERT TO SEARCHED_KEY SUCCESS");
} catch(IOException ioe)
{ System.out.println("insert to searched key table error: "+ioe);
}
}
//-----

public void insertToSearchedKey(String key,int type) throws SQLException
{
  String command = "INSERT INTO SEARCHED_KEY VALUES ("+
    key+ " ,"+
    type+ ")";
  stmt.executeUpdate(command);
  System.out.println("Complete to insert to searched key table");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 //-----
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public void initialLinkTable(File f1) throws SQLException
```

```
{ String command = "INSERT INTO LINKLIST VALUES (";
  System.out.println("Start to initial link table");
  try
  { FileReader fr = new FileReader(f1);
    BufferedReader br = new BufferedReader(fr);
    String temp="";
    boolean have = true;
    while(have)
    { temp=br.readLine();
      while( temp!=null&&temp.equals("") )
      { temp=br.readLine();
      }
      if (temp==null) have=false;
      else
      { StringTokenizer st = new StringTokenizer(temp,"|");
        String linkname="";
        String linkordertemp="";
        int count=1;
        while(st.hasMoreTokens())
        { if(count<3)
          { switch(count)
            { case 1:
              { linkname=st.nextToken().trim();
                count++;
                break;
              }
            case 2:
              { linkordertemp=st.nextToken().trim();
                count++;
                break;
              }
            }
          }
        }
      }
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }                // ent if count
}                // end while hasmore

int linkorder = Integer.parseInt(linkordertemp);

String command1 = command+ ""
    +linkname+ ","
    +linkorder+ ")";

stmt.executeUpdate(command1);
}                // else
}                // end while temp
br.close();
System.out.println("INITIAL LINK TABLE SUCCESSFUL");
} catch(IOException ioe)
{ System.out.println("io error: "+ioe);
}
}
//-----
}
//===== end class MyDatabase =====

```

6. SearchServlet.java

```

import javax.servlet.*;
import javax.servlet.http.*;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import oracle.jdbc.*;

```

```

public class SearchServlet extends HttpServlet

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ Connection con;
  Statement state;
//-----
public void init(ServletConfig config) throws ServletException
{ super.init(config);
  String url = "jdbc:oracle:thin:@PURE:1521:orcl";
  String user="purekik";
  String password="purekik";
  try
  { Class.forName("oracle.jdbc.driver.OracleDriver");
    con=DriverManager.getConnection(url,user,password);
    state = con.createStatement();

  } catch (Exception e)
  { printreport("checkpoint1",e);
  }
}
//-----
public void doPost(HttpServletRequest req,HttpServletResponse res)
  throws ServletException,IOException
{ res.setContentType("text/html");
  boolean ins = false;
  boolean only = false;

  try
  { OutputStream os = res.getOutputStream();
    OutputStreamWriter osw = new OutputStreamWriter(os,"cp874");
    PrintWriter toClient = new PrintWriter(osw);

    toClient.println("<html>");
    toClient.println("<head>");
    toClient.println("<title>Search Result from my Search engine</title>");
    toClient.println("</head>");

```

```

toClient.print("<body text=\"#000099\" bgcolor=\"#FFFFFF\" ");
toClient.println("alink=\"#FF00FF\" background=\"/snowbg.gif\" >");
toClient.print("<br><h3><b><a href=\"/searchpage.html\"> ");
toClient.print(" main page </a>&nbsp;&nbsp;&||&nbsp;&nbsp;&<a href=\"/realsearch.html\">");
toClient.print(" real time search </a> &nbsp;&nbsp;&||&nbsp;&nbsp;&<a href=\"/testlink.html\">");
toClient.println(" show link</a></b></h3>");
toClient.flush();

```

```

toClient.println("<br>");
toClient.println("<h1>RESULT FOR YOUR SEARCH</h1>");
toClient.println("<br>");

```

```

String value = req.getParameterValues("keyword")[0];
value = value.trim();

String matchtype = req.getParameterValues("matchtype")[0];
if( matchtype!=null )
{
    matchtype = matchtype.trim();

    if( matchtype.equalsIgnoreCase("yes") )
    {
        only = true;
    }
    else
    {
        only = false;
    }
}

```

```

String inserttype = req.getParameterValues("inserttype")[0];
if( inserttype!=null )
{
    inserttype = inserttype.trim();

```

```

    if( inserttype.equalsIgnoreCase("no") )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{
    ins = true;
}
}

StringBufferInputStream sbis = new StringBufferInputStream(value);
InputStreamReader isr = new InputStreamReader(sbis,"cp874");

int i = 0;
char c=' ';
String temp="";
while(i!=-1)
{
    i=isr.read();
    if(i!=-1)
    {
        c = (char) i;
        temp=temp+c;
    }
}

toClient.print("<i><big>You're searching for : </i></big><big>");
toClient.println("<big><font color='red'><b>"+temp+"</b></big></big></font><br>");
String searchQuery = "SELECT DISTINCT * "+
    "FROM KEY,DATA "+
    "WHERE KEY.URLNUMBER = DATA.URLNUMBER "+
    "AND KEY.KEYWORDS ";

String searchkey = "";

if( only )
{
    searchkey = searchQuery+ "= " +temp+ " "+
    "ORDER BY KEY.PRIORITY DESC";
}

else
{
    searchkey = searchQuery+ "LIKE '%" +temp+ "%' "+
    "ORDER BY KEY.PRIORITY DESC";
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

```

```
toClient.println("<h2>RESULTS ARE</h2>");

```

```
ResultSet rs = state.executeQuery(searchkey);

```

```
int count=1;

```

```
while(rs.next())

```

```
{

```

```
int priority = rs.getInt("PRIORITY");

```

```
String urlname = rs.getString("URLNAME");

```

```
String description = rs.getString("DESCRIPTION");

```

```
int pagesize = rs.getInt("PAGESIZE");

```

```
String title = rs.getString("TITLE");

```

```
toClient.println("<h2> "+count+ ". " + "<a href='"+urlname+"'">"+urlname+"</a></h2>");

```

```
toClient.print("<b><font color='red'"><big>RATING : </big></font></b>");

```

```
if( priority <= 20 )

```

```
{ toClient.println("<IMG SRC='/stars1.gif' border=0 height=12 width=64>");

```

```
}

```

```
if( (priority>20)&&(priority<=70) )

```

```
{ toClient.println("<IMG SRC='/stars2.gif' border=0 height=12 width=64>");

```

```
}

```

```
if( (priority>70)&&(priority<=120) )

```

```
{ toClient.println("<IMG SRC='/stars3.gif' border=0 height=12 width=64>");

```

```
}

```

```
if( (priority>120)&&(priority<=170) )

```

```
{ toClient.println("<IMG SRC='/stars4.gif' border=0 height=12 width=64>");

```

```
}

```

```
if( priority > 170 )

```

```
{ toClient.println("<IMG SRC='/stars5.gif' border=0 height=12 width=64>");

```

```
}

```

```
toClient.println("<br><b>SIZE :</b>"+pagesize+"<b> bytes</b>");

```

```
toClient.println("<br><b>TITLE : </b>"+title);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

toClient.println("<br><b>DESCRIPTION : </b>"+description);
count++;
}

if(count==1)
{ toClient.println("&nbsp;");
toClient.println("<center>");
toClient.println("<br><img SRC=\\\"/kik2.gif\" height=97 width=638>");
toClient.println("<br>&nbsp;");
toClient.println("<p><img SRC=\\\"/kik5.gif\" height=40 width=650>");
toClient.println("<p><img SRC=\\\"/pulleye.gif\" height=130 width=96></center>");

if( ins )
{ String checkupdate = "SELECT * FROM SEARCHED_KEY WHERE KEYWORDS = '"+
temp +"'";
ResultSet ck = state.executeQuery(checkupdate);
if( ck.next() )
{
}
else
{ String up = "INSERT INTO UNSEARCH_KEY VALUES ('"+
temp+ "',1)";
state.executeUpdate(up);
}
}
}
}

else
{ toClient.println("<br><center><h2><font color=\\\"red\\\">END SEARCH</h2></center>
</font>");
}

toClient.println("</body>");
toClient.println("</html>");

toClient.flush();
toClient.close();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    } catch(Exception exc)
    { printreport("checkpoint2",exc);
    }
} //end dopost
//-----

```

```

public void doGet(HttpServletRequest req,HttpServletResponse res)
    throws ServletException,IOException
{ res.setContentType("text/html");

```

```

    String command = "SELECT DISTINCT * "+
        "FROM KEY,DATA "+
        "WHERE KEY.URLNUMBER = DATA.URLNUMBER "+
        "AND KEY.KEYWORDS = ";

```

```

try

```

```

{ OutputStream os = res.getOutputStream();
  OutputStreamWriter osw = new OutputStreamWriter(os,"cp874");
  PrintWriter toClient = new PrintWriter(osw);

  toClient.println("<html>");
  toClient.println("<head>");
  toClient.println("<title>Search Result from my Search engine:Get method</title>");
  toClient.println("</head>");
  toClient.print("<body text=\"#000099\" bgcolor=\"#0FFFFFF\" ");
  toClient.println("alink=\"#FF00FF\" background=\"/snowbg.gif\" >");
  toClient.print("<br><h3><b><a href=\"/searchpage.html\"> ");
  toClient.print(" main page </a>&nbsp;&||&nbsp;&<a href=\"/realsearch.html\">");
  toClient.print(" real time search </a> &nbsp;&||&nbsp;&<a href=\"/testlink.html\">");
  toClient.println(" show link</a></b></h3>");
  toClient.flush();

```

```

  toClient.println("<br>");

```

```

  toClient.println("<h1>RESULT FOR YOUR SEARCH</h1>");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

toClient.println("<br>");
String value = req.getQueryString();
value = value.trim();

StringBufferInputStream sbis = new StringBufferInputStream(value);
InputStreamReader isr = new InputStreamReader(sbis,"cp874");

int i = 0;
char c=' ';
String temp="";
while(i!=-1)
{ i=isr.read();
  if(i!=-1)
  { c = (char) i;
    temp=temp+c;
  }
}
toClient.print("<i><big>You're searching for : </i></big>");
toClient.println("<big><big><font color='red'><b>" +temp+"</b></big></big></font><br>");

String searchkey = command+ "" +temp+ "" +
"ORDER BY KEY.PRIORITY DESC";

toClient.println("<h2>RESULTS ARE</h2>");

ResultSet rs = state.executeQuery(searchkey);

int count=1;

while(rs.next())
{
  int priority = rs.getInt("PRIORITY");
  String urlname = rs.getString("URLNAME");
  String description = rs.getString("DESCRIPTION");
}

```

```

int pagesize = rs.getInt("PAGESIZE");
String title = rs.getString("TITLE");
toClient.println("<h2>" +count+ ". " + "<a href=\""+urlname+ "\">"+urlname+"</a></h2>");

toClient.print("<b><font color=\"red\"><big>RATING : </big></font></b>");
if( priority <= 20 )
{ toClient.println("<IMG SRC=\"/stars1.gif\" border=0 height=12 width=64>");
}
if( (priority>20)&&(priority<=70) )
{ toClient.println("<IMG SRC=\"/stars2.gif\" border=0 height=12 width=64>");
}
if( (priority>70)&&(priority<=120) )
{ toClient.println("<IMG SRC=\"/stars3.gif\" border=0 height=12 width=64>");
}
if( (priority>120)&&(priority<=170) )
{ toClient.println("<IMG SRC=\"/stars4.gif\" border=0 height=12 width=64>");
}
if( priority > 170 )
{ toClient.println("<IMG SRC=\"/stars5.gif\" border=0 height=12 width=64>");
}

toClient.println("<br><b>SIZE : </b>"+pagesize+"<b> bytes</b>");
toClient.println("<br><b>TITLE : </b>"+title);
toClient.println("<br><b>DESCRIPTION : </b>"+description);
count++;
}
if(count==1)
{ toClient.println("&nbsp;");
toClient.println("<center>");
toClient.println("<br><img SRC=\"/kik2.gif\" height=97 width=638>");
toClient.println("<br>&nbsp;");
toClient.println("<p><img SRC=\"/kik5.gif\" height=40 width=650>");
toClient.println("<p><img SRC=\"/pulleye.gif\" height=130 width=96></center>");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else
        { toClient.println("<br><center><h2><font color='red'>END SEARCH</h2></center>
</font>");
        }

    toClient.println("</body>");
    toClient.println("</html>");
    toClient.flush();
    toClient.close();
} catch(Exception exc)
{ printreport("checkpoint A",exc);
}
}
//-----
public void printreport(String mes,Exception ee )
{ try
{ FileWriter fw = new FileWriter("d:\\project\\purelog1.txt",true);
BufferedWriter bw = new BufferedWriter(fw);
bw.write("have error at "+mes+" : "+ee.getMessage());
bw.newLine();
bw.flush();
bw.close();
} catch(IOException ioe)
{ System.out.println("have error while write error log.");
}
}

//-----
public String getServletInfo()
{ return "Search Engine Servlet";
}

//-----
public void destroy()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ try
  { con.close();
  } catch(Exception e)
  { printreport("checkpoint4",e);
  }
}
}
}
//===== END CLASS SEARCHSERVLET =====

```

7. SearchInfo.java

```

import javax.servlet.*;
import javax.servlet.http.*;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import search.*;
import oracle.jdbc.*;

public class SearchInfo extends HttpServlet
{ Connection con;
  Statement state;
//-----

  public void init(ServletConfig config) throws ServletException
  { super.init(config);

    String url = "jdbc:oracle:thin:@PURE:1521:orcl";
    String user="purekik";
    String password="purekik";

    try
    { Class.forName("oracle.jdbc.driver.OracleDriver");

```

เอกสารนี้เป็น con=DriverManager.getConnection(url,user,password); นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

state = con.createStatement();

} catch (Exception e)
{ printreport("checkpoint1",e);
}

}

//-----

public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
{ res.setContentType("text/html");
String mycheck = req.getQueryString();
if( mycheck!=null ) mycheck= mycheck.substring(mycheck.indexOf('=')+1);
String searchQuery = "SELECT DISTINCT * "+
"FROM ";
try
{ OutputStream os = res.getOutputStream();
OutputStreamWriter osw = new OutputStreamWriter(os,"cp874");
PrintWriter toClient = new PrintWriter(osw);

toClient.println("<html>");
toClient.println("<head>");
toClient.println("<title>KEYWORDS IN DATABASE</title>");
toClient.println("</head>");
toClient.print("<body text=\"\#000099\" bgcolor=\"\#0FFFFFF\" ");
toClient.println("<a link=\"\#FF00FF\" background=\"/snowbg.gif\" >");
toClient.print("<br><h3><b><a href=\"/searchpage.html\" > ");
toClient.print(" main page </a>&nbsp;&nbsp;&||&nbsp;&nbsp;&<a href=\"/realsearch.html\">");
toClient.print(" real time search </a> &nbsp;&nbsp;&||&nbsp;&nbsp;&<a href=\"/testlink.html\">");
toClient.println(" show link</a></b></h3>");
toClient.flush();

toClient.println("<br>");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

toClient.println("<h1>THESE ARE KEYWORDS IN <font color=\"red\" >"+mycheck+"</font>
</h1>");

toClient.println("<br>");

String searchkey = searchQuery+ mycheck;
ResultSet rs = state.executeQuery(searchkey);

int count=1;
while(rs.next())
{
String key = rs.getString("KEYWORDS");
toClient.print("<b>" +count+ ".</b>&nbsp;");
toClient.println("<a href=\"/servlet/SearchServlet?\" +key+ "\">" +key+ "</a><br>");
count++;
}
if(count==1)
{ toClient.println("<h2><font color=\"red\"><center> SORRY,DON'T HAVE KEY IN
DATABASE </h2></font></center>");
}
toClient.println("</body>");
toClient.println("</html>");
toClient.flush();
toClient.close();
} catch(Exception exc)
{ printreport("checkpoint2 ",exc);
}
} //end doGET
//-----

public void printreport(String mes,Exception ee )
{ try
{ FileWriter fw = new FileWriter("d:\\project\\SearchInfo.err",true);
BufferedWriter bw = new BufferedWriter(fw);
bw.write("have error at "+mes+" : "+ee.getMessage());
bw.newLine();
}
}

```

```

        bw.flush();
        bw.close();
    } catch(IOException ioe)
    { System.out.println("have error while write error log.");
    }
}

//-----
public String getServletInfo()
{ return "Search Information Engine Servlet";
}

//-----
public void destroy()
{ try
  { con.close();
  } catch(Exception e)
  { printreport("checkpoint4",e);
  }
}
}

```

8. SearchLink.java

```

import javax.servlet.*;
import javax.servlet.http.*;

```

```

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;

```

```

public class SearchLink extends HttpServlet
    implements SingleThreadModel

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----
public void init(ServletConfig config) throws ServletException
{
    super.init(config);
}
//-----

public void doGet(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException
{
    res.setContentType("text/html");

    boolean uerr = false;
    boolean cando = true;
    boolean malform = false;
    String urlname = req.getQueryString();
    urlname = urlname.toLowerCase();
    urlname = urlname.substring(urlname.indexOf('=')+1);
    if( urlname.length()== 0 )
    {
        uerr=true;
        cando = false;
    }
    else
    {
        int from =0;
        int to = 0;
        String temp="";
        while( (to=urlname.indexOf("%2f",from))>=0 )
        {
            temp += urlname.substring(from,to) +"/";
            from = to+3;
        }
        temp+=urlname.substring(from);
        if( temp.endsWith("%2f") )
        {
            temp = temp.substring(0,temp.length()-3)+" ";
        }
        from =0;
        to = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น ยกเว้นที่ มีมติเห็นชอบโดยคณะกรรมการ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { temp2 += temp.substring(from,to) +":";
      from = to+3;
    }

    temp2+=temp.substring(from);
    if( temp2.endsWith("%3a") )
    { temp2 = temp2.substring(0,temp.length()-3)+":";
    }

    urlname=temp2;
  }
  if( !uerr )
  { if( !(urlname.startsWith("http")) )
    { urlname="http://" +urlname;
    }
  }
  try
  { URL myu = new URL(urlname);
  }
  catch (MalformedURLException d)
  { malform = true;
    cando = false;
    printreport("checkpoint 1",d);
  }
  try
  { OutputStream os = res.getOutputStream();
    OutputStreamWriter osw = new OutputStreamWriter(os,"cp874");
    PrintWriter toClient = new PrintWriter(osw);

    toClient.println("<html>");
    toClient.println("<head>");
    toClient.println("<title>show link</title>");
    toClient.println("</head>");

    toClient.println("<body>");
    toClient.print("<body text=\"\#000099\" bgcolor=\"\#FFFFFFF\"");

```

```

toClient.println("alink=#FF00FF" background="/snowbg.gif" >");
toClient.print("<br><h3><b><a href='/searchpage.html'> ");
toClient.print(" main page </a>&nbsp;&nbsp;&||&nbsp;&nbsp;&<a href='/testlink.html'> ");
toClient.print("show link again </a>&nbsp;&nbsp;&||&nbsp;&nbsp;&<a href='/realsearch.html'>");
toClient.println("real time search</a></b></h3>");

```

```

toClient.println("<h1><font color='red'>THIS SERVLET SHOW FIND LINK FROM PAGE
</font></h1>");

```

```

if( uerr )

```

```

{ toClient.print("<b><big><big>SORRY, YOU DONT INSERT URL. PLEASE &nbsp;&nbsp;&");
  toClient.println("<a href='/testlink.html'> TRY AGAIN</a></b></big></big><br>");
}

```

```

if( !(uerr)&&(malform) )

```

```

{ toClient.print("<b><big>SORRY, YOUR URL IS MALFORMED. PLEASE &nbsp;&nbsp;&");
  toClient.println("<a href='/testlink.html'> TRY AGAIN</a></b></big><br>");
}

```

```

if ( cando )

```

```

{
  MyPageFile mypage = new MyPageFile(urlname);
  if( mypage.getConnectionErrorFlag() )
  { toClient.println("<h1><font color='red'> SORRY! CONNECTION ERROR</font></h1>");
  }
}

```

```

else

```

```

{ if( !(mypage.isUsable() )

```

```

  { toClient.print("<b><big><big><font color='red'>THIS PAGE IS NOT USABLE.

```

```

PLEASE &nbsp;&nbsp;&</font>");

```

```

  toClient.println("<a href='/testlink.html'> TRY AGAIN</a></b></big></big><br>");

```

```

}

```

```

else

```

```

{ toClient.print("SEARCHING FROM &nbsp;&nbsp;&<big><big><font color='red'>");

```

```

  toClient.println(urlname+"</big></big></font><br>");

```

```

  toClient.flush();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MyLink mylink = new MyLink(urlname,mypage.getURLFile());
FileReader fr = new FileReader(mylink.getLinkFile());
BufferedReader br = new BufferedReader(fr);
String mytemp = "";
while( (mytemp=br.readLine())!=null )
{ if( !(mytemp.equals("")) )
{ toClient.println("<LI><h3><a href=\""+mytemp+ "\">");
toClient.println(mytemp+ "</h3></a>");
toClient.flush();
}
}
}

}

toClient.println("<center><h2><font color=\"red\">END PAGE</h2></center></font>");
}
toClient.println("</body>");
toClient.println("</html>");
toClient.flush();
toClient.close();
} catch(Exception exc)
{ printreport("checkpoint2 ",exc);
}
} //end doGET
//-----

```

```

public void printreport(String mes,Exception ee )
{ try
{ FileWriter fw = new FileWriter("d:\\project\\SearchLink.err",true);
BufferedWriter bw = new BufferedWriter(fw);
bw.write("have error at "+mes+" : "+ee.getMessage());
bw.newLine();
bw.flush();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { System.out.println("have error while write error log.");
    }
}

//-----

public String getServletInfo()
{ return "This Servlet show how to find link from URL";
}

//-----

public void destroy()
{
}

}

//===== END CLASS SEARCHLINK =====

class MyPageFile
{ URL myurl;
  String urlname;
  File myfile = new File("ttlink.tmp");
  boolean errorflag;
}

//-----

public MyPageFile(String url)
{ clearFile();
  errorflag = false;
  urlname= url;
  getToFile();
}

//-----

public File getURLFile()
{ return myfile;
}

//-----

public boolean isUsable()
{ boolean doit=true;

```

เอกสารนี้เป็นเอกสารที่ลงลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

boolean page = false;
boolean content = false;

String temp = "";
String lower = "";
try
{
    FileReader fr = new FileReader(myfile);
    BufferedReader br = new BufferedReader(fr);
    while(doit)
    {
        temp=br.readLine();
        if( temp==null )
        {
            doit = false;
        }
        else
        {
            lower = temp.toLowerCase().trim();
            if( lower.startsWith("<html>") )
            {
                doit = false;
            }
            if( lower.startsWith("http/1.0 2") )
            {
                page = true;
            }
            if( (lower.equals("content-type: text/html")) ||
                (lower.equals("content-type: text/plain")) )
            {
                content = true;
            }
        }
    }
}
br.close();
} catch(IOException ioe)
{
    System.out.println("error: "+ioe);
}

System.out.println("check page complete");

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----
public synchronized void getToFile()
{
    InputStreamReader in;
    BufferedReader br;
    FileWriter fw;
    BufferedWriter bw;

    try
    {
        Socket socket = new Socket("proxy.ce.kmitl.ac.th",3128);
        socket.setSoTimeout(120000);

        OutputStream rawOut = socket.getOutputStream();
        InputStream rawIn = socket.getInputStream();

        DataOutputStream dos = new DataOutputStream(rawOut);
        dos.writeBytes("Get "+urlname+ " HTTP/1.0\r\n\r\n");
        in = new InputStreamReader(rawIn);
        br = new BufferedReader(in);
        fw = new FileWriter(myfile);
        bw = new BufferedWriter(fw);
        String temp="";
        while( (temp=br.readLine())!=null )
        {
            bw.write(temp);
            bw.newLine();
            bw.flush();
        }
        bw.close();
    } catch(IOException ioe)
    {
        System.out.println("getToFile error: "+ioe);
        errorflag=true;
        clearFile();
    }
}
//-----
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น หากท่านมีให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
//-----
private void clearFile()
{ try
  { FileWriter fw1 = new FileWriter(myfile);
    BufferedWriter bw1 = new BufferedWriter(fw1);
    bw1.write("");
    bw1.flush();
    bw1.close();
  } catch(IOException ioe)
  { System.out.println("clear file error: "+ioe);
  }
}
}
//===== end class MyPageFile =====

class MyLink
{ File pagefile;
  File lin = new File("tlink.lst");
  String urlname;
  String urlbase;
  String urlhost;

//-----

public MyLink(String uname,File myfile)
{ clearFile();
  pagefile = myfile;
  urlname =uname;
  getLink();
}

//-----

public File getLinkFile()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าวิธีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----
private String getDeciseBase(String oldbase)
{ String newbase = "";
  if( oldbase.endsWith("/") )
  { newbase = oldbase;
  }
  else
  { int from = oldbase.lastIndexOf("/");
    int comma = oldbase.indexOf(".",from+1);
    if (from >10)
    { if( comma!= -1 )
      { newbase = oldbase.substring(0,from+1);
      }
      else
      { newbase = oldbase+"/";
      }
    }
    else newbase = oldbase+"/";
  }
  return newbase;
}
}
//-----
```

```
private void getLink()
{ boolean islink = false;
  boolean have = true;

  int i=0;

  char input;

  String link="";
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

urlbase = getBase();
if( urlbase==null ) urlbase = urlname;

urlbase = getDeciseBase(urlbase);

int slash = urlbase.indexOf("/",10) ;
if( slash!= -1 ) urlhost = urlbase.substring(0,slash);
else urlhost = urlbase;

```

```

try
{
  FileReader fr = new FileReader(pagefile);
  BufferedReader br = new BufferedReader(fr);
  readlink:
  while(have)
  {
    i = br.read();
    if(i==32) i=br.read();
    if(i!=-1)
    {
      input = (char) i;
      compare = compare.substring(1,2)+input;

      if(compare.equalsIgnoreCase("<a"))
      {
        islink = true;
      }
      //end if(compare...

```

```

while(islink)
{
  i = br.read();
  if(i==32) i=br.read();
  if(i== -1) break readlink;
  input = (char) i;
  refer = refer.substring(1,5)+input;
  if(refer.equalsIgnoreCase("href="))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

boolean end = true;
i=br.read();
if(i==1) break readlink;
while((i==32)||(i==34)||(i==13))
{ if(i==32) i=br.read();
  if(i==13)
  { i=br.read();
    i=br.read();
  }
  if(i==34)
  { countfun++;
    i=br.read();
    if(countfun==2) end=false;
  }
  if(i==1) break readlink;
}
input = (char) i;
while(end)
{ link = link+input;
  i = br.read();
  if(i==1) break readlink;
  input = (char) i;
  if( (i==32)||(i==34)||(input=='>') ) end=false;
} // end while(input...)
boolean check = link.equals("");
if(!check)
{ writeFile(link);
}
islink=false;
} // end if(refer...)
} // end while(islink)
} //end if
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { have = false;
      }
    // end else if
  }
  // end while(have)

  br.close();
}
// end try
catch(IOException ioe)
{ System.out.println(ioe);
}
}
// end main

```

```

//-----
private void writeFile(String llist)
{ llist = llist.toLowerCase();
  String temp = "";
  boolean yes = true;
  try
  { FileWriter fw = new FileWriter("ftlink.lst",true);
    BufferedWriter bw = new BufferedWriter(fw);

    if( llist.startsWith("http://") )
    {
      temp=llist;
      if( yes )
      { if( (llist.endsWith(".exe")) || (llist.endsWith(".gif")) ||
          (llist.endsWith(".zip")) || (llist.endsWith(".jpg")) )
        { yes = false;
        }
      }
    }
  }
  else
  { if( llist.indexOf(":")==-1 )
    { if( llist.startsWith("/") )
      { temp = urlhost+llist;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ if( llist.startsWith("../") )
  { int j = urlbase.lastIndexOf("/",urlbase.length()-2);
    temp = urlbase.substring(0,j)+llist.substring(2);
  }
  else temp = urlbase+llist;
}
}
else
{ yes = false;
}
}
if( (llist.indexOf(".asp?")!=-1 )
{ yes = false;
}
if( yes )
{ if( temp.indexOf("#") != -1 )
  { temp = temp.substring(0, temp.indexOf("#") );
  }
temp = temp.trim();
if( !(temp.endsWith("/") ) )
{ int k = temp.lastIndexOf("/");
  if( k<10 )
  { temp = temp+"/";
  }
  else
  { int from = temp.indexOf(".",k);
    if( from == -1 )
    { temp = temp+"/";
    }
  }
}
}
}
bw.write(temp);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        bw.close();
    }
}
catch(IOException ioe)
{ System.out.println(ioe);
}
}
//-----
private String getBase()
{ int i=0;

char input = '';

String tag ="";
String base="";

boolean doit= true;
boolean nothavebase = true;
boolean istag = false;
boolean isbase = false;

try
{ FileReader fr = new FileReader(pagefile);
  BufferedReader br = new BufferedReader(fr);

while( (doit)&&(nothavebase) )
{ i = br.read();
  input = (char) i;
  if( i== -1 )
  { doit = false;
  }

  if( input=='<' )
  { istag = true;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( istag )
{ tag = tag+input;
}
if( isbase )
{ base = base +input;
}
if( tag.equalsIgnoreCase("<base") )
{ isbase = true;
}
if( tag.equalsIgnoreCase("</head>") )
{ doit = false;
}
if( input=='>' )
{ istag = false;
tag ="";
}
if( (isbase)&&(input=='>' ) )
{ nothavebase = false;
}
} //END while
} catch(IOException ioe)
{ System.out.println("error: "+ioe);
}

```

```

if( base.length()>0 )
{ base = base.trim();
String temp = base.toLowerCase();
int k= temp.indexOf("href=");
if( k!=-1 )
{
base = base.substring(k+5);
base = base.trim();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น ยกเว้นที่ ไม่มีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

base = base.replace('\n',' ');

StringTokenizer st = new StringTokenizer(base);
if( st.hasMoreTokens() )
{ base = st.nextToken();
}

while( base.startsWith("\"") )
{ base = base.substring(1);
  base = base.trim();
}
while( (base.endsWith("\"") || (base.endsWith(">"))) )
{ base = base.substring(0,base.length()-1);
  base = base.trim();
}
}
else base = null;
if( (base!=null) && (!(base.startsWith("http://"))) &&
    (!(base.startsWith("HTTP://"))) )
{ base = null;
}
}
else base = null;

return base;
}

```

```
//-----
```

```

private void clearFile()
{
try
{ FileWriter fw = new FileWriter(lin);
  BufferedWriter bw = new BufferedWriter(fw);

```

```
    bw.write("");
```

```
    bw.flush();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        bw.close();
    }
    catch(IOException ioe)
    { System.out.println(ioe);
    }
}
}
//===== END CLASS MYLINK =====

```

9. RealSearch.java

```

import javax.servlet.*;
import javax.servlet.http.*;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import oracle.jdbc.*;

public class RealSearch extends HttpServlet
    implements SingleThreadModel
{
//-----

    public void init(ServletConfig config) throws ServletException
    { super.init(config);
    }

//-----

    public void doPost(HttpServletRequest req,HttpServletResponse res)
        throws ServletException,IOException
    { res.setContentType("text/html");

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะในรูปแบบใดก็ตาม หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูงและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
}

```

```
String value = req.getParameterValues("key")[0];
```

```
value = value.trim();
```

```
String num = req.getParameterValues("count")[0];
```

```
if( num!=null )
```

```
{ num = num.trim();
```

```
try
```

```
{ turn = Integer.parseInt(num);
```

```
} catch(NumberFormatException n)
```

```
{ numerr=true;
```

```
printreport("checkpoint1",n);
```

```
}
```

```
}
```

```
StringBufferInputStream sbis = new StringBufferInputStream(value);
```

```
InputStreamReader isr = new InputStreamReader(sbis,"cp874");
```

```
int i = 0;
```

```
char c=' ';
```

```
String temp="";
```

```
while(i!=-1)
```

```
{ i=isr.read();
```

```
if(i!=-1)
```

```
{ c = (char) i;
```

```
temp=temp+c;
```

```
}
```

```
}
```

```
keyword=temp;
```

```
if( numerr )
```

```
{ toClient.println("<h1><font color='red'>SORRY, YOU ENTER WRONG ARGUMENTS
```

```
</h1></font>");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

else

```
{ toClient.print("<i><big>You're searching for : </i></big><big>");
toClient.print("<big><font color='red'><b>&nbsp;</b>"+keyword);
toClient.print("&nbsp; within &nbsp;  " + turn+ "&nbsp; page</b></big></big></font><br>");
toClient.flush();
```

try

```
{ RealDatabase real1 = new RealDatabase();
ResultSet rs1 = real1.getInSearchKey(keyword);
```

```
if( rs1.next() )
```

```
{ haveinsearch=true;
}
```

else

```
{ haveinsearch = false;
}
```

```
real1.getClose();
```

```
} catch(SQLException s1)
```

```
{ printreport("checkpoint2",s1);
}
```

```
if( haveinsearch )
```

```
{ toClient.print("<b><big><big>Your key have in our database, to view it click");
```

```
toClient.println(" &nbsp;<a href='\"/servlet/SearchServlet?\" +keyword+ \"'>here</a></b>
```

```
</big></big>");
```

```
}
```

else

```
{ try
```

```
{ RealDatabase real2 = new RealDatabase();
```

```
ResultSet rs2 = real2.getInRealSearch(keyword);
```

```
if( rs2.next() )
```

```
{ haveinrealsearch=true;
```

```
current = rs2.getInt("UPCURRENT");
```

```
}
```

```
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        current = order ;
    }          // end if mypage is usable
else
{ toClient.print("<font color=\"black\">");
  toClient.println("THIS PAGE IS NOT USABLE</font><br>");
  toClient.flush();
}

if( counter == turn )
{ break ENDING;
}
counter++;
}          // end while updatelist.next();
real3.getClose();
} catch(SQLException s3)
{ printreport("checkpoint4",s3);
}

if( current+1==max )
{ try
{ RealDatabase real6 = new RealDatabase();
  real6.insertToSearchedKey(keyword,0);
  real6.deleteFromRealSearch(keyword);
  real6.getClose();
} catch(SQLException s6)
{ printreport("checkpoint7",s6);
}

toClient.print("<br><h2><font color=\"red\">");
toClient.println("EMPTY LINK IN DATABASE!!!</h2></font>");
toClient.flush();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ try
  { RealDatabase real7 = new RealDatabase();
    real7.insertToRealSearch(keyword,current);
    real7.getClose();
  } catch(SQLException s7)
  { printreport("checkpoint8",s7);
  }
}
else
{ try
  { RealDatabase real8 = new RealDatabase();
    real8.changeUpCurrent(keyword,current);
    real8.getClose();
  } catch(SQLException s8)
  { printreport("checkpoint9",s8);
  }
}
} //END IF CURRENT+1== MAX
} //END ELSE IF HAVEINSEARCH
} //END ELSE IF NUMERR
toClient.println("<br>");
if( !(haveinsearch)&&!(numerr) )
{ toClient.println("<h2><font color=\"red\">OPERATION COMPLETE.</font></h2><b><big>
click &nbsp;<a href=\"/servlet/SearchServlet?\">
  toClient.println(keyword+\"><big> here </big> </a>&nbsp;<b><big><br>");
}
toClient.println("</body>");
toClient.println("</html>");
toClient.flush();
toClient.close();
} catch(Exception exc)
{ printreport("checkpoint2",exc);

```

เอกสารนี้เป็นลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} //end dopost
//-----
public void printreport(String mes,Exception ee )
{ try
  { FileWriter fw = new FileWriter("realsearch.err",true);
    BufferedWriter bw = new BufferedWriter(fw);
    bw.write("have error at "+mes+" : "+ee.getMessage());
    bw.newLine();
    bw.flush();
    bw.close();
  } catch(IOException ioe)
  { System.out.println("have error while write error log.");
  }
}
//-----
public String getServletInfo()
{ return "Real Time Search Engine Servlet";
}
//-----
public void destroy()
{
}
}
//===== END CLASS REALSEARCH =====

```

```
class RealDatabase
```

```
{ Statement stmt;
```

```
Connection con;
```

```
//-----
```

```
public RealDatabase()
```

```
{ try
```

```
{ Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
String url = "jdbc:oracle:thin:@PURE:1521:orcl";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

String user = "purekik";
String password = "purekik";
con = DriverManager.getConnection(url, user, password);
stmt = con.createStatement();

} catch(SQLException sqlx)
{ System.out.println("error: "+sqlx);
}
catch(java.lang.Exception ex)
{ System.out.println("exception error: "+ex);
}
}
}
//-----
public void getClose() throws SQLException
{ stmt.close();
con.close();
System.out.println("Close database complete");
}
//-----
public ResultSet getInSearchKey(String key) throws SQLException
{ String command = "SELECT * FROM SEARCHED_KEY WHERE KEYWORDS = '"+key+"'";
return stmt.executeQuery(command);
}
//-----
public ResultSet getInRealSearch(String key) throws SQLException
{ String command = "SELECT * FROM REALSEARCH WHERE KEYWORDS = '"+key+"'";
return stmt.executeQuery(command);
}
//-----
public ResultSet getUpdateList(int up) throws SQLException
{ String command = "SELECT DISTINCT * "+
"FROM DATA "+
"WHERE URLNUMBER > "+ up;
ResultSet linkset = stmt.executeQuery(command);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะในรูปแบบใดๆ ทั้งสิ้น อีกทั้งยังมีเนื้อหาที่ละเอียดอ่อน และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return linkset;
}
//-----
public int getMaxOrder() throws SQLException
{ String command = "SELECT STATEVALUE "+
    "FROM STATE "+
    "WHERE STATETYPE='PAGEORDER'";

    ResultSet stateset = stmt.executeQuery(command);
    int pageorder = 0;
    if( stateset.next() )
    { pageorder = stateset.getInt("STATEVALUE");
    }
    return pageorder;
}
//-----
public void insertToKey(String key,int unum,int pri) throws SQLException
{
    String command = "INSERT INTO KEY VALUES (" +
        key+ "," +
        unum+ "," +pri+ ")";
    stmt.executeUpdate(command);
}
//-----
public void insertToSearchedKey(String key,int type) throws SQLException
{
    String command = "INSERT INTO SEARCHED_KEY VALUES (" +
        key+ "," +
        type+ ")";
    stmt.executeUpdate(command);
}
//-----
public void deleteFromRealSearch(String key) throws SQLException
{ String command = "DELETE FROM REALSEARCH WHERE KEYWORDS=" +key+ """;
    stmt.executeUpdate(command);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
//-----
public void insertToRealSearch(String key,int cur) throws SQLException
{
    String command = "INSERT INTO REALSEARCH VALUES ("+
        key+ ","+
        cur+ ")";
    stmt.executeUpdate(command);
}
//-----
public void changeUpCurrent(String key,int cur) throws SQLException
{
    String command3 = "UPDATE REALSEARCH "+
        "SET UPCURRENT = "+ cur +
        " WHERE KEYWORDS = '"+key+"'";
    stmt.executeUpdate(command3);
}
}
}
//===== END CLASS REALDATABASE =====

class MyOp
{
    File htmlfile;
    String[] opentag = {"<comment>","<title>","<big>","<i>","<b>","<u>","<strong>","<em>","<h1>","
<h2>","<h3>","<h4>","<h5>","<h6>"};
    String[] closetag = {"</comment>","</title>","</big>","</i>","</b>","</u>","</strong>","</em>","
</h1>","</h2>","</h3>","</h4>","</h5>","</h6>"};
    int[] value = {0,30,7,7,7,7,7,20,17,14,11,9,8};
//-----
    public MyOp(File file)
    {
        htmlfile = file;
    }
//-----
    public MyOp(String tofile)
    {
        htmlfile = new File(tofile);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----
private int getTagValue(int nowvalue,String tag)
{
    for(int k=0; k<opentag.length; k++)
    { if(tag.equalsIgnoreCase(opentag[k]))
        { return nowvalue+value[k];
        }
    if(tag.equalsIgnoreCase(closetag[k]))
    { if(nowvalue>value[k])
        { nowvalue= nowvalue-value[k];
        }
    return nowvalue;
    }
}
return nowvalue;
}
//-----
```

```
public int getValue(String keyword)
{ int keyvalue=0;
  int tagvalue=2;
  int i =0;

  char input=' ';

  String matchkey ="";
  String matchtag="";
  String matchmeta="";

  boolean ismeta=false;
  boolean istag = false;

  for(int j=0; j<keyword.length(); j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

try
{
    FileReader fr = new FileReader(htmlfile);
    BufferedReader br = new BufferedReader(fr);
    while(i!=-1)
    {
        i = br.read();
        input = (char) i;
        matchkey = matchkey.substring(1)+input;
        if(input=='<') istag=true;
        if(matchkey.equals(keyword))
        {
            keyvalue+=tagvalue;
        }
        if(istag)
        {
            matchtag = matchtag+input;
            if(matchtag.equalsIgnoreCase("<meta")) // meta tag
            {
                ismeta = true; //
                tagvalue+=50;
            }
            if(ismeta) matchmeta=matchtag;
            if(input=='>')
            {
                if(ismeta)
                {
                    ismeta=false; //check about meta tag
                    tagvalue-=50;
                }
                istag = false;
                tagvalue = getTagValue(tagvalue,matchtag);
                matchtag=""; // CLEAR MATCHTAG
            }
        }
        // END IF(ISTAG)
    }
    // END WHILE (I!=-1)
    br.close();
} catch(IOException ioe)
{
    System.out.println("error: "+ioe);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
//-----
}
//===== END CLASS MyOp =====

class MyP
{ URL myurl;
  String urlname;
  File myfile = new File("real.tmp");
  boolean errorflag;
//-----

public MyP(String url)
{ clearFile();
  errorflag = false;
  urlname= url;
  getToFile();
}
//-----

public File getURLFile()
{ return myfile;
}
//-----

public boolean isUsable()
{ boolean doit=true;
  boolean page = false;
  boolean content = false;
  String temp = "";
  String lower="";
  try
  { FileReader fr = new FileReader(myfile);
    BufferedReader br = new BufferedReader(fr);
    while(doit)
    { temp=br.readLine();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
else
{ lower = temp.toLowerCase().trim();
  if( lower.startsWith("<html>") )
  { doit = false;
  }
  if( lower.startsWith("http/1.0 2") )
  { page = true;
  }
  if( (lower.equals("content-type: text/html")) ||
      (lower.equals("content-type: text/plain")) )
  { content = true;
  }
}
}
br.close();
} catch(IOException ioe)
{ System.out.println("error: "+ioe);
}
System.out.println("check page complete");
return (page&&content);
}
//-----
public synchronized void getToFile()
{ InputStreamReader in;
  BufferedReader br;
  FileWriter fw;
  BufferedWriter bw;
  try
  { Socket socket = new Socket("proxy.ce.kmitl.ac.th",3128);
    socket.setSoTimeout(120000);
    OutputStream rawOut = socket.getOutputStream();
    InputStream rawIn = socket.getInputStream();

```

```

DataOutputStream dos = new DataOutputStream(rawOut);
dos.writeBytes("Get "+urlname+ " HTTP/1.0\r\n\r\n");
in = new InputStreamReader(rawIn);
br = new BufferedReader(in);
fw = new FileWriter(myfile);
bw = new BufferedWriter(fw);
String temp="";
while( (temp=br.readLine())!=null )
{
    bw.write(temp);
    bw.newLine();
    bw.flush();
}
bw.close();
} catch(IOException ioe)
{
    System.out.println("getToFile error: "+ioe);
    errorflag=true;
    clearFile();
}
}
//-----
public boolean getConnectionErrorFlag()
{
    return errorflag;
}
//-----
private void clearFile()
{
    try
    {
        FileWriter fw1 = new FileWriter(myfile);
        BufferedWriter bw1 = new BufferedWriter(fw1);
        bw1.write("");
        bw1.flush();
        bw1.close();
    } catch(IOException ioe)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

}

//===== end class MyP =====



.เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. CAY S. HORSTMANN, GARY CORNELL, " CORE JAVA 1.1 VOLUME I ", SUNSOFT PRESS, 1998.
2. CAY S. HORSTMANN, GARY CORNELL, " CORE JAVA 1.1 VOLUME II ", SUNSOFT PRESS, 1998.
3. CHRIS TAYLOR, TIM KIMMET, " CORE JAVA WEB SERVER ", PRENTICE HALL, 1999.
4. BERNARD VAN HAECKE, " JDBC, JAVA DATABASE CONNECTIVITY ", IDG BOOKS, 1998.
5. RICK DARNELL, ET AL., " HTML4 UNLEASHED SECOND EDITION ", SAMS PUBLISHING, 1999.
6. PETER VAN DER LINDEN, " JUST JAVA 1.1 THIRD EDITION ", SUNSOFT PRESS, 1998.
7. C. J. DATE, " AN INTRODUCTION TO DATABASE CONCEPT. SIXTH EDITION ", ADDISON-WESLEY, 1995.
8. KEN ARNOLD, JAMES GOSLING, " THE JAVA PROGRAMMING LANGUAGE ", ADDISON-WESLEY, 1996.
9. MERLIN AND CONRAD HUGHES, MICHAEL SHOFFNER, MARIA WINSLOW, " JAVA NETWORK PROGRAMMING ", MANNING PUBLICATIONS, 1997.
10. DAVID FLANAGAN, " JAVA IN A NUTSHELL ", O'REILLY & ASSOCIATES, 1997.
11. RICK DECKER, STUART HIRSHFIELD, " PROGRAMMING.JAVA, AN INTRODUCTION TO PROGRAMMING USING JAVA ", PWS PUBLISHING, 1998.