

การพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสาร XML  
ด้วยไวยากรณ์ XQuery ผ่านเว็บ

The Development of a Web-Based XML Query Program using XQuery



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคเรียนที่ 2 ปีการศึกษา 2546  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วัน เดือน ปี.....	06.11.2550
เลขทะเบียน.....	02102
เลขเรียกหนังสือ.....	วท. ๗๖๖๓ ๕๔๖
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

ชื่อหัวข้อ	การพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสาร XML ด้วยไวยากรณ์ XQuery ผ่านเว็บ
นักศึกษา	นางสาวชนิษฐสรณ์ เวียรศิลป์
อาจารย์ที่ปรึกษา	ดร. ภัทรชัย ลลิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

### บทคัดย่อ

จากที่ XML ได้เข้ามามีบทบาท และได้กลายเป็นมาตรฐานในการแลกเปลี่ยนข้อมูลระหว่างองค์กร ทำให้ในวันนี้ได้มีมาตรฐานใหม่ที่ใช้ในการค้นหาข้อมูลภายในเอกสาร XML นั่นคือ XQuery ซึ่งการค้นหานี้จะมิใช่การค้นหาภายในฐานข้อมูลอีกต่อไป แต่จะเป็นการค้นหาข้อมูลจากไฟล์ที่มีเพียงแต่ข้อความที่เป็นตัวอักษรเท่านั้น ซึ่งจุดประสงค์ของการพัฒนาโปรแกรมค้นหาข้อมูลภายในเอกสาร XML ก็เพื่อที่จะให้ผู้ใช้สามารถทำงานได้โดยไม่ต้องอาศัยการค้นหาด้วยระบบฐานข้อมูล ออกแบบการใช้งานให้เหมาะกับผู้ใช้ระดับนักพัฒนา และผู้ใช้ทั่วไป (ไม่มีความรู้เกี่ยวกับไวยากรณ์ XQuery) โดยมีหน้าจอช่วยเหลือการใช้งานสำหรับผู้ทั่วไป และสามารถใช้งานได้ผ่านเครือข่ายอินเทอร์เน็ต

<b>Title</b>	The Development of a Web-Based XML Query Program using XQuery
<b>Student</b>	Miss Kanitsorn Vearasilp
<b>Advisor</b>	Dr. Pattarachai Lalitrojwong
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	2003



## ABSTRACT

As we know, XML has played a significant role as a standard on data interchange between organizations. Today there is a new standard for searching data in XML documents called XQuery. By this method, the search is performed with text files only, not databases. This project proposes the development of a web-based XML query program using XQuery. Its objective is to enable users search for desired data in XML documents with no need of a database. It is designed for both skilled developers and non-skilled to work through the Internet with a friendly user interface.

# สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
สารบัญ .....	III
สารบัญภาพ .....	V
บทที่	
1. บทนำ .....	1
1.1. ความเป็นมา และความสำคัญของปัญหา .....	1
1.2. วัตถุประสงค์ .....	2
1.3. ขอบเขตของโครงการพัฒนาระบบงาน .....	2
1.4. ประโยชน์ที่คาดว่าจะได้รับ .....	2
1.5. ขั้นตอนการดำเนินงาน .....	3
1.6. เครื่องมือที่ใช้ในการดำเนินงาน .....	3
2. ทฤษฎีที่เกี่ยวข้องกับการพัฒนาโครงการ .....	4
2.1. เทคโนโลยี XML .....	4
2.2. XQuery .....	7
2.3. XQuery Core .....	17
2.4. แบบจำลองการทำงานของ XQuery (XQuery Processing Model) .....	19
3. ทฤษฎีที่เกี่ยวข้องกับเครื่องมือที่ใช้ในการพัฒนาโครงการ .....	21
3.1. UML (Unified Modeling Language) .....	21
3.2. จาวา .....	22
4. วิเคราะห์ออกแบบโครงการพัฒนาระบบงาน .....	24

## สารบัญ (ต่อ)

	หน้า
4.1. ความต้องการ และข้อจำกัดของโปรแกรม .....	24
4.2. ยูสเคสวิว .....	26
4.3. ลอจิคออลวิว .....	36
4.4. ดีพลอยแมนตัวิว .....	54
4.5. การออกแบบหน้าจอ .....	54
5. การใช้งานโปรแกรม .....	58
5.1. เริ่มใช้งานโปรแกรม .....	58
5.2. หน้าจอหลัก .....	59
5.3. หน้าจอ XQuery Editor .....	60
5.4. หน้าจอ XQuery Builder .....	61
5.5. หน้าจอ Result .....	67
6. บทสรุป .....	69
6.1. สรุปผลการพัฒนาระบบ .....	69
6.2. ปัญหาที่พบในการทำงาน และข้อเสนอแนะ .....	69
บรรณานุกรม .....	71
ภาคผนวก .....	72
ประวัติผู้เขียน .....	75

# สารบัญภาพ

หน้า

ภาพที่

2.1	โครงสร้างเอกสาร XML sample.xml ในลักษณะลำดับชั้นต้นไม้ .....	6
2.2	แบบจำลองการทำงานของ XQuery .....	20
3.1	ขั้นตอนการสร้างจาวาแอปพลิเคชัน .....	22
3.2	ขั้นตอนการสร้างจาวาแอปพลิเคชัน .....	23
4.1	ยูสเคสไดอะแกรม .....	32
4.2	แอคทิวิตีไดอะแกรมของโปรแกรมค้นหาข้อมูลภายในเอกสาร XML .....	35
4.3	ซีเควนซ์ไดอะแกรมการค้นหาข้อมูล .....	36
4.4	ซีเควนซ์ไดอะแกรมการแสดงผลเอกสาร XML ในรูปแบบโครงสร้างต้นไม้ .....	37
4.5	ซีเควนซ์ไดอะแกรมการแปลงรูปแบบการค้นหาจากหน้าจอช่วยเหลือเป็นชุดคำสั่ง XQuery .....	39
4.6	ซีเควนซ์ไดอะแกรมการกำหนดฟังก์ชันในหน้าจอช่วยเหลือ .....	40
4.7	ซีเควนซ์ไดอะแกรมการกำหนดเงื่อนไขในหน้าจอช่วยเหลือ .....	41
4.8	ซีเควนซ์ไดอะแกรมการเคลียร์หน้าจอเอดิเตอร์ .....	42
4.9	ซีเควนซ์ไดอะแกรมการเคลียร์หน้าจอช่วยเหลือ .....	42
4.10	ซีเควนซ์ไดอะแกรมการเปิดไฟล์ .....	43
4.11	ซีเควนซ์ไดอะแกรมการบันทึกไฟล์ .....	44
4.12	คลาสไดอะแกรมของโปรแกรม .....	45
4.13	คลาสไดอะแกรมของแพคเกจ com .....	46
4.14	คลาสไดอะแกรมของแพคเกจ pr .....	46
4.15	คลาสไดอะแกรมของแพคเกจ ui .....	47

## สารบัญภาพ (ต่อ)

หน้า

ภาพที่

4.16	คลาสไคอะแกรมของแพคเกจ graphical .....	47
4.17	คลาสไคอะแกรมของแพคเกจ parser.....	52
4.18	คลาสไคอะแกรมของแพคเกจ coretree .....	53
4.19	ดีพลอยเมนต์ไคอะแกรม .....	54
4.20	หน้าจอ XQuery Editor.....	55
4.21	หน้าจอ XQuery Builder.....	56
4.22	หน้าจอ Result.....	57
5.1	ไคอะล็อกเตือนเรื่องความปลอดภัยในการเข้าถึงข้อมูลภายในเครื่องคอมพิวเตอร์ผู้ใช้งาน .....	58
5.2	หน้าจอหลักของโปรแกรม .....	59
5.3	หน้าจอ XQuery Editor พร้อมรับชุดคำสั่ง XQuery จากผู้ใช้.....	61
5.4	หน้าจอ XQuery Builder พร้อมรับการสร้างชุดคำสั่ง XQuery จากผู้ใช้.....	62
5.5	ไคอะล็อกสำหรับกำหนดโหนดเริ่มต้นของการค้นหาข้อมูล.....	63
5.6	ไคอะล็อกสำหรับการกำหนดฟังก์ชัน.....	64
5.7	ไคอะล็อกสำหรับการกำหนดเงื่อนไข .....	65
5.8	หน้าจอ XQuery Builder ที่ผ่านการสร้างชุดคำสั่ง XQuery .....	66
5.9	หน้าจอ XQuery Editor ที่แสดงชุดคำสั่ง XQuery .....	67
5.10	หน้าจอ Result แสดงผลลัพธ์การค้นหาข้อมูล .....	68

# บทที่ 1

## บทนำ

### 1.1. ความเป็นมา และความสำคัญของปัญหา

ในโลกของธุรกิจปัจจุบันนี้ การติดต่อสื่อสารข้อมูลระหว่างองค์กรนับว่ามีความสำคัญ และเป็นแรงผลักดันให้ธุรกิจเดินหน้าต่อไป ทำให้ความพยายามในการแลกเปลี่ยนข้อมูลระหว่างกัน ถูกหยิบยกขึ้นมาเป็นประเด็นสำคัญในการพัฒนาเทคโนโลยีให้รองรับกับความต้องการนี้ให้ดีขึ้น และจากในช่วง 2-3 ปีที่ผ่านมา XML ได้เข้ามาเป็นมาตรฐานใหม่ และเป็นที่ยอมรับมากสำหรับการแลกเปลี่ยนข้อมูล เสริมให้การแลกเปลี่ยนข้อมูลนั้นมีประสิทธิภาพ และมีความยืดหยุ่นต่อการใช้งานได้มากที่สุด

เมื่อ XML ได้รับการยอมรับเป็นอย่างดี และมีผู้ใช้งานมากมาย ทำให้เกิดเป็นความต้องการใหม่ เพื่อให้เพิ่มประสิทธิภาพในการทำงานมากขึ้น ซึ่งนั่นก็คือ การค้นหาข้อมูลภายในเอกสาร XML และนี่เอง จึงเป็นจุดกำเนิดของเทคโนโลยีสำหรับการค้นหาข้อมูลภายในเอกสาร XML การพัฒนารูปแบบของการค้นหาข้อมูลนั้น ได้ถูกพัฒนาต่อเนื่องมาอย่างเป็นลำดับ ซึ่งในระยะเริ่มแรก ได้ถูกออกแบบมาในรูปแบบของไวยากรณ์ที่ง่ายต่อการใช้งานเพียงอย่างเดียวเท่านั้น แต่ต่อมา ก็ได้มีการพัฒนาเพิ่มขึ้น เพื่อให้มีประสิทธิภาพมากขึ้น โดยเริ่มมาจาก XQL (Robie. 1999), XML-QL (Deutsch et al. 1998), Quilt (Robie et al. 2000) จนมาถึง XQuery ซึ่งได้กลายมาเป็นมาตรฐานสำหรับการค้นหาข้อมูลภายในเอกสาร XML โดยผู้ที่กำหนดมาตรฐานนี้คือองค์กรไม่มุ่งหวังผลกำไรที่มีชื่อว่า World Wide Web Consortium (W3C)

ในปัจจุบันได้มีซอฟต์แวร์ที่เป็นเครื่องมือสำหรับสร้างสรรคงานต่างๆ ที่เกี่ยวกับเอกสาร XML โดยเฉพาะ ซึ่งเป็นการอำนวยความสะดวกให้กับผู้ใช้งานเป็นอย่างมาก แต่สำหรับ XQuery เองกลับยังไม่มีการอิมพลีเมนต์ทฤษฎีขึ้นมาใช้งานอย่างจริงจัง ทั้งที่ความจำเป็นในการค้นหาข้อมูลภายในเอกสาร XML เริ่มมีมากขึ้นเรื่อยๆ ต่อการปฏิบัติงาน เช่น ในกรณีที่ผู้ใช้งานได้รับเอกสาร XML มาจากองค์กรหนึ่ง และต้องการส่งต่อเอกสาร XML ไปยังอีกองค์กรหนึ่ง โดยที่คัดเลือกเฉพาะข้อมูลที่ต้องการเท่านั้น หรือในกรณีที่ต้องการส่งเอกสาร XML ที่คัดเลือกเฉพาะข้อมูลที่ต้องการจาก 2 เอกสาร XML เป็นต้น ดังนั้นโครงการนี้จะจึงพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสาร XML ด้วยไวยากรณ์

XQuery ผ่านเว็บ โดยเพิ่มการทำงานให้สะดวกต่อผู้ใช้งานที่ไม่มีความรู้ในเรื่องการเขียนชุดคำสั่ง XQuery

## 1.2. วัตถุประสงค์

วัตถุประสงค์ในการทำโครงการพัฒนาระบบงาน มีดังนี้

1. เพื่อเป็นการศึกษาเทคโนโลยี XML ที่มีใช้งานในปัจจุบัน
2. เพื่อเป็นการศึกษาทฤษฎี XQuery สำหรับเป็นแนวทางในการพัฒนาโครงการพัฒนาระบบงาน
3. เพื่อเป็นการศึกษาการวิเคราะห์ห่อแบบโครงการพัฒนาระบบงานในรูปแบบของวิซวลโมเดลโดยใช้ UML (Unified Modeling Language)
4. เพื่อเป็นการศึกษาวิธีการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) โดยใช้ภาษาจาวา
5. เพื่อเป็นการประยุกต์ใช้ทฤษฎี XQuery ตามมาตรฐานขององค์กร W3C ไปสร้างโปรแกรมที่สามารถนำไปใช้ในการทำงานได้จริง

## 1.3. ขอบเขตของโครงการพัฒนาระบบงาน

ขอบเขตในการทำโครงการพัฒนาระบบงาน มีดังนี้

1. สามารถใช้งาน โปรแกรมได้ผ่านทางอินเทอร์เน็ต
2. สนับสนุนทฤษฎี XQuery ตามมาตรฐานขององค์กร W3C
3. มีหน้าจอช่วยเหลือการทำงานเพื่อให้ผู้ใช้ที่ไม่มีความรู้ในไวยากรณ์ของ XQuery ใช้งานได้ง่าย
4. สามารถค้นหาข้อมูลจากเอกสาร XML ที่อยู่ภายในเครื่องผู้ใช้ หรือผ่านทาง URL ก็ได้
5. สามารถบันทึกชุดคำสั่ง XQuery ไว้ใช้งานในครั้งต่อไปได้
6. สามารถบันทึกผลลัพธ์ที่ได้จากการค้นหาข้อมูลเพื่อนำไปใช้ประโยชน์ต่อไปได้

## 1.4. ประโยชน์ที่คาดว่าจะได้รับ

จากการพัฒนาระบบงานตามโครงการที่เสนอมานี้ คาดว่าจะได้รับประโยชน์ดังนี้

1. ได้ศึกษาเทคโนโลยี XML และเทคโนโลยีที่เกี่ยวข้อง
2. ได้ศึกษาทฤษฎี XQuery ตามมาตรฐานขององค์กร W3C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ได้ศึกษาหลักการเขียนโปรแกรมเชิงวัตถุเพื่อนำมาใช้ในการพัฒนาโครงการ
4. ได้ศึกษา UML เพื่อนำมาใช้เป็นเครื่องมือในการวิเคราะห์ออกแบบโครงการ
5. ได้วิเคราะห์ออกแบบ และพัฒนาโปรแกรมตามทฤษฎีให้ใช้งานได้จริง

### 1.5. ขั้นตอนการดำเนินงาน

ขั้นตอนในการดำเนินงานโครงการพัฒนาระบบงาน มีดังนี้

1. ศึกษาเทคโนโลยี XML และเทคโนโลยีที่เกี่ยวข้อง
2. ศึกษาทฤษฎี XQuery ตามมาตรฐานขององค์กร W3C
3. กำหนดขอบเขตของโครงการพัฒนาระบบงาน
4. คัดเลือก และศึกษาเครื่องมือที่จะใช้ในการพัฒนาโครงการ
5. วิเคราะห์ และออกแบบโครงการพัฒนาระบบงานโดยใช้ UML
6. พัฒนาโปรแกรมโดยใช้ภาษาจาวา
7. ทดสอบความถูกต้อง และประสิทธิภาพของโปรแกรม

### 1.6. เครื่องมือที่ใช้ในการดำเนินงาน

เครื่องมือที่ใช้ในการดำเนินงานโครงการพัฒนาระบบงาน มีดังนี้

1. Rational Rose Enterprise Edition
2. IBM VisualAge for Java for Windows Version 4.0
3. Java™ 2 SDK, Standard Edition Version 1.4.1
4. Apache HTTP Server Version 1.3

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้องกับการพัฒนาโครงการ

การพัฒนาโครงการระบบงานนี้ เป็นเรื่องที่เกี่ยวข้องกับการค้นหาข้อมูลภายในเอกสาร XML ดังนั้นการศึกษาทฤษฎีที่เกี่ยวข้องกับโครงการนี้จึงเริ่มจากการเรียนรู้เทคโนโลยี XML และ XQuery เพื่อนำมาประยุกต์ใช้ในการพัฒนาโครงการ พร้อมกับศึกษา และคัดเลือกเครื่องมือที่ใช้ในการพัฒนาระบบ

#### 2.1. เทคโนโลยี XML

XML (eXtensible Markup Language) เป็นเทคโนโลยีที่เกี่ยวกับการกำหนดลักษณะ และโครงสร้างของข้อมูล ซึ่งเป็นข้อมูลที่สามารถอธิบายความหมายของตัวข้อมูลเองได้ (Self-Described Data) โดยอาศัยแท็กที่ผู้ใช้กำหนดขึ้นเพื่อบรรยายข้อมูล ซึ่งมีประโยชน์สำหรับการนำมาวางรูปแบบเอกสาร ทำให้สามารถนำมาเป็นรูปแบบในการแลกเปลี่ยนเอกสารระหว่างหน่วยงาน หรือองค์กรได้ โดยไม่ต้องกังวลเรื่องความแตกต่างของรูปแบบข้อมูล (Homer. 1999)

##### 2.1.1. โครงสร้างเอกสาร XML

XML มีการจัดแบ่งข้อมูลออกเป็นลำดับขั้นต้นไม่ ซึ่งมีกฎเกณฑ์เบื้องต้นเพื่อให้เอกสาร XML มีคุณสมบัติ Well-Formed ดังนี้ (Homer. 1999 ; Simpson. 1999)

1. วิธีการประกาศความเป็น XML จะเริ่มต้นด้วยอักษร <?xml และจบด้วยอักษร ?> และจำเป็นต้องอยู่ในตำแหน่งแรกของไฟล์
2. เอกสาร XML หนึ่งๆ จะมีรูตอิลิเมนต์ (Root Element) ได้เพียงหนึ่งเดียวเท่านั้น ซึ่งทำหน้าที่คลุมอิลิเมนต์อื่นๆ ทั้งหมด
3. แท็กเปิด และแท็กปิดต้องเหมือนกัน ต่างกันแต่เพียงในแท็กปิดต้องมีเครื่องหมายสแลช (/) นำหน้าชื่อแท็กเท่านั้น
4. ห้ามระบุแท็กเหลื่อมซ้อนกัน นั่นคือแท็กที่เปิดก่อน ต้องปิดทีหลัง
5. ชื่อแท็กให้ความสำคัญกับตัวอักษรเล็ก-ใหญ่
6. ค่าของแอตทริบิวต์ต้องอยู่ในเครื่องหมายคำพูดแบบ ("") หรือ (') อย่างใดอย่างหนึ่ง จากคำอธิบายข้างต้นสามารถยกตัวอย่างเพื่อประกอบความเข้าใจ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

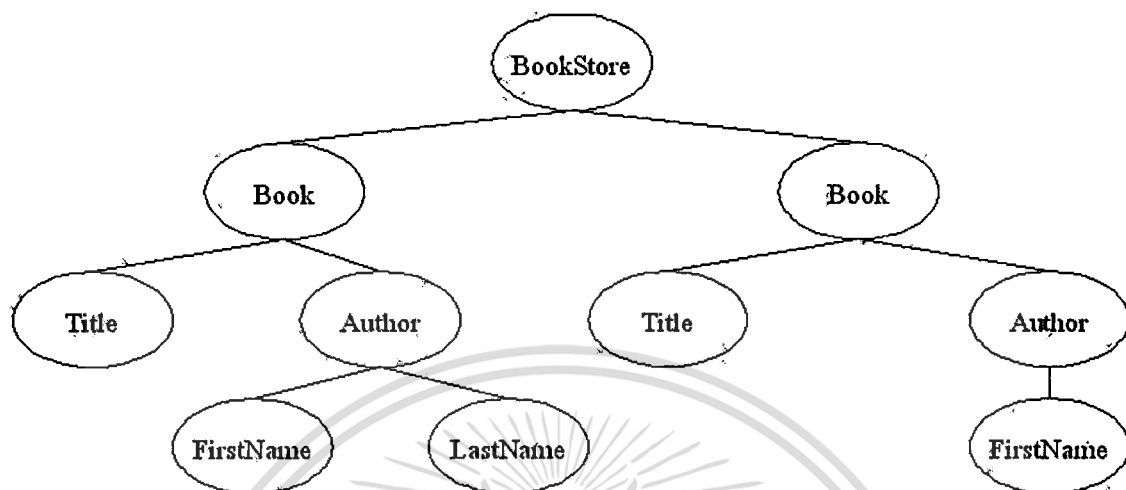
## ตัวอย่างที่ 2.1 ตัวอย่างเอกสาร XML sample.xml

```

<?xml version="1.0"?>
<BookStore>
  <Book category='Computer'>
    <Title>XML for Beginner</Title>
    <Author>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
    </Author>
  </Book>
  <Book category='Math'>
    <Title>Calculus I</Title>
    <Author>
      <FirstName>James</FirstName>
    </Author>
  </Book>
</BookStore>

```

จากตัวอย่างที่ 2.1 สามารถนำเอกสาร XML sample.xml มาแสดงในลักษณะของลำดับขั้นต้น  
ไม่ได้ ดังนี้



ภาพที่ 2.1 โครงสร้างเอกสาร XML sample.xml ในลักษณะลำดับชั้นต้นไม้

### 2.1.2. การกำหนดโครงสร้างเอกสาร

จากที่เอกสาร XML กลายเป็นมาตรฐานที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างหน่วยงาน หรือองค์กร ดังนั้นทั้งฝ่ายส่งและฝ่ายรับเอกสาร XML ของแต่ละองค์กร จะต้องมั่นใจว่า เอกสารนั้นมีความถูกต้องสามารถนำไปใช้งานได้ตามข้อตกลงร่วมกัน ซึ่งการกำหนดโครงสร้างเอกสารก็เพื่อให้เอกสาร XML นั้นมีรูปแบบที่ถูกต้องพร้อมที่จะนำไปใช้งาน วิธีการกำหนดโครงสร้างเอกสารที่นิยมใช้กันมี 2 วิธี ดังนี้

#### 2.1.2.1. การกำหนดโครงสร้างเอกสารแบบวิธี DTD

การกำหนดโครงสร้างเอกสาร XML แบบวิธี DTD (Document Type Declaration) สามารถทำได้ 2 แบบคือ แทรก DTD ไว้ในตัวเอกสาร XML หรือแยกเป็นไฟล์ที่มีนามสกุล .dtd ต่างหาก สำหรับวิธีการสร้าง DTD นั้นแบ่งออกเป็น 3 หัวข้อหลักคือ การประกาศค่าอิลิเมนต์ การประกาศค่าแอตทริบิวต์ และการประกาศค่าเอ็นทิตี (Simpson. 1999)

#### 2.1.2.2. การกำหนดโครงสร้างเอกสารแบบวิธี XML Schema

การกำหนดโครงสร้างเอกสาร XML ด้วยวิธี XML Schema สามารถทำได้โดยแยกไฟล์ออกมาต่างหากจากเอกสาร XML โดยไฟล์ดังกล่าวจะมีนามสกุลเป็น .xsd และมีการใช้แท็กเหมือนเอกสาร XML สำหรับมาตรฐานของ XML Schema ที่ประกาศโดย W3C แบ่งออกเป็น 3 หัวข้อหลักคือ ข้อมูลเบื้องต้น (Primer) โครงสร้าง และชนิดข้อมูล (Datatypes) (Simpson. 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3. DOM (Document Object Model)

DOM เป็นกระบวนการดึงข้อมูลในเอกสาร XML มาใช้งานในแอปพลิเคชัน โดยมีหลักการคือจะนำข้อมูลจากเอกสาร XML ทั้งเอกสารมาวางเป็นโครงสร้างลักษณะลำดับขั้นต้นไม้ในหน่วยความจำของเครื่องคอมพิวเตอร์ แล้วใช้วิธีการท่องไปในต้นไม้เพื่อดึงข้อมูลมา ซึ่งวิธีการโหลดข้อมูลทั้งเอกสาร XML เข้ามาไว้ในหน่วยความจำนั้นทำให้เกิดเป็นข้อเสียอย่างหนึ่งของ DOM เพราะจะกลายเป็นข้อจำกัดในเรื่องของหน่วยความจำในกรณีที่เอกสาร XML มีขนาดใหญ่หลายๆ ก็จะต้องเปลืองเนื้อที่ในหน่วยความจำมาก และอาจจะไม่มีเนื้อที่เหลือพอที่จะทำงาน (Homer. 1999)

## 2.2. XQuery

XQuery คือภาษาที่ใช้ค้นหาข้อมูลภายในเอกสาร XML ถูกคิดค้น และพัฒนาโดยองค์กร W3C เพื่อให้เป็นมาตรฐานสำหรับค้นหาข้อมูลภายในเอกสาร XML โดย XQuery เป็นภาษาในรูปแบบเชิงฟังก์ชัน (Functional Language) ที่ผู้ใช้สามารถอ่านแล้วเข้าใจ (Human-readable) และผลลัพธ์จะยังคงอยู่ในรูปแบบตามโครงสร้างของ XML ด้วยเช่นกัน (Boag et al. 2002) สำหรับการเขียนชุดคำสั่งต่างๆ ในการค้นหาข้อมูลนั้นจะอยู่ในรูปของนิพจน์ (Expression) ซึ่งนิพจน์ที่เป็นส่วนประกอบใน XQuery มีดังนี้ รูปแบบการเข้าถึงข้อมูล (Path Expression) ไวยากรณ์ FLWOR (อ่านออกเสียงว่า ฟลาวเวอร์) การใช้โอเปอเรเตอร์ และฟังก์ชันพื้นฐาน การใช้เงื่อนไขในการแสดงผลลัพธ์ (Condition Expression) และการใช้เงื่อนไขในการจำกัดจำนวนข้อมูล (Quantified Expression) (Bakker and Widarto. 2001) พร้อมกันนี้ XQuery ยังมีความสามารถในการค้นหาข้อมูลจากเอกสาร XML มากกว่า 1 เอกสารในการค้นหาแต่ละครั้ง โดยที่เอกสาร XML เหล่านั้นไม่จำเป็นต้องถูกเก็บไว้ในที่เก็บข้อมูลที่เดียวกัน และความสามารถในการสร้างฟังก์ชันขึ้นใช้งานเองได้ที่นอกเหนือจากฟังก์ชันพื้นฐาน

### 2.2.1. รูปแบบการเข้าถึงข้อมูล

XQuery มีวิธีการชี้ตำแหน่งของส่วนต่างๆ ในเอกสาร XML โดยอิงตามหลักไวยากรณ์ของ XPath ซึ่งยังคงมองเอกสาร XML ในลักษณะของลำดับขั้นต้นไม้ (Boag et al. 2002) ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.2 การเขียนนิพจน์สำหรับการเข้าถึงข้อมูล Author

```
/BookStore/Book/Author
```

คำอธิบาย: เข้าถึงข้อมูลของอิลิเมนต์ Author ทั้งหมดที่อยู่ภายใต้อิลิเมนต์ Book และอิลิเมนต์ Book ก็ต้องอยู่ภายใต้อิลิเมนต์ BookStore

## 2.2.2. ไวยากรณ์ FLWOR

ไวยากรณ์ที่ใช้ในการค้นหาข้อมูลของ XQuery นั้นพยายามลอกเลียนแบบการทำงานมาจาก SQL ให้มากที่สุด เพื่อให้สามารถเรียนรู้ได้ง่าย ไวยากรณ์ที่กล่าวมานี้เรียกว่า FLWOR ซึ่งย่อมาจาก For, Let, Where, Order By และ Return สำหรับการค้นหาข้อมูลในเอกสาร XML นั้น For และ/หรือ Let จะใช้ในการกำหนดเอกสาร XML ที่ต้องการค้นหาข้อมูล และเป็นการชี้ตำแหน่งเริ่มต้น Where จะใช้สำหรับการกรองข้อมูลที่ต้องการค้นหา Order By จะใช้สำหรับกำหนดการเรียงลำดับของข้อมูลผลลัพธ์ และ Return จะใช้ในการกำหนดรูปแบบของผลลัพธ์ตามโครงสร้างของ XML ใน XQuery นั้นให้ความสำคัญกับตัวอักษรเล็ก-ใหญ่เหมือนกับการกำหนดชื่อแท็กใน XML ดังนั้นจึงต้องมีการกำหนดรูปแบบของการใช้คำสั่งใน XQuery ซึ่งจะใช้ตัวอักษรตัวเล็กในทุกๆ คำสั่งของ XQuery (Boag et al. 2002)

หมายเหตุ: ตัวอย่างประกอบความเข้าใจทั้งหมดจากส่วนนี้เป็นต้นไป จะเป็นการค้นหาข้อมูลจาก

เอกสาร XML โดยอ้างอิงมาจากภาคผนวก

### 2.2.2.1. FOR และ LET

For ใช้สำหรับกำหนดตัวแปร และ/หรือชุดของตัวแปร ซึ่งในแต่ละชุดคำสั่งของ For นั้นจะคืนค่าออกมาเป็นกลุ่มของโหนด และค่าผลลัพธ์ในแต่ละโหนด (รวมถึงโหนดลูกหลานของโหนดดังกล่าวด้วย) ที่ได้มาก็จะถูกเก็บไว้ในตัวแปรที่กำหนด และมีการวนซ้ำในการคืนค่าของผลลัพธ์ด้วย (Bakker and Widarto. 2001)

Let ใช้สำหรับกำหนดตัวแปร และ/หรือชุดของตัวแปรเหมือนกับ For พร้อมกับการคืนค่าข้อมูลก็เหมือนกับ For ด้วยเช่นกัน ซึ่งถ้ามองดูแบบผิวเผินแล้วการทำงานของ Let นั้นจะคล้ายกับการทำงานของ For แต่ในความเป็นจริงแล้วทั้งสองยังคงมีสิ่งที่แตกต่างกันอยู่คือ Let จะไม่มีการวนซ้ำในการคืนค่าของผลลัพธ์ (Bakker and Widarto. 2001)

ถึงแม้ว่าการทำงานของ For และ Let จะคล้ายคลึงกันมาก แต่อย่างไรแล้วทั้งสองยังคงมีความแตกต่างกันในเรื่องของการคืนค่าผลลัพธ์ เพื่อให้เข้าใจถึงความแตกต่างของ For และ Let จะแสดงในตัวอย่างต่อไปนี้

### ตัวอย่างที่ 2.3 การใช้ For และการคืนค่าผลลัพธ์

```
for $s in (<one/>, <two/>, <three/>)
```

```
return <out> { $s } </out>
```

ผลลัพธ์ที่ได้คือ

```

<out>
  <one/>
</out>
<out>
  <two/>
</out>
<out>
  <three/>
</out>

```

#### ตัวอย่างที่ 2.4 การใช้ Let และการคืนค่าผลลัพธ์

```

let $s := (<one/>, <two/>, <three/>)
return <out> { $s } </out>

```

ผลลัพธ์ที่ได้คือ

```

<out>
  <one/>
  <two/>
  <three/>
</out>

```

#### 2.2.2.2. WHERE

Where ใช้สำหรับกำหนดเงื่อนไขในการกรองข้อมูลที่ต้องการจากที่ได้กำหนดไว้ในตัวแปร (จากการใช้ For และ/หรือ Let) สำหรับการคืนค่าของการใช้ Where นี้จะเป็นในลักษณะของบูลีน (Bakker and Widarto. 2001) การทำงานในส่วนนี้เปรียบเทียบกับการใช้ Where ใน SQL นั่นเอง ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.5 ค้นหาชื่อหนังสือจากเอกสาร books.xml โดยหนังสือเล่มนั้นจะต้องมาจากสำนักพิมพ์ Harper and Row

```
for $b in doc("books.xml") //book
where $b/pubinfo/publisher = "Harper and Row"
return $b/title
```

ผลลัพธ์ที่ได้คือ

```
<title>Harold and the Purple Crayon</title>
<title>Harold's Fairy Tale</title>
```

### 2.2.2.3. ORDER BY

Order By ใช้ในการกำหนดการเรียงลำดับของข้อมูล และ/หรือชุดของข้อมูล และการใช้งานของ Order By นั้นจะต้องถูกกำหนดไว้ก่อนการใช้ Return การทำงานในส่วนนี้เปรียบเทียบกับการใช้ Order By ใน SQL แต่ความพิเศษของ Order By ใน XQuery คือสามารถเรียงลำดับข้อมูลที่ไม่มีอยู่ในผลลัพธ์ได้ (Boag et al. 2002) ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.6 ค้นหาชื่อหนังสือจากเอกสาร books.xml โดยหนังสือเล่มนั้นจะต้องมาจากสำนักพิมพ์ Harper and Row และผลลัพธ์ที่ได้จะต้องเรียงลำดับตามปีที่พิมพ์ของหนังสือเล่มนั้น ซึ่งในผลลัพธ์ไม่ต้องแสดงปีที่พิมพ์หนังสือ

```
for $b in doc("books.xml") //book
where $b/pubinfo/publisher = "Harper and Row"
order by $b/pubinfo/year
return $b/title
```

ผลลัพธ์ที่ได้คือ

```
<title>Harold and the Purple Crayon</title>
<title>Harold's Fairy Tale</title>
```

#### 2.2.2.4. RETURN

Return ใช้ในการกำหนดรูปแบบการแสดงผลของผลลัพธ์ ซึ่งอิลิเมนต์ที่ปรากฏในผลลัพธ์นั้น ไม่จำเป็นต้องเป็นอิลิเมนต์ที่มีอยู่ในเอกสาร XML ที่ทำการค้นหาก็ได้ และถ้าก่อนหน้า Return ไม่ปรากฏ Order by อยู่ก่อน การเรียงลำดับของข้อมูลในผลลัพธ์จะเป็นไปตามลำดับในเอกสาร XML ที่ใช้ในการค้นหา หรือเป็นไปตามลำดับที่ได้ถูกกำหนดไว้จากการใช้ชุดคำสั่ง For (Boag et al. 2002) ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.7 ค้นหาชื่อหนังสือจากเอกสาร books.xml โดยหนังสือเล่มนั้นจะต้องมาจากสำนักพิมพ์ Harper and Row และมีการกำหนดโครงสร้างของผลลัพธ์ โดยสร้างอิลิเมนต์ใหม่ที่ไม่ได้อยู่ในเอกสาร books.xml

```
for $b in doc("books.xml") //book
where $b/pubinfo/publisher = "Harper and Row"
return <HR_book> { $b/title } <HR_book>
```

ผลลัพธ์ที่ได้คือ

```
<HR_book>
  <title>Harold and the Purple Crayon</title>
</HR_book>
<HR_book>
  <title>Harold's Fairy Tale</title>
</HR_book>
```

#### 2.2.3. โอเปอเรเตอร์ และฟังก์ชันพื้นฐาน

XQuery มีการกำหนดโอเปอเรเตอร์ และฟังก์ชันพื้นฐานให้ใช้งาน ซึ่งประกอบไปด้วย (Bakker and Widarto. 2001)

- โอเปอเรเตอร์ทางคณิตศาสตร์ ได้แก่ เครื่องหมายบวก ลบ เป็นต้น
- โอเปอเรเตอร์สำหรับการเปรียบเทียบ ได้แก่ เครื่องหมายมากกว่า น้อยกว่า เป็นต้น
- โอเปอเรเตอร์ในเชิงตรรกะ ได้แก่ AND, OR เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับฟังก์ชันพื้นฐานที่มีให้ใช้ จะประกอบด้วย (Bakker and Widarto. 2001)

- ฟังก์ชันการหาค่าเฉลี่ย (AVG)
- ฟังก์ชันการหาผลรวม (SUM)
- ฟังก์ชันการนับจำนวน (COUNT)
- ฟังก์ชันการหาค่าสูงสุด (MAX)
- ฟังก์ชันการหาค่าต่ำสุด (MIN)
- ฟังก์ชันสำหรับการอ้างอิงเอกสาร XML ที่ต้องการค้นหาข้อมูล (DOC)
- ฟังก์ชันสำหรับการนำเสนอเฉพาะข้อมูลภายในแท็ก (TEXT)
- ฟังก์ชันการตรวจสอบค่าว่าง (EMPTY)
- ฟังก์ชันการตรวจสอบค่าที่มีอยู่ (EXISTS)

ตัวอย่าง 2.8 ค้นหาราคาหนังสือที่แพงที่สุดจากเอกสาร books.xml

```
let $b := doc("data/books.xml") //book
return <HR_book> { max($b/pubinfo/price) } </HR_book>
ผลลัพธ์ที่ได้คือ
<HR_book>15.45</HR_book>
```

#### 2.2.4. การกำหนดเงื่อนไขสำหรับการแสดงผลของผลลัพธ์

การกำหนดเงื่อนไขสำหรับการแสดงผลของผลลัพธ์จะถูกใช้ภายใน Return เพื่อให้กำหนดเงื่อนไขในการแสดงค่าผลลัพธ์ ซึ่งจะต้องอยู่ในรูปแบบของ If...Then...Else (Chamberlin. 2002) ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.9 ค้นหาชื่อหนังสือจากเอกสาร books.xml โดยในผลลัพธ์จะแสดงปีที่พิมพ์หนังสือถ้าหนังสือเล่มนั้นพิมพ์หลังจากปี 1955 แต่จะแสดงราคาหนังสือถ้าหนังสือเล่มนั้นพิมพ์ก่อน หรือพิมพ์ในปี 1955

```
for $b in doc("books.xml") //book
return <HR_book> {
```

```

$b/title,
if ($b/pubinfo/year > "1955") then $b/pubinfo/year
else $b/pubinfo/price
} </HR_book>

```

ผลลัพธ์ที่ได้คือ

```

<HR_book>
  <title>Harold and the Purple Crayon</title>
  <price>4.76</price>
</HR_book>
<HR_book>
  <title>Harold's Fairy Tale</title>
  <year>1956</year>
</HR_book>
<HR_book>
  <title>Rise Up Singing</title>
  <year>1988</year>
</HR_book>

```

### 2.2.5. การกำหนดเงื่อนไขในการจำกัดจำนวนข้อมูล

การใช้เงื่อนไขประเภทนี้ใช้สำหรับการค้นหาข้อมูลในลักษณะดังนี้ คัดเลือกข้อมูลที่มีเพียงบางอิเลเมนต์ตรงกับเงื่อนไขที่ต้องการ หรือคัดเลือกข้อมูลทีทุกๆ อิเลเมนต์จะต้องตรงกับเงื่อนไขที่ต้องการ โดยใช้คำสั่ง Some หรือ Every ตามลำดับ สำหรับการใช้งานด้วยคำสั่งทั้งสองนั้นจะต้องตามด้วยคำสั่ง Satisfies (Bakker and Widarto. 2001) ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.10 ค้นหาชื่อหนังสือจากเอกสาร books.xml โดยที่จะต้องมียังหนังสืออย่างน้อย 1 เล่มที่พิมพ์หลังจากปี 1970

```
let $b in doc("books.xml") //book
```

**where some \$y in \$b//year satisfies \$y > “1970”**

return \$b/title

ผลลัพธ์ที่ได้คือ

<title>Harold and the Purple Crayon</title>

<title>Harold’s Fairy Tale</title>

<title>Rise Up Singing</title>

ตัวอย่างที่ 2.11 ค้นหาชื่อหนังสือจากเอกสาร books.xml โดยที่หนังสือทุกๆ เล่มจะต้องพิมพ์หลังจากปี 1955

let \$b in doc(“books.xml”) //book

**where every \$y in \$b//year satisfies \$y > “1955”**

return \$b/title

ผลลัพธ์ที่ได้คือ ไม่มีคำตอบสำหรับการค้นหา

### 2.2.6. การค้นหาข้อมูลภายในเอกสาร XML ที่มีมากกว่า 1 เอกสาร

ในกรณีที่ต้องทำการค้นหาข้อมูลจากเอกสาร XML 2 เอกสารขึ้นไป เพื่อนำผลลัพธ์ที่ได้จากทั้งสองเอกสารมารวมกันเป็นเอกสารผลลัพธ์เดียว โดยการรวมผลลัพธ์จากทั้งสองเอกสารนั้นจะต้องมีการกำหนดเงื่อนไขเพื่อให้เอกสารทั้งสองอ้างอิงถึงในสิ่งเดียวกัน (Boag et al. 2002) ซึ่งสามารถเปรียบเทียบได้กับการค้นหาข้อมูลจาก 2 ตารางขึ้นไป (Join) ใน SQL นั่นเอง ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.12 ค้นหาชื่อหนังสือ และชื่อผู้แต่งจากเอกสาร books.xml รวมกับการค้นหาคำวิจารณ์หนังสือเล่มนั้นจากเอกสาร reviews.xml

for \$b in doc(“books.xml”) //book,

**\$r in doc(“reviews.xml”) //book**

**where \$b/title = \$r/title**

return <HR\_book> {

    \$b/title,

```

    $b/author,
    $r/review
  } </HR_book>

```

ผลลัพธ์ที่ได้คือ

```

<HR_book>
  <title>Rise Up Singing</title>
  <author>
    <lastname>Blood</lastname>
    <firstname>Peter</firstname>
  </author>
  <author>
    <lastname>Patterson</lastname>
    <firstname>Annie</firstname>
  </author>
  <review>This is great!</review>
</HR_book>

```

### 2.2.7. การสร้างฟังก์ชัน

นอกจากฟังก์ชันพื้นฐานที่มีให้ไว้แล้ว XQuery ยังเปิดให้ผู้ใช้สามารถสร้างฟังก์ชันขึ้นใช้งานเองได้ โดยในการสร้างฟังก์ชันหนึ่งๆ นั้นจะต้องเป็นไปตามฟอร์แมตดังนี้ (Boag et al. 2002)

Function-name(\$Parameter-name,...) as Return-type

ตัวอย่างที่ 2.13 การสร้างฟังก์ชัน depth เพื่อค้นหาจำนวนลำดับชั้นของเอกสาร books.xml

```

define function depth($e) as integer
{ if (empty($e/*)) then 1
  else max(depth($e/*)) + 1
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
depth(doc("books.xml"))//book)
```

ผลลัพธ์ที่ได้คือ

3

### 2.2.8. การทำงานกับแอดทริบิวต์

ในกรณีที่เอกสาร XML ที่ต้องการค้นหาแอดทริบิวต์เป็นส่วนประกอบหนึ่งในเอกสารด้วยนั้น สามารถนำมาใช้เป็นข้อมูลหนึ่งในการกำหนดเงื่อนไข และการแสดงผลได้ดังนี้

#### 2.2.8.1. การกำหนดเงื่อนไขด้วยแอดทริบิวต์

XQuery สามารถค้นหาข้อมูลโดยมีการกำหนดเงื่อนไขในการกรองข้อมูลด้วยแอดทริบิวต์ ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.14 ค้นหาชื่อหนังสือจากเอกสาร books.xml โดยหนังสือเล่มนั้นจะต้องมีแอดทริบิวต์ number มากกว่า 2000

```
for $b in doc("books.xml") //book
where $b/@number > "2000"
return <HR_book> { $b/title } </HR_book>
ผลลัพธ์ที่ได้คือ
<HR_book>
  <title>Rise Up Singing</title>
</HR_book>
```

#### 2.2.8.2. การแสดงแอดทริบิวต์ในผลลัพธ์

XQuery สามารถแสดงแอดทริบิวต์ในอิลิเมนต์ที่สร้างใหม่ของการแสดงผลได้ ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.15 ค้นหาชื่อหนังสือจากเอกสาร books.xml โดยมีการกำหนดโครงสร้างของผลลัพธ์ โดยสร้างอิลิเมนต์ใหม่ที่ไม่มีอยู่ในเอกสาร และภายในอิลิเมนต์ใหม่นี้แสดงข้อมูลของแอตทริบิวต์ number ด้วย

```
for $b in doc("books.xml")//book
return
<HR_book> { $b/@number }
  { $b/title }
</HR_book>
ผลลัพธ์ที่ได้คือ
<HR_book number="1001">
  <title>Harold and the Purple Crayon</title>
</HR_book>
<HR_book number="1002">
  <title>Harold's Fairy Tale</title>
</HR_book>
<HR_book number="2001">
  <title>Rise Up Singing</title>
</HR_book>
```

### 2.3. XQuery Core

XQuery Core เป็นรูปแบบภาษาในเชิงฟังก์ชันที่มีพื้นฐานบนพีชคณิต ซึ่งนิพจน์ XQuery ที่ได้รับการตรวจสอบว่าถูกต้องตามไวยากรณ์แล้วจะถูกแปลงให้อยู่ในรูปของ XQuery Core เพื่อนำไปประมวลผลหาผลลัพธ์ต่อไป โดยรูปแบบของ XQuery Core ก็คือฟอรัลูปที่ซ้อนกัน (Nested For-Loop) (Fankhauser, 2001) ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.16 การแปลงนิพจน์การเข้าถึงข้อมูลให้เป็น XQuery Core

```
/bib/book
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงเป็น XQuery Core ได้ดังนี้

```
FOR $v1 IN /bib RETURN
```

```
  FOR $v2 IN NODES ($v1) RETURN
```

```
    TYPESWITCH ($v2) AS $v3
```

```
      CASE ELEMENT book {ANYTYPE}
```

```
        RETURN $v3
```

```
      DEFAULT RETURN ()
```

คำอธิบาย

รูปนอกสุดจะเป็นการวนซ้ำทุกๆ โหนดใน /bib ด้วยตัวแปร \$v1 ซึ่งจากเอกสาร books.xml จะเห็นว่า /bib นั้นเป็นรูตอิลิเมนต์อยู่แล้ว เพราะฉะนั้นจะเป็นการวนซ้ำเพียงแค่ครั้งเดียว และสำหรับรูปในจะเป็นการวนซ้ำภายใน \$v1 อีกทีเพื่อหาว่ามีอิลิเมนต์ใดใน \$v1 ที่มีอิลิเมนต์เป็น book

จากตัวอย่างข้างต้นจะเป็นตัวอย่างง่ายๆ ในการแปลงนิพจน์การเข้าถึงข้อมูลของ XQuery ให้อยู่ในรูปของ XQuery Core แต่นอกจากการใช้ฟอร์มูลที่ซ้อนกันนั้นยังมีการใช้ If...Then...Else มาเสริมในส่วนของการแปลงนิพจน์ XQuery ด้วย ซึ่งจะถูกใช้ในการกรองข้อมูลด้วย WHERE (Fankhauser. 2001) ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.17 การแปลงนิพจน์ XQuery ให้อยู่ในรูปของ XQuery Core

```
for $b in doc("books.xml")/bib/book,
```

```
    $r in doc("reviews.xml")/reviews/book
```

```
where $b/title = $r/title
```

```
return <HR_book> { $b/title, $b/author, $r/review } <HR_book>
```

แปลงเป็น XQuery Core ได้ดังนี้

```
FOR $b IN NODES (doc("books.xml")/bib) RETURN
```

```
  FOR $r IN NODES (doc("reviews.xml")/reviews) RETURN
```

```
    NOT (EMPTY(
```

```
      FOR $v1 IN $b/title RETURN
```

```
        FOR $v2 IN $r/title RETURN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF EQ($v1,$v2) THEN
    ELEMENT book { $b/title, $b/author, $r/review }
ELSE ( )

```

## 2.4. แบบจำลองกระบวนการทำงานของ XQuery

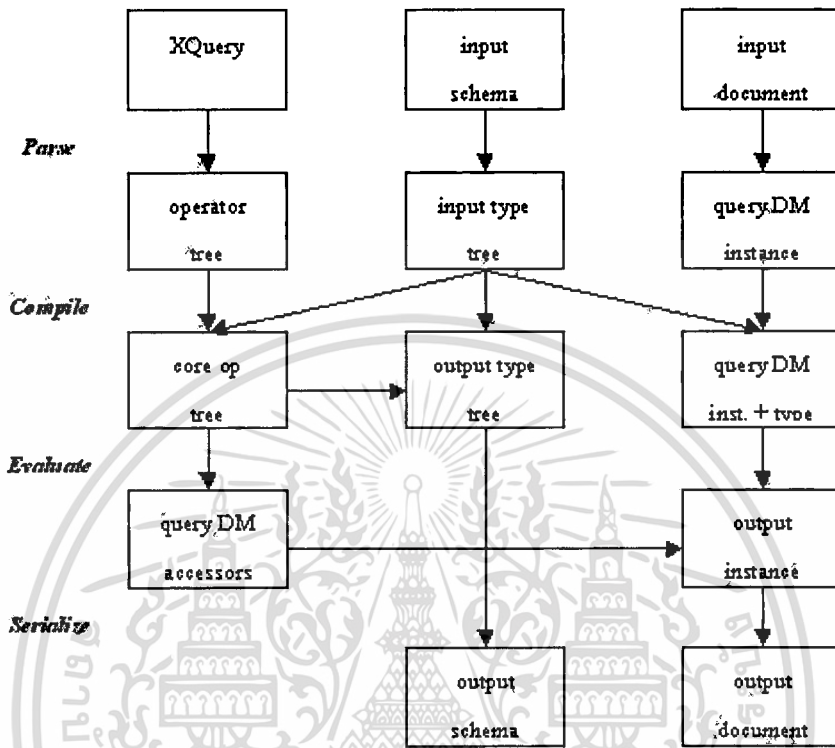
แบบจำลองกระบวนการทำงาน (XQuery Processing Model) ประกอบด้วย 4 ขั้นตอน ซึ่งในแต่ละขั้นตอนจะนำผลลัพธ์ที่ได้จากขั้นตอนก่อนหน้ามาเป็นอินพุต เพื่อนำมาเข้าสู่กระบวนการทำงานเพื่อให้ได้เอาต์พุตไปทำงานในขั้นตอนต่อไป ดังแสดงในภาพที่ 2.2 ซึ่งรายละเอียดในแต่ละขั้นตอนมีดังนี้ (Fankhauser. 2001)

กระบวนการวิภาษข้อมูล (Parse) มีหน้าที่ในการตรวจสอบความถูกต้องของนิพจน์ชุดคำสั่ง XQuery (XQuery Expression) ความถูกต้องของเค้าร่างเอกสาร (XML Schema) และความถูกต้องของรูปแบบเอกสาร XML (XML Document) หลังจากตรวจสอบความถูกต้องเรียบร้อยแล้ว จะสร้างโครงสร้างโอเปอเรเตอร์ในรูปแบบต้นไม้ สร้างโครงสร้างเอกสารในรูปแบบต้นไม้ และสร้างโครงสร้างข้อมูลเอกสารในรูปแบบต้นไม้ตามลำดับ

กระบวนการแปลโปรแกรม (Compile) มีหน้าที่แปลงนิพจน์ชุดคำสั่ง XQuery ให้อยู่ในรูปของ XQuery Core และนำโครงสร้างเอกสารกับเอกสาร XML มาตรวจสอบความถูกต้องร่วมกัน

กระบวนการประเมินหาคำตอบ (Evaluate) มีหน้าที่นำ XQuery Core มาประมวลผลเพื่อค้นหาข้อมูลผลลัพธ์ และสร้างโครงสร้างผลลัพธ์

กระบวนการเรียงลำดับสร้างผลลัพธ์ (Serialize) มีหน้าที่นำข้อมูลผลลัพธ์ และโครงสร้างผลลัพธ์มาสร้างเป็นเอกสาร XML



ภาพที่ 2.2 แบบจำลองการทำงานของ XQuery

## บทที่ 3

### เครื่องมือที่ใช้ในการพัฒนาโครงการ

หลังจากที่ได้เรียนรู้เทคโนโลยี XML และ XQuery เพื่อนำมาเป็นพื้นฐานขั้นต้นในการพัฒนาโครงการแล้ว ขั้นตอนต่อไปคือทำการศึกษา และคัดเลือกเครื่องมือที่ใช้ในการพัฒนาระบบ โดยมีทฤษฎีที่เกี่ยวข้องกับเครื่องมือที่เลือกใช้ในการพัฒนาโครงการ ดังนี้

#### 3.1. UML (Unified Modeling Language)

UML เป็นภาษาพิมพ์เขียวที่ใช้เป็นเครื่องมือในการวิเคราะห์ออกแบบ เพื่อสร้างแบบจำลองเชิงวัตถุสำหรับการพัฒนาโปรแกรม ที่ได้รับการรับรองจาก Object Management Group (OMG) (Quatrani. 2000) สนับสนุนการวิเคราะห์ออกแบบเชิงแนวคิด และเชิงตรรกะ ซึ่งรูปแบบของโมเดลสามารถใช้อธิบายได้ว่าระบบทำอะไร (What) และทำงานอย่างไร (How) สำหรับใน UML จะประกอบด้วยไดอะแกรมต่างๆ ดังต่อไปนี้ (Booch et al. 2000)

- คลาส ไดอะแกรม (Class Diagram): แสดงคลาส และความสัมพันธ์ของคลาสทั้งหมดในระบบ
- ออบเจกต์ ไดอะแกรม (Object Diagram): แสดงออบเจกต์ และความสัมพันธ์ของออบเจกต์ทั้งหมดในระบบ
- ยูสเคส ไดอะแกรม (Use Case Diagram): แสดงยูสเคส แอคเตอร์ และความสัมพันธ์ระหว่างยูสเคสกับแอคเตอร์ทั้งหมดในระบบ
- อินเทอร์แอคชัน ไดอะแกรม (Interaction Diagram): แสดงลำดับการทำงานของระบบในเชิงวัตถุ ซึ่งสามารถเขียนไดอะแกรมลักษณะนี้ได้ทั้งสองรูปแบบคือ ซีควเอนซ์ ไดอะแกรม (Sequence Diagram) หรือคอลแลบอเรชัน ไดอะแกรม (Collaboration Diagram)
- สเตทชาร์ท ไดอะแกรม (Statechart Diagram): แสดงลำดับสถานะการทำงานของระบบ
- แอกทิวิตี ไดอะแกรม (Activity Diagram): แสดงลำดับการทำงานของระบบในรูปแบบกิจกรรม
- คอมโพเนนต์ ไดอะแกรม (Component Diagram): แสดงคอมโพเนนต์ และความสัมพันธ์ของคอมโพเนนต์ทั้งหมดในระบบ
- ดีพลอยเม้นต์ ไดอะแกรม (Deployment Diagram): แสดงโครงแบบ (Configuration) ของระบบ

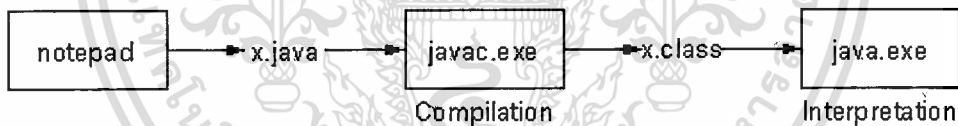
### 3.2. จาวา

ภาษาจาวามีไวยากรณ์ของภาษาส่วนใหญ่คล้ายกับภาษา C และ C++ แต่มีรายละเอียด และโครงสร้างบางอย่างแตกต่างกัน เนื่องจากภาษาจาวาถูกสร้างขึ้นสำหรับเขียนโปรแกรมในแบบเชิงวัตถุ อีกทั้งยังเน้นในเรื่องความปลอดภัย และ โปรแกรมที่ไม่ขึ้นกับระบบปฏิบัติการ หรือหน่วยประมวลผล (วีระศักดิ์ ซึ่งถาวร. 2541 : 25) โปรแกรมภาษาจาวาแบ่งออกได้เป็นสองประเภท ขึ้นกับวิธีที่เรียกโปรแกรมนั้นให้ทำงาน คือ (วีระศักดิ์ ซึ่งถาวร. 2541 : 11)

1. จาวาแอปพลิเคชัน (Java application) จะถูกทำงาน โดย java.exe
2. แอปเพล็ต (Applets) จะถูกทำงานโดยจาวาอินเทอร์เน็ตที่มีอยู่ในเว็บเบราว์เซอร์

#### 3.2.1. จาวาแอปพลิเคชัน

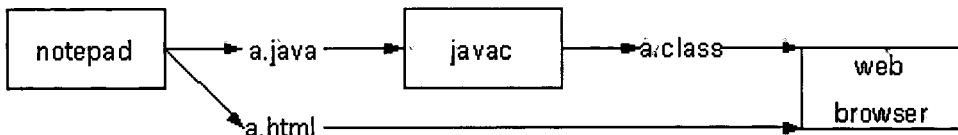
จาวาแอปพลิเคชัน คือ โปรแกรมที่ถูกสร้างขึ้นเพื่อทำงาน โดยถูก java.exe ทำการแปลคำสั่ง ถือว่าเป็นโปรแกรมที่สมบูรณ์ เนื่องจากไม่จำเป็นต้องมีโปรแกรมอื่นช่วยในการทำงาน สามารถควบคุมการดำเนินไปของตัวเองได้ โดยมีเมธอด main() เป็นจุดเริ่มต้นของโปรแกรม บางคนเรียกโปรแกรมแบบนี้ว่า Stand alone program (วีระศักดิ์ ซึ่งถาวร. 2541 : 12) ซึ่งมีขั้นตอนวิธีการสร้างดังแสดงในภาพที่ 3.1



ภาพที่ 3.1 ขั้นตอนการสร้างจาวาแอปพลิเคชัน

#### 3.2.2. จาวาแอปเพล็ต

แอปเพล็ต คือ โปรแกรมที่สร้างขึ้นโดยภาษาจาวาเพื่อส่งไปทำงานที่เครื่องต่างๆ เช่น ทางระบบเครือข่าย ภายใต้เว็บเบราว์เซอร์ที่มีจาวาอินเทอร์เน็ต โดยคล้ายกับว่าเป็นโปรแกรมที่ฝากมากับหน้า HTML นั่นคือ จากในหน้า HTML จะสามารถเรียกแอปเพล็ตมาทำงาน และแสดงผลในพื้นที่ของหน้านั้น โดยที่เว็บเบราว์เซอร์จะเป็นผู้โหลด โปรแกรมของแอปเพล็ตที่ถูกเรียกนั้นมาจากเซิร์ฟเวอร์ที่ให้หน้า HTML มา และจัดการทำงานให้โดยอัตโนมัติ วิธีนี้ทำให้เราสามารถส่งแอปเพล็ตกับหน้า HTML ไปทำงานบนเครื่องของผู้เรียกดูหน้า HTML ได้ (วีระศักดิ์ ซึ่งถาวร. 2541 : 14) ซึ่งมีขั้นตอนวิธีการสร้างดังแสดงในภาพที่ 3.2



ภาพที่ 3.2 ขั้นตอนการสร้างจาวาแอปเพล็ต

การเรียกให้แอปเพล็ตทำงานนั้นต้องทำจากภายในหน้า HTML โดยใช้แท็กที่ชื่อว่า APPLET และแอปเพล็ตนั้นจะทำงานในสถานะแวดล้อมของเว็บเบราว์เซอร์ อย่างเช่น Netscape Navigator, Microsoft Internet Explorer หรือ Hot Java ซึ่งทั้งหมดเป็นเว็บเบราว์เซอร์ที่มีจาวาอินเทอร์พรีเตอร์อยู่ภายใน หรือ โปรแกรมสำหรับใช้ทดสอบแอปเพล็ตที่มากับ JDK คือ appletviewer (วีระศักดิ์ ชิงถาวร. 2543 : 1)

เนื่องจากแอปเพล็ตเป็นโปรแกรมที่จะถูกแจกจ่ายไปในระบบเครือข่าย เพื่อไปทำงานบนเครื่องของผู้ใช้ซึ่งอยู่ในที่ต่างๆ ดังนั้นจึงต้องมีข้อจำกัดบางอย่างในการสร้างแอปเพล็ต เพื่อให้ผู้ใช้ประสงค์ร้ายบางคนสามารถสร้างแอปเพล็ตที่ไปทำลายโปรแกรม หรือข้อมูลบนเครื่องคอมพิวเตอร์ของผู้รับแอปเพล็ตนั้นไปทำงาน เช่น ทำการลบ ไฟล์ หรือส่งอีเมลปลอม โดยใช้ชื่อของเจ้าของเครื่อง หรือแอบใช้เครื่องคอมพิวเตอร์นั้นเป็นรีโมตไฟล์เซิร์ฟเวอร์ เป็นต้น (วีระศักดิ์ ชิงถาวร. 2543 : 27)

โดยปกติแล้วเว็บเบราว์เซอร์จะถือว่าแอปเพล็ตที่รับมาจากเครื่องอื่นในระบบเครือข่ายเป็นโปรแกรมที่ไม่ปลอดภัยไว้ก่อนซึ่งเรียกว่าแอปเพล็ตที่ไม่น่าเชื่อถือ (Untrusted Applet) และจะให้แอปเพล็ตเหล่านี้ทำงานภายใต้สภาพแวดล้อมที่ปลอดภัย ซึ่งมีการติดตั้งตัวควบคุมความปลอดภัย (Security Manager) ให้คอยตรวจสอบการทำงานของแอปเพล็ต และไม่ยอมให้มีการกระทำที่ไม่สมควรเกิดขึ้น (วีระศักดิ์ ชิงถาวร. 2543 : 28) ทำให้แอปเพล็ตที่ไม่น่าเชื่อถือถูกจำกัดด้วยกฎเกณฑ์ต่างๆ เช่น

- ห้ามออกคำสั่งกับระบบ ไฟล์ของเครื่องที่รับแอปเพล็ตนั้นไปทำงาน เช่น ห้ามเขียน อ่าน หรือลบไฟล์
- ห้ามออกคำสั่งแก่ระบบเครือข่าย เช่น ห้ามติดต่อไปยังคอมพิวเตอร์เครื่องอื่นที่ไม่ใช่เครื่องที่ให้แอปเพล็ตนั้นมา

ในกรณีที่ไม่ต้องการให้แอปเพล็ตถูกจำกัดด้วยกฎเกณฑ์ข้างต้นนั้น สามารถทำได้โดยการสร้างความน่าเชื่อถือให้กับแอปเพล็ตนั้น (Signed Applet) ซึ่งเป็นการกำหนดสิทธิการเข้าถึงต่างๆ โดยผู้ใช้จะต้องยินยอมเสียก่อนถึงจะสามารถใช้งานแอปเพล็ตนั้นได้

## บทที่ 4

### การวิเคราะห์ออกแบระบบงาน

หลังจากที่ได้ทำการศึกษาค้นคว้าทฤษฎีของ XQuery แล้วในขั้นตอนต่อไปจะเกี่ยวข้องกับการกำหนดความต้องการ และข้อจำกัดของโปรแกรมให้ชัดเจน เพื่อใช้เป็นพื้นฐานในการวิเคราะห์ และออกแบบ โครงการพัฒนาระบบงานต่อไป

#### 4.1. ความต้องการ และข้อจำกัดของโปรแกรม

โครงการพัฒนาระบบงานนี้สามารถกำหนดความต้องการ และข้อจำกัดของโปรแกรมเพื่อใช้เป็นขอบเขตในการพัฒนาโครงการได้ดังต่อไปนี้

##### 4.1.1. ความต้องการเชิงฟังก์ชัน

การทำงานของโปรแกรมนั้นจะต้องรองรับการทำงานต่างๆ ดังต่อไปนี้

- โปรแกรมจะต้องมีหน้าจอเอดิเตอร์สำหรับผู้ใช้ที่มีความรู้ในการเขียนชุดคำสั่ง XQuery
- โปรแกรมจะต้องสนับสนุนการทำงานในกรณีที่ใช้ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery โดยมีหน้าจอช่วยเหลือเป็นลำดับขั้นตอนเพื่อเพิ่มความสะดวก และใช้งานง่ายให้กับผู้ใช้ทั่วไป
- โปรแกรมจะต้องสามารถตรวจสอบคุณสมบัติ Well-Formed ของเอกสาร XML ที่ต้องการค้นหา พร้อมกับแสดงข้อความแจ้งผู้ใช้ในกรณีที่ตรวจพบความไม่ถูกต้องของเอกสาร XML
- โปรแกรมจะต้องสามารถค้นหาข้อมูลภายในเอกสาร XML ได้มากกว่า 1 เอกสารในการค้นหา 1 ครั้ง และเอกสารเหล่านั้นไม่จำเป็นต้องอยู่ในที่เก็บข้อมูลที่เดียวกัน
- โปรแกรมจะต้องสามารถค้นหาข้อมูลภายในเอกสาร XML ได้ผ่านทางงานบันทึกเฉพาะที่ และ URL
- โปรแกรมจะต้องสามารถตรวจสอบไวยากรณ์ของชุดคำสั่ง XQuery ซึ่งไวยากรณ์ดังกล่าวนั้นจะต้องเป็นไปตามมาตรฐานขององค์กร W3C พร้อมกับแสดงข้อความแจ้งผู้ใช้ในกรณีที่ตรวจพบความไม่ถูกต้องของชุดคำสั่ง XQuery
- โปรแกรมจะต้องเข้าใจ และทำงานได้เมื่อผู้ใช้มีการใช้โอเปอเรเตอร์ และฟังก์ชันพื้นฐานในการเขียนชุดคำสั่ง XQuery

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมจะต้องเปิดให้ผู้ใช้สามารถสร้างฟังก์ชันเองได้ในการเขียนชุดคำสั่ง XQuery
- โปรแกรมจะต้องสามารถแสดงโครงสร้างเอกสาร XML ในรูปแบบของลำดับขั้นต้นไม้ กรณีที่ผู้ใช้ทำงานผ่านหน้าจอช่วยเหลือ
- โปรแกรมจะต้องสามารถแปลงสิ่งที่ผู้ใช้ทำงานในหน้าจอช่วยเหลือเป็นชุดคำสั่ง XQuery ได้ เพื่อให้ผู้ใช้สามารถเปลี่ยนแปลงแก้ไขชุดคำสั่ง XQuery เพิ่มเติมผ่านหน้าจอเอดิเตอร์
- โปรแกรมจะต้องคืนผลลัพธ์เป็นเอกสาร XML โดยเอกสารนั้นจะต้องถูกต้องตามคุณสมบัติ Well-Formed
- โปรแกรมจะต้องสามารถบันทึกชุดคำสั่ง XQuery เพื่อนำมาใช้ในภายหลังได้
- โปรแกรมจะต้องสามารถบันทึกผลลัพธ์ที่ได้จากการค้นหา เพื่อนำไปใช้ประโยชน์ต่อไปได้
- โปรแกรมจะต้องสามารถเปิดดูเอกสาร XML, XML สกิวมา และไฟล์ชุดคำสั่ง XQuery

#### 4.1.2. ความต้องการที่ไม่ใช่เชิงฟังก์ชัน

ข้อตกลง เงื่อนไข และข้อจำกัดของโปรแกรม มีดังต่อไปนี้

- ผู้ใช้โปรแกรมควรมีความรู้ และความเข้าใจใน โครงสร้างเอกสาร XML เบื้องต้นเป็นอย่างน้อย
- เครื่องคอมพิวเตอร์ และเครื่องมือที่จะใช้ทำงานกับ โปรแกรมควรมีความต้องการดังนี้
  - CPU Pentium IV 1.6G
  - RAM 256M
  - JAVA Plug-in 1.4.0
  - Microsoft Internet Explorer Version 6
- โปรแกรมสามารถค้นหาข้อมูลภายในเอกสาร XML ได้ที่มีขนาดไม่เกิน 1.8 MB หรืออาจมากกว่าซึ่งขึ้นกับขนาดหน่วยความจำ และประสิทธิภาพของเครื่องคอมพิวเตอร์
- ความเร็ว ในการค้นหาผลลัพธ์จะขึ้นอยู่กับขนาดหน่วยความจำ และประสิทธิภาพของเครื่องคอมพิวเตอร์
- โปรแกรมไม่สามารถใช้คำสั่ง Order By ในการสร้างชุดคำสั่ง XQuery
- หน้าจอช่วยเหลือสำหรับผู้ใช้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery สามารถเลือกเอกสารที่ต้องการค้นหาได้ไม่เกิน 6 เอกสารต่อการค้นหาในแต่ละครั้ง
- หน้าจอช่วยเหลือสำหรับผู้ใช้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery จะไม่สามารถกำหนดเงื่อนไขการแสดงผลของผลลัพธ์

- หน้าจอช่วยเหลือสำหรับผู้ใช้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery จะไม่สามารถกำหนดเงื่อนไขสำหรับการจำกัดจำนวนข้อมูล
- หน้าจอช่วยเหลือสำหรับผู้ใช้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery จะไม่สามารถใช้สร้างฟังก์ชัน

## 4.2. ยูสเคสวิว

จากความต้องการในหัวข้อ 4.1 สามารถนำมาวิเคราะห์ในเชิงแนวคิด ซึ่งนำเสนอด้วยยูสเคสไดอะแกรม โดยมีขั้นตอนในการสร้างไดอะแกรมดังนี้

### 4.2.1. กำหนดแอกเตอร์

แอกเตอร์ และบทบาทหน้าที่ของแอกเตอร์ที่ติดต่อกับโปรแกรม มีดังนี้

- User: เป็นบุคคลที่เข้าใช้โปรแกรม เพื่อที่จะทำการค้นหาข้อมูลภายในเอกสาร XML ซึ่งจะมีบทบาทในการป้อนชุดคำสั่ง XQuery เข้าโปรแกรม ซึ่งอาจจะเป็นผู้ใช้ระดับนักพัฒนา (ผู้ใช้ที่มีความรู้ในการเขียนชุดคำสั่ง XQuery) หรือผู้ใช้ทั่วไป (ผู้ใช้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery)

### 4.2.2. กำหนดยูสเคส

กิจกรรมหลัก และรายละเอียดของกิจกรรมที่ทำงานภายในโปรแกรม มีดังนี้

- Create XQuery expression using editor  
ยูสเคสนี้สำหรับผู้ที่มีความรู้ในการเขียนชุดคำสั่ง XQuery เพื่อใช้เป็นหน้าจอในการสร้าง และ/หรือแก้ไขชุดคำสั่ง XQuery โดยผู้ใช้งานจะต้องเขียนชุดคำสั่ง XQuery ขึ้นเอง
- Create XQuery expression using builder  
ยูสเคสนี้สำหรับผู้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery เพื่อเป็นตัวช่วยในการเขียนชุดคำสั่ง ซึ่งสามารถทำการสร้าง และ/หรือแก้ไขชุดคำสั่งในรูปแบบกราฟฟิกที่เข้าใจง่าย และสุดท้ายโปรแกรมจะแปลงให้อยู่ในรูปแบบของชุดคำสั่ง XQuery
- Generate XML tree(s)  
ยูสเคสนี้มีหน้าที่อ่านโครงสร้างของเอกสาร XML ที่จะใช้ค้นหาข้อมูล แล้วนำมาสร้างในรูปแบบของลำดับขั้นต้นไม้ โดยแสดงในรูปแบบกราฟฟิกโครงสร้างต้นไม้ เพื่อช่วยผู้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery มองเห็นภาพของเอกสารก่อนค้นหาข้อมูล
- Parse XQuery expression

ยูสเคสนี้มีหน้าที่ตรวจสอบความถูกต้องของชุดคำสั่ง XQuery ที่ได้รับมาจากยูสเคส “Create XQuery expression using editor” หรือยูสเคส “Create XQuery expression using builder” และในกรณีที่ตรวจพบความไม่ถูกต้องของชุดคำสั่งจะทำการแสดงข้อความแจ้งกับผู้ใช้

- Compile XQuery expression to XQuery core

ยูสเคสนี้มีหน้าที่แปลงชุดคำสั่ง XQuery ให้อยู่ในรูปแบบของ XQuery Core เพื่อนำไปเข้ากระบวนการค้นหาข้อมูล

- Parse XML document(s)

ยูสเคสนี้มีหน้าที่ตรวจสอบความถูกต้องของเอกสาร XML ที่จะใช้ค้นหาข้อมูล และอ่านข้อมูลภายในเอกสารเข้ามาเก็บไว้ในหน่วยความจำ สำหรับในกรณีที่ตรวจพบความไม่ถูกต้องของเอกสาร XML จะทำการแสดงข้อความแจ้งกับผู้ใช้

- Evaluate query

ยูสเคสนี้มีหน้าที่ประมวลผล XQuery Core ซึ่งเป็นการค้นหาข้อมูลภายในเอกสาร XML จากที่ได้บันทึกไว้ในหน่วยความจำ

- Generate XML document

ยูสเคสนี้มีหน้าที่นำผลลัพธ์ที่ได้จากยูสเคส “Evaluate query” มาจัดให้อยู่ในรูปแบบของเอกสาร XML

#### 4.2.3. ลำดับเหตุการณ์ของยูสเคส

ลำดับเหตุการณ์ของในแต่ละกิจกรรม และรายละเอียด มีดังนี้

##### 4.2.3.1. ลำดับเหตุการณ์ของยูสเคส “Create XQuery expression using editor”

- เงื่อนไขก่อนเข้ายูสเคส

ไม่มี

- ลำดับเหตุการณ์ของกิจกรรมหลัก

โปรแกรมพร้อมรับผู้ใช้ป้อนชุดคำสั่ง XQuery (S-1) โดยผู้ใช้สามารถเลือกทำกิจกรรม NEW (S-2) หรือ OPEN (S-3) เมื่อผู้ใช้ทำงานกิจกรรม S-1 เรียบร้อยแล้ว (A-1) กดปุ่ม Execute เพื่อจบการทำงานของยูสเคสนี้

- ลำดับเหตุการณ์ของกิจกรรมย่อย

S-1: ผู้ใช้สร้าง และ/หรือแก้ไขชุดคำสั่ง XQuery

ผู้ใช้เขียนชุดคำสั่ง XQuery ในหน้าจอเอดิเตอร์ตามไวยากรณ์ของ XQuery

S-2: NEW window

โปรแกรมเคลียร์หน้าจอเอดิเตอร์พร้อมรับผู้ใช้ป้อนชุดคำสั่ง XQuery ชุดใหม่

S-3: OPEN file

ผู้ใช้เลือกเปิดไฟล์ชุดคำสั่ง XQuery ที่เคยบันทึกเอาไว้ (ไฟล์นามสกุล xquery) (A-2)

- ลำดับเหตุการณ์ของกิจกรรมทางเลือก

A-1: ผู้ใช้สามารถบันทึกชุดคำสั่ง XQuery ลงในจานบันทึกเฉพาะที่ และยังคงทำงานยูสเคสนี้ต่อได้ตามปกติ

A-2: นอกจากไฟล์นามสกุล xquery แล้วโปรแกรมยังสามารถเปิดเอกสาร XML ในจานบันทึกเฉพาะที่ (ไฟล์นามสกุล xml) หรือ XML สกิม่า (ไฟล์นามสกุล dtd และ xsd) แล้วแสดงข้อมูลในหน้าจอเอดิเตอร์

#### 4.2.3.2. ลำดับเหตุการณ์ของยูสเคส “Create XQuery expression using builder”

- เงื่อนไขก่อนเข้ายูสเคส

ไม่มี

- ลำดับเหตุการณ์ของกิจกรรมหลัก

โปรแกรมพร้อมให้ผู้ใช้เลือกเอกสาร XML ที่ต้องการค้นหา (S-1) โปรแกรมพร้อมให้ผู้ใช้เลือกโหนดที่ต้องการค้นหา และกำหนดเงื่อนไขของการค้นหาข้อมูล (S-2) เมื่อผู้ใช้ทำงานกิจกรรม (S-2) เรียบร้อยแล้ว โปรแกรมจะทำการแปลงให้อยู่ในรูปแบบของชุดคำสั่ง XQuery (S-3) และจบการทำงานของยูสเคสนี้

- ลำดับเหตุการณ์ของกิจกรรมย่อย

S-1: เลือกเอกสาร XML ที่ต้องการค้นหา

ผู้ใช้ใส่ชื่อเอกสาร XML พร้อมกับที่อยู่ของเอกสาร และโปรแกรมจะตรวจสอบความมีอยู่จริงของเอกสาร (A-1)

S-2: สร้าง และ/หรือแก้ไขการเลือก และการกำหนดเงื่อนไข

ผู้ใช้เลือกสิ่งที่ต้องการค้นหาจากโครงสร้างข้อมูลต้นไม้ที่โปรแกรมแสดง (A-2) และกำหนดเงื่อนไขสำหรับกรองข้อมูลเฉพาะที่ต้องการ

S-3: แปลงให้อยู่ในรูปแบบของชุดคำสั่ง XQuery

โปรแกรมนำข้อมูลที่ได้จากผู้ใช้มาทำการแปลงให้เป็นชุดคำสั่ง XQuery เพื่อส่งต่อไปยังยูสเคส “Parse XQuery expression” และจบการทำงานของยูสเคสนี้

- ลำดับเหตุการณ์ของกิจกรรมทางเลือก

A-1: เมื่อตรวจพบว่าไม่มีเอกสารอยู่จริงตามที่ผู้ใช้ป้อนเข้ามา โปรแกรมจะแสดงข้อความแจ้งผู้ใช้ และผู้ใช้สามารถเลือกทำงานกิจกรรม (S-1) ได้ใหม่อีกครั้ง หรือจบการทำงานของยูสเคสนี้

A-2: ผู้ใช้สามารถกำหนดฟังก์ชันให้กับ โหนดที่ต้องการค้นหาได้ และยังคงทำงานยูสเคสนี้ต่อไปได้ตามปกติ

#### 4.2.3.3. ลำดับเหตุการณ์ของยูสเคส “Generate XML tree(s)”

- เงื่อนไขก่อนเข้ายูสเคส

ผู้ใช้เลือกการทำงานค้นหาข้อมูลผ่านหน้าจอช่วยเหลือ และกำหนดเอกสาร XML ที่ผู้ใช้ต้องการค้นหา

- ลำดับเหตุการณ์ของกิจกรรมหลัก

โปรแกรมจะต้องอ่าน โครงสร้างข้อมูลภายในเอกสาร XML มาจัดแสดงเป็น โครงสร้างข้อมูลในรูปของลำดับขั้นต้นไม้ และจบการทำงานของยูสเคสนี้

- ลำดับเหตุการณ์ของกิจกรรมย่อย

ไม่มี

- ลำดับเหตุการณ์ของกิจกรรมทางเลือก

ไม่มี

#### 4.2.3.4. ลำดับเหตุการณ์ของยูสเคส “Parse XQuery expression”

- เงื่อนไขก่อนเข้ายูสเคส

จะต้องผ่านการทำงานจากยูสเคส “Create XQuery expression using editor” หรือยูสเคส “Create XQuery expression using builder” ก่อน

- ลำดับเหตุการณ์ของกิจกรรมหลัก

โปรแกรมทำการตรวจสอบความถูกต้องของไวยากรณ์ในชุดคำสั่ง XQuery (A-1) และจบการทำงานของยูสเคสนี้

- ลำดับเหตุการณ์ของกิจกรรมย่อย

ไม่มี

- ลำดับเหตุการณ์ของกิจกรรมทางเลือก

A-1: ตรวจพบชุดคำสั่ง XQuery ไม่ถูกต้องตามหลักไวยากรณ์ โปรแกรมจะแสดงข้อความแจ้งผู้ใช้ และจบการทำงานของยูสเคสนี้

#### 4.2.3.5. ลำดับเหตุการณ์ของยูสเคส “Compile XQuery expression to XQuery core”

- เงื่อนไขก่อนเข้ายูสเคส
 

จะต้องผ่านการทำงานจากยูสเคส “Parse XQuery expression” ก่อน เพราะต้องผ่านการตรวจสอบความถูกต้องของชุดคำสั่ง XQuery ว่ามีความถูกต้องทั้งหมดแล้วค่อยทำงานยูสเคสนี้
- ลำดับเหตุการณ์ของกิจกรรมหลัก
 

โปรแกรมนำชุดคำสั่ง XQuery มาแปลงให้อยู่ในรูปแบบของ XQuery Core และจบการทำงานของยูสเคสนี้
- ลำดับเหตุการณ์ของกิจกรรมย่อย
 

ไม่มี
- ลำดับเหตุการณ์ของกิจกรรมทางเลือก
 

ไม่มี

#### 4.2.3.6. ลำดับเหตุการณ์ของยูสเคส “Parse XML document(s)”

- เงื่อนไขก่อนเข้ายูสเคส
 

จะต้องผ่านการทำงานจากยูสเคส “Parse XQuery Expression” ก่อน เพราะเมื่อผ่านการตรวจสอบว่าไวยากรณ์ในชุดคำสั่ง XQuery ถูกต้องแล้ว ถึงจะทำงานยูสเคสนี้ เหตุที่ต้องผ่านยูสเคส “Parse XQuery Expression” ก่อนเนื่องจากโปรแกรมจะได้ไม่ต้องเสียเวลาทำงานยูสเคสนี้ทุกๆ ที่ชุดคำสั่ง XQuery ไม่ถูกต้อง
- ลำดับเหตุการณ์ของกิจกรรมหลัก
 

โปรแกรมอ่านชื่อเอกสาร XML ที่ต้องการค้นหา (S-1) โปรแกรมทำการตรวจสอบความถูกต้องของโครงสร้างข้อมูลในเอกสาร XML (A-2) อ่านข้อมูลภายในเอกสาร XML เก็บไว้ในหน่วยความจำ และจบการทำงานของยูสเคสนี้
- ลำดับเหตุการณ์ของกิจกรรมย่อย
 

S-1: โปรแกรมอ่านชื่อเอกสาร XML ที่ต้องการค้นหา

โปรแกรมอ่านชื่อเอกสาร XML พร้อมกับที่อยู่ของเอกสาร และตรวจสอบความมีอยู่จริงของเอกสาร (A-1)

- ลำดับเหตุการณ์ของกิจกรรมทางเลือก

A-1: เมื่อตรวจพบว่าไม่มีเอกสารอยู่จริง โปรแกรมจะแสดงข้อความแจ้งผู้ใช้ และจบการทำงานของยูสเคสนี้

A-2: เมื่อตรวจพบเอกสาร XML ไม่ถูกต้องตามคุณสมบัติ Well-Formed โปรแกรมจะแสดงข้อความแจ้งผู้ใช้ และจบการทำงานของยูสเคสนี้

#### 4.2.3.7. ลำดับเหตุการณ์ของยูสเคส “Evaluate query”

- เงื่อนไขก่อนเข้ายูสเคส

จะต้องผ่านการทำงานจากยูสเคส “Parse XML document(s)” และ “Compile XQuery expression to XQuery core” ก่อนตามลำดับ

- ลำดับเหตุการณ์ของกิจกรรมหลัก

โปรแกรมประมวลผลตามคำสั่งใน XQuery Core เพื่อค้นหาข้อมูลที่ต้องการจากเอกสาร XML ที่ได้นับที่เก็บไว้ในหน่วยความจำ และจบการทำงานของยูสเคสนี้

- ลำดับเหตุการณ์ของกิจกรรมย่อย

ไม่มี

- ลำดับเหตุการณ์ของกิจกรรมทางเลือก

ไม่มี

#### 4.2.3.8. ลำดับเหตุการณ์ของยูสเคส “Generate XML document”

- เงื่อนไขก่อนเข้ายูสเคส

จะต้องผ่านการทำงานจากยูสเคส “Evaluate query” ก่อน

- ลำดับเหตุการณ์ของกิจกรรมหลัก

โปรแกรมจะนำผลลัพธ์ที่หาได้จากยูสเคส “Evaluate query” มาจัดให้อยู่ในรูปแบบของเอกสาร XML ตามที่ผู้ใช้ได้กำหนดมาในชุดคำสั่ง XQuery (A-1) พร้อมกับแสดงผลในหน้าจอผลลัพธ์ (A-2) และจบการทำงานของยูสเคสนี้

- ลำดับเหตุการณ์ของกิจกรรมย่อย

ไม่มี

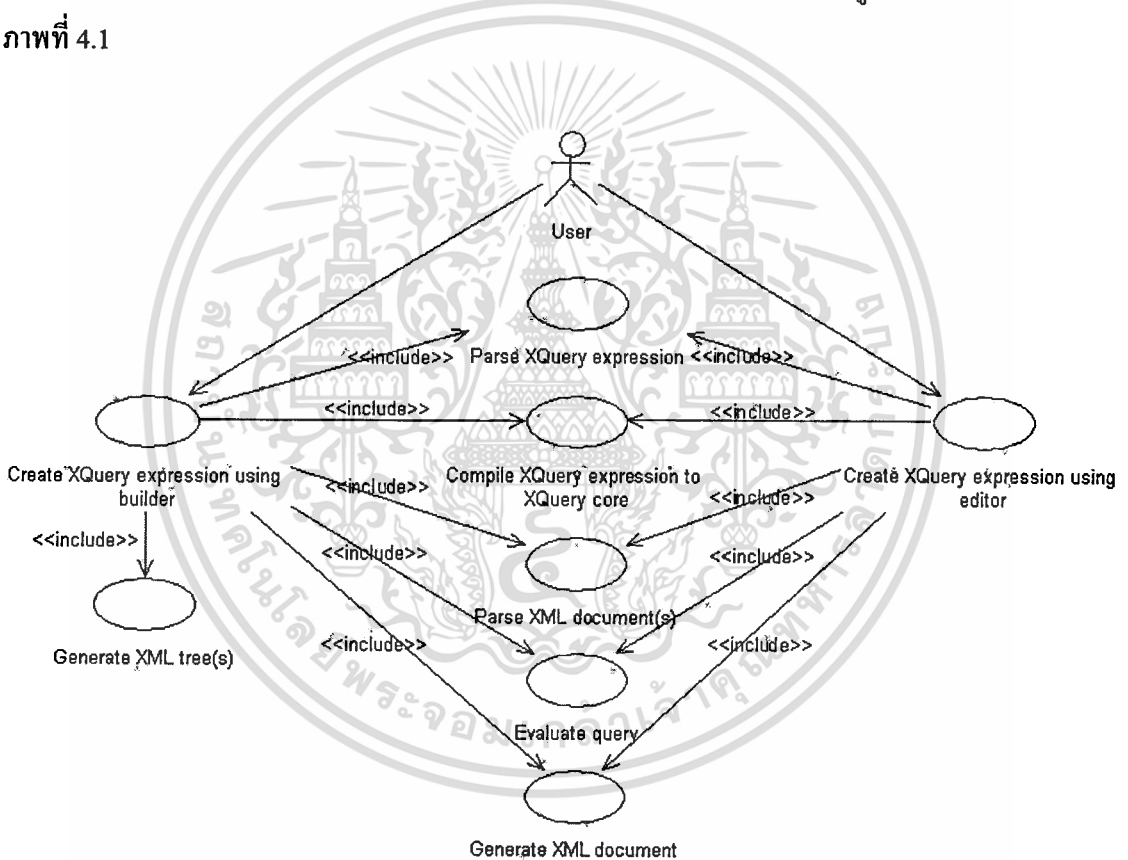
- ลำดับเหตุการณ์ของกิจกรรมทางเลือก

A-1: เมื่อตรวจพบว่ารูปแบบของเอกสาร XML ที่ผู้ใช้กำหนดมาในชุดคำสั่ง XQuery ไม่ถูกต้องตามคุณสมบัติ Well-Formed โปรแกรมจะแสดงข้อความแจ้งผู้ใช้ และจบการทำงานของยูสเคสนี้

A-2: ผู้ใช้สามารถบันทึกเอกสาร XML ผลลัพธ์เก็บไว้ในเครื่องของผู้ใช้

#### 4.2.4. ยูสเคสไดอะแกรม

จากที่ได้วิเคราะห์มาในหัวข้อ 4.2.1 ถึง 4.2.3 สามารถนำมาวาดเป็นยูสเคสไดอะแกรมได้ดังภาพที่ 4.1



ภาพที่ 4.1 ยูสเคสไดอะแกรม

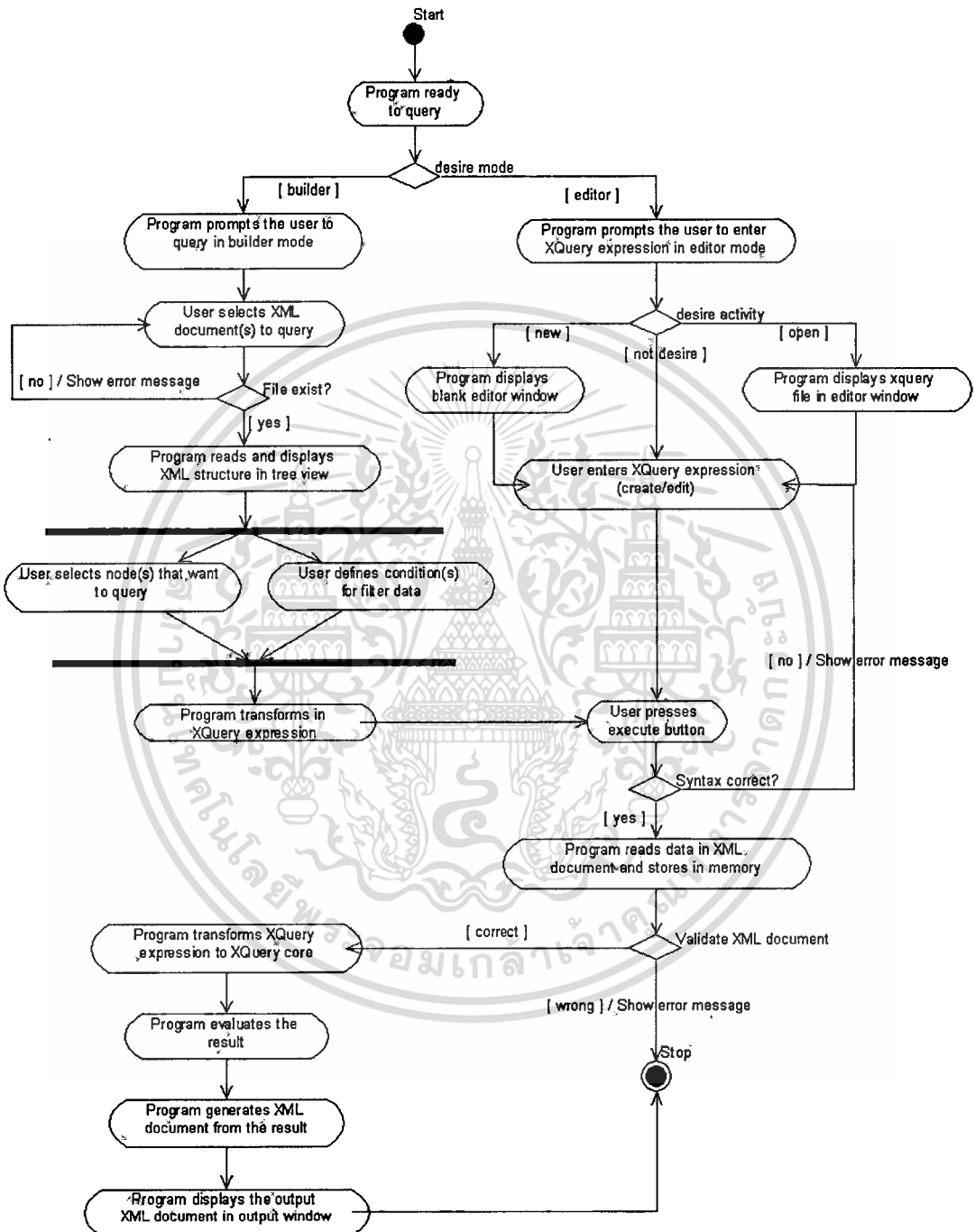
#### 4.2.5. แอคทิวิตีไดอะแกรม

จากที่ได้วิเคราะห์มาในหัวข้อ 4.2.3 สามารถนำมาเป็นพื้นฐาน และขยายรายละเอียดออกเป็นลำดับกิจกรรมตั้งแต่ต้นจนจบได้ดังนี้

- Program ready to query: เมื่อผู้ใช้เปิดโปรแกรมเริ่มใช้งาน โปรแกรมจะแสดงหน้าจอหลัก

- **Desire mode:** ผู้ใช้เลือกการทำงานระหว่างการใช้งานผ่านหน้าจอเอดิเตอร์ หรือหน้าจอช่วยเหลือ ซึ่งความแตกต่างของทั้งสองคือ หน้าจอเอดิเตอร์จะเหมาะกับผู้ใช้ที่มีความรู้ในการเขียนชุดคำสั่ง XQuery และหน้าจอช่วยเหลือจะเหมาะกับผู้ใช้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery
- ในกรณีที่ผู้ใช้เลือกทำงานผ่านหน้าจอช่วยเหลือจะมีลำดับกิจกรรมดังนี้
  - Program prompts the user to query in builder mode: โปรแกรมแสดงหน้าจอช่วยเหลือพร้อมให้ผู้ใช้งาน
  - User selects XML document(s) to query: ผู้ใช้กำหนดเอกสาร XML ที่ต้องการค้นหาข้อมูล
  - File exist?: โปรแกรมตรวจสอบว่าเอกสาร XML ที่ผู้ใช้กำหนดมีอยู่จริงหรือไม่ ถ้าไม่ ต้องกลับไปกำหนดเอกสาร XML ใหม่
  - Program reads and displays XML structure in tree view: โปรแกรมอ่านโครงสร้างข้อมูลในเอกสาร XML แล้วนำมาแปลงเป็นรูปแบบโครงสร้างต้นไม้
  - User selects node(s) that want to query: ผู้ใช้เลือกโหนดที่ต้องการค้นหา (กิจกรรมนี้สามารถทำก่อนหรือหลังกิจกรรม “User defines condition(s) for filter data” ก็ได้)
  - User defines condition(s) for filter data: ผู้ใช้กำหนดเงื่อนไขในการค้นหาข้อมูล (กิจกรรมนี้สามารถทำก่อนหรือหลังกิจกรรม “User selects node(s) that want to query” ก็ได้)
  - Program transforms in XQuery expression: โปรแกรมแปลงรูปแบบการค้นหาจากที่ผู้ใช้งานผ่านหน้าจอช่วยเหลือเป็นชุดคำสั่ง XQuery
- ในกรณีที่ผู้ใช้เลือกทำงานผ่านหน้าจอเอดิเตอร์จะมีลำดับกิจกรรมดังนี้
  - Program prompts the user to enter XQuery expression in editor mode: โปรแกรมแสดงหน้าจอเอดิเตอร์ว่างๆ พร้อมรับผู้ใช้งาน
  - Desire activity: ผู้ใช้เลือกที่จะเขียนชุดคำสั่ง XQuery ชุดใหม่ หรือเลือกที่จะเปิดไฟล์ชุดคำสั่ง XQuery (นามสกุล .xquery) จากที่เคยเขียนเก็บไว้
  - Program displays blank editor window: ในกรณีที่เลือก “Desire activity” เป็น new โปรแกรมแสดงหน้าจอเอดิเตอร์ว่างๆ พร้อมรับผู้ใช้ที่ต้องการจะเขียนชุดคำสั่ง XQuery ชุดใหม่

- Program displays xquery file in editor window: ในกรณีที่เลือก “Desire activity” เป็น open โปรแกรมแสดงชุดคำสั่ง XQuery ของไฟล์นามสกุล .xquery ที่เปิดในหน้าจอเอดิเตอร์
  - User enters XQuery expression (create/edit): ผู้ใช้ทำการป้อนชุดคำสั่ง XQuery ในหน้าจอเอดิเตอร์
    - User presses execute button: เมื่อผู้ใช้เขียนชุดคำสั่ง XQuery เรียบร้อยแล้ว (ไม่ว่าจะผ่านทางหน้าจอเอดิเตอร์ หรือหน้าจอช่วยเหลือ) ก็จะกดปุ่ม Execute เพิ่มเริ่มการทำงานค้นหาข้อมูล
    - Syntax correct?: โปรแกรมตรวจสอบไวยากรณ์ของชุดคำสั่ง XQuery และเมื่อพบว่าไม่ถูกต้อง โปรแกรมจะแสดงข้อความแจ้งผู้ใช้ให้ย้อนกลับไปแก้ไขในหน้าจอเอดิเตอร์ หรือหน้าจอช่วยเหลือ (แล้วแต่ว่าผู้ใช้ทำงานอยู่ในหน้าจอใด)
    - Program reads data in XML document and stores in memory: โปรแกรมอ่านข้อมูลทั้งหมดในเอกสาร XML และเก็บไว้ในหน่วยความจำ
    - Validate XML document: โปรแกรมตรวจสอบความถูกต้องของโครงสร้างเอกสาร XML ถ้าพบว่าไม่ถูกต้อง โปรแกรมจะแสดงข้อความแจ้งผู้ใช้ และจบการทำงานค้นหาข้อมูล
    - Program transforms XQuery expression to XQuery core: โปรแกรมแปลงชุดคำสั่ง XQuery ให้อยู่ในรูปแบบของ XQuery core
    - Program evaluates the result: โปรแกรมทำการค้นหาข้อมูลจากเอกสาร XML ตามที่ผู้ใช้กำหนด
    - Program generates XML document from the result: โปรแกรมนำข้อมูลที่ค้นหาได้มาจัดให้อยู่ในรูปแบบของเอกสาร XML
    - Program displays the output XML document in output window: โปรแกรมแสดงเอกสาร XML ผลลัพธ์ในหน้าจอผลลัพธ์
- จากที่วิเคราะห์ในข้างต้นสามารถนำมาวาดเป็นเอกทิวทัศน์ไดอะแกรมได้ดังภาพที่ 4.2



ภาพที่ 4.2 แอกทिवิตีไดอะแกรมของโปรแกรมค้นหาข้อมูลภายในเอกสาร XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3. ลอจิกอลวิว

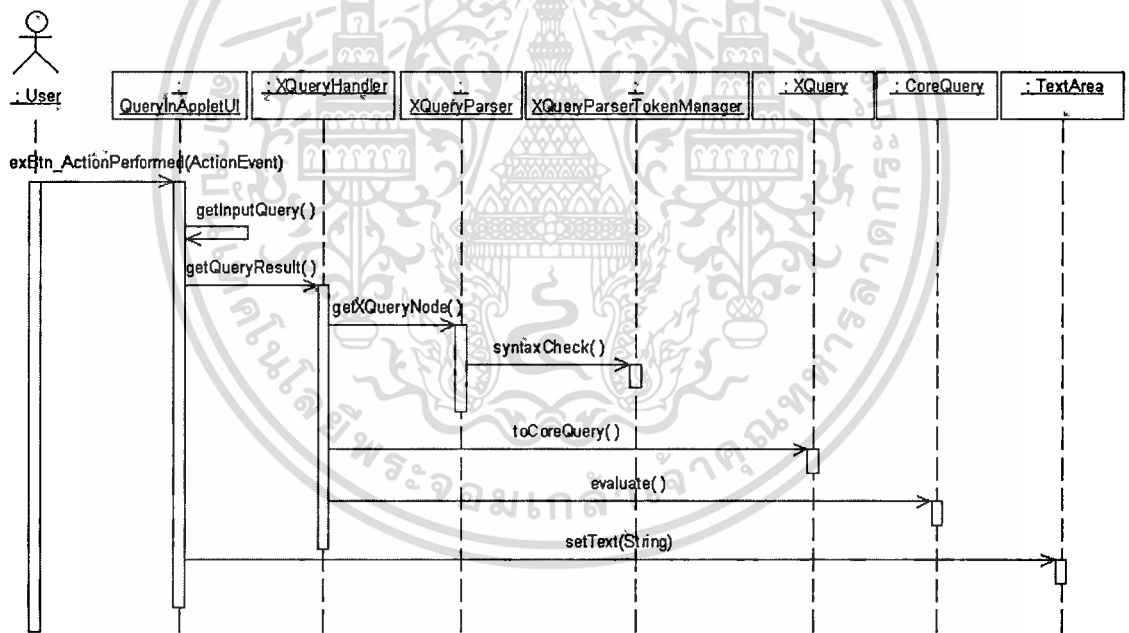
หลังจากที่ได้วิเคราะห์เชิงแนวคิดเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการวิเคราะห์ลงลึกในเชิงฟังก์ชันการทำงานของโปรแกรม เพื่อนำไปเป็นพื้นฐานในการเขียนโปรแกรม เพราะผลลัพธ์ที่ได้จากการวิเคราะห์ในขั้นตอนนี้จะได้เป็นข้อกำหนดคุณลักษณะของโปรแกรม (Program Specification)

#### 4.3.1. ซีเควนซ์ไดอะแกรม

ลำดับการทำงานในเชิงฟังก์ชันของโปรแกรม มีดังนี้

##### 4.3.1.1. ลำดับการทำงานของการค้นหาข้อมูล

ลำดับการทำงานของการค้นหาข้อมูลเริ่มต้นจากที่ผู้ใช้กดปุ่ม Execute และ โปรแกรมจะทำการค้นหาข้อมูลแล้วแสดงผลในหน้าจอผลลัพธ์ ดังภาพที่ 4.3



ภาพที่ 4.3 ซีเควนซ์ไดอะแกรมการค้นหาข้อมูล

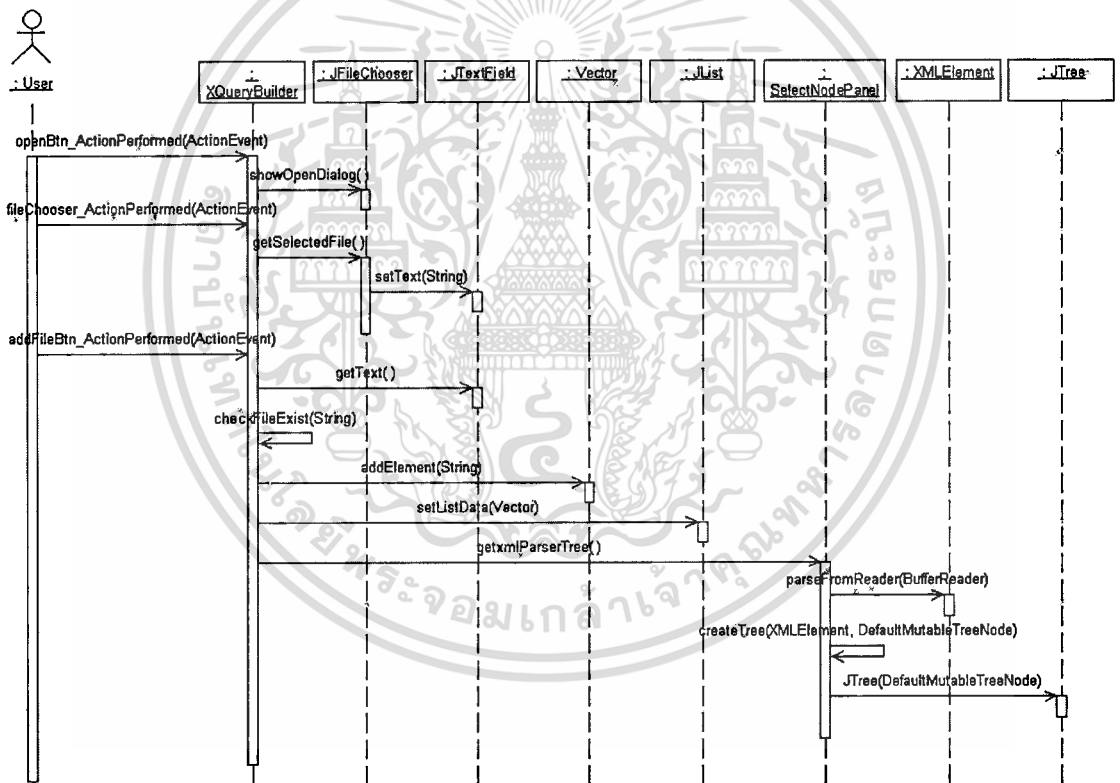
ขั้นตอนลำดับการทำงานของการค้นหาข้อมูลมีดังนี้

- เริ่มทำงานค้นหาข้อมูลเมื่อผู้ใช้กดปุ่ม Execute
- อ่านชุดคำสั่ง XQuery จากหน้าจอเอดิเตอร์
- ตรวจสอบความถูกต้องของชุดคำสั่ง XQuery

- แปลงชุดคำสั่ง XQuery ให้อยู่ในรูปแบบของ XQuery Core
- ค้นหาผลลัพธ์
- คืนค่าผลลัพธ์แสดงในหน้าจอผลลัพธ์

#### 4.3.1.2. ลำดับการทำงานของ การแสดงเอกสาร XML ในรูปแบบโครงสร้างต้นไม้

ลำดับการทำงานของ การแสดงเอกสาร XML ในรูปแบบ โครงสร้างต้นไม้เริ่มต้นจากที่ผู้ใช้เลือกเอกสาร XML ที่ต้องการค้นหา และกดปุ่ม “Add XML” โปรแกรมจะทำการอ่านข้อมูลในเอกสาร XML จากที่ผู้ใช้เลือก แล้วแปลงให้อยู่ในรูปแบบ โครงสร้างต้นไม้ ดังภาพที่ 4.4



ภาพที่ 4.4 ซีควเอนซ์ไดอะแกรมการแสดงผลเอกสาร XML ในรูปแบบโครงสร้างต้นไม้

ขั้นตอนลำดับการแสดงผลเอกสาร XML ในรูปแบบโครงสร้างต้นไม้มีดังนี้

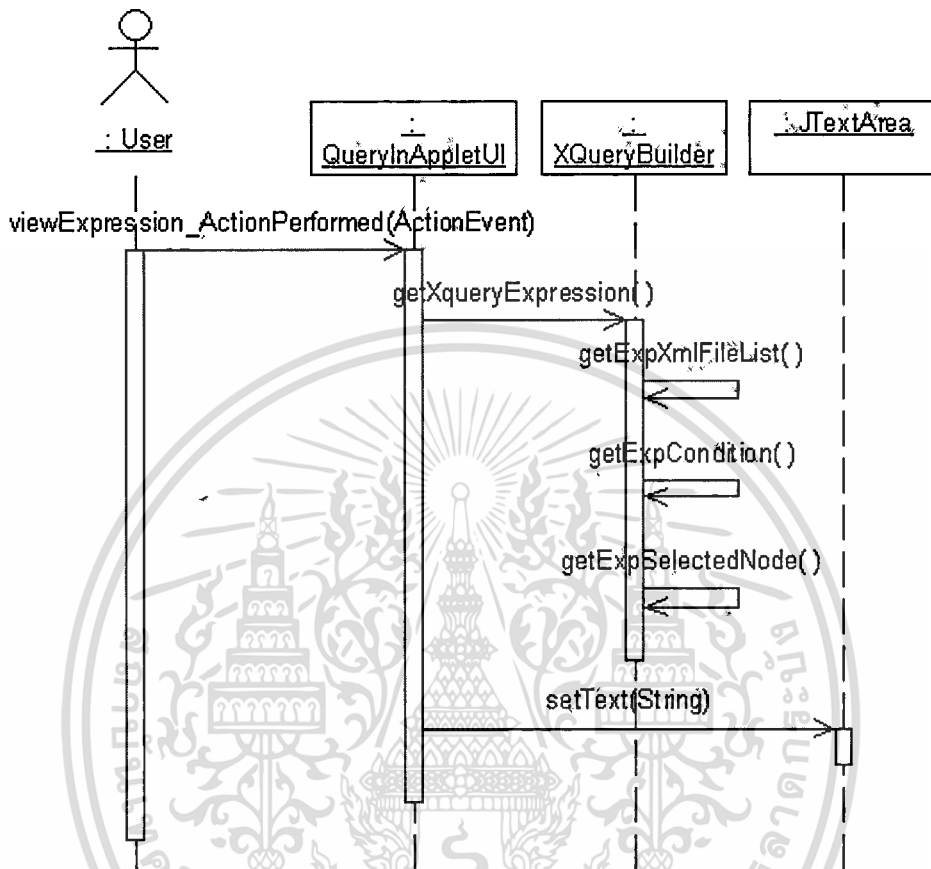
- ผู้ใช้กดปุ่ม Open เพื่อเลือกเอกสาร XML ที่ต้องการค้นหาข้อมูล
- แสดงไดอะล็อกสำหรับการเปิดไฟล์
- ผู้ใช้เลือกเอกสาร XML และกดปุ่ม Open ในไดอะล็อก

- แสดงชื่อไฟล์ที่ถูกเลือกในหน้าจอช่วยเหลือ
- ผู้ใช้คลิกปุ่ม “Add File” เพื่อเริ่มต้นการทำงานแสดงเอกสาร XML ในรูปแบบโครงสร้างต้นไม้
- อ่านชื่อไฟล์ที่ผู้ใช้เลือกไว้จากหน้าจอช่วยเหลือ
- ตรวจสอบความมีอยู่จริงของเอกสาร
- แสดงชื่อไฟล์ในรายการเอกสารที่ต้องการค้นหา
- อ่านเอกสาร XML แล้วแปลงเป็นรูปแบบโครงสร้างต้นไม้
- แสดงรูปแบบโครงสร้างต้นไม้ในหน้าจอช่วยเหลือ

#### 4.3.1.3. ลำดับการทำงานของการแปลงรูปแบบการค้นหาจากหน้าจอช่วยเหลือเป็นชุดคำสั่ง XQuery

ลำดับการทำงานของการแปลงรูปแบบการค้นหาจากหน้าจอช่วยเหลือเป็นชุดคำสั่ง XQuery เริ่มต้นจากที่ผู้ใช้คลิกปุ่ม “Show XQuery Expression” และโปรแกรมจะทำการแปลงข้อมูลการค้นหาจากที่ผู้ใช้ทำงานผ่านหน้าจอช่วยเหลือเป็นชุดคำสั่ง XQuery แล้วนำเสนอแสดงในหน้าจอเอดิเตอร์ ดังภาพที่ 4.5

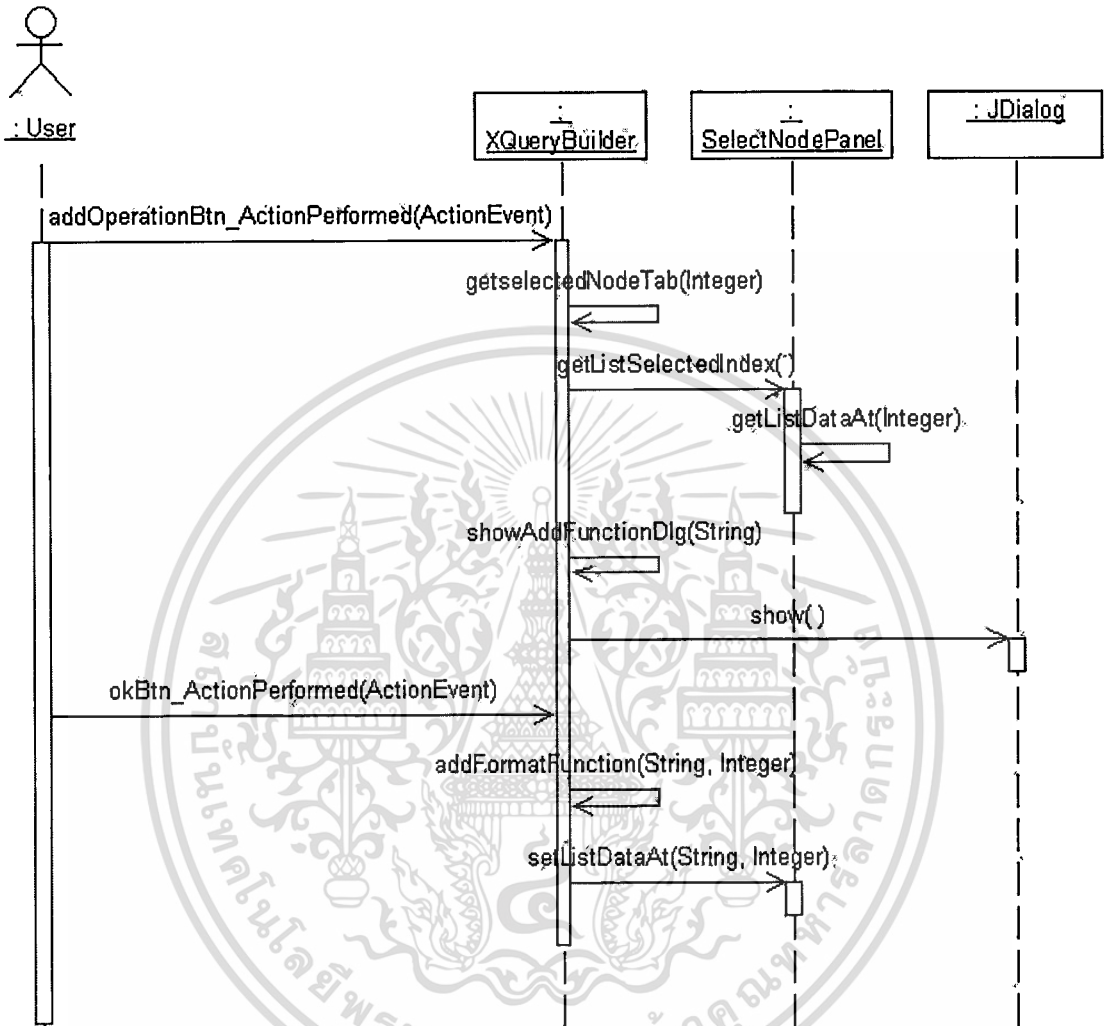
- ขั้นตอนลำดับการแปลงรูปแบบการค้นหาจากหน้าจอช่วยเหลือเป็นชุดคำสั่ง XQuery มีดังนี้
- เริ่มต้นทำงานเมื่อผู้ใช้คลิกปุ่ม “Show XQuery Expression”
  - อ่านรายการเอกสารจากที่ผู้ใช้เลือกไว้
  - อ่านรายการเงื่อนไขจากที่ผู้ใช้กำหนดไว้
  - อ่านรายการโหนดจากที่ผู้ใช้เลือกไว้
  - แปลงให้เป็นชุดคำสั่ง XQuery
  - แสดงชุดคำสั่ง XQuery ในหน้าจอเอดิเตอร์



ภาพที่ 4.5 ซีควเอนซ์โคอะเกรมการแปลงรูปแบบการค้นหาคำสั่ง XQuery

#### 4.3.1.4. ลำดับการทำงานของกาหนดฟังก์ชันในหน้าจอช่วยเหลือ

ลำดับการทำงานของกาหนดฟังก์ชันในหน้าจอช่วยเหลือเริ่มต้นจากที่ผู้ใช้เลือก โหนดที่ต้องการกาหนดฟังก์ชัน และกดปุ่ม “Operation with selected node” โปรแกรมจะแสดงโคอะลือก “Add function to field” เพื่อให้ผู้ใช้กาหนดฟังก์ชันตามที่ต้องการ แล้วนำแสดงโหนดที่ถูกกาหนดฟังก์ชันเรียบร้อยแล้วในหน้าจอช่วยเหลือ ดังภาพที่ 4.6



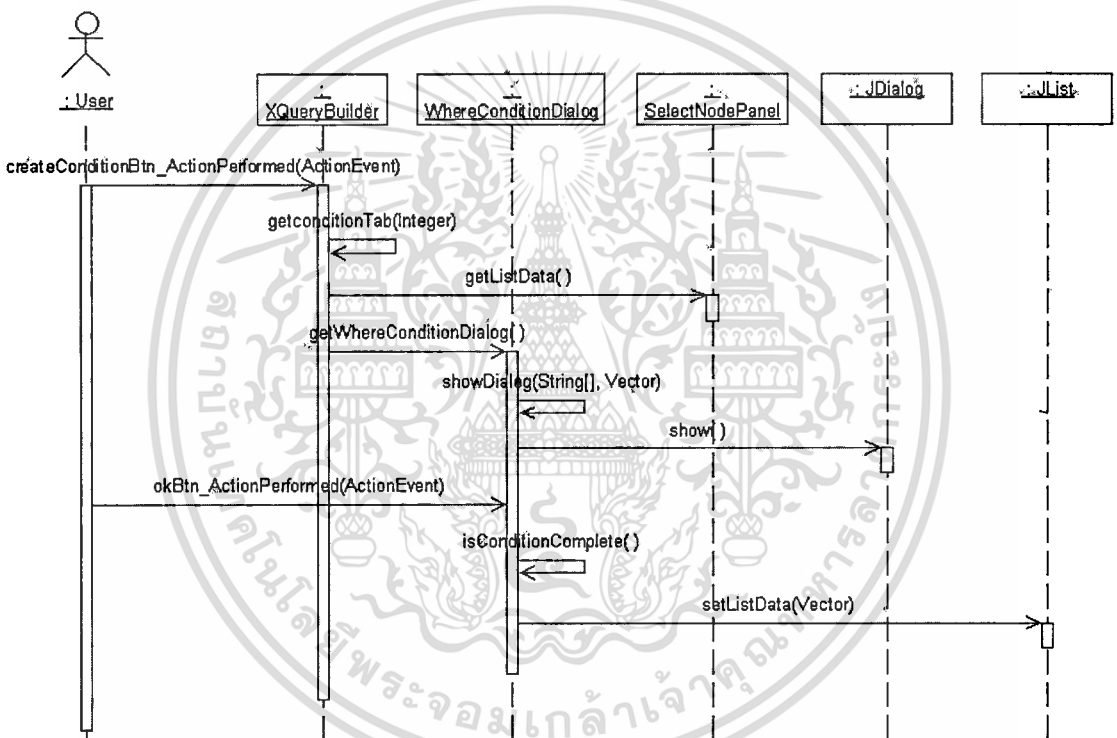
ภาพที่ 4.6 ซีควเอนซ์ไดอะแกรมการกำหนดฟังก์ชันในหน้าจอช่วยเหลือ

ขั้นตอนลำดับการกำหนดฟังก์ชันในหน้าจอช่วยเหลือมีดังนี้

- เริ่มต้นทำงานเมื่อผู้ใช้กดปุ่ม “Operation with selected node”
- อ่านโหนดจากที่ผู้ใช้เลือกไว้เพื่อกำหนดฟังก์ชัน
- แสดงไดอะล็อกสำหรับการกำหนดฟังก์ชัน
- ผู้ใช้เลือกฟังก์ชันที่ต้องการ และกดปุ่ม OK ในไดอะล็อก
- กำหนดฟังก์ชันให้กับ โหนดที่ผู้ใช้เลือกไว้
- แสดงโหนดที่กำหนดฟังก์ชันแล้วในรายการ โหนด

#### 4.3.1.5. ลำดับการทำงานของกาหนดเงื่อนไขในหน้าจอช่วยเหลือ

ลำดับการทำงานของกาหนดเงื่อนไขในหน้าจอช่วยเหลือเริ่มต้นจากที่ผู้ใช้กดปุ่ม “Create condition” หรือ “Edit condition” โปรแกรมจะแสดงไดอะล็อก “Add or Edit Condition” เพื่อให้ผู้ใช้กำหนดเงื่อนไขในการกรองข้อมูลตามที่ต้องการ แล้วนำแสดงเงื่อนไขการกรองข้อมูลในหน้าจอช่วยเหลือ ดังภาพที่ 4.7



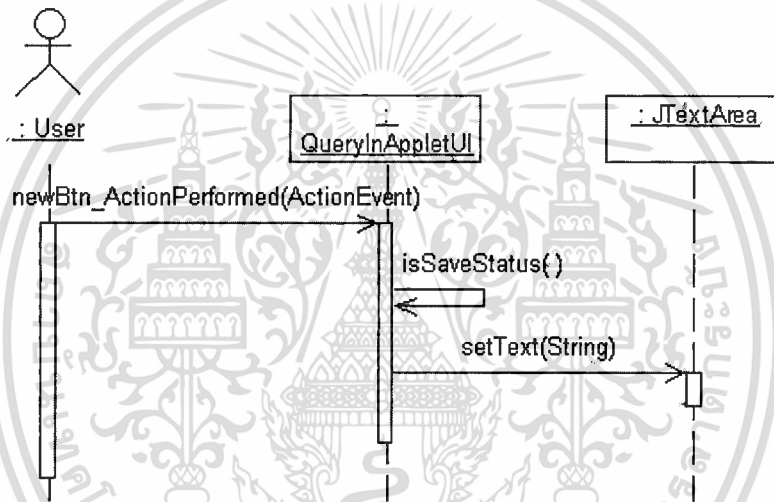
ภาพที่ 4.7 ซีเควนซ์ไดอะแกรมการกำหนดเงื่อนไขในหน้าจอช่วยเหลือ

ขั้นตอนลำดับการกำหนดเงื่อนไขในหน้าจอช่วยเหลือมีดังนี้

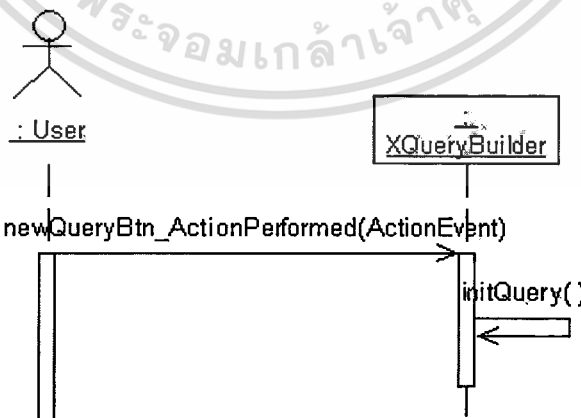
- เริ่มต้นทำงานเมื่อผู้ใช้กดปุ่ม “Create condition” หรือ “Edit condition”
- อ่าน โหนดทั้งหมดในรายการที่ผู้ใช้เลือกไว้เพื่อต้องการกำหนดเงื่อนไข
- แสดง ไดอะล็อกสำหรับการกำหนดเงื่อนไข
- ผู้ใช้กำหนดเงื่อนไขตามที่ต้องการ และกดปุ่ม OK ในไดอะล็อก
- ตรวจสอบการกำหนดเงื่อนไขว่าครบถ้วนหรือไม่
- แสดงชุดคำสั่งที่กำหนดเงื่อนไขเรียบร้อยแล้วในรายการเงื่อนไข

#### 4.3.1.6 ลำดับการทำงานของการเคลียร์หน้าจอเอดิเตอร์ และหน้าจอช่วยเหลือเพื่อรองรับชุดคำสั่ง XQuery ชุดใหม่

ลำดับการทำงานของการเคลียร์หน้าจอเอดิเตอร์ และหน้าจอช่วยเหลือเพื่อรองรับชุดคำสั่ง XQuery ชุดใหม่เริ่มต้นจากที่ผู้ใช้กดปุ่ม “New XQuery” หรือคปุ่ม “New XQuery Builder” โปรแกรมจะเคลียร์หน้าจอเอดิเตอร์ หรือหน้าจอช่วยเหลือ (แล้วแต่ผู้ใช้กดปุ่ม New จากหน้าจอใด) เพื่อรองรับชุดคำสั่ง XQuery ชุดใหม่ ดังภาพที่ 4.8 และภาพที่ 4.9



ภาพที่ 4.8 ซีควเอนซ์ไดอะแกรมการเคลียร์หน้าจอเอดิเตอร์



ภาพที่ 4.9 ซีควเอนซ์ไดอะแกรมการเคลียร์หน้าจอช่วยเหลือ

ขั้นตอนลำดับการเคลียร์หน้าจอเอดิเตอร์มีดังนี้

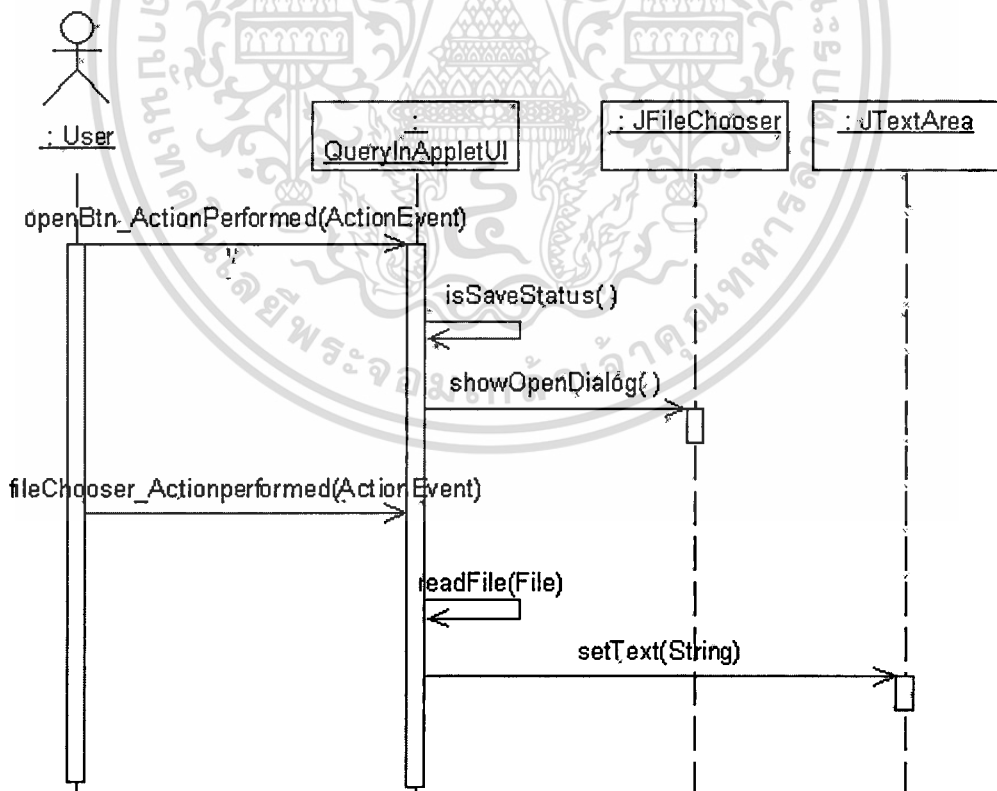
- เริ่มต้นทำงานเมื่อผู้ใช้คลิกปุ่ม “New XQuery”
- ตรวจสอบว่าชุดคำสั่ง XQuery ที่กำลังทำงานอยู่ถูกบันทึกแล้วหรือยัง
- แสดงหน้าจอเอดิเตอร์ว่างเพื่อรอรับชุดคำสั่ง XQuery ชุดต่อไป

ขั้นตอนลำดับการเคลียร์หน้าจอช่วยเหลือมีดังนี้

- เริ่มต้นทำงานเมื่อผู้ใช้คลิกปุ่ม “New XQuery Builder”
- เคลียร์ค่าตัวแปร และรายการต่างๆ
- แสดงหน้าจอช่วยเหลือว่างเพื่อรอรับชุดคำสั่ง XQuery ชุดต่อไป

#### 4.3.1.7. ลำดับการทำงานของการเปิดไฟล์

ลำดับการทำงานของการเปิดไฟล์เริ่มต้นจากที่ผู้ใช้คลิกปุ่ม Open และ โปรแกรมจะทำการอ่านข้อมูลแล้วนำมาแสดงในหน้าจอเอดิเตอร์ ดังภาพที่ 4.10



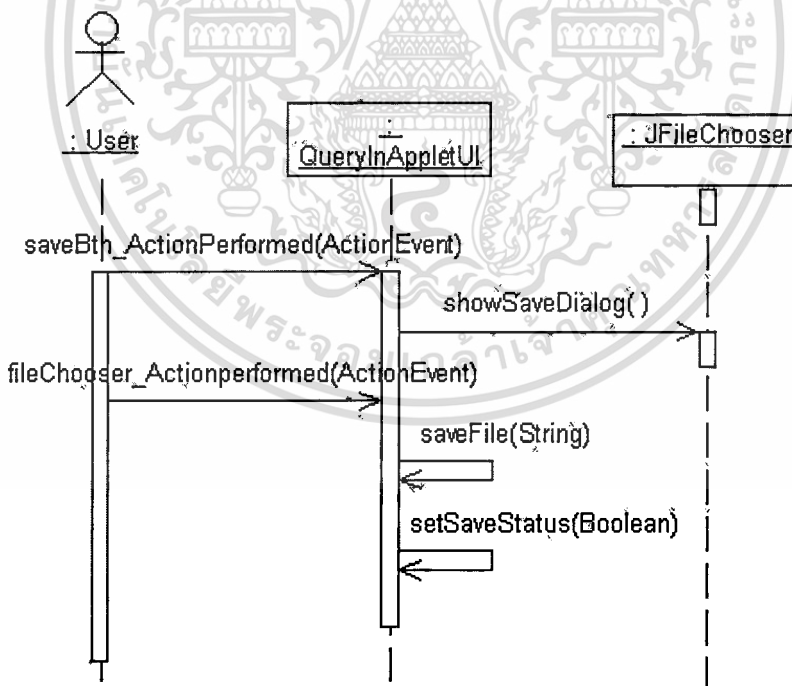
ภาพที่ 4.10 ซีเควนซ์ไดอะแกรมการเปิดไฟล์

ขั้นตอนลำดับการเปิดไฟล์มีดังนี้

- เริ่มต้นทำงานเมื่อผู้ใช้กดปุ่ม Open
- ตรวจสอบว่าชุดคำสั่ง XQuery ที่กำลังทำงานอยู่ถูกบันทึกแล้วหรือยัง
- แสดงไดอะล็อกสำหรับการเปิดไฟล์
- ผู้ใช้เลือกไฟล์ที่ต้องการ และกดปุ่ม Open ในไดอะล็อก
- อ่านไฟล์ที่ผู้ใช้เลือกไว้
- แสดงข้อมูลจากไฟล์ในหน้าจอเอดิเตอร์

#### 4.3.1.8. ลำดับการทำงานของการบันทึกไฟล์

ลำดับการทำงานของการบันทึกไฟล์เริ่มต้นจากที่ผู้ใช้กดปุ่ม Save แล้วโปรแกรมจะทำการบันทึกข้อมูลจากหน้าจอเอดิเตอร์ หรือหน้าจอผลลัพธ์ (แล้วแต่ว่าผู้ใช้ต้องการบันทึกข้อมูลในหน้าจอใด) ลงในที่เก็บข้อมูล ดังภาพที่ 4.11



ภาพที่ 4.11 ซีควเอนซ์ไดอะแกรมการบันทึกไฟล์

ขั้นตอนลำดับการบันทึกไฟล์มีดังนี้

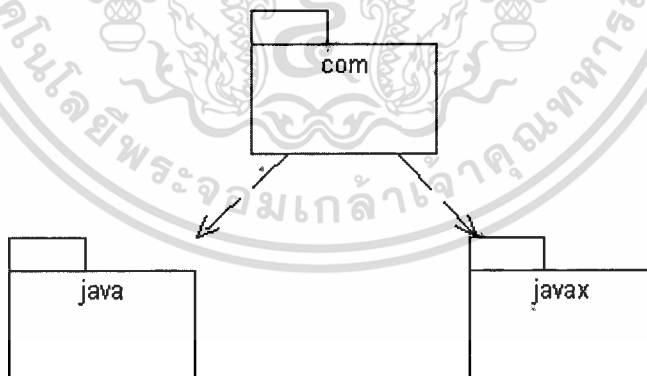
- เริ่มต้นทำงานเมื่อผู้ใช้กดปุ่ม Save
- แสดงไดอะล็อกสำหรับการบันทึกข้อมูล
- ผู้ใช้ตั้งชื่อไฟล์ที่ต้องการบันทึก และกดปุ่ม Save ในไดอะล็อก
- บันทึกข้อมูลจากหน้าจอเอดิเตอร์ หรือหน้าจอผลลัพธ์ (แล้วแต่ว่าผู้ใช้ต้องการบันทึกข้อมูล ในหน้าจอใด) ลงในที่เก็บข้อมูล

#### 4.3.2. คลาสไดอะแกรม

การวิเคราะห์ห่ออกแบบในขั้นต่อไปจะเป็นการนำซีควেনซ์ไดอะแกรมมาจัดเรียงให้อยู่ในรูปแบบคลาส เพื่อนำเสนอความสัมพันธ์ และรวบรวมเมธอดที่ได้นำเสนอไว้ในซีควেনซ์ไดอะแกรมในรูปแบบคลาส โดยมีการจัดกลุ่มรวบรวมคลาสเป็นแพ็คเกจเพื่อให้ง่ายต่อการจัดการดูแล คลาสไดอะแกรมในโปรแกรม มีดังนี้

##### 4.3.2.1. คลาสไดอะแกรมของโปรแกรม

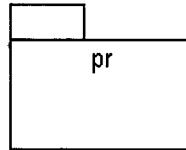
คลาสไดอะแกรมของโปรแกรมในมุมมองกว้าง ซึ่งแสดงเป็นแพ็คเกจที่รวบรวมคลาสต่างๆ โดยแบ่งเป็น 3 แพ็คเกจ คือ แพ็คเกจ com รวบรวมคลาสของโปรแกรม แพ็คเกจ java และแพ็คเกจ javax รวบรวมคลาสมาตรฐานของจาวาที่โปรแกรมเรียกใช้ ดังภาพที่ 4.12



ภาพที่ 4.12 คลาสไดอะแกรมของโปรแกรม

##### 4.3.2.2. คลาสไดอะแกรมของแพ็คเกจ com

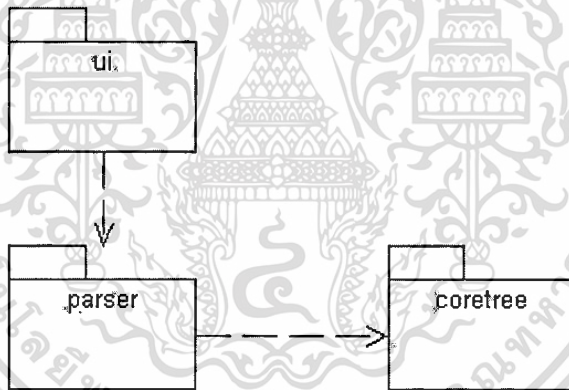
คลาสไดอะแกรมของแพ็คเกจ com มีแพ็คเกจย่อยคือ pr ซึ่งเป็นแพ็คเกจที่รวบรวมคลาสของโปรแกรม ดังภาพที่ 4.13



ภาพที่ 4.13 คลาสไดอะแกรมของแพ็คเกจ com

#### 4.3.2.3. คลาสไดอะแกรมของแพ็คเกจ pr

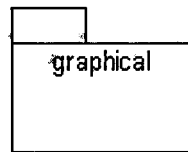
คลาสไดอะแกรมของแพ็คเกจ pr มีแพ็คเกจย่อย 3 แพ็คเกจ คือ แพ็คเกจ ui รวบรวมคลาสที่เกี่ยวข้องกับอินเทอร์เฟซของโปรแกรม แพ็คเกจ parser รวบรวมคลาสที่เกี่ยวกับการควบคุมการค้นหาผลลัพธ์ และแพ็คเกจ coretree รวบรวมคลาสที่เกี่ยวกับการค้นหาผลลัพธ์ ดังภาพที่ 4.14



ภาพที่ 4.14 คลาสไดอะแกรมของแพ็คเกจ pr

#### 4.3.2.4. คลาสไดอะแกรมของแพ็คเกจ ui

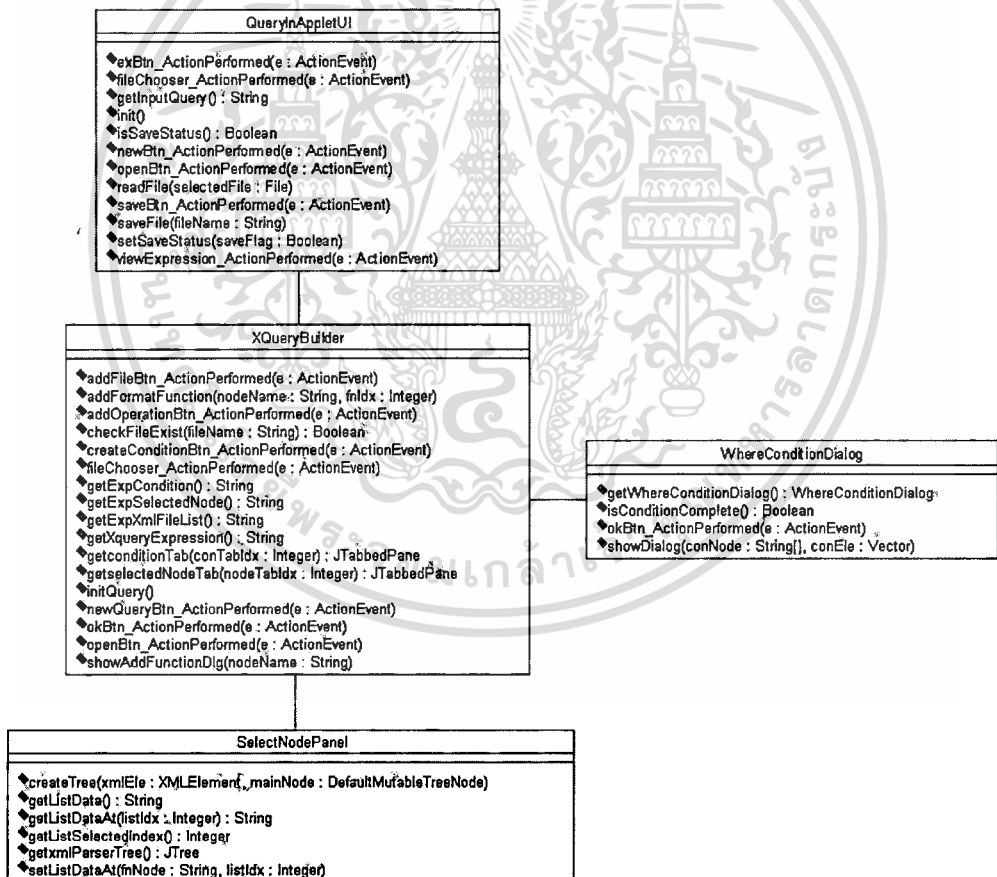
คลาสไดอะแกรมของแพ็คเกจ ui มีแพ็คเกจย่อย 1 แพ็คเกจ คือ แพ็คเกจ graphical รวบรวมคลาสที่เกี่ยวข้องกับอินเทอร์เฟซของโปรแกรม ดังภาพที่ 4.15



ภาพที่ 4.15 คลาสไดอะแกรมของแพ็คเกจ ui

#### 4.3.2.5. คลาสไดอะแกรมของแพ็คเกจ graphical

คลาสไดอะแกรมของแพ็คเกจ graphical ประกอบด้วย 4 คลาส คือ คลาส QueryInAppletUI, คลาส XQueryBuilder, คลาส SelectNodePanel และคลาส WhereConditionDialog ดังภาพที่ 4.16



ภาพที่ 4.16 คลาสไดอะแกรมของแพ็คเกจ graphical

สำหรับรายละเอียดของแต่ละคลาสนี้

- คลาส `com.pr.ui.graphical.QueryInAppletUI` ประกอบด้วยเมธอด ดังต่อไปนี้
  - `exBtn_ActionPerformed(ActionEvent e)`  
 รองรับเหตุการณ์จากการกดปุ่ม Execute
  - `fileChooser_ActionPerformed(ActionEvent e)`  
 รองรับเหตุการณ์จากการกดปุ่มในไดอะล็อก
  - `getInputQuery()`  
 เก็บชุดคำสั่ง XQuery จากหน้าจอเอดิเตอร์  
 คืนค่าเป็น สตริง
  - `init()`  
 กำหนดค่าเริ่มต้นเมื่อแอปพลิเคชันเริ่มทำงาน
  - `isSaveStatus()`  
 ตรวจสอบสถานะการบันทึกข้อมูล  
 คืนค่าเป็น บูลีน
  - `newBtn_ActionPerformed(ActionEvent e)`  
 รองรับเหตุการณ์จากการกดปุ่ม “New XQuery”
  - `openBtn_ActionPerformed(ActionEvent e)`  
 รองรับเหตุการณ์จากการกดปุ่ม Open
  - `readFile(File selectedFile)`  
 อ่านข้อมูลในไฟล์
  - `saveBtn_ActionPerformed(ActionEvent e)`  
 รองรับเหตุการณ์จากการกดปุ่ม Save
  - `saveFile(String fileName)`  
 บันทึกข้อมูลลงไฟล์
  - `setSaveStatus(Boolean saveFlag)`  
 กำหนดสถานะการบันทึกข้อมูล
  - `viewExpression_ActionPerformed(ActionEvent e)`  
 รองรับเหตุการณ์จากการกดปุ่ม “Show XQuery Expression”
- คลาส `com.pr.ui.graphical.XQueryBuilder` ประกอบด้วยเมธอด ดังต่อไปนี้

`addFileBtn_ActionPerformed(ActionEvent e)`

รองรับเหตุการณ์จากการกดปุ่ม “Add file”

`addFormatFunction(String nodeName, Integer fnIdx)`

กำหนดฟังก์ชันให้กับ โหนดตามฟอร์แมตที่ถูกต้อง

`checkFileExist(String fileName)`

ตรวจสอบความมีอยู่จริงของไฟล์

คืนค่าเป็น บูลีน

`createConditionBtn_ActionPerformed(ActionEvent e)`

รองรับเหตุการณ์จากการกดปุ่ม “Create condition”

`fileChooser_ActionPerformed(ActionEvent e)`

รองรับเหตุการณ์จากการกดปุ่มใน โคะะล็อก

`getExpCondition()`

นำแต่ละชุดคำสั่งที่เป็นเงื่อนไขในรายการเงื่อนไขมาต่อเป็นชุดเดียวกัน โดยค้นแต่ละชุดคำสั่งเงื่อนไขด้วยการขึ้นบรรทัดใหม่

คืนค่าเป็น สตริง

`getExpSelectedNode()`

นำแต่ละโหนดในรายการโหนดมาต่อเป็นชุดเดียวกัน โดยค้นแต่ละโหนดด้วยเครื่องหมายคอมมา “,”

คืนค่าเป็น สตริง

`getExpXmlFileList()`

นำแต่ละไฟล์ XML ในรายการไฟล์มาทำเป็นชุดคำสั่ง FOR หรือ LET (ตามที่ผู้ใช้เลือกใน “Loop at result”) โดยค้นแต่ละชุดคำสั่ง FOR หรือ LET ด้วยการขึ้นบรรทัดใหม่

คืนค่าเป็น สตริง

`getXQueryExpression()`

นำผลลัพธ์ที่ได้คืนค่ามาจากเมธอด `getExpCondition`, `getExpSelectedNode` และ `getExpXmlFileList` มาต่อกันให้เป็นชุดคำสั่ง XQuery

คืนค่าเป็น สตริง

getConditionTab(Integer conTabIdx)

เก็บค่ารายการเงื่อนไข

คืนค่าเป็น JTabbedPane

getSelectedNodeTab(Integer nodeTabIdx)

เก็บค่ารายการโหนด

คืนค่าเป็น JTabbedPane

initQuery()

กำหนดค่าเริ่มต้นของหน้าจอช่วยเหลือ

newQueryBtn\_ActionPerformed(ActionEvent e)

รองรับเหตุการณ์จากการกดปุ่ม “New XQuery Builder”

okBtn\_ActionPerformed(ActionEvent e)

รองรับเหตุการณ์จากการกดปุ่ม OK ในไดอะล็อก

openBtn\_ActionPerformed(ActionEvent e)

รองรับเหตุการณ์จากการกดปุ่ม Open

showAddFunctionDlg(String nodeName)

นำโหนดไปแสดงในไดอะล็อกกำหนดฟังก์ชัน

- คลาส com.pr.ui.graphical.SelectNodePanel ประกอบด้วยเมธอด ดังต่อไปนี้

createTree(XMLElement xmlEle, DefaultMutableTreeNode mainNode)

นำแต่ละอิลิเมนต์ของเอกสาร XML มาสร้างเป็นโหนดในโครงสร้างต้นไม้

getListData()

เก็บค่าข้อมูลทั้งหมดในรายการเงื่อนไข

คืนค่าเป็น สตริง

getListDataAt(Integer listIdx)

เก็บค่าข้อมูลแบบเจาะจงตำแหน่งในรายการโหนด

คืนค่าเป็น สตริง

getListSelectedIndex()

เก็บค่าตำแหน่งในรายการโหนด

คืนค่าเป็น Integer

getxmlParserTree()

นำข้อมูลในเอกสาร XML มาเป็นสร้างโครงสร้างต้นไม้  
คืนค่าเป็น JTree

setListDataAt(String fnNode, Integer listIdx)

กำหนดโหนดที่ผ่านการกำหนดฟังก์ชันแล้วในรายการโหนด

- คลาส com.pr.ui.graphical.WhereConditionDialog ประกอบด้วยเมธอด ดังต่อไปนี้

getWhereConditionDialog()

เก็บรายการเงื่อนไขนำไปสร้างไดอะล็อกกำหนดเงื่อนไข

คืนค่าเป็น WhereConditionDialog

isConditionComplete()

ตรวจสอบความครบถ้วนของการกำหนดเงื่อนไข

คืนค่าเป็น บูลีน

okBtn\_ActionPerformed(ActionEvent e)

รองรับเหตุการณ์จากการกดปุ่ม OK ในไดอะล็อก

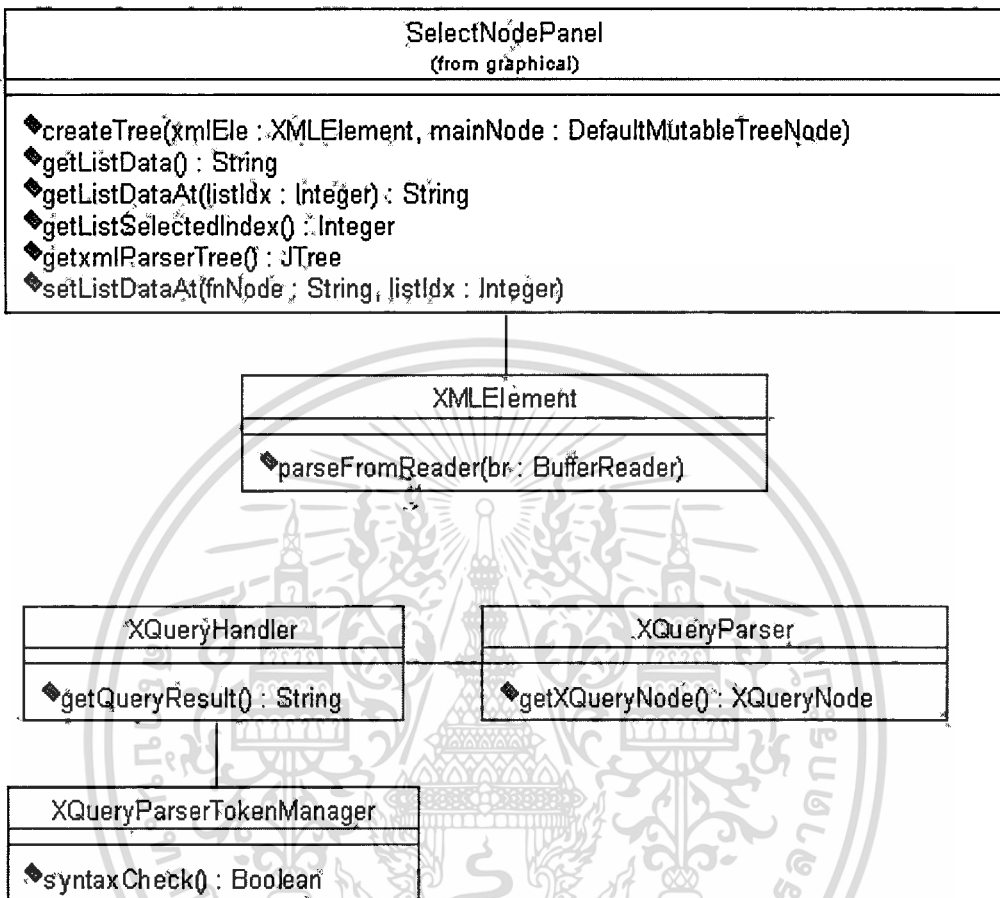
showDialog(String[] conNode, Vector conEle)

นำรายการเงื่อนไขทั้งหมดไปแสดงในไดอะล็อกกำหนดเงื่อนไข

#### 4.3.2.6. คลาสไดอะแกรมของแพ็คเกจ parser

คลาสไดอะแกรมของแพ็คเกจ parser ประกอบด้วย 5 คลาส คือ คลาส XMLParser, คลาส XMLElement, คลาส XQueryHandler, คลาส XQueryParser และคลาส XQueryParserTokenManager ดังภาพที่ 4.17

\*



ภาพที่ 4.17 คลาสโคดอะแกรมของแพ็คเกจ parser

สำหรับรายละเอียดของแต่ละคลาสมีดังนี้

- คลาส `com.pr.parser.XMLElement` ประกอบด้วยเมธอด ดังต่อไปนี้  
`parseFromReader(BufferedReader br)`  
 อ่านข้อมูลในเอกสาร XML
- คลาส `com.pr.parser.XQueryHandler` ประกอบด้วยเมธอด ดังต่อไปนี้  
`getResult()`  
 นำชุดคำสั่ง XQuery ไปหาผลลัพธ์  
 คืนค่าเป็น สตริง
- คลาส `com.pr.parser.XQueryParser` ประกอบด้วยเมธอด ดังต่อไปนี้  
`getXQueryNode()`

อ่านชุดคำสั่ง XQuery

คืนค่าเป็น XQueryNode

- คลาส `com.pr.parser.XQueryParserTokenManager` ประกอบด้วยเมธอด ดังต่อไปนี้  
`syntaxCheck()`

ตรวจสอบความถูกต้องของชุดคำสั่ง XQuery

คืนค่าเป็น บูลีน

#### 4.3.2.7. คลาสไดอะแกรมของแพ็คเกจ `coretree`

คลาสไดอะแกรมของแพ็คเกจ `coretree` ประกอบด้วย 2 คลาส คือ คลาส `XQuery` และคลาส `CoreQuery` ดังภาพที่ 4.18

สำหรับรายละเอียดของแต่ละคลาสมีดังนี้

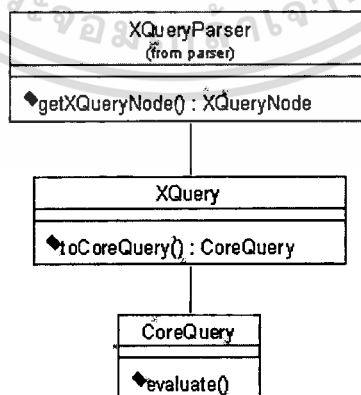
- คลาส `com.pr.coretree.XQuery` ประกอบด้วยเมธอด ดังต่อไปนี้  
`toCoreQuery()`

แปลงชุดคำสั่ง XQuery ให้อยู่ในรูปแบบ XQuery Core

คืนค่าเป็น `CoreQuery`

- คลาส `com.pr.coretree.CoreQuery` ประกอบด้วยเมธอด ดังต่อไปนี้  
`evaluate()`

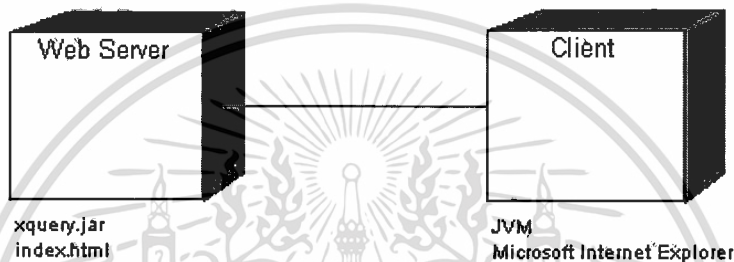
ค้นหาผลลัพธ์



ภาพที่ 4.18 คลาสไดอะแกรมของแพ็คเกจ `coretree`

#### 4.4. ดีพลอยเมนต์วิว

หลังจากที่ได้วิเคราะห์ในมุมมองเชิงแนวคิด และเชิงตรรกะเรียบร้อยแล้ว มุมมองต่อไปที่จะนำเสนอคือมุมมองของการใช้งานจริง นั่นคือ จะนำโปรแกรมที่พัฒนาไปเก็บไว้ที่เว็บเซิร์ฟเวอร์ และเมื่อผู้ใช้ต้องการใช้งาน โปรแกรมจะต้องใช้งานผ่านเว็บเบราว์เซอร์ ซึ่งโปรแกรมเว็บเบราว์เซอร์ของเครื่องผู้ใช้ จะทำการโหลดโปรแกรมแอปพลิเคชันจากเซิร์ฟเวอร์มาทำงานบนเครื่องผู้ใช้ ดังภาพที่ 4.19



ภาพที่ 4.19 ดีพลอยเมนต์ไดอะแกรม

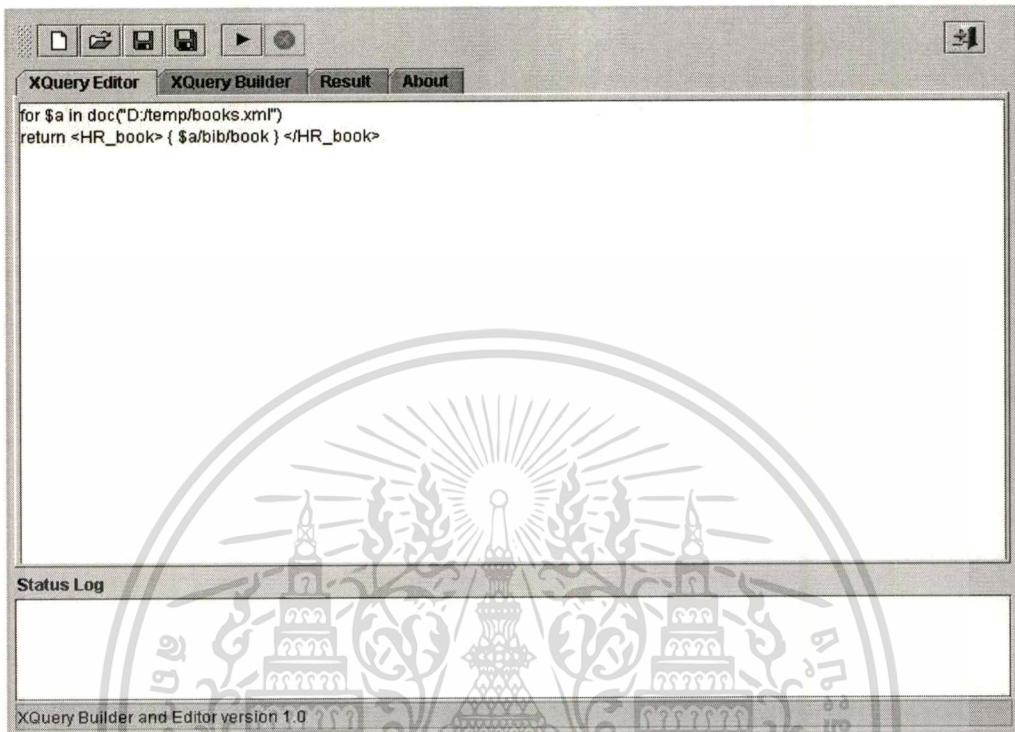
#### 4.5. การออกแบบหน้าจอ

การออกแบบหน้าจอเพื่อให้ผู้ใช้สามารถใช้งานโปรแกรมได้ง่าย และสะดวก ซึ่งคำนึงถึงความสอดคล้องกันกับความต้องการในหัวข้อ 4.1 และลำดับการทำงานที่ได้วิเคราะห์ไว้ในหัวข้อ 4.2 และ 4.3 เพื่อช่วยสนับสนุนการทำงานของทั้งผู้ใช้ระดับนักพัฒนา และผู้ใช้ทั่วไป โดยหน้าจอทั้งหมดในโปรแกรมมีดังนี้

##### 4.5.1. หน้าจอ XQuery Editor

หน้าจอ XQuery Editor เหมาะสำหรับผู้ใช้ที่มีความรู้เกี่ยวกับการเขียนชุดคำสั่ง XQuery ในหน้าจอนี้ผู้ใช้สามารถทำงานได้ 4 ฟังก์ชัน ดังนี้

- New Query: การเขียนชุดคำสั่ง XQuery ชุดใหม่
- Open: การเปิดไฟล์เพื่อแสดงในหน้าจอ XQuery Editor
- Save/Save as: การบันทึกไฟล์
- Execute: การค้นหาข้อมูลเมื่อผู้ใช้เขียนชุดคำสั่ง XQuery เรียบร้อยแล้ว



ภาพที่ 4.20 หน้าจอ XQuery Editor

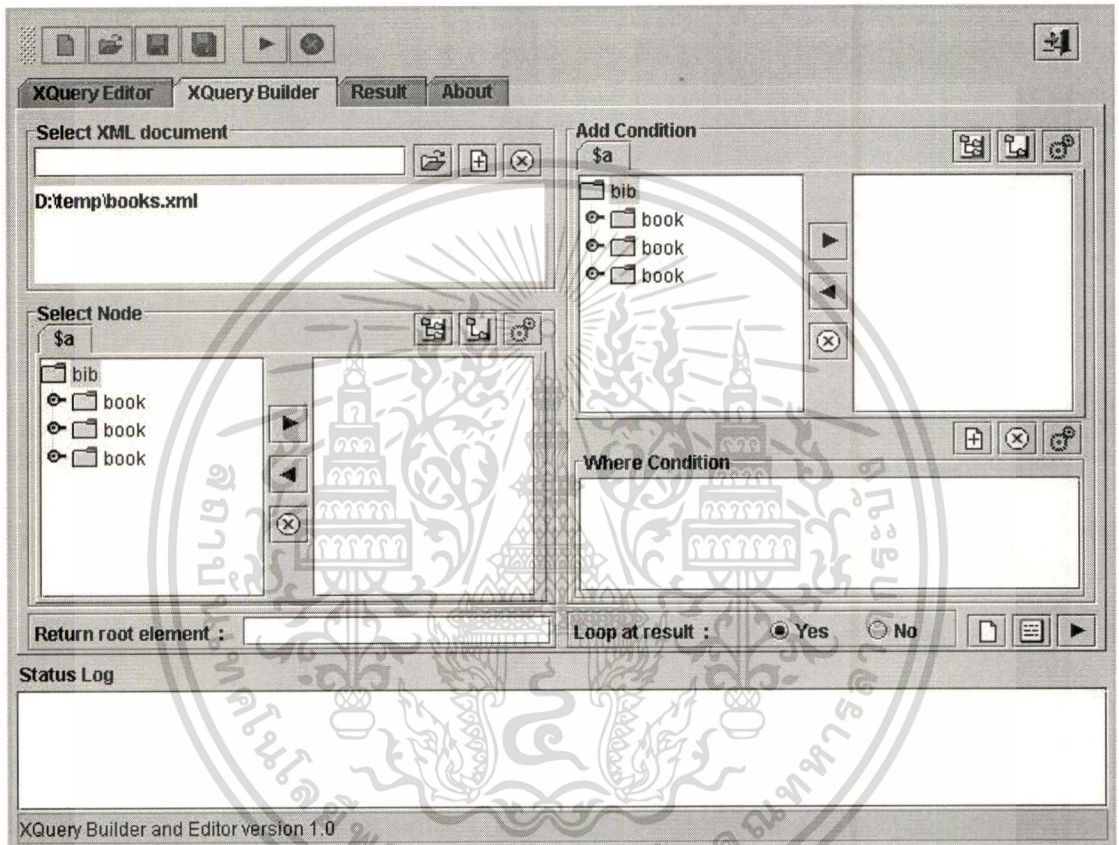
#### 4.5.2. หน้าจอ XQuery Builder

หน้าจอ XQuery Builder เหมาะสำหรับผู้ใช้ที่ไม่มีความรู้เกี่ยวกับการเขียนชุดคำสั่ง XQuery ในหน้าจอนี้ผู้ใช้จะต้องทำงานตามขั้นตอน ดังนี้

- New XQuery Builder: สร้างชุดคำสั่ง XQuery ชุดใหม่ผ่านหน้าจอ XQuery Builder
- Select XML document: เลือกเอกสาร XML ที่ต้องการค้นหา
- Select Node: เลือกโหนดที่ต้องการค้นหา
- Add Condition: เลือกโหนดที่ต้องการกำหนดเงื่อนไข
- Where Condition: กำหนดเงื่อนไขสำหรับการค้นหา
- Return root element: กำหนดครูดิเมนต์ของผลลัพธ์ (ถ้าต้องการ)
- Loop at result: กำหนดการวนซ้ำในการคืนค่าของผลลัพธ์
- Show XQuery Expression: แปลงข้อมูลที่ผู้ใช้ทำงานผ่านหน้าจอ XQuery Builder ไปเป็นชุดคำสั่ง XQuery ในหน้าจอ XQuery Editor (ถ้าต้องการ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Execute: ค้นหาข้อมูลเมื่อผู้ใช้เขียนชุดคำสั่ง XQuery ผ่านหน้าจอ XQuery Builder เรียบร้อยแล้ว

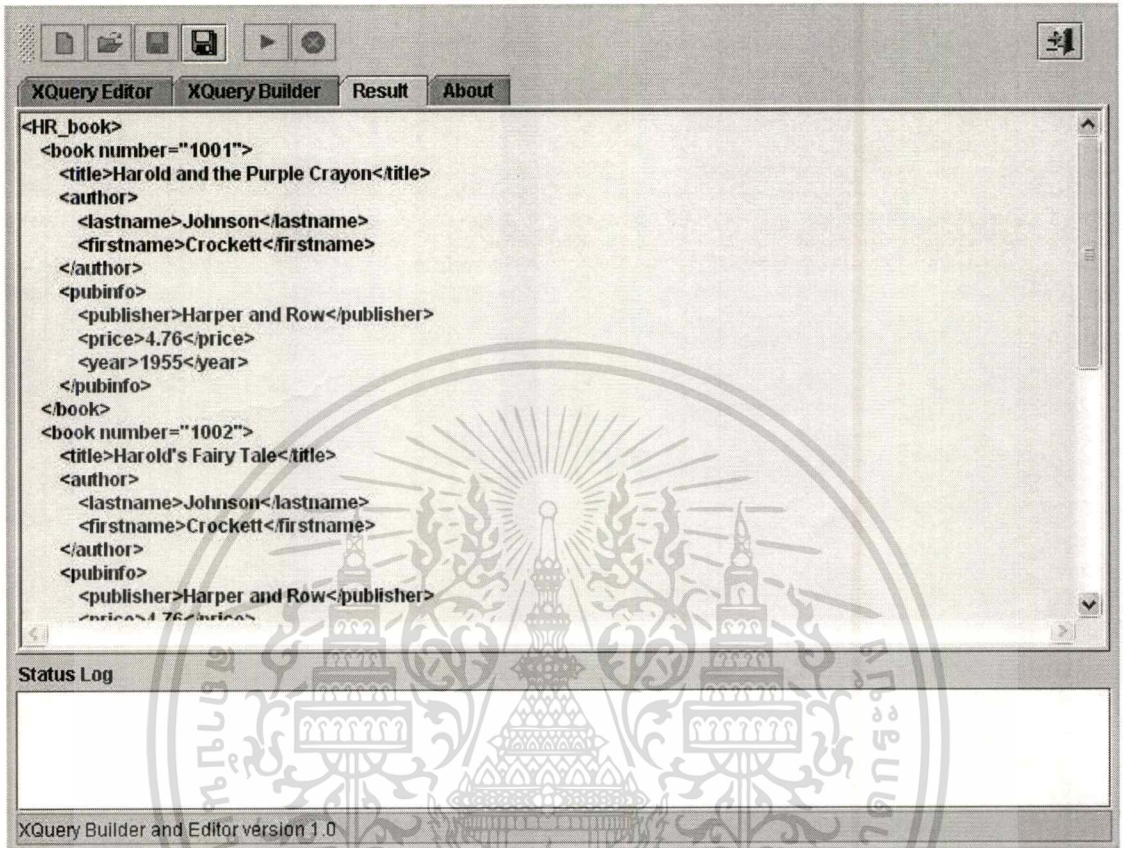


ภาพที่ 4.21 หน้าจอ XQuery Builder

#### 4.5.3. หน้าจอ Result

หน้าจอ Result ใช้สำหรับแสดงค่าผลลัพธ์ หรือแสดงข้อมูลความผิดพลาดที่เกิดขึ้นจากการค้นหาข้อมูล ในหน้าจอนี้ผู้ใช้สามารถทำงานได้ 1 ฟังก์ชัน ดังนี้

- Save as: การบันทึกผลลัพธ์



ภาพที่ 4.22 หน้าจอ Result

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

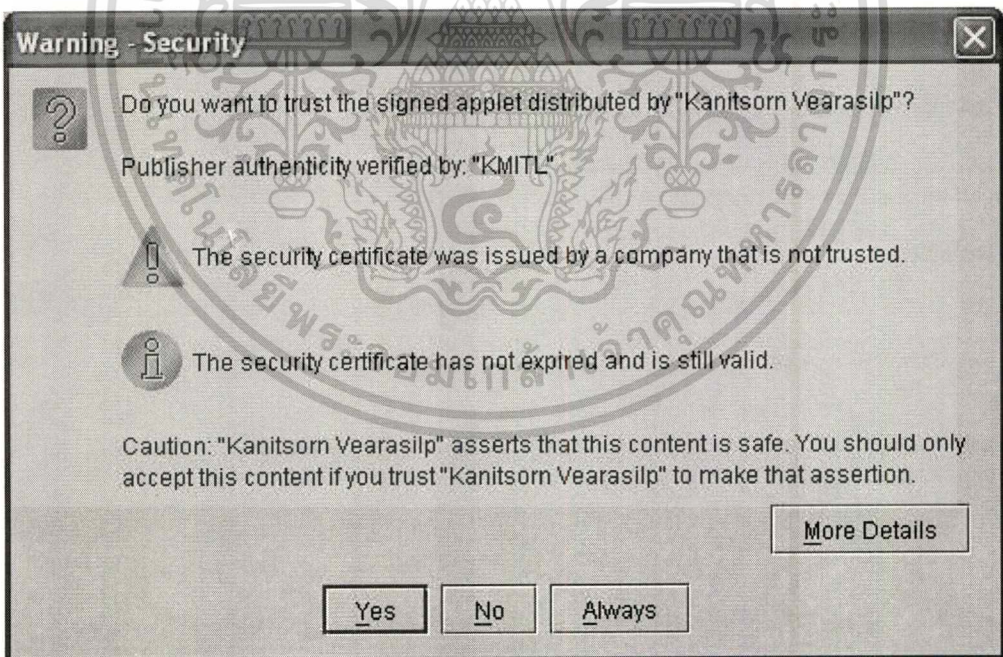
## บทที่ 5

### การใช้งานโปรแกรม

โปรแกรม XQuery Builder and Editor มีหน้าจอ และวิธีการใช้งาน ดังนี้

#### 5.1. เริ่มใช้งานโปรแกรม

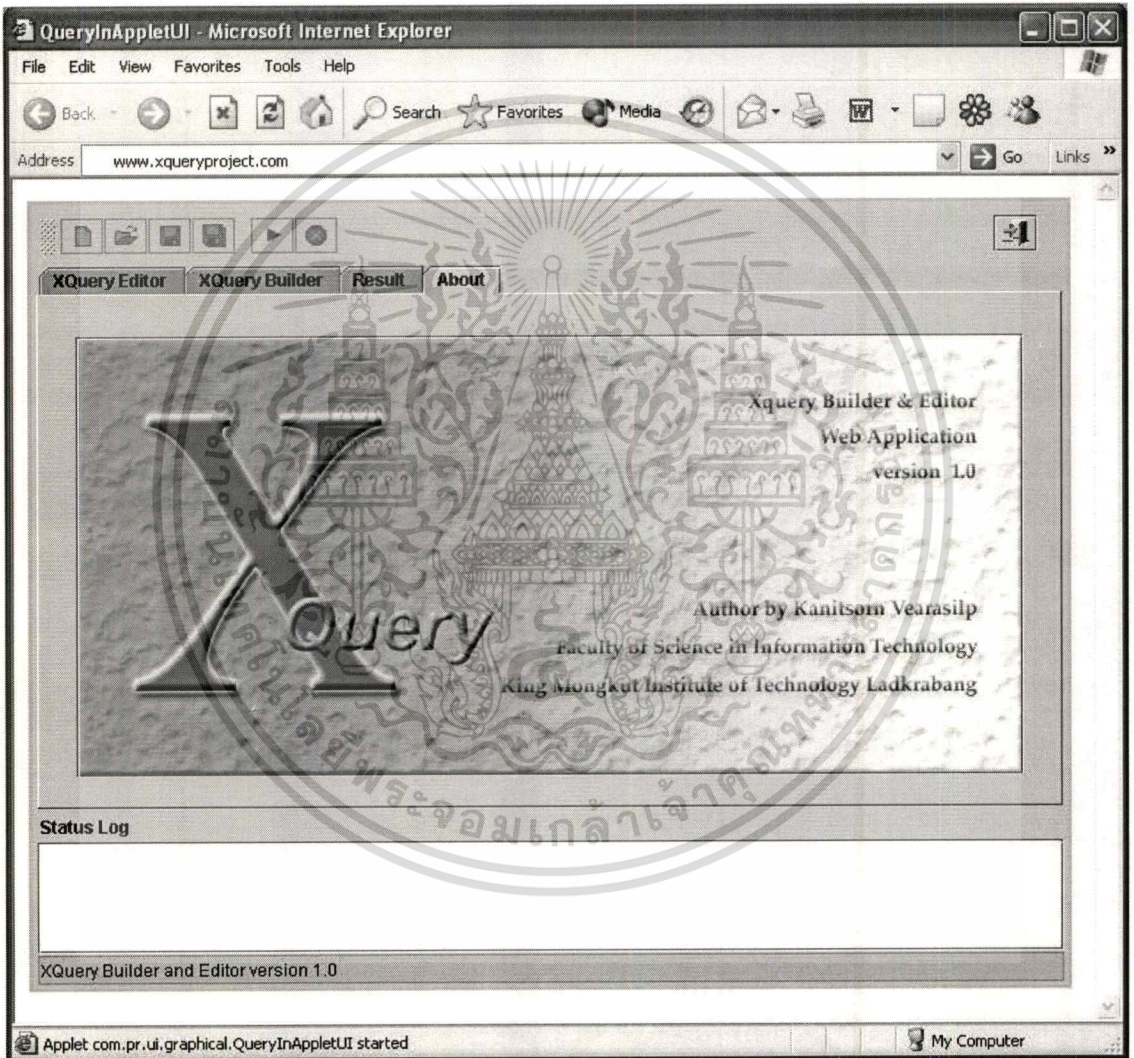
ผู้ใช้เปิดโปรแกรม Microsoft Internet Explorer แล้วพิมพ์ URL [www.xqueryproject.com](http://www.xqueryproject.com) เพื่อเข้าใช้งานโปรแกรม ซึ่งก่อนเข้าใช้งานจะปรากฏไดอะล็อกเตือนเรื่องความปลอดภัยในการเข้าถึงข้อมูลของเครื่องคอมพิวเตอร์ผู้ใช้งาน ดังภาพที่ 5.1 และผู้ใช้จะต้องยินยอมถึงจะสามารถใช้งานโปรแกรม XQuery Builder and Editor ได้



ภาพที่ 5.1 ไดอะล็อกเตือนเรื่องความปลอดภัยในการเข้าถึงข้อมูลภายในเครื่องคอมพิวเตอร์ผู้ใช้งาน

## 5.2. หน้าจอหลัก

เมื่อผู้ใช้งานยินยอมให้โปรแกรมสามารถเข้าถึงข้อมูลภายในเครื่องคอมพิวเตอร์เรียบร้อยแล้ว โปรแกรมจะเริ่มทำงานโดยแสดงหน้าจอหลัก ดังภาพที่ 5.2



ภาพที่ 5.2 หน้าจอหลักของโปรแกรม

ภายในหน้าจอประกอบด้วย 4 ส่วนดังต่อไปนี้

- แถบเครื่องมือ ประกอบด้วยปุ่ม New, Open, Save, Save as, Execute และ Stop
- แท็บหน้าจอ ประกอบด้วยแท็บหน้าจอ XQuery Editor, XQuery Builder และ Result

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน้าต่างแสดงสถานะการทำงาน
- ปุ่มออกจากโปรแกรม

### 5.3. หน้าจอ XQuery Editor

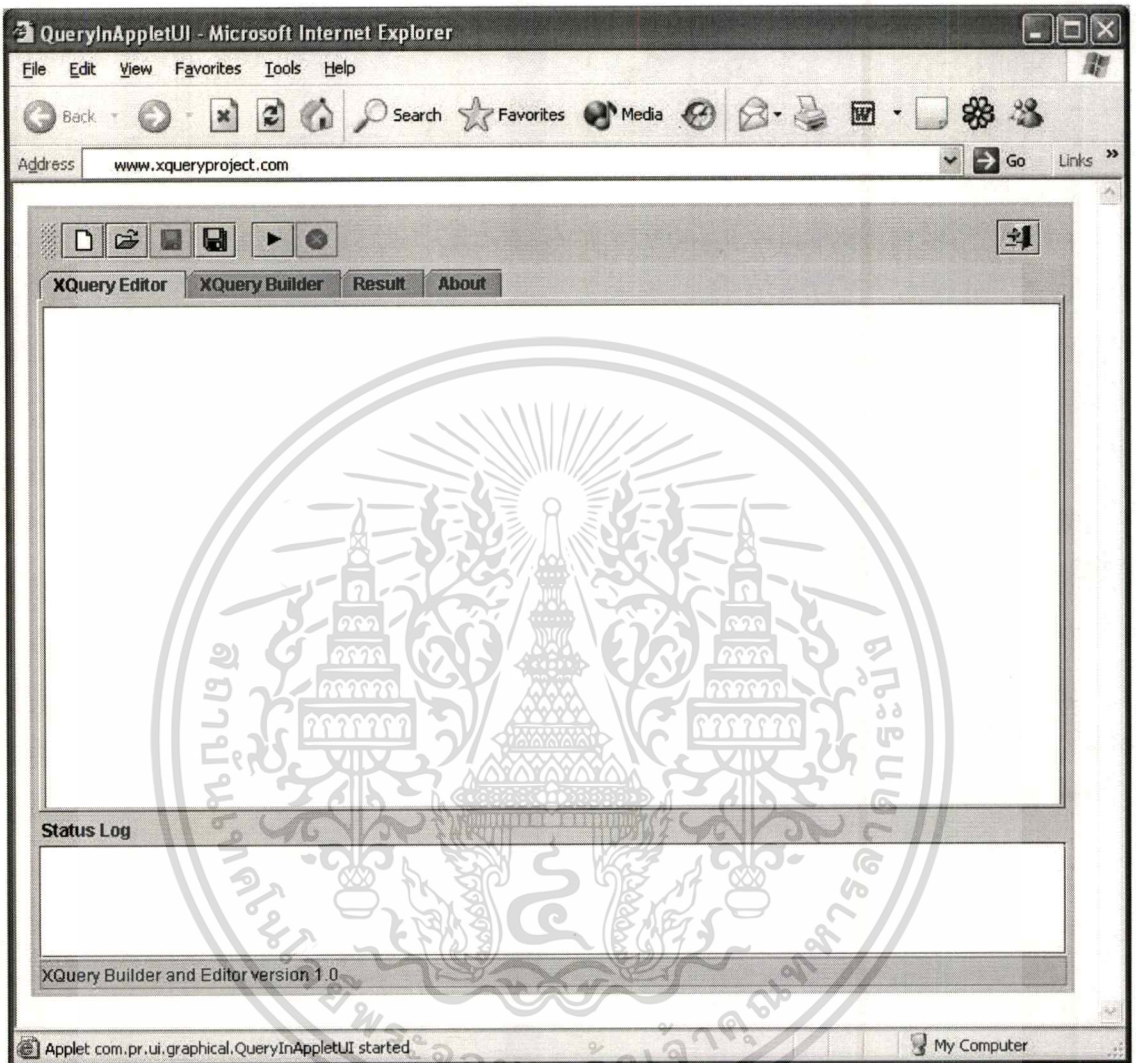
เมื่อผู้ใช้เลือกแท็บหน้าจอ XQuery Editor โปรแกรมแสดงหน้าจอพร้อมรับชุดคำสั่ง XQuery ดังภาพที่ 5.3

ภายในแท็บหน้าจอ XQuery Editor ประกอบด้วยพื้นที่หน้าจอว่างเพื่อรอรับผู้ใช้เขียนชุดคำสั่ง XQuery หรือใช้แสดงข้อมูลในเอกสาร XML โดยหน้าจอสามารถรองรับการแสดงผลข้อมูลเอกสาร XML ได้มากถึงขนาด 4 เมกะไบต์ หรืออาจมากกว่านี้ซึ่งขึ้นอยู่กับขนาดหน่วยความจำ และประสิทธิภาพของเครื่องคอมพิวเตอร์ ผู้ใช้สามารถเขียนชุดคำสั่ง XQuery ได้ 2 วิธี คือ

- กดปุ่ม Open ที่แถบเครื่องมือ เพื่อนำชุดคำสั่ง XQuery ที่เคยบันทึกไว้มาใช้งาน
- กดปุ่ม New ที่แถบเครื่องมือ เพื่อเขียนชุดคำสั่ง XQuery ขึ้นใหม่

เมื่อเขียนชุดคำสั่ง XQuery เรียบร้อยแล้วให้กดปุ่ม Execute ที่แถบเครื่องมือเพื่อเริ่มการทำงาน ค้นหาข้อมูล โดยผลลัพธ์ที่ได้จากการค้นหาจะแสดงผลในหน้าจอ Result ในกรณีที่ผู้ใช้ต้องการหยุดการทำงานค้นหาข้อมูลในขณะนั้นก็สามารถทำได้โดยกดปุ่ม Stop ที่แถบเครื่องมือ

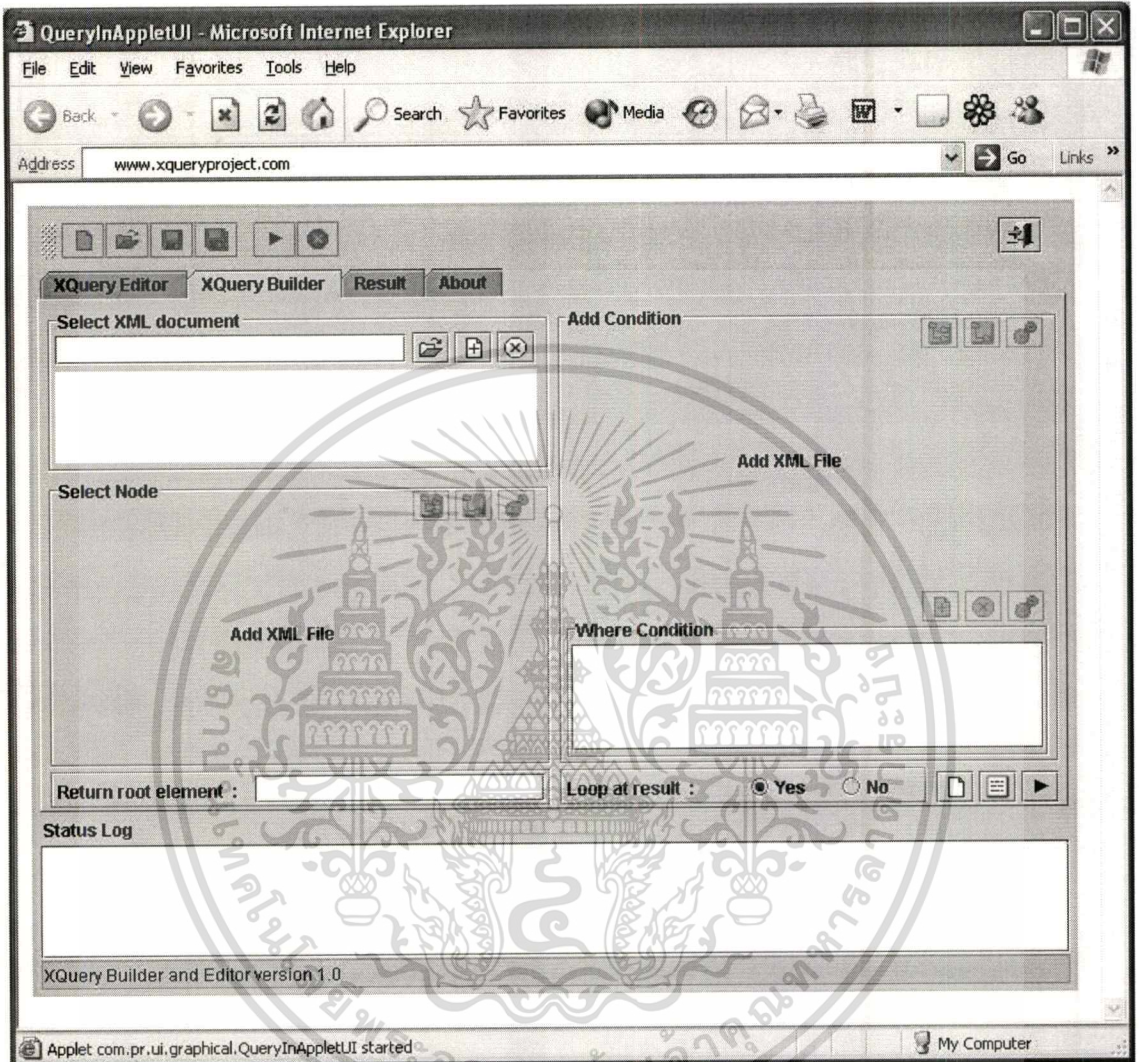
สำหรับในกรณีที่ต้องการบันทึกชุดคำสั่ง XQuery ไว้ทำงานในครั้งต่อไป ให้ผู้ใช้กดปุ่ม Save หรือปุ่ม Save as เพื่อบันทึกข้อมูลเก็บในที่เก็บข้อมูลตามที่ต้องการ โดยลำดับการทำงานเปิดไฟล์ และบันทึกไฟล์จะถูกแสดงในหน้าต่างแสดงสถานะการทำงาน



ภาพที่ 5.3 หน้าจอ XQuery Editor พร้อมรับชุดคำสั่ง XQuery จากผู้ใช้

#### 5.4. หน้าจอ XQuery Builder

เมื่อผู้ใช้เลือกแท็บหน้าจอ XQuery Builder โปรแกรมแสดงหน้าจอพร้อมบริการสร้างชุดคำสั่ง XQuery ดังภาพที่ 5.4



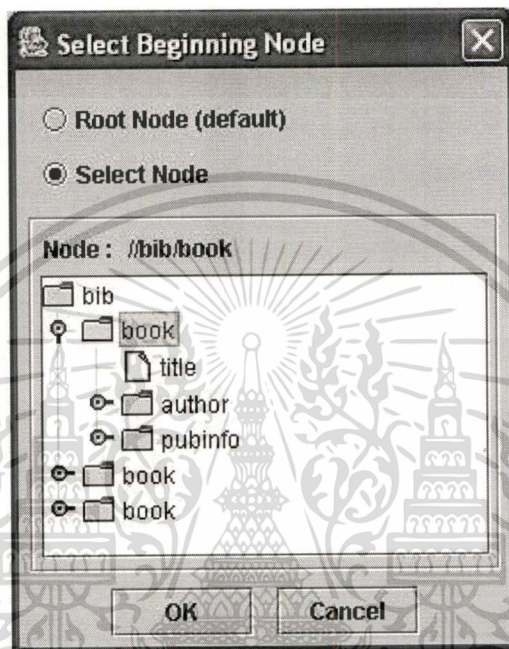
ภาพที่ 5.4 หน้าจอ XQuery Builder พร้อมรับการสร้างชุดคำสั่ง XQuery จากผู้ใช้

ภายในแท็บหน้าจอ XQuery Builder ประกอบด้วย 5 ส่วนย่อยๆ คือ การเลือกเอกสาร XML (Select XML document) การเลือกโหนดที่ต้องการค้นหา (Select Node) การกำหนดเงื่อนไขสำหรับการค้นหา (Add Condition และ Where Condition) การกำหนดครูดิเมนต์สำหรับผลลัพธ์ (Return root element) และการกำหนดการวนซ้ำในการคืนค่าผลลัพธ์ (Loop at result) โดยมีขั้นตอนการใช้งาน ดังนี้

- เลือกเอกสาร XML ที่ต้องการค้นหา โดยการกดปุ่ม “Browse XML File” ในแท็บหน้าจอ XQuery Builder เมื่อเลือกเอกสาร XML เรียบร้อยแล้วให้กดปุ่ม “Add XML”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลังจากที่ผู้ใช้กดปุ่ม “Add XML” แล้ว โปรแกรมจะแสดงไดอะล็อก “Select Beginning Node” ดังแสดงตัวอย่างในภาพที่ 5.5 เพื่อให้ผู้ใช้กำหนดโหนดเริ่มต้นของการค้นหาข้อมูล



ภาพที่ 5.5 ไดอะล็อกสำหรับกำหนดโหนดเริ่มต้นของการค้นหาข้อมูล

- เมื่อผู้ใช้กำหนดโหนดเริ่มต้นเรียบร้อยแล้ว โปรแกรมจะแสดงชื่อไฟล์ในรายการเอกสาร และแปลงข้อมูลในเอกสารให้อยู่ในรูปแบบของโครงสร้างต้นไม้แสดงในพื้นที่ของการเลือกโหนด และพื้นที่ของการกำหนดเงื่อนไข สำหรับการเลือกเอกสารนั้นสามารถเลือกได้มากกว่า 1 เอกสาร แต่จะต้องไม่เกิน 6 เอกสาร โดยในแต่ละเอกสารที่เลือกนั้นจะแสดงจำนวนโครงสร้างต้นไม้ตามจำนวนเอกสารที่เลือก ซึ่งแยกกันได้ด้วยแท็บตัวแปรภายในพื้นที่ของการเลือกโหนด และพื้นที่ของการกำหนดเงื่อนไข ในกรณีที่ต้องการยกเลิกเอกสารที่เลือกไว้ให้ผู้ใช้กดปุ่ม “Remove XML” ในแท็บหน้าจอก XQuery Builder
- เลือกโหนดที่ต้องการค้นหา โดยการเลือกโหนดที่ต้องการจากโครงสร้างต้นไม้ แล้วกดปุ่ม “Add Node” ในแท็บหน้าจอก XQuery Builder โปรแกรมจะแสดงโหนดที่ผู้ใช้เลือกในรายการโหนดที่อยู่ทางด้านขวาของโครงสร้างต้นไม้ สำหรับการเลือกโหนดในโครงสร้างต้นไม้ โปรแกรมมีปุ่มช่วยเหลือ 2 ปุ่มในการขยาย และหุบต้นไม้โดยใช้ปุ่ม “Expand node”

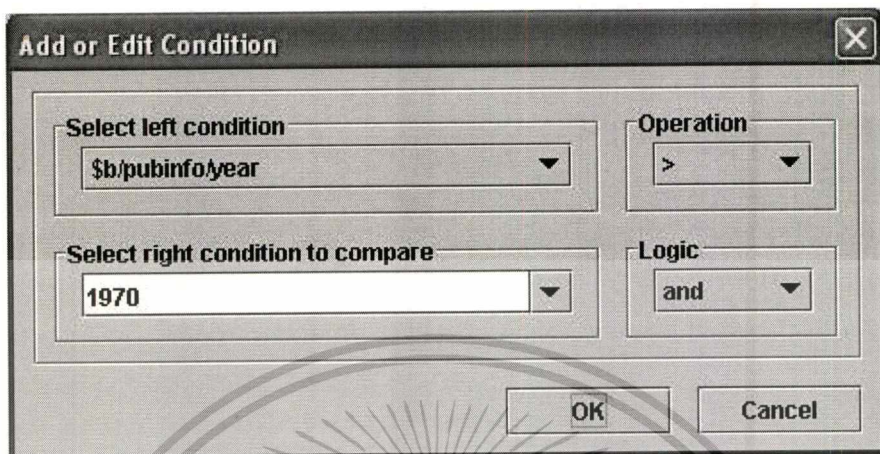
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ “Collapse node” ตามลำดับ ในกรณีที่ต้องการยกเลิกโหนดที่เลือกไว้ให้ผู้ใช้เลือกโหนดที่ต้องการยกเลิกในรายการโหนด และกดปุ่ม “Remove Node” หรือถ้าต้องการยกเลิกโหนดที่เลือกไว้ทั้งหมดให้ผู้ใช้กดปุ่ม “Remove All” และนอกจากนี้ยังสามารถกำหนดฟังก์ชันพื้นฐานให้กับโหนดที่เลือกได้โดยการเลือกโหนดที่ต้องการกำหนดฟังก์ชันในรายการโหนด และกดปุ่ม “Operation with selected node” ซึ่งโปรแกรมจะแสดงไดอะล็อก “Add function to field” ดังแสดงตัวอย่างในภาพที่ 5.6 เพื่อให้ผู้ใช้กำหนดฟังก์ชัน โดยฟังก์ชันที่มีให้เลือกใช้มีดังนี้ AVG, SUM, COUNT, MAX, MIN และ TEXT



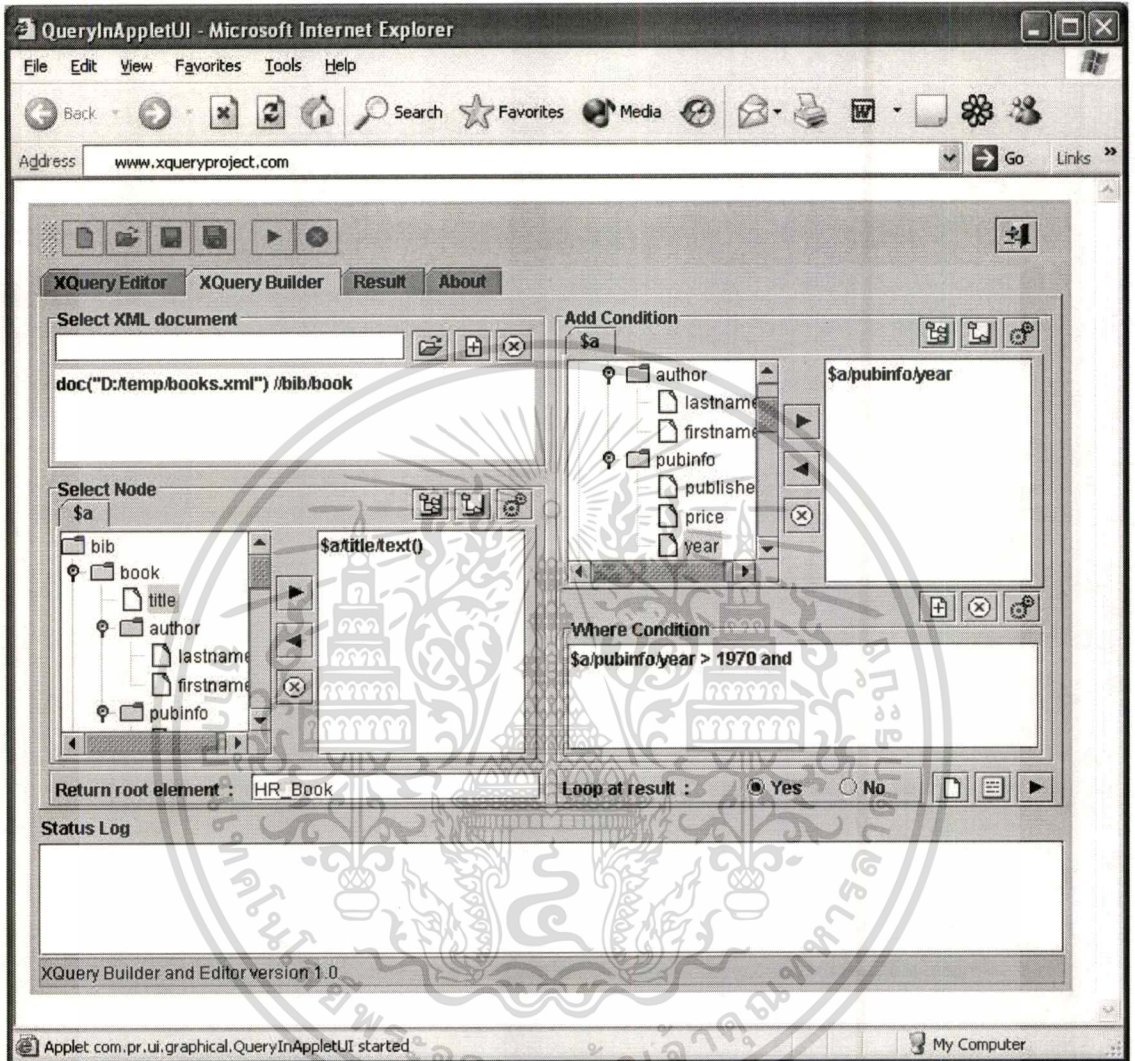
ภาพที่ 5.6 ไดอะล็อกสำหรับการกำหนดฟังก์ชัน

- กำหนดเงื่อนไขสำหรับการค้นหา จะเริ่มจากการเลือกโหนดที่ใช้กำหนดเงื่อนไขก่อน ซึ่งในขั้นตอนการเลือกโหนดนั้นจะใช้วิธีเดียวกันกับวิธีการเลือกโหนดที่ต้องการค้นหาในหัวข้อก่อนหน้า แล้วตามด้วยการกำหนดเงื่อนไขโดยเลือกโหนดในรายการโหนด และกดปุ่ม “Create condition” ซึ่งโปรแกรมจะแสดงไดอะล็อก “Add or Edit Condition” ดังแสดงตัวอย่างในภาพที่ 5.7 เพื่อให้ผู้ใช้กำหนดเงื่อนไขตามที่ต้องการ แล้วนำเสนอเงื่อนไขที่ผู้ใช้กำหนดในรายการเงื่อนไข ในกรณีที่ต้องการแก้ไขเงื่อนไขที่กำหนดไปแล้วให้ผู้ใช้เลือกเงื่อนไขในรายการเงื่อนไข และกดปุ่ม “Edit condition” หรือถ้าต้องการยกเลิกเงื่อนไขให้ผู้ใช้เลือกเงื่อนไขในลิสต์เงื่อนไข และกดปุ่ม “Remove condition”



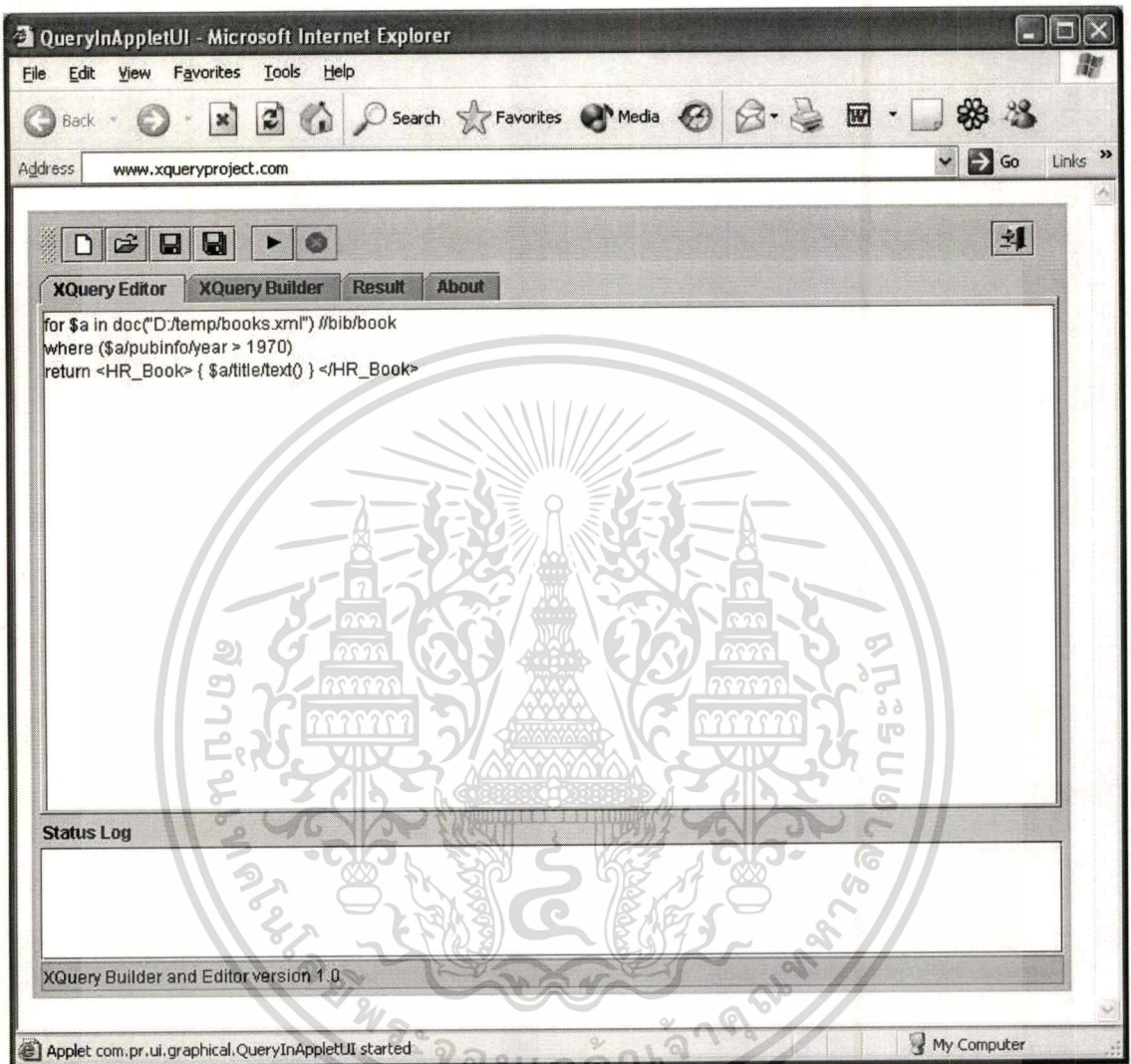
ภาพที่ 5.7 ไอคอนสำหรับกำหนดเงื่อนไข

- กำหนดรูตอิลิเมนต์สำหรับผลลัพธ์ (ถ้าต้องการ) โดยการใส่ชื่อรูตอิลิเมนต์ที่ต้องการในช่องรับ “Return root element”
- กำหนดการวนซ้ำในการคืนค่าผลลัพธ์ โดยการเลือก Yes หรือ No ซึ่งโปรแกรมจะกำหนดค่าเริ่มต้นให้เป็น Yes
- เมื่อสร้างชุดคำสั่ง XQuery ตามที่ต้องการแล้ว ดังแสดงตัวอย่างในภาพที่ 5.8 ผู้ใช้สามารถกดปุ่ม Execute เพื่อทำการค้นหาข้อมูล โดยผลลัพธ์ที่ได้จะแสดงในหน้าจอ Result
- สำหรับผู้ใช้ที่ต้องการแก้ไขชุดคำสั่ง XQuery เพิ่มเติมจากที่ได้ทำในหน้าจอ XQuery Builder นั้นสามารถทำได้โดยกดปุ่ม “Show XQuery Expression” โปรแกรมจะแปลงรูปแบบชุดคำสั่ง XQuery จากหน้าจอ XQuery Builder ไปเป็นชุดคำสั่ง XQuery ที่ถูกต้องและนำเสนอแสดงในหน้าจอ XQuery Editor ดังแสดงตัวอย่างในภาพที่ 5.9
- สำหรับปุ่ม “New XQuery Builder” ในแท็บหน้าจอ XQuery Builder นั้น ใช้เมื่อต้องการสร้างชุดคำสั่ง XQuery ชุดใหม่ ซึ่งโปรแกรมจะทำการเคลียร์หน้าจอให้พร้อมรับการสร้างชุดคำสั่ง XQuery ชุดใหม่



ภาพที่ 5.8 หน้าจอ XQuery Builder ที่ผ่านการสร้างชุดคำสั่ง XQuery

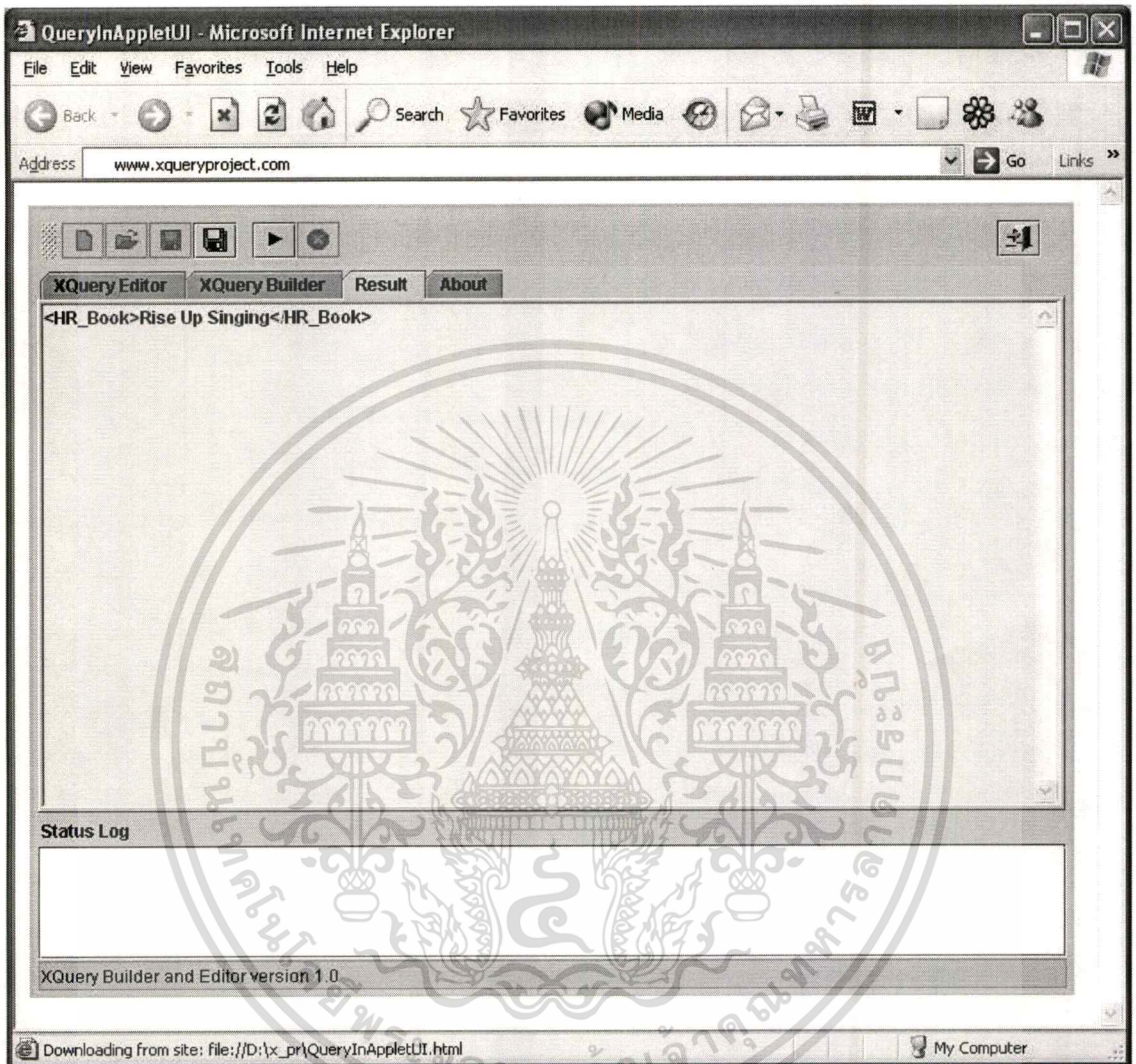
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.9 หน้าจอ XQuery Editor ที่แสดงชุดคำสั่ง XQuery

### 5.5. หน้าจอ Result

เมื่อผู้ใช้คลิกปุ่ม Execute ผลลัพธ์ของการค้นหาจะถูกนำเสนอในหน้าจอ Result จากตัวอย่างในภาพที่ 5.9 จะได้ผลลัพธ์ของการค้นหาข้อมูล ดังแสดงตัวอย่างในภาพที่ 5.10



ภาพที่ 5.10 หน้าจอ Result แสดงผลลัพธ์การค้นหาค้นหาข้อมูล

ภายในแท็บหน้าจอ Result จะประกอบด้วยพื้นที่สำหรับแสดงผลจากการค้นหาค้นหาข้อมูล โดยผลลัพธ์ดังกล่าวสามารถบันทึกเก็บไว้ใช้งานต่อไปได้ โดยการกดปุ่ม Save as เพื่อบันทึกเป็นเอกสาร XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### บทสรุป

ผลที่ได้จากการทำโครงการพัฒนาระบบงานในหัวข้อเรื่อง “การพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสาร XML ด้วยไวยากรณ์ XQuery ผ่านเว็บ” สามารถสรุปผลได้ดังต่อไปนี้

#### 6.1. สรุปผลการพัฒนาระบบ

การศึกษาค้นคว้า วิเคราะห์ออกแบบ และการพัฒนาโครงการเป็นไปอย่างที่น่าพึงพอใจ ซึ่งได้เลือกใช้กระบวนการวิเคราะห์ออกแบบ และกระบวนการพัฒนาไปในแนวทางเดียวกัน นั่นคือในลักษณะเชิงวัตถุเพื่อให้ง่ายต่อการทำงาน และต่อเนื่อง โดยเลือกใช้ UML ผ่านเครื่องมือ Rational Rose ในการวิเคราะห์ออกแบบ และใช้ภาษาจาวาผ่านเครื่องมือ IBM VisualAge ในการพัฒนาโปรแกรม ซึ่งจำเป็นต้องศึกษาเรียนรู้ทั้งทฤษฎี และวิธีการใช้เครื่องมือในการพัฒนาโครงการเพิ่มขึ้น โดยการวิเคราะห์ออกแบบด้วย UML นั้นจะอาศัยโคแอมแกรมต่างๆ เป็นตัวถ่ายทอดกระบวนการทำงานของโปรแกรมในมุมมองผู้ใช้ มุมมองผู้ออกแบบ และมุมมองของนักพัฒนา ทั้งนี้ข้อดีของการวิเคราะห์ออกแบบในเชิงวัตถุทำให้การพัฒนาโปรแกรมในเชิงวัตถุเป็นไปได้ง่ายขึ้นนั่นเอง สำหรับผลการพัฒนาโครงการออกมาเป็น โปรแกรมที่ทำงานผ่านเว็บนั้นก็สามารถทำงาน ได้ดีอย่างเป็นที่น่าพอใจ และโดยรวมก็เป็นไปตามที่ได้วิเคราะห์ออกแบบไว้

#### 6.2. ปัญหาที่พบในการทำงาน และข้อเสนอแนะ

ปัญหาที่พบในการพัฒนาโครงการก็คือขาดความรู้ และความเข้าใจในการพัฒนาเชิงวัตถุ เพราะเนื่องมาจากขาดประสบการณ์ ส่วนการออกแบบหน้าจอช่วยเหลือสำหรับผู้ใช้ที่ไม่มีความรู้ในการเขียนชุดคำสั่ง XQuery นั้นยังไม่ยืดหยุ่นพอที่จะใช้กับการเขียนชุดคำสั่งในรูปแบบที่ยาก เนื่องจากเน้นความง่ายของการใช้งาน เพราะถ้าออกแบบให้มีความยืดหยุ่นสูงในการใช้งานจะส่งผลให้หน้าจอมีความซับซ้อน ใช้งานยากเกินไป และอาจเป็นไปได้ยากในการอิมพลีเมนต์จริง ดังนั้นจึงได้เน้นถึงความง่ายในการใช้งานแทน อนึ่งถ้าผู้ใช้ต้องการที่จะแก้ไขชุดคำสั่งเพิ่มเติมจากที่ได้ใช้หน้าจอช่วยเหลือแล้วก็ยังสามารถทำได้ เพราะ โปรแกรมสามารถแปลงข้อมูลที่ผู้ใช้ทำงานผ่านหน้าจอช่วยเหลือ ไปเป็นชุดคำสั่ง

XQuery ได้อย่างถูกต้อง และแสดงผลในหน้าจอเคคิเตอร์ให้ผู้ใช้แก้ไข เพื่อเป็นการเปิดช่องทางให้ผู้ใช้สามารถทำงานได้อย่างอิสระ



## บรรณานุกรม

- วีระศักดิ์ ชิงฉาวร. 2541. **Fundamental of JAVA Programming Volume I**. กรุงเทพฯ: ชัม ชิสเท็ม.
- วีระศักดิ์ ชิงฉาวร. 2543. **Fundamental of JAVA Programming Volume 2**. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- Bakker, B. and Widarto, I. 2001. **An Introduction to XQuery**. [Online]. Available:  
<http://perfectxml.com/articles/xml/xquery.asp>.
- Boag, S. et al. 2002. **XQuery 1.0: An XML Query Language**. [Online]. Available:  
<http://www.w3.org/TR/xquery>.
- Booch, G. et al. 2000. **The Unified Modeling Language User Guide**. Massachusetts: Addison-Wesley.
- Chamberlin, D. 2002. "XQuery: An XML Query Language." **IBM System Journal**. Vol. 41, No. 4.
- Deutsch, A. et al. 1998. **XML-QL: A Query Language for XML**. [Online]. Available:  
<http://www.w3.org/TR/1998/NOTE-xml-ql-19980819>.
- Fankhauser, P. 2001. **XQuery Formal Semantics State and Challenges**. [Online]. Available:  
<http://www.acm.org/sigmod/record/issues/0109/SPECIAL/fankhauser2.pdf>
- Homer, A. 1999. **XML in IE5 Programmer's Reference**. Indianapolis: Wrox Press.
- Quatrani, T. 2000. **Visual Modeling with Rational Rose 2000 and UML**. 2<sup>nd</sup> ed. Massachusetts: Addison-Wesley.
- Robie, J. 1999. **XQL Tutorial**. [Online]. Available: <http://www.ibiblio.org/xql/xql-tutorial.html>.
- Robie, J. et al. 2000. **Quilt: an XML Query Language**. [Online]. Available:  
[http://www.almaden.ibm.com/cs/people/chamberlin/quilt\\_euro.html](http://www.almaden.ibm.com/cs/people/chamberlin/quilt_euro.html).
- Simpson, J. E. 1999. **Just XML**. New Jersey: Prentice-Hall.

## ภาคผนวก

### ตัวอย่างเอกสาร XML

#### books.xml

```
<?xml version="1.0"?>
<bib>
  <book number="1001">
    <title>Harold and the Purple Crayon</title>
    <author>
      <lastname>Johnson</lastname>
      <firstname>Crockett</firstname>
    </author>
    <pubinfo>
      <publisher>Harper and Row</publisher>
      <price>4.76</price>
      <year>1955</year>
    </pubinfo>
  </book>
  <book number="1002">
    <title>Harold's Fairy Tale</title>
    <author>
      <lastname>Johnson</lastname>
      <firstname>Crockett</firstname>
    </author>
    <pubinfo>
```

```

    <publisher>Harper and Row</publisher>
    <price>4.76</price>
    <year>1956</year>
  </pubinfo>
</book>
<book number="2001">
  <title>Rise Up Singing</title>
  <author>
    <lastname>Blood</lastname>
    <firstname>Peter</firstname>
  </author>
  <author>
    <lastname>Patterson</lastname>
    <firstname>Annie</firstname>
  </author>
  <pubinfo>
    <publisher>Sing Out Corporation</publisher>
    <price>15.45</price>
    <year>1988</year>
  </pubinfo>
</book>
</bib>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**reviews.xml**

```

<?xml version="1.0"?>
<reviews>
  <book>
    <title>Rise Up Singing</title>
    <review>This is great!</review>
  </book>
</review>

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาวนินธุสรณ์ เวียรศิลป์
วันเกิด	28 มีนาคม 2521
สถานที่เกิด	กรุงเทพฯ
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (วิทยาการคอมพิวเตอร์) คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
การทำงาน	บริษัท ฟรีวิลล์ โซลูชั่น จำกัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้