

วิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส



นายเศรษฐพงศ์ ลีพรัตน์รักษ์

# สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2547

ISBN 974-17-6306-9

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

AN APPROACH FOR AUTOMATICALLY GENERATING TEST CASES  
FROM USE CASES

Mr. Setapong Leeraharattanarak



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2004

ISBN 974-17-6306-9

หัวข้อวิทยานิพนธ์

วิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

โดย

นายเศรษฐพงศ์ ลีพรัตน์รักษ์

สาขาวิชา

วิทยาศาสตร์คอมพิวเตอร์

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. ธราทิพย์ สุวรรณศาสตร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยานิพนธ์ฉบับนี้  
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์  
(ศาสตราจารย์ ดร. ดิเรก ลาวัญศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ)

..... อาจารย์ที่ปรึกษา  
(ผู้ช่วยศาสตราจารย์ ดร. ธราทิพย์ สุวรรณศาสตร์)

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. พรศิริ หมั่นไชยศิริ)

..... กรรมการ  
(อาจารย์ ดร. อาทิตย์ ทองทักษ์)

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

เศรษฐพงศ์ ลิพัหัตนรักษ์ : วิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส. (AN APPROACH FOR AUTOMATICALLY GENERATING TEST CASES FROM USE CASES) อ. ที่ปรึกษา : ผศ. ดร. ธาวาทิพย์ สุวรรณศาสตร์, 137 หน้า. ISBN 974-17-6306-9.

การพัฒนาซอฟต์แวร์ในปัจจุบันมักใช้วิธีการพัฒนาซอฟต์แวร์เชิงวัตถุ ซึ่งขั้นตอนการวิเคราะห์ถือเป็นขั้นตอนแรกของการพัฒนาซอฟต์แวร์เชิงวัตถุ โดยนักวิเคราะห์จะนำเอกสารความต้องการซอฟต์แวร์มาสร้างเอกสารการออกแบบซึ่งอยู่ในรูปแบบของแผนภาพยูเอ็มแอลเพื่ออธิบายความสัมพันธ์ของระบบ แผนภาพยูสเคสเป็นแผนภาพหนึ่งที่นักวิเคราะห์จะสร้างขึ้นในขั้นตอนการวิเคราะห์ ซึ่งเป็นแผนภาพที่แสดงฟังก์ชันการทำงานต่างๆ ของระบบและความสัมพันธ์ระหว่างฟังก์ชันการทำงาน วิทยานิพนธ์นี้นำเสนอวิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสในแผนภาพยูสเคส กรณีทดสอบที่สร้างขึ้นตามวิธีการที่นำเสนอนี้ช่วยให้ขั้นตอนการทดสอบซอฟต์แวร์ในกระบวนการพัฒนาซอฟต์แวร์เริ่มต้นได้เร็วขึ้น

นักวิเคราะห์ต้องให้ข้อมูลเพิ่มเติมแก่ยูสเคสได้แก่ ข้อมูลนำเข้า และลำดับเหตุการณ์สำหรับประกอบกรณีสการสร้างกรณีทดสอบ หลังจากนั้นจึงนำข้อมูลรายละเอียดยูสเคสมาช่วยในการสร้างกรณีทดสอบ โดยกรณีทดสอบที่สร้างขึ้นจะมีครอบคลุมทุกลำดับเหตุการณ์ที่เป็นไปได้ของแต่ละยูสเคส

วิทยานิพนธ์นี้ได้พัฒนาเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสตามวิธีการที่นำเสนอไว้ ผู้ทดสอบสามารถสร้างกรณีทดสอบได้โดยอัตโนมัติด้วยเครื่องมือที่พัฒนาขึ้น ทำให้เวลาและค่าใช้จ่ายในการสร้างกรณีทดสอบและขั้นตอนการทดสอบซอฟต์แวร์ลดลง

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....  
สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์.....ลายมือชื่ออาจารย์ที่ปรึกษา.....  
ปีการศึกษา...2547.....ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

# # 4570569921 : MAJOR COMPUTER SCIENCE

KEYWORD: SOFTWARE TESTING / USE CASES / TEST CASES

SETAPONG LEERAHARATTANARAK : AN APPROACH FOR AUTOMATICALLY  
GENERATING TEST CASES FROM USE CASES. THESIS ADVISOR : ASST.  
PROF. TARATIP SUWANNASART PH.D., 137 pp. ISBN 974-17-6306-9.

Currently, software is often developed using object-oriented technology. In the analysis phase, which is the first phase of software development process, a system analyst creates design documents, such as UML diagrams, from software requirements. A use case diagram, which is one of UML diagrams, shows system functions and relationships among use cases within a system. This thesis proposes an approach for automatically generating test cases from use cases. These test cases can help testers launch the test process early in the software development life cycle.

System analyst has to supply additional information related to use cases such as inputs and scenarios in order to generate test cases. All generated test cases are covered all possible use case scenarios.

According to this approach, a test case generating tool is developed. With this tool, testers can automatically generate test cases, so it reduces the cost and effort of test case generation and the software testing process.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Department.....Computer Engineering..... Student's signature.....

Field of study.....Computer Science..... Advisor's signature.....

Academic year...2004..... Co-advisor's signature.....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลืออย่างดียิ่งจากผู้ช่วยศาสตราจารย์ ดร. ธาราทิพย์ สุวรรณศาสตร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้เสียสละเวลาอันมีค่าให้คำแนะนำเกี่ยวกับแนวทางการวิจัย และคำปรึกษาที่มีประโยชน์อย่างมากกับวิทยานิพนธ์นี้ ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ ผู้ช่วยศาสตราจารย์ ดร. พรศิริ หมั่นไชยศรี และอาจารย์ ดร. อาทิตย์ ทองทักษ์ คณะกรรมการสอบวิทยานิพนธ์ ที่ท่านได้กรุณาให้คำแนะนำ และชี้แนะในการทำงานวิจัย รวมถึงตรวจสอบวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณคณาจารย์ทุกท่านที่ให้ความรู้เพื่อนำมาใช้ในการทำงานวิจัยและทำวิทยานิพนธ์

ขอขอบคุณพี่ชาติชาย ดวงสะอาด และพี่ศิริส สุภาวิตา พี่ร่วมอาจารย์ที่ปรึกษาวิทยานิพนธ์สำหรับคำแนะนำ คำปรึกษาการทำวิจัย

ขอขอบคุณเพื่อนๆ สำหรับคำแนะนำ และเป็นกำลังใจในการทำงานวิจัย และขอบคุณเพื่อนคนหนึ่งที่ทำให้ข้าพเจ้าอาศัยรถกลับบ้านเป็นประจำตลอดระยะเวลาที่มาทำงานวิจัยที่มหาวิทยาลัย

สุดท้ายนี้ ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดา และพี่น้องทุกคนในครอบครัว ที่สนับสนุนด้านการเงิน ให้ความหวังใจ และให้กำลังใจแก่ข้าพเจ้าจนสำเร็จการศึกษา

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฐ
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 ขั้นตอนการดำเนินการวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 แผนภาพยูสเคส (Use Case Diagram).....	5
2.2 รูปแบบของรายละเอียดยูสเคส.....	7
2.3 การทดสอบซอฟต์แวร์.....	9
2.4 เอกซ์เอ็มแอล (XML: eXtensible Markup Language).....	10
2.5 งานวิจัยที่เกี่ยวข้อง.....	12
2.5.1 งานวิจัย “Use Case-based Testing of Product Lines”.....	12
2.5.2 งานวิจัย “Generating Test Cases from Use Cases”.....	13
2.5.3 เครื่องมือ “SpecStudio”.....	13
3. การออกแบบการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส.....	15
3.1 แนวคิดการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส.....	15
3.2 แผนภาพยูสเคสและรายละเอียดยูสเคส.....	17

บทที่	หน้า
3.3 การรวมยูสเคสที่มีความสัมพันธ์กัน.....	27
3.4 การสร้างกรณีทดสอบโดยอัตโนมัติ.....	34
3.5 การวิเคราะห์และออกแบบเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติ.....	42
3.5.1 แผนภาพยูสเคส.....	42
3.5.2 แผนภาพคลาส.....	46
3.5.3 แผนภาพซีควเอนซ์.....	53
3.5.4 แผนภาพลำดับกิจกรรม.....	57
4. การพัฒนาเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส.....	59
4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ.....	59
4.2 ฐานข้อมูลของเครื่องมือ.....	60
4.3 โครงสร้างของเครื่องมือ.....	61
5. การทดสอบเครื่องมือ.....	66
5.1 ขั้นตอนการทดสอบเครื่องมือที่พัฒนาขึ้น.....	66
5.2 สภาพแวดล้อมที่ใช้ในการทดสอบเครื่องมือ.....	66
5.3 แผนภาพยูสเคสที่ใช้ในการทดสอบเครื่องมือ.....	67
5.4 ผลการทดสอบ.....	72
5.5 สรุปผลการทดสอบ.....	74
6. สรุปผลการวิจัยและข้อเสนอแนะ.....	75
6.1 สรุปผลการวิจัย.....	75
6.2 ข้อจำกัดและแนวทางการวิจัยต่อ.....	76
รายการอ้างอิง.....	77
ภาคผนวก.....	78
ภาคผนวก ก.....	79
ภาคผนวก ข.....	82



สารบัญ (ต่อ)

ณ

บทที่	หน้า
ภาคผนวก ค.....	105
ภาคผนวก ง.....	123
ภาคผนวก จ.....	130
ประวัติผู้เขียนวิทยานิพนธ์.....	136



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 รายละเอียดยูสเคสรูปแบบตารางหนึ่งสดมภ์.....	7
ตารางที่ 3.1 ตารางเปรียบเทียบรายละเอียดยูสเคสของงานวิจัยนี้กับรายละเอียดยูสเคส ของ Cockburn.....	17
ตารางที่ 3.2 ตัวอย่างรายละเอียดยูสเคสหมายเลข 1: Add contact.....	20
ตารางที่ 3.3 เครื่องหมายเปรียบเทียบที่ใช้ในประโยคเงื่อนไข.....	24
ตารางที่ 3.4 ตัวอย่างรายละเอียดยูสเคสหมายเลข 2: Authenticate client.....	30
ตารางที่ 3.5 ตัวอย่างรายละเอียดยูสเคสหมายเลข 3: Invalid phone no.....	31
ตารางที่ 3.6 ตัวอย่างรายละเอียดยูสเคสที่รวมความสัมพันธ์ยูสเคสหมายเลข 1: Add contact.....	32
ตารางที่ 3.7 ตัวอย่างกรณีทดสอบที่สร้างขึ้นจากยูสเคส: Add contact.....	41
ตารางที่ 3.8 รายละเอียดยูสเคสวิเคราะห์แผนภาพยูสเคส.....	43
ตารางที่ 3.9 รายละเอียดยูสเคสการรวมยูสเคส.....	44
ตารางที่ 3.10 รายละเอียดยูสเคสการแยกรายละเอียดยูสเคส.....	44
ตารางที่ 3.11 รายละเอียดยูสเคสการสร้างกรณีทดสอบ.....	45
ตารางที่ 3.12 รายละเอียดยูสเคสการสุ่มข้อมูลทดสอบ.....	46
ตารางที่ 5.1 ตารางสรุปยูสเคสของระบบสั่งซื้อสินค้าหลังจากวิเคราะห์ด้วยเครื่องมือ.....	73
ตารางที่ 5.2 ตารางสรุปยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์หลังจากวิเคราะห์ ด้วยเครื่องมือ.....	74
ตารางที่ 5.3 ตารางสรุปยูสเคสของระบบให้บริการเช่าวีซีดีหลังจากวิเคราะห์ด้วยเครื่องมือ.....	75
ตารางที่ ก-1 พจนานุกรมข้อมูลตาราง uc_context.....	81
ตารางที่ ก-2 พจนานุกรมข้อมูลตาราง required_item.....	81
ตารางที่ ก-3 พจนานุกรมข้อมูลตาราง uc_item.....	81
ตารางที่ ก-4 พจนานุกรมข้อมูลตาราง condition.....	82
ตารางที่ ก-5 พจนานุกรมข้อมูลตาราง success.....	82
ตารางที่ ก-6 พจนานุกรมข้อมูลตาราง alternative.....	82
ตารางที่ ข-1 รายละเอียดยูสเคสหมายเลข 1: Login.....	87
ตารางที่ ข-2 รายละเอียดยูสเคสหมายเลข 2: Place Order.....	87
ตารางที่ ข-3 รายละเอียดยูสเคสหมายเลข 3: Get Product Information.....	88

## สารบัญตาราง (ต่อ)

๗

ตาราง	หน้า
ตารางที่ ข-4 รายละเอียดบัญชีหมายเลข 4: Update Account.....	89
ตารางที่ ข-5 รายละเอียดบัญชีหมายเลข 5: Return Product.....	89
ตารางที่ ข-6 รายละเอียดบัญชีหมายเลข 6: Update Product Quantities.....	90
ตารางที่ ข-7 รายละเอียดบัญชีหมายเลข 7: Cancel Order.....	90
ตารางที่ ข-8 รายละเอียดบัญชีหมายเลข 8: Get Status on Order.....	91
ตารางที่ ข-9 รายละเอียดบัญชีหมายเลข 9: Get Catalog.....	91
ตารางที่ ข-10 รายละเอียดบัญชีหมายเลข 10: Register Complaint.....	92
ตารางที่ ข-11 รายละเอียดบัญชีหมายเลข 11: Run Sales Report.....	92
ตารางที่ ข-12 รายละเอียดบัญชีหมายเลข 12: Fill and Ship Order.....	93
ตารางที่ ข-13 รายละเอียดบัญชีหมายเลข 13: Receive Back-Ordered Items.....	94
ตารางที่ ข-14 รายละเอียดของข้อมูลนำเข้าจาก Customer.....	94
ตารางที่ ข-15 รายละเอียดของข้อมูลนำเข้าจาก Customer Rep.....	95
ตารางที่ ข-16 รายละเอียดของข้อมูลนำเข้าจาก Clerk.....	95
ตารางที่ ข-17 รายละเอียดของข้อมูลนำเข้าจาก Inventory System.....	96
ตารางที่ ข-18 รายละเอียดบัญชีหมายเลข 1: Create Session.....	96
ตารางที่ ข-19 รายละเอียดบัญชีหมายเลข 3: Submit Session News.....	96
ตารางที่ ข-20 รายละเอียดบัญชีหมายเลข 4: Add Register.....	97
ตารางที่ ข-21 รายละเอียดบัญชีหมายเลข 5: Submit Order.....	97
ตารางที่ ข-22 รายละเอียดบัญชีหมายเลข 6: Check Bid.....	97
ตารางที่ ข-23 รายละเอียดบัญชีหมายเลข 7: Match Call Market.....	98
ตารางที่ ข-24 รายละเอียดบัญชีหมายเลข 8: Match ATO.....	98
ตารางที่ ข-25 รายละเอียดของข้อมูลนำเข้าจาก Session_Administrator.....	99
ตารางที่ ข-26 รายละเอียดของข้อมูลนำเข้าจาก Investor.....	99
ตารางที่ ข-27 รายละเอียดบัญชีหมายเลข 1: Create member.....	100
ตารางที่ ข-28 รายละเอียดบัญชีหมายเลข 2: Search member.....	100
ตารางที่ ข-29 รายละเอียดบัญชีหมายเลข 3: Edit member.....	101
ตารางที่ ข-30 รายละเอียดบัญชีหมายเลข 4: Rent VCD.....	101
ตารางที่ ข-31 รายละเอียดบัญชีหมายเลข 5: Print detail.....	102

## สารบัญตาราง (ต่อ)

๘

ตาราง	หน้า
ตารางที่ ข-32 รายละเอียดบัญชีหมายเลข 6: Return VCD.....	102
ตารางที่ ข-33 รายละเอียดบัญชีหมายเลข 7: Print overdue report.....	103
ตารางที่ ข-34 รายละเอียดบัญชีหมายเลข 8: Add VCD.....	103
ตารางที่ ข-35 รายละเอียดบัญชีหมายเลข 9: Change category VCD.....	103
ตารางที่ ข-36 รายละเอียดบัญชีหมายเลข 10: Delete VCD.....	104
ตารางที่ ข-37 รายละเอียดบัญชีหมายเลข 11: Search VCD.....	104
ตารางที่ ข-38 รายละเอียดบัญชีหมายเลข 12: Print daily report.....	105
ตารางที่ ข-39 รายละเอียดของข้อมูลนำเข้าจาก Member-Staff.....	105



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญภาพ

ภาพประกอบ	หน้า
รูปที่ 2.1 ตัวอย่างแผนภาพยูสเคส.....	5
รูปที่ 2.2 สัญลักษณ์แทนผู้ใช้งานระบบ.....	6
รูปที่ 2.3 สัญลักษณ์แทนยูสเคส.....	6
รูปที่ 2.4 ความสัมพันธ์แบบอินคลูด.....	6
รูปที่ 2.5 ความสัมพันธ์แบบเอ็กซ์เทน.....	7
รูปที่ 2.6 ตัวอย่างเอกสารเอ็กซ์เอ็มแอล.....	12
รูปที่ 3.1 ภาพรวมของแนวคิดการสร้างกรณีทดสอบจากยูสเคส.....	16
รูปที่ 3.2 ตัวอย่างแผนภาพยูสเคส : Contact list manager.....	20
รูปที่ 3.3 รูปแบบของหมายเลขประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่น.....	22
รูปที่ 3.4 รูปแบบของหมายเลขลำดับการทำงานทางเลือกอื่น.....	23
รูปที่ 3.4 ภาพแสดงยูสเคสที่มีความสัมพันธ์ที่ซ้อนกันแบบที่ 1.....	26
รูปที่ 3.5 ภาพแสดงยูสเคสที่มีความสัมพันธ์ที่ซ้อนกันแบบที่ 2.....	27
รูปที่ 3.6 แผนภาพความสัมพันธ์ของเอนทิตี (ER-Diagram) ของรายละเอียดยูสเคส.....	35
รูปที่ 3.7 ภาพรวมขั้นตอนการสร้างกรณีทดสอบ.....	38
รูปที่ 3.8 แผนภาพยูสเคสของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส.....	42
รูปที่ 3.9 แผนภาพคลาสของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส.....	47
รูปที่ 3.10 คลาส DBMng.....	47
รูปที่ 3.11 คลาส XMIParse.....	48
รูปที่ 3.12 คลาส ArrangeInfo.....	48
รูปที่ 3.13 คลาส GenData.....	48
รูปที่ 3.14 คลาส RandomData.....	49
รูปที่ 3.15 คลาส RPNParser.....	49
รูปที่ 3.16 คลาส Operator.....	49
รูปที่ 3.17 คลาส ArithmeticOperator.....	50
รูปที่ 3.18 คลาส ComparisonOperator.....	50
รูปที่ 3.19 คลาส LogicalOperator.....	50
รูปที่ 3.20 คลาส StringOperator.....	50
รูปที่ 3.21 คลาส Operand.....	51

ภาพประกอบ	หน้า
รูปที่ 3.22 คลาส BooleanOperand.....	51
รูปที่ 3.23 คลาส DateOperand.....	51
รูปที่ 3.24 คลาส DecimalOperand.....	52
รูปที่ 3.25 คลาส StringOperand.....	52
รูปที่ 3.26 คลาส LongOperand.....	53
รูปที่ 3.27 คลาส exportHTML.....	53
รูปที่ 3.28 แผนภาพที่คอนซ์ของการวิเคราะห์แผนภาพยูสเคส.....	54
รูปที่ 3.29 แผนภาพที่คอนซ์ของการสร้างกรณีทดสอบ.....	56
รูปที่ 3.30 แผนภาพลำดับกิจกรรมการวิเคราะห์แผนภาพยูสเคส.....	58
รูปที่ 3.31 แผนภาพลำดับกิจกรรมการสร้างกรณีทดสอบ.....	59
รูปที่ 4.1 แผนภาพแสดงความสัมพันธ์ระหว่างเอนทิตีของเครื่องมือ.....	61
รูปที่ 4.2 โครงสร้างของเครื่องมือ.....	62
รูปที่ 4.3 หน้าจอหลักของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส.....	63
รูปที่ 4.4 หน้าจอสำหรับเลือกไฟล์เอ็กซ์เอ็มแอล.....	63
รูปที่ 4.5 ปุ่มอ่านข้อมูล (สำหรับวิเคราะห์ไฟล์).....	64
รูปที่ 4.6 หน้าจอแสดงรายละเอียดยูสเคส.....	64
รูปที่ 4.7 ปุ่มสร้างกรณีทดสอบ.....	65
รูปที่ 4.8 หน้าจอแสดงกรณีทดสอบ.....	65
รูปที่ 4.9 ปุ่มออก.....	66
รูปที่ 5.1 แผนภาพยูสเคสของระบบสั่งซื้อสินค้า.....	69
รูปที่ 5.2 ตัวอย่างรายละเอียดยูสเคสของระบบสั่งซื้อสินค้า.....	70
รูปที่ 5.3 แผนภาพยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์.....	71
รูปที่ 5.4 ตัวอย่างรายละเอียดยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์.....	71
รูปที่ 5.5 แผนภาพยูสเคสของระบบให้บริการเช่าวีซีดี.....	72
รูปที่ 5.6 ตัวอย่างรายละเอียดยูสเคสของระบบให้บริการเช่าวีซีดี.....	73
รูปที่ ค-1 กรณีทดสอบหมายเลข 1.1 ของระบบสั่งซื้อสินค้า.....	107
รูปที่ ค-2 กรณีทดสอบหมายเลข 2.1 ของระบบสั่งซื้อสินค้า.....	107
รูปที่ ค-3 กรณีทดสอบหมายเลข 2.2 ของระบบสั่งซื้อสินค้า.....	108



## สารบัญญภาพ (ต่อ)

ณ

ภาพประกอบ	หน้า
รูปที่ ค-32 กรณียกทดสอบหมายเลข 6.3 ของระบบให้บริการเช่าวีซีดี.....	121
รูปที่ ค-33 กรณียกทดสอบหมายเลข 8.1 ของระบบให้บริการเช่าวีซีดี.....	122
รูปที่ ค-34 กรณียกทดสอบหมายเลข 9.1 ของระบบให้บริการเช่าวีซีดี.....	122
รูปที่ ค-35 กรณียกทดสอบหมายเลข 10.1 ของระบบให้บริการเช่าวีซีดี.....	122
รูปที่ ค-36 กรณียกทดสอบหมายเลข 12.1 ของระบบให้บริการเช่าวีซีดี.....	123
รูปที่ ง-1 สคริปต์ของแบบฟอร์มรายละเอียดยูสเคส.....	125
รูปที่ จ-1 หน้าจอของเครื่องมือสร้างกรณียกทดสอบโดยอัตโนมัติจากยูสเคส.....	132
รูปที่ จ-2 หน้าต่างเลือกไฟล์เอ็กซ์เอ็มแอล.....	133
รูปที่ จ-3 หน้าต่างรายงานผลการอ่านไฟล์เอ็กซ์เอ็มแอล.....	133
รูปที่ จ-4 หน้าต่างแสดงรายละเอียดยูสเคสและลำดับเหตุการณ์ที่เป็นไปได้ของยูสเคส.....	134
รูปที่ จ-5 ส่วนที่ใช้กำหนดสถานที่เก็บไฟล์เอ็กซ์เอ็มแอล.....	134
รูปที่ จ-6 หน้าต่างเลือกสถานที่เก็บไฟล์เอ็กซ์เอ็มแอล.....	135
รูปที่ จ-7 หน้าต่างรายงานผลการสร้างกรณียกทดสอบ.....	135
รูปที่ จ-8 หน้าต่างแสดงกรณียกทดสอบที่สร้างขึ้น.....	135

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

การพัฒนาซอฟต์แวร์ตามหลักวิศวกรรมซอฟต์แวร์นั้น ขั้นตอนการวิเคราะห์ (Analysis) ถือเป็นขั้นตอนเริ่มแรกของการพัฒนาซอฟต์แวร์ โดยนักวิเคราะห์ระบบจะเก็บความต้องการของระบบมาสร้างเอกสารความต้องการซอฟต์แวร์ จากนั้นจะนำเอกสารความต้องการซอฟต์แวร์มาสร้างเอกสารการออกแบบ (Design Document) ซึ่งอาจอยู่ในรูปแบบของแผนภาพมาอธิบายความสัมพันธ์ของระบบต่างๆ เช่นแผนภาพยูเอ็มแอล (UML Diagram) หรืออาจอยู่ในรูปแบบภาษาธรรมชาติ (Natural language) ในการบรรยายระบบต่างๆ หลังจากนั้นนำเอกสารออกแบบไปให้ผู้พัฒนาโปรแกรม (Programmer) ใช้เป็นแนวทางในการพัฒนาโปรแกรม และให้ผู้ทดสอบ (Tester) ทดสอบโปรแกรมนั้นว่าพัฒนาตามความต้องการของระบบหรือไม่

การทดสอบซอฟต์แวร์เป็นขั้นตอนที่สำคัญมากในกระบวนการพัฒนาซอฟต์แวร์ ซึ่งการทดสอบซอฟต์แวร์แบ่งเป็น 3 ระดับ คือการทดสอบระดับหน่วย (Unit Testing) การทดสอบแบบบูรณาการ (Integration Testing) และการทดสอบระบบ (System Testing) การทดสอบซอฟต์แวร์แต่ละระดับต้องสร้างกรณีทดสอบ (Test cases) เพื่อใช้ทดสอบการทำงานของซอฟต์แวร์ ซึ่งกรณีทดสอบประกอบด้วยค่าของข้อมูลทดสอบ (Test data) และผลลัพธ์ที่คาดหวัง (Expected output)

ในปัจจุบันมีการนำเสนอแนวคิดการสร้างกรณีทดสอบจากรายละเอียดยูสเคส (Use Cases Description) ที่สร้างจากการเอกสารความต้องการซอฟต์แวร์ อาทิเช่น Jim Heumann [1] เสนองานวิจัยเรื่อง “Generating Test Cases From Use Cases” โดยเตรียมรายละเอียดยูสเคสซึ่งระบุลำดับเหตุการณ์การทำงานยูสเคสเป็นข้อๆ จากนั้นกำหนดกรณีทดสอบตามเงื่อนไขของเหตุการณ์ และระบุชื่อข้อมูลนำเข้า จากนั้นกำหนดผลลัพธ์ที่คาดหวัง แล้วสร้างข้อมูลทดสอบ ในขณะที่ ImplSoftware, Inc. [2] พัฒนาเครื่องมือชื่อ SpecStudio ซึ่งเป็นเครื่องมือสร้างกรณีทดสอบจากข้อกำหนดเชิงพฤติกรรม (Behavior Specification) เครื่องมือนี้ช่วยสร้างข้อกำหนดเชิงพฤติกรรม แล้วผู้ใช้งานนำข้อกำหนดเชิงพฤติกรรมมาสร้างกรณีทดสอบ และเครื่องมือนี้สามารถวิเคราะห์ประสิทธิภาพของกรณีทดสอบที่สร้างขึ้นนั้นว่าครอบคลุมทุกเงื่อนไขที่เป็นไปได้หรือไม่

การสร้างกรณีทดสอบจากรายละเอียดยูสเคสหรือเอกสารความต้องการซอฟต์แวร์ที่นำเสนอข้างต้นไม่สามารถสร้างกรณีทดสอบโดยอัตโนมัติได้ ผู้พัฒนาระบบอ่านและวิเคราะห์เอกสารความต้องการซอฟต์แวร์หรือรายละเอียดยูสเคส แล้วสร้างข้อมูลทดสอบและผลลัพธ์ที่คาดหวังเพื่อใช้เป็นกรณีทดสอบ งานวิจัยนี้จึงมีเป้าหมายที่จะนำเสนอวิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสในแผนภาพยูสเคส ซึ่งกรณีทดสอบจะครอบคลุมทุกยูสเคสและครอบคลุมทุกลำดับเหตุการณ์ที่เป็นไปได้ ผู้ทดสอบสามารถนำกรณีทดสอบนี้ไปใช้ในขั้นตอนการทดสอบระบบได้ทันที

## 1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 เพื่อออกแบบวิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส
- 1.2.2 เพื่อพัฒนาเครื่องมือสำหรับสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

## 1.3 ขอบเขตของการวิจัย

- 1.3.1 แผนภาพยูสเคสที่ใช้ในการสร้างกรณีทดสอบต้องประกอบด้วยรายละเอียดยูสเคสของงานวิจัยนี้
- 1.3.2 หมายเลขลำดับของเงื่อนไขสำหรับลำดับเหตุการณ์สำเร็จกำหนดให้เป็น 0 เสมอ
- 1.3.3 หมายเลขลำดับของเงื่อนไขสำหรับลำดับเหตุการณ์ทางเลือกอื่นต้องขึ้นต้นด้วยหมายเลขลำดับการทำงานที่ไม่สามารถทำงานได้ตามปกติของ Success scenario ตามด้วยเครื่องหมายจุด และตามด้วยหมายเลขตั้งแต่ 1 เป็นต้นไป
- 1.3.4 แต่ละยูสเคสต้องมีข้อมูลนำเข้าอย่างน้อย 1 ตัว
- 1.3.5 ชนิดของข้อมูลนำเข้ามี 5 ชนิดคือ ชนิดข้อมูลจำนวนเต็ม (Integer) ชนิดข้อมูลจำนวนจริง (Float) ชนิดข้อมูลตรรกะ (Boolean) ชนิดข้อมูลชุดข้อความ (String) และชนิดข้อมูลวันที่ (Date)
- 1.3.6 เครื่องหมายทางคณิตศาสตร์ในประโยคเงื่อนไขมี 4 ชนิดได้แก่ เครื่องหมายบวก (+) เครื่องหมายลบ (-) เครื่องหมายคูณ (\*) และเครื่องหมายหาร (/)
- 1.3.7 จำนวนวงเล็บในประโยคเงื่อนไขต้องไม่เกิน 5 วงเล็บ และวงเล็บต้องซ้อนกันอย่างถูกต้อง

- 1.3.8 ประโยคเงื่อนไขนั้นมีพจน์ที่เป็นการคำนวณทางคณิตศาสตร์ได้เพียงพจน์เดียวเท่านั้น
- 1.3.9 ความหมายของแต่ละพจน์ในประโยคเงื่อนไขต้องไม่ขัดแย้งกัน เช่น  $(X \leq 10) \& (X \geq 20)$  เป็นต้น
- 1.3.10 ลำดับเหตุการณ์ทางเลือกอื่นเกิดขึ้นเมื่อไม่สามารถทำงานตามลำดับเหตุการณ์สำเร็จได้เท่านั้น
- 1.3.11 ความสัมพันธ์ระหว่างยูสเคสกับยูสเคสในแผนภาพยูสเคสต้องไม่เป็นความสัมพันธ์แบบซ้ำซ้อนกัน
- 1.3.12 กรณีทดสอบที่สร้างขึ้นอยู่ในรูปแบบเอกสารเอกซ์เอ็มแอล (HTML) ซึ่งประกอบด้วยหมายเลขกรณีทดสอบ ชื่อกรณีทดสอบ ข้อความอธิบายกรณีทดสอบ ข้อมูลทดสอบ ผลลัพธ์คาดหวัง และผลลัพธ์จากการทำงานตามเงื่อนไข
- 1.3.13 แฟ้มเอกสารเอกซ์เอ็มแอลที่นำมาสร้างกรณีทดสอบต้องอยู่ในรูปแบบที่สมบูรณ์ และถูกต้องตามข้อกำหนดของเอกซ์เอ็มแอล
- 1.3.14 แฟ้มเอกสารเอกซ์เอ็มแอลที่นำมาสร้างกรณีทดสอบต้องบันทึกด้วย UML 1.3 XMI addin ซึ่งเป็นส่วนเสริมของโปรแกรมเรชั่นนอลโรส
- 1.3.15 พัฒนาเครื่องมือและใช้เครื่องมือบนระบบปฏิบัติการวินโดวส์ (Windows) ตั้งแต่รุ่น XP ขึ้นไป

#### 1.4 ขั้นตอนการดำเนินการวิจัย

- 1.4.1 ศึกษาแผนภาพยูสเคส และศึกษารายละเอียดยูสเคสสำหรับสร้างกรณีทดสอบ
- 1.4.2 ศึกษาการทดสอบซอฟต์แวร์
- 1.4.3 ศึกษาเอกซ์เอ็มแอล
- 1.4.4 กำหนดรูปแบบรายละเอียดยูสเคสสำหรับสร้างกรณีทดสอบโดยอัตโนมัติ
- 1.4.5 ออกแบบวิธีการสร้างกรณีทดสอบจากยูสเคส
- 1.4.6 พัฒนาเครื่องมือสำหรับสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส
- 1.4.7 ตรวจสอบผลการวิจัย
- 1.4.8 สรุปผลงานวิจัย และข้อเสนอแนะ
- 1.4.9 จัดทำรายงานวิทยานิพนธ์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 วิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส ซึ่งสามารถนำไปเป็นแนวทางในการพัฒนาเครื่องมือสำหรับสร้างกรณีทดสอบโดยอัตโนมัติได้
- 1.5.2 เครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส ที่ช่วยสร้างกรณีทดสอบสำหรับการทดสอบระบบได้สะดวกยิ่งขึ้น
- 1.5.3 ผู้ทดสอบสามารถนำกรณีทดสอบที่ได้ไปใช้ในการทดสอบระบบส่วนของการทดสอบฟังก์ชันได้ทันที
- 1.5.4 ลดเวลาของกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุในส่วนของกรณีเตรียมกรณีทดสอบสำหรับการทดสอบระบบ



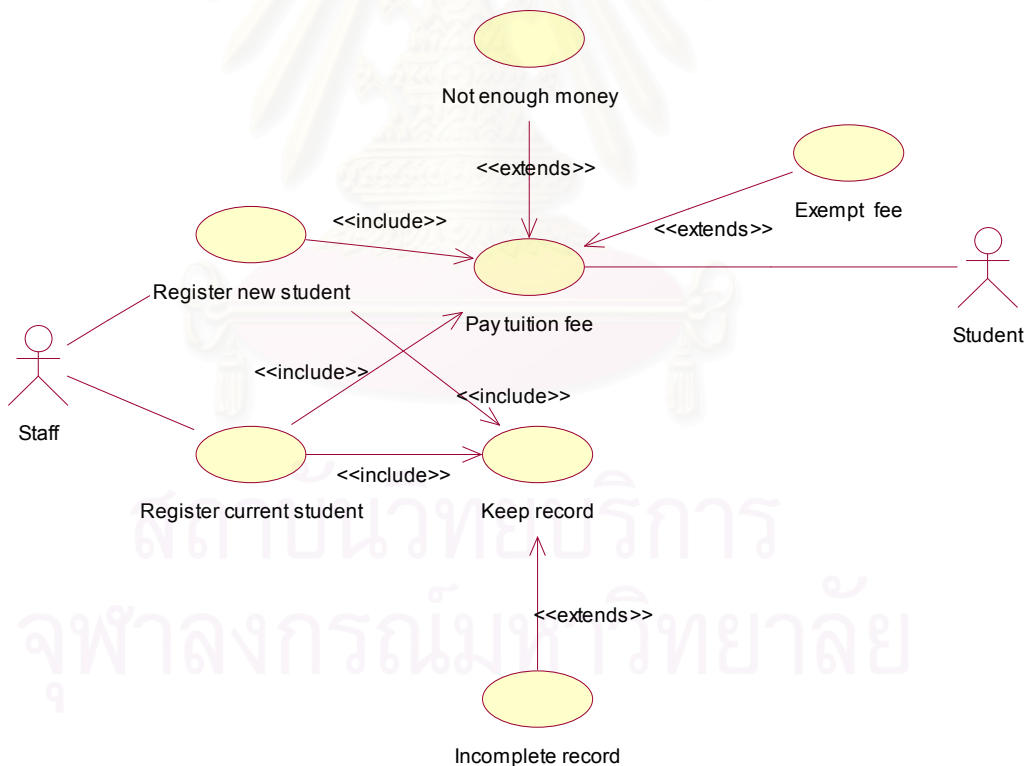
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพยูสเคส [3] เป็นแผนภาพแทนการเล่าเรื่องราวทั้งหมดของขอบเขตของปัญหา (Problem Domain) ว่าต้องประกอบด้วยกิจกรรมใด และแต่ละกิจกรรมมีความสัมพันธ์กันเป็นระบบได้อย่างไร การเขียนแผนภาพยูสเคสช่วยให้ผู้พัฒนาระบบสามารถแยกแยะกิจกรรมที่จะเกิดขึ้นในระบบโดยแผนภาพยูสเคสใช้สัญลักษณ์ และรูปภาพที่ไม่ซับซ้อน และเข้าใจง่าย แผนภาพยูสเคสที่มีประสิทธิภาพช่วยให้การวิเคราะห์ระบบมีความสมบูรณ์ และถูกต้อง เนื่องจากแผนภาพยูสเคสสร้างขึ้นหลังจากการเก็บความต้องการของระบบจากผู้ใช้งาน ซึ่งเป็นขั้นตอนแรกของกระบวนการพัฒนาซอฟต์แวร์ ตัวอย่างแผนภาพยูสเคสแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างแผนภาพยูสเคส

แผนภาพยูสเคสประกอบด้วย ผู้ใช้งานระบบ (Actor) ยูสเคส (Use Case) และความสัมพันธ์ (Relationship) ซึ่งมีรายละเอียดดังนี้

### 2.1.1 ผู้ใช้งานระบบ

ผู้ใช้งานระบบเป็นคนที่ใช้ของ หรือระบบอื่น ที่ใช้งานยูสเคส แผนภาพยูสเคสใช้สัญลักษณ์รูปคนแทนผู้ใช้งานระบบ ดังรูปที่ 2.2



รูปที่ 2.2 สัญลักษณ์แทนผู้ใช้งานระบบ

### 2.1.2 ยูสเคส

แผนภาพยูสเคสใช้สัญลักษณ์รูปวงรีแทนยูสเคส ดังรูปที่ 2.3 แต่ละยูสเคสอธิบายเงื่อนไขทั้งหมดที่เป็นไปได้และลำดับเหตุการณ์ที่เกิดขึ้นเมื่อเงื่อนไขเป็นจริง ชื่อของยูสเคสใช้คำกริยาขึ้นต้นเพื่ออธิบายการทำงานของยูสเคส เช่น กรอกรหัสผ่าน (Enter PIN) เป็นต้น



Enter PIN

รูปที่ 2.3 สัญลักษณ์แทนยูสเคส

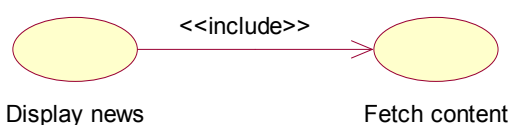
### 2.1.3 ความสัมพันธ์

แผนภาพยูสเคสแบ่งความสัมพันธ์เป็น 2 แบบคือความสัมพันธ์ระหว่างผู้ใช้งานระบบกับยูสเคส และความสัมพันธ์ระหว่างยูสเคสกับยูสเคส ซึ่งแผนภาพยูสเคสใช้เส้นตรงแทนความสัมพันธ์ระหว่างผู้ใช้งานระบบกับยูสเคส

ส่วนความสัมพันธ์ระหว่างยูสเคสกับยูสเคสมี 2 ประเภทคือความสัมพันธ์แบบอินคลูด (Include) และความสัมพันธ์แบบเอ็กซ์เทน (Extend)

#### 2.1.3.1 ความสัมพันธ์แบบอินคลูด

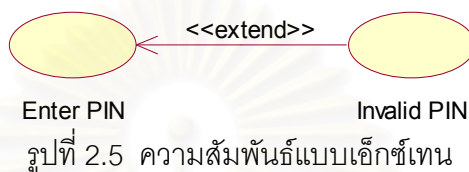
ความสัมพันธ์แบบอินคลูดเกิดขึ้นเมื่อยูสเคสหนึ่งเรียกใช้งานอีกยูสเคสหนึ่งซึ่งคล้ายกับการเรียกใช้งานโปรแกรมย่อยโดยโปรแกรมหลัก แผนภาพยูสเคสใช้เส้นตรงที่มีลูกศรชี้ไปหายูสเคสที่ถูกเรียกใช้ แทนความสัมพันธ์แบบอินคลูด และมีคำว่า <<include>> กำกับบนเส้นลูกศร ดังรูปที่ 2.4



รูปที่ 2.4 ความสัมพันธ์แบบอินคลูด

### 2.1.3.2 ความสัมพันธ์แบบเอ็กซ์เทน

ความสัมพันธ์แบบเอ็กซ์เทนเกิดขึ้นเมื่อยูสเคสหนึ่งไม่สามารถทำงานได้ตามปกติจึงมียูสเคสอีกอันทำงานแทนเพื่อเป็นทางเลือก แผนภาพยูสเคสใช้เส้นตรงที่มีหัวลูกศรชี้ไปหายูสเคสที่ถูกเอ็กซ์เทนแทนความสัมพันธ์แบบเอ็กซ์เทนและมีคำว่า <<extend>> กำกับบนเส้นลูกศร ดังรูปที่ 2.5



รูปที่ 2.5 ความสัมพันธ์แบบเอ็กซ์เทน

## 2.2 รูปแบบของรายละเอียดยูสเคส

รูปแบบของการเขียนรายละเอียดยูสเคสที่ Cockburn[4] เสนอมีหลายรูปแบบ แต่ Cockburn แนะนำว่ารายละเอียดยูสเคสในรูปแบบตารางหนึ่งสดมภ์ (One-Column Table) เป็นรูปแบบของรายละเอียดยูสเคสที่เข้าใจง่าย และปรับใช้กับเครื่องมือจัดทำเอกสารเช่น โลดัสโน้ต (Lotus Notes) และเวิร์ด (Word) เป็นต้น รายละเอียดยูสเคสรูปแบบตารางหนึ่งสดมภ์แสดงดังตารางที่ 2.1 โดยมีรายละเอียดดังนี้

ตารางที่ 2.1 รายละเอียดยูสเคสรูปแบบตารางหนึ่งสดมภ์ [4]

USE CASE #	<the name is the goal as a short active verb phrase>	
Context of use	<a longer statement of the context of use if needed>	
Scope	<what system is being considered black box under design>	
Level	<one of summary, primary task, subfunction>	
Primary Actor	<a role name for the primary actor, or a description>	
Stakeholder and interests	Stakeholder	Interests
	<stakeholder name>	<put here the interest of the stakeholder>
	<stakeholder name>	<put here the interest of the stakeholder>

ตารางที่ 2.1 รายละเอียดยูสเคสรูปแบบตารางหนึ่งสดมภ์ [4] (ต่อ)

Preconditions	<what we expect is already the state of the world>	
Minimal Guarantees	<the interests as protected on any exit>	
Success Guarantees	<the interests as satisfied on a successful ending>	
Trigger	<the action upon the system that starts the use case>	
Description	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery and any cleanup after>
	2	<...>
	3	
Extensions	Step	Branching Action
	1a	<condition causing branching>: 1a1 <action or name of sub use case> 1a2 <...>
Technology and Data Variations		
	1	<list of variations>

รายละเอียดยูสเคสรูปแบบตารางหนึ่งสดมภ์อธิบายดังนี้

- USE CASE# เป็นข้อความที่แสดงชื่อยูสเคส
- Context of use เป็นข้อความแสดงรายละเอียดของยูสเคสให้มากขึ้น
- Scope เป็นข้อความอธิบายขอบเขตที่ยูสเคสนี้ทำงานได้
- Primary Actor เป็นข้อความแสดงชื่อบุคคล หรือระบบที่ติดต่อใช้งานยูสเคส
- Stakeholder เป็นข้อความแสดงชื่อบุคคล หรือระบบที่ได้รับผลประโยชน์จากการทำงานของยูสเคส ซึ่งอาจเป็น Primary Actor ได้
- Interest เป็นข้อความแสดงผลประโยชน์ที่ Stakeholder ได้รับ
- Precondition เป็นข้อความแสดงกิจกรรมที่ต้องปฏิบัติก่อนเริ่มการทำงานยูสเคส
- Minimal guarantees เป็นข้อความอธิบายกิจกรรมที่ยูสเคสปฏิบัติเมื่อเกิดข้อผิดพลาด เช่น การทำเพิ่ม log เพื่อระบุข้อผิดพลาดและวันเวลาที่เกิดข้อผิดพลาด



- Success guarantees เป็นข้อความอธิบายผลลัพธ์ เมื่อเกิดเหตุการณ์สำเร็จ Success guarantees เรียกอีกอย่างว่า Post-condition
- Trigger เป็นข้อความแสดงเงื่อนไข หรือการทำงานที่ทำให้ยูสเคสเริ่มต้นทำงานได้
- Main Success Scenario เป็นข้อความแสดงลำดับเหตุการณ์สำเร็จ ซึ่งระบุเหตุการณ์ที่เกิดขึ้นเป็นข้อๆ
- Extension เป็นข้อความแสดงเงื่อนไขที่ทำให้เกิดเหตุการณ์อื่น และลำดับเหตุการณ์ที่เกิดขึ้นเมื่อเงื่อนไขนั้นเป็นจริง ลำดับเหตุการณ์ที่เกิดขึ้นจะถูกระบุเป็นข้อๆ
- Technology and Data variation เป็นข้อความแสดงการให้ข้อมูลนำเข้าของผู้ใช้งาน ยูสเคสด้วยวิธีอื่น เช่น ผู้ใช้งานระบุหมายเลขสมาชิกโดยอ่านแถบแม่เหล็กของบัตรสมาชิก หรืออ่านลายนิ้วมือ เป็นต้น

รายละเอียดยูสเคสรูปแบบตารางหนึ่งสดมภ์ใช้อธิบายยูสเคสหนึ่งยูสเคสของแผนภาพยูสเคส รายละเอียดยูสเคสแบบนี้มีลำดับเหตุการณ์สำเร็จ และอาจมีลำดับเหตุการณ์อื่นเพื่ออธิบายเหตุการณ์ที่อาจเกิดขึ้นจากเงื่อนไขที่ทำให้ยูสเคสทำงานไม่เป็นไปตามเป้าหมาย

## 2.3 การทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์ [5][6] เป็นกระบวนการประเมินผลซอฟต์แวร์เพื่อตรวจสอบว่าซอฟต์แวร์ทำงานเป็นไปตามความต้องการของซอฟต์แวร์

การทดสอบซอฟต์แวร์แบ่งระดับของการทดสอบเป็น 3 ระดับ คือ การทดสอบระดับหน่วย การทดสอบแบบบูรณาการ และการทดสอบระบบ การทดสอบแต่ละระดับมีรายละเอียดดังนี้

### 2.3.1 การทดสอบระดับหน่วย

การทดสอบระดับหน่วยเป็นการทดสอบความสามารถของโมดูล ซึ่งเป็นหน่วยที่เล็กที่สุดของการออกแบบซอฟต์แวร์ การทดสอบระดับหน่วยทำหลังจากพัฒนาโมดูลนั้นเสร็จ

### 2.3.2 การทดสอบแบบบูรณาการ

การทดสอบแบบบูรณาการเป็นการทดสอบการทำงานร่วมกันของโมดูลที่ผ่านการทดสอบระดับหน่วยแล้ว การทดสอบบูรณาการมีหลายวิธี เช่น การรวมจากบนลงล่าง (Top-down approach) การรวมจากล่างขึ้นบน (Bottom-up approach) การรวมแบบแซนด์วิช (Sandwich approach) และการรวมแบบบิกแบง (Big-bang approach) เป็นต้น

### 2.3.3 การทดสอบระบบ

การทดสอบระบบเป็นการทดสอบว่าซอฟต์แวร์มีการทำงานเป็นไปตามความต้องการของผู้ใช้งาน และข้อกำหนดของซอฟต์แวร์หรือไม่ การทดสอบระบบประกอบด้วย 5 การทดสอบดังนี้

2.3.3.1 การทดสอบฟังก์ชัน (Functional Testing) เป็นการทดสอบที่ทดสอบการทำงานของซอฟต์แวร์ให้เป็นไปตามความต้องการของผู้ใช้งานด้วยเทคนิคแบบแบล็กบ็อกซ์ (Black Box) ซึ่งการทดสอบแบบแบล็กบ็อกซ์ไม่สนใจกระบวนการทำงานภายในของซอฟต์แวร์

2.3.3.2 การทดสอบความกดดัน (Stress Testing) เป็นการทดสอบซอฟต์แวร์ภายใต้สภาวะการใช้อย่างหนัก เพื่อตรวจหาความสามารถสูงสุดของซอฟต์แวร์ภายใต้ทรัพยากรที่กำหนด

2.3.3.3 การทดสอบประสิทธิภาพ (Performance Testing) เป็นการทดสอบประสิทธิภาพซอฟต์แวร์ตามที่ระบุในข้อกำหนดของซอฟต์แวร์ การทดสอบประสิทธิภาพตรวจสอบจากเวลาตอบสนองการทำงานของซอฟต์แวร์ การทดสอบประสิทธิภาพจะทำงานที่สภาวะการใช้งานปกติ

2.3.3.4 การทดสอบความปลอดภัย (Security Testing) เป็นการทดสอบระบบรักษาความปลอดภัยของซอฟต์แวร์จากผู้ไม่มีสิทธิใช้งานซอฟต์แวร์

2.3.3.5 การทดสอบพื้นหลัง (Background Testing) เป็นการทดสอบซอฟต์แวร์ด้านความสามารถการรองรับรายการทำงานหลายรายการในเวลาเดียวกัน

## 2.4 เอ็กซ์เอ็มแอล (XML: eXtensible Markup Language)

ในปัจจุบันการจัดเก็บข้อมูลที่มีโครงสร้างเป็นลำดับชั้น (Hierarchy) ได้นำไปใช้ในการสื่อสารระหว่างโปรแกรมต่างๆ บนอินเทอร์เน็ตมากขึ้น ทำให้การส่งผ่านข้อมูลระหว่างโปรแกรมที่อยู่ต่างระบบปฏิบัติการนั้นค่อนข้างยาก ดังนั้นจำเป็นต้องมีการกำหนดภาษากลางเพื่อเป็นมาตรฐานที่ใช้นิยาม และอธิบายข้อมูล ภาษาเอ็กซ์เอ็มแอล [7] จึงเป็นภาษาที่สร้างขึ้นเพื่อเป็นภาษามาตรฐานใช้อธิบายข้อมูล เอกสารเอ็กซ์เอ็มแอลมีการประกาศใช้ที่ดี (DTD: Document Type Definitions) เพื่อนิยามโครงสร้างของข้อมูลภายในเอกสารเอ็กซ์เอ็มแอล

เนื่องจากภาษาเอ็กซ์เอ็มแอลเป็นภาษามาร์คอัพ (Markup Language) ดังนั้นข้อมูลภายในเอกสารเอ็กซ์เอ็มแอลประกอบด้วยอีลีเมนต์ (Elements) และแอททริบิวต์ (Attribute) ต่างๆ

ของแต่ละอีลีเมนต์ที่ใช้ในการอธิบายข้อมูล แต่ละอีลีเมนต์เขียนแทนด้วยแท็ก (Tag) และเมื่อจบการใช้งานอีลีเมนต์ต้องมีแท็กปิดเสมอ เอกสารเอ็กซ์เอ็มแอลแบ่งเป็น 2 ชนิดคือ

2.4.1 เอกสารเอ็กซ์เอ็มแอลที่มีรูปแบบถูกต้อง (Well-Formed XML Document) คือ เอกสารเอ็กซ์เอ็มแอลที่มีคุณสมบัติถูกต้องตามกฎหมายข้อบังคับพื้นฐานของการเขียนเอกสารเอ็กซ์เอ็มแอล ซึ่งกฎข้อบังคับพื้นฐานของการเขียนเอ็กซ์เอ็มแอลมีดังนี้

2.4.1.1 เอกสารต้องมีอีลีเมนต์ระดับบนสุดเพียงอีลีเมนต์เดียวเท่านั้น ซึ่งเรียกว่า อีลีเมนต์เอกสาร (Document Element) หรืออีลีเมนต์ราก (Root Element) โดยอีลีเมนต์อื่นทั้งหมดต้องซ้อนอยู่ภายในอีลีเมนต์ราก เช่น จากรูปที่ 2.6 อีลีเมนต์รากคือ <Student>

2.4.1.2 อีลีเมนต์ต้องซ้อนอยู่อย่างถูกต้อง หมายความว่าถ้ามีอีลีเมนต์ซ้อนอยู่ภายในอีลีเมนต์อื่น ต้องปิดอีลีเมนต์ที่ซ้อนในอีลีเมนต์ที่ถูกซ้อน เช่น จากรูปที่ 2.6 อีลีเมนต์ <FirstName> ซ้อนอยู่ในอีลีเมนต์ <Student> อย่างถูกต้อง

2.4.1.3 แต่ละอีลีเมนต์ต้องมีแท็กเปิด และแท็กปิดท้าย

2.4.1.4 ชื่อของอีลีเมนต์ในแท็กเปิด ต้องมีชื่อเดียวกับชื่อของแท็กปิดท้าย เช่น อีลีเมนต์ Student มีแท็กเปิดเป็น <Student> ต้องมีแท็กปิดเป็น </Student> เป็นต้น

2.4.1.5 ชื่อของอีลีเมนต์มีลักษณะเป็น Case sensitive คืออักษรตัวใหญ่หรือตัวเล็กในชื่อมีความหมายแตกต่างกัน ข้อความทั้งหมดภายในเอกสารเอ็กซ์เอ็มแอลจะมีลักษณะเป็น Case sensitive เช่นกัน

2.4.2 เอกสารเอ็กซ์เอ็มแอลที่ถูกต้องสมบูรณ์ (Valid XML Document) คือเอกสารเอ็กซ์เอ็มแอลที่มีรูปแบบถูกต้องที่เพิ่มกฎอีก 2 ข้อคือ

2.4.2.1 ต้องมีการประกาศการใช้ดีทีดี ข้อมูลภายในดีทีดีเป็นข้อกำหนดโครงสร้างของเอกสารนั้น เช่น จากรูปที่ 2.6 อีลีเมนต์ <!DOCTYPE> คืออีลีเมนต์ที่ประกาศการใช้ดีทีดี ซึ่งภายในดีทีดีอธิบายอีลีเมนต์ <Student> ต้องประกอบด้วยอีลีเมนต์ <FirstName> อีลีเมนต์ <LastName> อีลีเมนต์ <Age> อีลีเมนต์ <ID> และอีลีเมนต์ <Faculty> โดยเนื้อหาภายในอีลีเมนต์เป็นข้อมูลแบบอักขระ เป็นต้น

2.4.2.2 ส่วนต่างๆ ภายในเอกสารต้องถูกต้องตามโครงสร้างที่กำหนดในดีทีดี จากรูปที่ 2.6 ข้อมูลภายในอีลีเมนต์ <Student> เป็นไปตามโครงสร้างที่กำหนดในดีทีดี

```

<? xml version="1.0" standalone='yes' ?>

<!DOCTYPE Student
[
  <!ELEMENT Student (FirstName,LastName,Age,ID,Faculty)>
  <!ELEMENT FirstName (#PCDATA)>
  <!ELEMENT LastName (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT ID (#PCDATA)>
  <!ELEMENT Faculty (#PCDATA)>
]
>

<Student>
  <FirstName> Andryi </FirstName>
  <LastName> Shevchenko </LastName>
  <Age> 25 </Age>
  <ID> 250746352 </ID>
  <Faculty> Sciences </Faculty>
</Student>

```

รูปที่ 2.6 ตัวอย่างเอกสารเอ็กซ์เอ็มแอล

## 2.5 งานวิจัยที่เกี่ยวข้อง

2.5.1 งานวิจัย “Use Case-based Testing of Product Lines” โดย Bertolino และ Gnesi [8]

งานวิจัยนี้เสนอวิธีการ PLUTO (Product Line Use case Test Optimisation) สำหรับการทดสอบสายการผลิต (Product Line) วิธีการนี้ได้รับอิทธิพลจากวิธีการแบ่งหมวดหมู่ (Category Partition) แต่วิธีการ PLUTO เพิ่มความสามารถให้จัดการสายการผลิตที่เปลี่ยนแปลงบ่อยและสามารถสร้างกรณีทดสอบได้ รายละเอียดยูนิตเคสที่ใช้ในงานวิจัยนี้เรียกว่า PLUCs (Product Line Use Cases) เป็นรายละเอียดยูนิตเคสที่ประยุกต์จากรายละเอียดยูนิตเคสที่นำเสนอโดย Cockburn [4] รายละเอียดยูนิตเคสนี้เพิ่มเติมข้อมูลบางประการเพื่อใช้สร้างกรณีทดสอบ วิธีการนี้ตั้งอยู่บนพื้นฐานความต้องการของระบบในรูปแบบภาษาธรรมชาติ (Natural language)

การสร้างกรณีทดสอบในส่วนของการระบุหมวดหมู่และการระบุตัวเลือก (Choice) ขึ้นอยู่กับการตัดสินใจของผู้ทดสอบ อย่างไรก็ตามงานวิจัยนี้แนะนำให้ใช้เครื่องมือวิเคราะห์คำศัพท์ (Lexical analyzer) และเครื่องมือวิเคราะห์โครงสร้างไวยากรณ์ (Syntactical analyzer) แยกข้อมูลสำคัญเพื่อใช้ระบุหมวดหมู่ที่สำคัญ

ผลลัพธ์ของงานวิจัยนี้คือ วิธีการทดสอบของสายการผลิตและการสร้างกรณีทดสอบที่เรียกว่า PLUTO แต่การสร้างกรณีทดสอบที่งานวิจัยนี้นำเสนอไม่ใช่การสร้างกรณีทดสอบโดยอัตโนมัติ ซึ่งการสร้างกรณีทดสอบขึ้นอยู่กับการตัดสินใจของผู้ทดสอบ

## 2.5.2 งานวิจัย “Generating Test Cases from Use Cases” โดย Heumann [1]

งานวิจัยนี้นำเสนอวิธีการนำยูสเคสมาใช้สร้างกรณีทดสอบ โดยขั้นตอนการสร้างกรณีทดสอบเริ่มจากนำเอกสารยูสเคสซึ่งมีรูปแบบของรายละเอียดยูสเคสที่ประกอบด้วยชื่อยูสเคส คำอธิบายยูสเคส คำอธิบายเหตุการณ์ที่ระบบต้องปฏิบัติ สิ่งที่ต้องทำก่อนทำงานยูสเคส และสิ่งที่ได้รับเมื่อจบการทำงานยูสเคส ซึ่งคำอธิบายเหตุการณ์แบ่งออกเป็น 2 แบบคือลำดับการทำงานตามปกติ และลำดับการทำงานที่เป็นทางเลือกอื่นหรือการทำงานผิดปกติ จากนั้นสร้างตารางแสดงลำดับเหตุการณ์ที่เป็นไปได้ของยูสเคส โดยกำหนดให้แต่ละลำดับเหตุการณ์จะสร้างกรณีทดสอบได้อย่างน้อยหนึ่งกรณีทดสอบ จากนั้นสร้างตารางกรณีทดสอบโดยสดมภ์แรกของตารางเป็นรหัสกรณีทดสอบ สดมภ์ที่สองเป็นคำอธิบายลำดับเหตุการณ์ที่เกิดขึ้นอย่างคร่าวๆ สดมภ์ที่สามเหลือ(ยกเว้นสดมภ์สุดท้าย)เป็นข้อมูลนำเข้าของยูสเคส และสดมภ์สุดท้ายเป็นค่าคาดหวังของกรณีทดสอบ จากนั้นสร้างข้อมูลทดสอบสำหรับกรณีทดสอบ สำหรับวิธีการสร้างข้อมูลทดสอบนั้นอยู่นอกเหนือจากขอบเขตงานวิจัยนี้

ผลลัพธ์ของงานวิจัยนี้คือ กรณีทดสอบที่สร้างจากเอกสารยูสเคส ซึ่งต้องทำความเข้าใจเอกสารยูสเคสก่อนเพื่อสร้างลำดับการทำงาน อย่างไรก็ตามการสร้างกรณีทดสอบที่งานวิจัยนี้เสนอไม่เป็นการสร้างกรณีทดสอบโดยอัตโนมัติ

## 2.5.3 เครื่องมือ “SpecStudio” โดย Impl Software, Inc. [2]

SpecStudio เป็นเครื่องมือช่วยวิเคราะห์และสร้างเอกสารเชิงพฤติกรรมของระบบ (Behavior Specification) เครื่องมือนี้สามารถสร้างกรณีทดสอบได้เร็วและมีคุณภาพซึ่งมีผลให้เวลาในการตรวจสอบ (Review) และการบำรุงรักษา (Maintenance) น้อยลง เครื่องมือนี้สร้างกรณีทดสอบจากเอกสารเชิงพฤติกรรมของระบบ การสร้างเอกสารเชิงพฤติกรรมสร้างโดยเริ่มจากการกำหนดส่วนที่โปรแกรมติดต่อกับบุคคลหรือระบบภายนอก จากนั้นกำหนดวัตถุที่ใช้ติดต่อกับ

ผู้ใช้งานภายนอก (Interface Object) และวัตถุที่รองรับข้อมูลภายในของระบบ (Internal Object) แล้วจัดทำเอกสารเชิงพฤติกรรมของระบบสำหรับอธิบายผลกระทบจากพฤติกรรมของแต่ละวัตถุที่รองรับข้อมูลภายใน ซึ่งเอกสารเชิงพฤติกรรมของระบบนั้นประกอบด้วยชื่อของวัตถุที่ใช้ติดต่อกับผู้ใช้งานภายนอกหรือวัตถุที่รองรับข้อมูลภายใน คุณสมบัติของวัตถุ เงื่อนไขของการทำงานของวัตถุ ผลของการทำงานของวัตถุ และการทำงานของวัตถุต่อไปที่เป็นไปได้ หลังจากนั้นจึงสร้างกรณีทดสอบจากเอกสารเชิงพฤติกรรมของระบบ ความสัมพันธ์ระหว่างกรณีทดสอบและเอกสารเชิงพฤติกรรมของระบบช่วยวิเคราะห์ความครอบคลุมของกรณีทดสอบ (Test case coverage) ได้ การวิเคราะห์คุณภาพของกรณีทดสอบพิจารณาจากการทำงานที่ยังไม่ครอบคลุมกรณีทดสอบที่สร้างขึ้น ความสัมพันธ์นี้ช่วยตรวจสอบกรณีทดสอบง่ายขึ้น

ผลลัพธ์ของเครื่องมือนี้คือเอกสารเชิงพฤติกรรมของระบบ และกรณีทดสอบ ซึ่งกรณีทดสอบนี้สร้างจากเอกสารเชิงพฤติกรรมของระบบ แต่เครื่องมือนี้ไม่ได้สร้างกรณีทดสอบโดยอัตโนมัติผู้ใช้งานต้องทำด้วยตนเองจากเอกสารเชิงพฤติกรรมที่ออกแบบไว้

## บทที่ 3

### การออกแบบการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

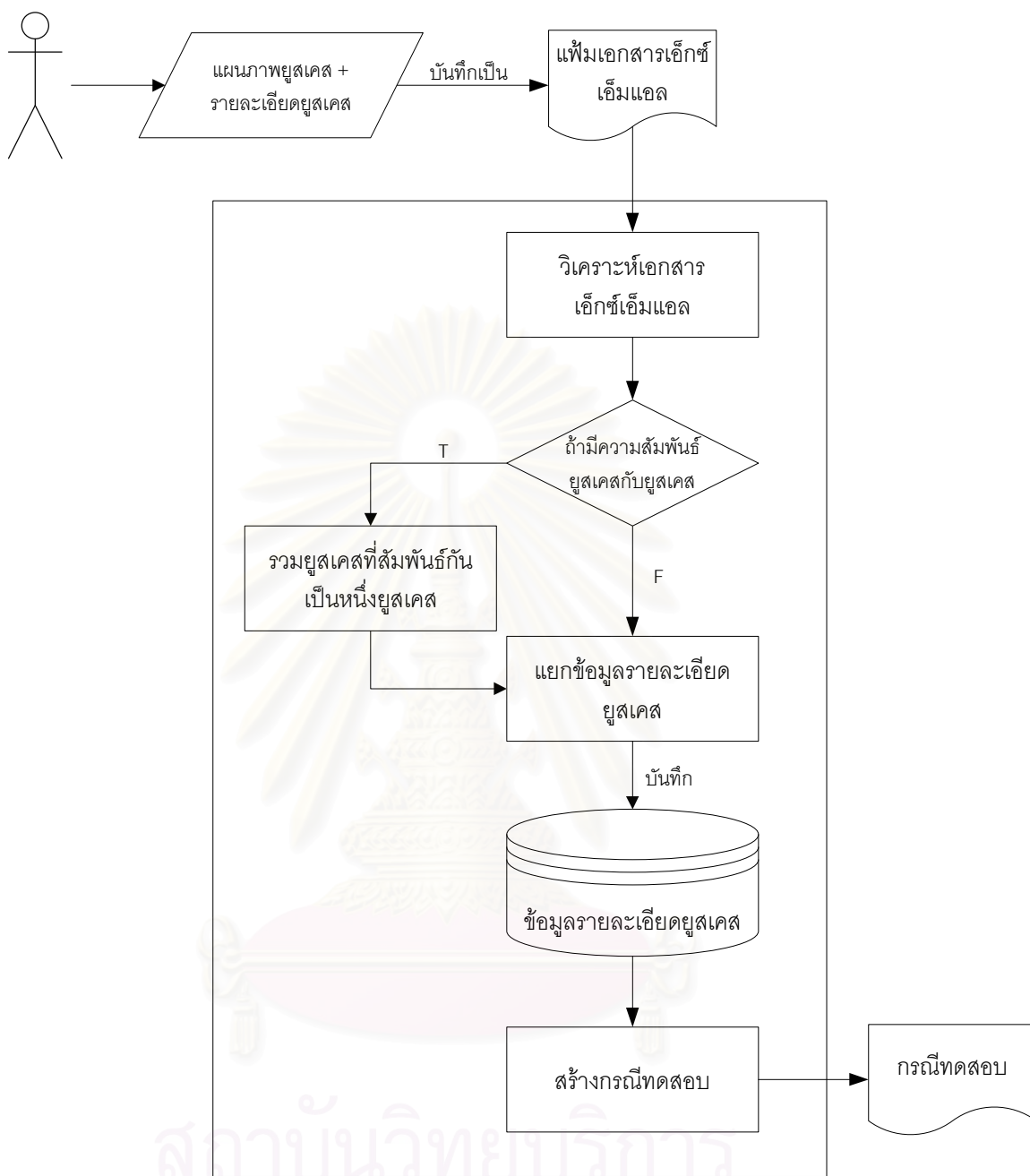
บทนี้เริ่มต้นโดยการเสนอแนวทางการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส ประกอบด้วย การเขียนแผนภาพยูสเคส การเขียนรายละเอียดยูสเคส การรวมยูสเคสที่มีความสัมพันธ์กัน และการสร้างกรณีทดสอบโดยอัตโนมัติ จากนั้นจะอธิบายถึงการวิเคราะห์และออกแบบเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติโดยใช้แผนภาพยูสเคส แผนภาพคลาส แผนภาพซีควเอนซ์ และแผนภาพลำดับกิจกรรม

#### 3.1 แนวคิดการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

Heumann [1] เสนอวิธีการนำยูสเคสมาใช้สร้างกรณีทดสอบ โดยผู้ทดสอบจะอ่านรายละเอียดยูสเคสแล้วจึงสร้างลำดับการทำงานของยูสเคส จากนั้นจึงสร้างกรณีทดสอบได้ ซึ่งการสร้างกรณีทดสอบวิธีนี้ไม่เป็นวิธีการสร้างกรณีทดสอบโดยอัตโนมัติ ดังนั้นงานวิจัยนี้จึงนำเสนอวิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส ซึ่งรายละเอียดยูสเคสที่ใช้ในงานวิจัยนี้จะระบุการทำงานของลำดับเหตุการณ์ที่เป็นไปได้ของยูสเคสพร้อมทั้งกำหนดหมายเลขลำดับการทำงาน เพื่อช่วยให้สร้างกรณีทดสอบได้โดยอัตโนมัติ

โดยทั่วไปแผนภาพยูสเคสประกอบด้วยสัญลักษณ์แทนผู้ใช้งานระบบ ยูสเคส และเส้นแสดงความสัมพันธ์ของยูสเคสและผู้ใช้งานระบบเท่านั้น ซึ่งทำให้ไม่สามารถสร้างกรณีทดสอบสำหรับแต่ละยูสเคสได้ งานวิจัยนี้จึงกำหนดให้เพิ่มข้อมูลรายละเอียดของแต่ละยูสเคสเพื่อช่วยให้สร้างกรณีทดสอบได้ ดังนั้นแนวทางการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสจึงประกอบด้วยขั้นตอนหลัก 3 ส่วนคือ การเขียนแผนภาพยูสเคสและรายละเอียดยูสเคส การรวมยูสเคสที่มีความสัมพันธ์กัน และการสร้างกรณีทดสอบโดยอัตโนมัติ ภาพรวมของแนวคิดของงานวิจัยนี้แสดงดังรูปที่ 3.1

ขั้นตอนการทำงานเริ่มต้นจากผู้วิเคราะห์ระบบออกแบบแผนภาพยูสเคสพร้อมทั้งกำหนดรายละเอียดยูสเคส แล้วบันทึกเป็นเอกสารเอ็กซ์เอ็มแอล จากนั้นวิเคราะห์เอกสารหาความสัมพันธ์ระหว่างยูสเคสกับยูสเคส ถ้าพบความสัมพันธ์จะรวมยูสเคสเหล่านั้นเป็นหนึ่งยูสเคส แล้วแยกรายละเอียดยูสเคสลงฐานข้อมูล และสร้างกรณีทดสอบจากข้อมูลในฐานข้อมูล โดยกรณีทดสอบจะอยู่ในรูปแบบเอกสารเอ็กซ์เอ็มแอล (HTML) ประกอบด้วยข้อมูลทดสอบ และผลลัพธ์ที่คาดหวัง



รูปที่ 3.1 ภาพรวมของแนวคิดการสร้างกรณีทดสอบจากยูสเคส



### 3.2 แผนภาพยูสเคสและรายละเอียดยูสเคส

งานวิจัยนี้ได้กำหนดรูปแบบรายละเอียดข้อมูลยูสเคสเพิ่มเติมสำหรับใช้สร้างกรณีทดสอบได้โดยอัตโนมัติ โดยมีส่วนประกอบบางส่วนซึ่งนำมาจากรายละเอียดยูสเคสของ Cockburn (ได้แก่ ชื่อของยูสเคส ชื่อผู้ใช้งานระบบ Precondition ลำดับเหตุการณ์สำเร็จ ลำดับเหตุการณ์ทางเลือกอื่น และ Postcondition) และส่วนประกอบที่จำเป็นซึ่งงานวิจัยกำหนดขึ้น (ได้แก่ หมายเลขยูสเคส ข้อมูลนำเข้าสำหรับยูสเคส ประโยคเงื่อนไขของลำดับเหตุการณ์ หมายเลขประโยคเงื่อนไข หมายเลขลำดับการทำงาน และคุณสมบัติการเป็นยูสเคสต้นแบบ) ซึ่งส่วนประกอบของรายละเอียดยูสเคสของงานวิจัยนี้เปรียบเทียบกับของ Cockburn แสดงได้ดังตารางที่ 3.1

ตารางที่ 3.1 ตารางเปรียบเทียบรายละเอียดยูสเคสของงานวิจัยนี้กับรายละเอียดยูสเคสของ Cockburn

ส่วนประกอบของรายละเอียดยูสเคส	งานวิจัยนี้	Cockburn
หมายเลขยูสเคส	มี	ไม่มี
ชื่อของยูสเคส	มี	มี
ข้อความอธิบายยูสเคส	มี	มี
ชื่อผู้ใช้งานระบบ	มี	มี
Pre-condition	มี	มี
ข้อมูลนำเข้า	มี	ไม่มี
ลำดับเหตุการณ์สำเร็จ	มี	มี
ลำดับเหตุการณ์ทางเลือกอื่น	มี	มี
Post-condition	มี	มี
คุณสมบัติการเป็นยูสเคสต้นแบบ	มี	ไม่มี
ขอบเขตการทำงานของยูสเคส	ไม่มี	มี
ระดับความสำคัญของยูสเคส	ไม่มี	มี
ผู้ได้รับผลประโยชน์	ไม่มี	มี
วิธีการให้ข้อมูลนำเข้า	ไม่มี	มี

ส่วนประกอบของรายละเอียดยูสเคส มีรายละเอียดดังนี้

1) Use case no. เป็นหมายเลขยูสเคส โดยแต่ละหมายเลขยูสเคสในแผนภาพยูสเคส ต้องไม่ซ้ำกัน

2) Use case name เป็นชื่อของยูสเคส ซึ่งบอกลักษณะการทำงานของยูสเคส โดยปกติจะขึ้นต้นด้วยคำกริยา ตัวอย่างเช่น “Add contact” เป็นต้น

3) Description เป็นข้อความอธิบายยูสเคสเพิ่มเติม เพื่อให้เข้าใจยิ่งขึ้น

4) Actor เป็นชื่อผู้ใช้งานระบบ หรือระบบที่ติดต่อใช้งานยูสเคส ซึ่งเป็นผู้ให้ข้อมูลนำเข้าสำหรับให้ยูสเคสเริ่มต้นทำงานได้

5) Pre-condition เป็นข้อความอธิบายกิจกรรมที่ต้องปฏิบัติก่อนเริ่มการทำงานของยูสเคส

6) Required-item เป็นข้อมูลนำเข้าสำหรับยูสเคสโดย Actor เป็นผู้ให้ข้อมูลเหล่านี้ ซึ่ง Required-item ประกอบด้วย

- Item name เป็นชื่อของข้อมูลนำเข้าของยูสเคส
- Item type เป็นชนิดของข้อมูลนำเข้าของยูสเคส โดยงานวิจัยนี้กำหนดให้ชนิดของข้อมูลนำเข้ามีทั้งหมด 4 ชนิดได้แก่

- Integer เป็นข้อมูลชนิดตัวเลขจำนวนเต็ม
- Float เป็นข้อมูลชนิดตัวเลขจำนวนจริง
- String เป็นข้อมูลชุดอักขระ หรือข้อความ
- Boolean เป็นข้อมูลชนิดตรรกะ มีค่าเป็นไปได้สำหรับข้อมูลชนิดนี้ 2 แบบคือ ค่าจริง (True) หรือเท็จ (False)

- Item size เป็นขนาดของข้อมูลนำเข้าของยูสเคส ซึ่งการกำหนดขนาดของข้อมูลนำเข้าขึ้นอยู่กับชนิดของข้อมูลนำเข้าดังนี้

- ข้อมูลชนิดตัวเลขจำนวนเต็ม ให้ใส่ตัวเลขกำหนดจำนวนหลักสูงสุดที่เป็นไปได้สำหรับการสร้างข้อมูลทดสอบ ถ้าไม่ได้กำหนดตัวเลขจำนวนหลักสูงสุดแล้วจำนวนหลักของข้อมูลทดสอบจะถูกสุ่มโดยอัตโนมัติ

- ข้อมูลชนิดตัวเลขจำนวนจริง ให้ใส่ตัวเลขกำหนดจำนวนหลักหลังจุดทศนิยมของข้อมูลทดสอบ ถ้าไม่กำหนดจำนวนหลักหลังจุดทศนิยมของข้อมูลทดสอบจะถูกสุ่มโดยอัตโนมัติ

- ข้อมูลชุดอักขระ ให้ใส่ตัวเลขกำหนดความยาวของข้อมูลชุดอักขระ ความยาวของข้อมูลทดสอบจะถูกสุ่มโดยอัตโนมัติในกรณีที่ไม่ได้กำหนดความยาวของข้อมูลชุดอักขระ

- ข้อมูลชนิดตรรกะไม่ต้องระบุขนาดของข้อมูลนำเข้า

- Max value เป็นค่าสูงสุดที่เป็นไปได้ของข้อมูลนำเข้าของยูสเคส ซึ่งข้อมูลทดสอบที่สุ่มได้ต้องมีค่าไม่มากกว่าค่านี้

- Min value เป็นค่าต่ำสุดที่เป็นไปได้ของข้อมูลนำเข้าของยูสเคส ซึ่งข้อมูลทดสอบที่สุ่มได้ต้องมีค่าไม่น้อยกว่าค่านี้

7) Success scenario เป็นลำดับเหตุการณ์สำเร็จของการทำงานยูสเคส ที่เกิดขึ้นหลังจากที่ผู้ใช้งานระบบให้ข้อมูลนำเข้าซึ่งสอดคล้องกับเงื่อนไขปกติของยูสเคส โดย Success scenario ต้องระบุรายละเอียดต่อไปนี้

- Condition เป็นประโยคเงื่อนไข (Condition clause) ซึ่งใช้ตรวจสอบข้อมูลนำเข้าก่อนการทำงานยูสเคส ถ้าข้อมูลนำเข้าสอดคล้องกับประโยคเงื่อนไขแล้วลำดับเหตุการณ์ทำงานของยูสเคสจะเริ่มทำงาน รูปแบบการเขียนประโยคเงื่อนไขสำหรับงานวิจัยนี้คล้ายกับการเขียนประโยคเงื่อนไขของการเขียนโปรแกรมคอมพิวเตอร์ ซึ่งวิธีการเขียนอธิบายเพิ่มเติมในหัวข้อ 3.2.3

- Condition no. เป็นหมายเลขของประโยคเงื่อนไข เนื่องจากในแต่ละยูสเคสมีลำดับเหตุการณ์สำเร็จเพียงหนึ่งลำดับเหตุการณ์ ดังนั้นจึงกำหนดให้หมายเลขของประโยคเงื่อนไขของลำดับเหตุการณ์สำเร็จเป็นหมายเลข 0 เสมอ

- Step เป็นหมายเลขลำดับการทำงานของยูสเคส ซึ่งการทำงานของยูสเคสอธิบายเพิ่มเติมใน Action และการเขียนหมายเลขลำดับการทำงานของยูสเคสอธิบายเพิ่มเติมในหัวข้อ 3.2.2

- Action เป็นการทำงานของยูสเคส เมื่อประโยคเงื่อนไขของลำดับเหตุการณ์เป็นจริง โดยแต่ละลำดับเหตุการณ์สำเร็จของยูสเคสอาจมีมากกว่า 1 การทำงาน

8) Alternative scenario เป็นลำดับเหตุการณ์ทางเลือกอื่นของการทำงานยูสเคส ที่เกิดขึ้นเมื่อการทำงานบางขั้นตอนของลำดับเหตุการณ์สำเร็จทำงานไม่เป็นไปตามปกติ

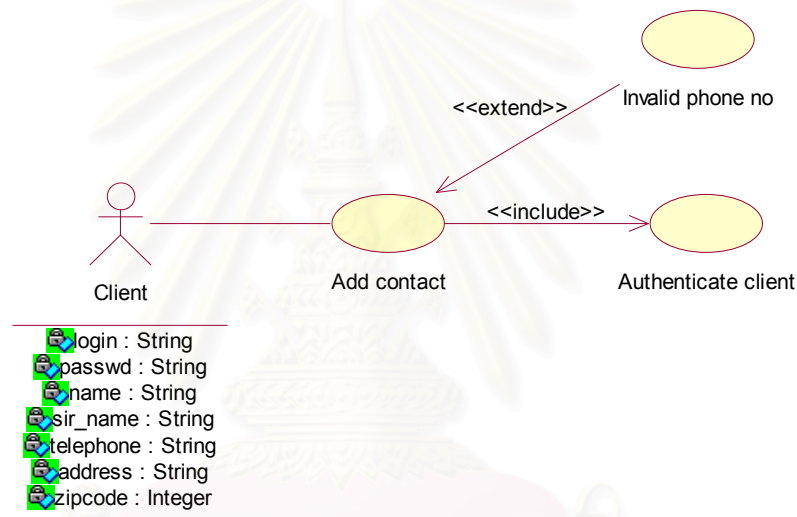
ตัวอย่างเช่น การทำงานเพื่อจัดการข้อผิดพลาด (Exception handling) ของข้อมูลนำเข้า เป็นต้น หรือการทำงานบางขั้นตอนของลำดับเหตุการณ์สำเร็จสามารถเลือกทำงานได้มากกว่า 1 วิธี โดย Alternative scenario ต้องระบุรายละเอียดเช่นเดียวกับ Success scenario แต่แตกต่างกันตรงวิธีการกำหนดหมายเลขของประโยคเงื่อนไข และหมายเลขลำดับการทำงานของยูสเคส ซึ่งวิธีการกำหนดหมายเลขของประโยคเงื่อนไขอธิบายเพิ่มเติมในหัวข้อ 3.2.1 และวิธีการกำหนดหมายเลขการทำงานของยูสเคสอธิบายเพิ่มเติมในหัวข้อ 3.2.2

9) Post-condition เป็นข้อความอธิบายผลลัพธ์หลังจากการทำงานยูสเคส ซึ่งแต่ละยูสเคสอาจมีหลายการทำงาน (ทั้งการทำงานสำเร็จ และการทำงานทางเลือกอื่น) ทำให้ผลลัพธ์อาจแตกต่างกัน ดังนั้น Post-condition จึงแยกเป็นผลลัพธ์ของแต่ละการทำงานของยูสเคส โดยหมายเลข

ของผลลัพธ์ของลำดับเหตุการณ์สำเร็จ และเหตุการณ์ทางเลือกอื่นเป็นหมายเลขเดียวกับหมายเลขประโยคเงื่อนไขของลำดับเหตุการณ์สำเร็จ และเหตุการณ์ทางเลือกอื่นตามลำดับ

10) Is abstract เป็นคุณสมบัติที่กำหนดการเป็น abstract use case หรือยูสเคสที่เป็นต้นแบบสำหรับยูสเคสอื่น โดยในงานวิจัยนี้ระบุให้เป็น 0 เมื่อต้องการกำหนดให้เป็นยูสเคสต้นแบบ แต่ถ้าไม่ต้องการกำหนดให้เป็นยูสเคสต้นแบบให้ระบุเป็น 1 ในงานวิจัยนี้จะไม่สร้างกรณีทดสอบจากยูสเคสที่เป็นยูสเคสต้นแบบ

ตัวอย่างแผนภาพยูสเคสแสดงดังรูปที่ 3.2 และตัวอย่างรายละเอียดยูสเคสหมายเลข 1: Add contact ของยูสเคส Add contact ในแผนภาพยูสเคสแสดงดังตารางที่ 3.2



รูปที่ 3.2 ตัวอย่างแผนภาพยูสเคส : Contact list manager

ตารางที่ 3.2 ตัวอย่างรายละเอียดยูสเคสหมายเลข 1: Add contact

Use case no.:	1
Use case name:	Add contact
Description:	Adding new contact to list
Actor:	Client
Pre-condition:	Client enters the contact information: name, surname, telephone number, address and zip code.

ตารางที่ 3.2 ตัวอย่างรายละเอียดคุณสมบัติหมายเลข 1: Add contact (ต่อ)

Required-item:		Name	Type	Size	Max	Min
		name	String	15	-	-
		surname	String	20	-	-
		telephone	String	7	-	-
		address	String	20	-	-
		zipcode	Integer	5	10000	99999
Is abstract:		0				
Success scenario:						
Condition no:	0	(name.length > 0) && (surname.length > 0) && (telephone>="0000000") && (telephone<="9999999")				
Step		Action				
1		{UC2}				
2		System submits the contact information from client.				
3		System saves the contact information into database.				
4		System shows a message "Save new contact complete".				
Alternative scenario:						
Condition no:	2.1	(name.length <= 0)				
Step		Action				
2.1.1		System shows an error message "Please enter name".				
Condition no:	2.2	(surname.length <= 0)				
Step		Action				
2.2.1		System shows an error message "Please enter surname"				
Condition no:	2.3	n/a				
Step		Action				
2.3		{UC3}				
Post-condition	0	System saves new contact and shows a message "Save new contact complete".				
	2.1	System shows an error message "Please enter name"				
	2.2	System shows an error message "Please enter surname"				

งานวิจัยนี้กำหนดรูปแบบรายละเอียดยูสเคสตามที่กล่าวมา ซึ่งการกำหนดหมายเลข ประโยคเงื่อนไข หมายเลขลำดับของการทำงานยูสเคส และการเขียนประโยคเงื่อนไขของ การทำงานยูสเคส เป็นส่วนสำคัญมากในการสร้างกรณีทดสอบ มีรายละเอียดดังนี้

### 3.2.1 วิธีการกำหนดหมายเลขประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่น

จากที่ได้กล่าวแล้วว่าหมายเลขประโยคเงื่อนไขของลำดับเหตุการณ์สำเร็จของ การทำงานยูสเคสถูกกำหนดให้เป็นหมายเลข 0 เสมอ อันเนื่องมาจากในแต่ละยูสเคสมีลำดับ เหตุการณ์สำเร็จได้เพียงหนึ่งลำดับเหตุการณ์ แต่การกำหนดหมายเลขประโยคเงื่อนไขของลำดับ เหตุการณ์ทางเลือกอื่นนั้นแตกต่างกับของลำดับเหตุการณ์สำเร็จ โดยหมายเลขประโยคเงื่อนไข ของลำดับเหตุการณ์ทางเลือกอื่นนี้ถูกกำหนดขึ้นเพื่อเป็นจุดอ้างอิงถึงเหตุการณ์ หรือการทำงานใน ลำดับเหตุการณ์สำเร็จที่ไม่สามารถทำงานได้ตามปกติอันเนื่องจากข้อมูลนำเข้า การกำหนด หมายเลขประโยคเงื่อนไขมีรูปแบบดังรูปที่ 3.3

*<Order no. of Success scenario> . <Running no.>*

### รูปที่ 3.3 รูปแบบของหมายเลขประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่น

จากรูปที่ 3.3 รูปแบบของหมายเลขประโยคเงื่อนไขประกอบด้วยหมายเลข 2 ชุด ซึ่งคั่นด้วยเครื่องหมายมหัพภาค ได้แก่

- Order no. of Success scenario เป็นหมายเลขลำดับการทำงานสำเร็จที่ไม่สามารถทำงานได้ตามปกติ หรือหมายเลขลำดับการทำงานสำเร็จที่เลือกทำงานได้มากกว่า 1 วิธี

- Running no. เป็นหมายเลขซึ่งเริ่มต้นตั้งแต่หมายเลข 1 เป็นต้นไป

จากรายละเอียดยูสเคส: Add contact ในตารางที่ 3.1 ลำดับเหตุการณ์สำเร็จมี การทำงานจำนวน 4 ข้อ ในกรณีที่ Client ไม่ใส่ข้อมูล name เมื่อยูสเคสทำงานตามลำดับ เหตุการณ์สำเร็จถึงการทำงานหมายเลข 2 (ระบบรับข้อมูลจาก Client) หลังจากนั้นจะไม่สามารถ ทำงานตามเหตุการณ์ปกติได้ จึงต้องทำงานลำดับเหตุการณ์ทางเลือกอื่นสำหรับกรณีที่ข้อมูล name มีความยาวน้อยกว่าหรือเท่ากับศูนย์ ( $\text{name.length} \leq 0$ ) โดยหมายเลขประโยคเงื่อนไข สำหรับเหตุการณ์ทางเลือกอื่นนี้คือ 2.1

### 3.2.2 วิธีการกำหนดหมายเลขลำดับการทำงานยูสเคส

หมายเลขลำดับการทำงานยูสเคสเป็นหมายเลขแสดงลำดับการทำงานแต่ละขั้นตอนของยูสเคส โดยต้องกำหนดหมายเลขลำดับการทำงานให้กับการทำงานแรกจนกระทั่งการทำงานสุดท้าย ซึ่งการกำหนดหมายเลขลำดับการทำงานยูสเคสแบ่งออกเป็น 2 แบบคือ

- หมายเลขลำดับการทำงานของลำดับเหตุการณ์สำเร็จ

เป็นหมายเลขลำดับการทำงานซึ่งอยู่ในลำดับเหตุการณ์สำเร็จ โดยกำหนดให้หมายเลขลำดับการทำงานสำเร็จเริ่มต้นตั้งแต่หมายเลข 1 เป็นต้นไป ดังนั้นถ้าลำดับเหตุการณ์สำเร็จประกอบด้วยการทำงาน 5 ขั้นตอน หมายเลขลำดับการทำงานจะเริ่มตั้งแต่หมายเลข 1 ถึงหมายเลข 5

- หมายเลขลำดับการทำงานของลำดับเหตุการณ์ทางเลือกอื่น

เป็นหมายเลขลำดับการทำงานซึ่งอยู่ในลำดับเหตุการณ์ทางเลือกอื่น โดยหมายเลขลำดับการทำงานทางเลือกอื่นมีรูปแบบดังรูปที่ 3.4

<Condition no.> . <Running no.>

รูปที่ 3.4 รูปแบบของหมายเลขลำดับการทำงานทางเลือกอื่น

จากรูปที่ 3.4 รูปแบบของหมายเลขลำดับการทำงานทางเลือกอื่นประกอบด้วยหมายเลข 2 ชุดคือ Condition no. ซึ่งเป็นหมายเลขประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่นและ Running no. เป็นหมายเลขซึ่งเริ่มต้นตั้งแต่หมายเลข 1 เป็นต้นไป โดยที่ค้นด้วยเครื่องหมายมหัพภาค ซึ่งจากตัวอย่างรายละเอียดยูสเคส: Add contact ในตารางที่ 3.1 หมายเลขประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่น 2.1 ดังนั้นหมายเลขลำดับการทำงานเริ่มต้นตั้งแต่หมายเลข 2.1.1 เป็นต้นไป

### 3.2.3 การเขียนประโยคเงื่อนไข

ในแต่ละยูสเคสอาจมีลำดับเหตุการณ์ที่เกิดขึ้นได้หลายลำดับเหตุการณ์ การสร้างข้อมูลทดสอบเพื่อใช้ทดสอบลำดับเหตุการณ์เหล่านั้นจึงต้องกำหนดประโยคเงื่อนไขซึ่งใช้เป็นข้อกำหนดในการร่วมสร้างข้อมูลทดสอบ การเขียนประโยคเงื่อนไขของงานวิจัยนี้กำหนดให้เป็นไปตามข้อกำหนดดังนี้

- 1) กำหนดให้ข้อมูลนำเข้าที่ระบุไว้ใน required-item 1 ตัวคือตัวแปร 1 ตัวแปร

- 2) ชื่อตัวแปรในประโยคเงื่อนไขต้องถูกระบุไว้ใน required-item และต้องสะกดเหมือนกัน
- 3) แต่ละพจน์ในประโยคเงื่อนไขต้องเขียนไว้ภายในวงเล็บเท่านั้น เช่น (card\_id==1234) เป็นต้น
- 4) เครื่องหมายเปรียบเทียบที่ใช้ในประโยคเงื่อนไขมีดังตารางที่ 3.3

ตารางที่ 3.3 เครื่องหมายเปรียบเทียบที่ใช้ในประโยคเงื่อนไข

เครื่องหมาย	คำอธิบาย	ชนิดข้อมูลของนำเข้า			
		Integer	Float	String	Boolean
==	เท่ากับ	ใช้ได้	ใช้ได้	ใช้ได้	ใช้ได้
<>	ไม่เท่ากับ	ใช้ได้	ใช้ได้	ใช้ได้	ใช้ไม่ได้
>	มากกว่า	ใช้ได้	ใช้ได้	ใช้ไม่ได้	ใช้ไม่ได้
>=	มากกว่าหรือเท่ากับ	ใช้ได้	ใช้ได้	ใช้ไม่ได้	ใช้ไม่ได้
<	น้อยกว่า	ใช้ได้	ใช้ได้	ใช้ไม่ได้	ใช้ไม่ได้
<=	น้อยกว่าหรือเท่ากับ	ใช้ได้	ใช้ได้	ใช้ไม่ได้	ใช้ไม่ได้

- 5) การระบุขนาดความยาวของข้อมูลนำเข้าชนิดชุดอักขระ ให้เขียนชื่อข้อมูลนำเข้า ตามด้วยเครื่องหมายจุลภาค และตามด้วยคำว่า length เช่น (FirstName.length <= 20) เป็นต้น
- 6) ค่าของข้อมูลนำเข้าชนิดชุดอักขระ ต้องกำหนดอยู่ภายในเครื่องหมายอัญประกาศ(“) เท่านั้น เช่น (UnivName=="chula") เป็นต้น
- 7) สำหรับข้อมูลนำเข้าชนิดตรรกะ กำหนดให้แทนค่าจริงด้วย True และแทนค่าเท็จด้วย False
- 8) ถ้าประโยคเงื่อนไขมีจำนวนพจน์มากกว่า 1 พจน์สามารถเชื่อมพจน์ได้ 2 แบบคือ แบบและ (&&) และแบบหรือ (||) เช่น ((X<=0)||(X>=10)) เป็นต้น
- 9) กำหนดให้ใช้เครื่องหมายทางคณิตศาสตร์ในประโยคเงื่อนไขได้ดังนี้บวก (+) ลบ (-) คูณ (\*) และหาร (/)
- 10) แต่ละพจน์ต้องประกอบด้วยชื่อข้อมูลนำเข้าหรือการคำนวณทางคณิตศาสตร์ เครื่องหมายเปรียบเทียบ และค่าขอบเขตของข้อมูลนำเข้า ตัวอย่างเช่น (Height - Weight < 110) เป็นต้น



11) จำนวนวงเล็บในประโยคเงื่อนไขต้องไม่เกิน 5 วงเล็บ และวงเล็บต้องซ้อนกันอย่างถูกต้อง เช่น  $((X \leq 0) \parallel (X \geq 11))$  วงเล็บของประโยคเงื่อนไขนี้ถูกต้องเพราะมีจำนวนวงเล็บ 3 วงเล็บและซ้อนกันอย่างถูกต้อง แต่  $((X \leq 0) \parallel (X \geq 11))$  วงเล็บของประโยคเงื่อนไขนี้ไม่ถูกต้อง เพราะมีวงเล็บเปิด 3 อันแต่มีวงเล็บปิด 2 อัน ส่วน  $((X \leq 0) \&\& X \geq 11))$  แม้ว่าวงเล็บของประโยคเงื่อนไขนี้มีวงเล็บ 3 วงเล็บ และซ้อนกันอย่างถูกต้อง แต่ตำแหน่งของวงเล็บไม่ถูกต้อง เป็นต้น

12) ประโยคเงื่อนไขมีพจน์ที่เป็นการคำนวณทางคณิตศาสตร์ได้เพียงพจน์เดียวเท่านั้น เช่น  $((Weight > 30) \&\& (Height - Weight < 110))$  เป็นต้น

### 3.2.4 การกำหนดจุดเชื่อมโยงยูสเคส

ความสัมพันธ์ระหว่างยูสเคสกับยูสเคสในแผนภาพยูสเคสมี 2 แบบคือ ความสัมพันธ์แบบอินคลูด และความสัมพันธ์แบบเอ็กซ์เทน ซึ่งแสดงด้วยเส้นเชื่อมโยงระหว่างยูสเคสหนึ่งไปอีกยูสเคสหนึ่งโดยแสดงชนิดความสัมพันธ์บนเส้นเชื่อมโยง

เอกสารข้อกำหนดของยูเอ็มแอล (UML Specification) [9] ระบุว่ายูสเคสที่มีความสัมพันธ์กันแบบเอ็กซ์เทนต้องมีการกำหนดจุดอ้างอิง (Extension point) ไว้ที่ยูสเคสหลัก (Base use case) เพื่อให้ยูสเคสที่เอ็กซ์เทนยูสเคสหลักอ้างอิงถึงขั้นตอนการทำงานในยูสเคสหลักได้ โดยจุดอ้างอิงในยูสเคสอาจมีได้หลายตำแหน่งในลำดับเหตุการณ์ที่เกิดขึ้นตามที่ถูกออกแบบกำหนด ในงานวิจัยนี้ไม่ได้มีการใช้จุดอ้างอิงเพื่อให้ยูสเคสอื่นอ้างอิงถึงตามเอกสารข้อกำหนดของยูเอ็มแอล เนื่องจากรูปแบบของรายละเอียดยูสเคสและวิธีการสร้างกรณีทดสอบโดยอัตโนมัติที่งานวิจัยนี้นำเสนอไม่รองรับกับการกำหนดจุดอ้างอิง ดังนั้นงานวิจัยนี้ออกแบบจุดเชื่อมโยงยูสเคสเพื่อระบุว่ายูสเคสที่ต้องการเชื่อมโยง โดยจุดเชื่อมโยงจะระบุอยู่ในลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสหลักเพื่อให้ยูสเคสที่เชื่อมโยงทำงานแทนยูสเคสหลัก

การกำหนดจุดเชื่อมโยงความสัมพันธ์ของยูสเคสในรายละเอียดยูสเคสของงานวิจัยนี้ทำได้ดังนี้

#### 3.2.4.1 จุดเชื่อมโยงยูสเคสสำหรับความสัมพันธ์แบบอินคลูด

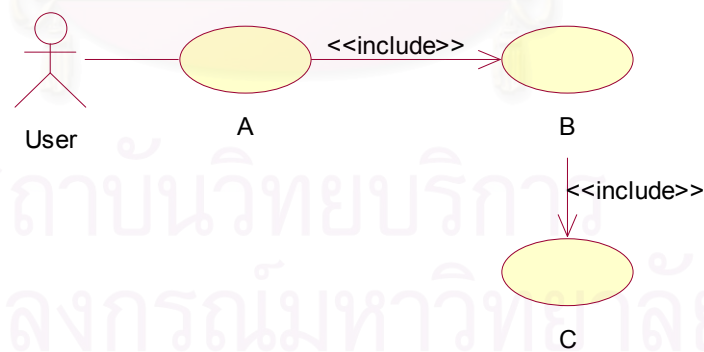
สำหรับความสัมพันธ์แบบอินคลูด งานวิจัยนี้กำหนดให้จุดเชื่อมโยงไปยังยูสเคสอื่นสามารถกำหนดได้เฉพาะในลำดับเหตุการณ์สำเร็จเท่านั้น การเขียนจุดเชื่อมโยงทำได้โดยเขียนเครื่องหมายปีกกาเปิด และ “UC” แล้วระบุหมายเลขยูสเคสที่ต้องการเรียกใช้ และปิดด้วยเครื่องหมายปีกกาปิด ดังในตัวอย่างรายละเอียดยูสเคสหมายเลข 1: Add contact จากตาราง

ที่ 3.1 การทำงานข้อที่ 1 ของลำดับเหตุการณ์สำเร็จมีการเรียกใช้งานยูสเคสหมายเลข 2 จึงเขียนจุดเชื่อมโยงเป็น "{UC2}"

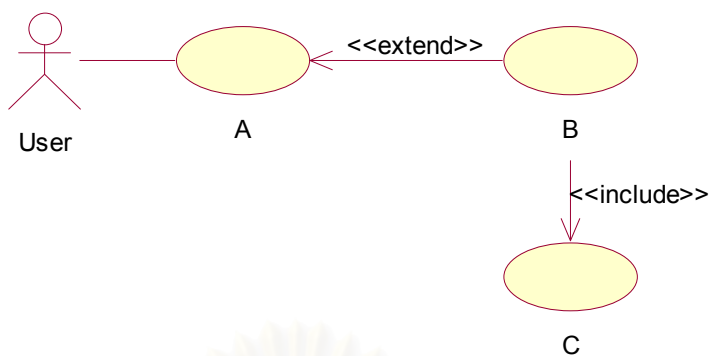
### 3.2.4.2 จุดเชื่อมโยงยูสเคสสำหรับความสัมพันธ์แบบเอ็กซ์เทน

การเขียนจุดเชื่อมโยงสำหรับความสัมพันธ์แบบเอ็กซ์เทนมีรูปแบบเช่นเดียวกับการเขียนจุดเชื่อมโยงสำหรับความสัมพันธ์แบบอินคลูด โดยงานวิจัยนี้กำหนดให้จุดเชื่อมโยงสำหรับความสัมพันธ์แบบเอ็กซ์เทนสามารถกำหนดได้เฉพาะในลำดับเหตุการณ์ทางเลือกอื่นเท่านั้น และไม่ต้องระบุประโยคเงื่อนไขสำหรับลำดับเหตุการณ์ทางเลือกอื่นที่มีจุดเชื่อมโยง ดังตัวอย่างจากตารางที่ 3.1 รายละเอียดยูสเคสหมายเลข 1: Add contact ลำดับเหตุการณ์ทางเลือกอื่นที่มีหมายเลขประโยคเงื่อนไข 2.3 มีการกำหนดจุดเชื่อมโยงเพื่อเรียกใช้งานยูสเคสหมายเลข 3 จึงเขียนจุดเชื่อมโยงเป็น "{UC3}" และไม่ระบุประโยคเงื่อนไข

สำหรับงานวิจัยนี้กำหนดให้ความสัมพันธ์ระหว่างยูสเคสกับยูสเคสต้องไม่ซ้อนกัน กล่าวคือถ้ายูสเคสหนึ่งเรียกใช้งานอีกยูสเคสหนึ่ง แล้วยูสเคสที่ถูกเรียกใช้งานต้องไม่มีการเรียกใช้งานยูสเคสอื่นดังตัวอย่างรูปที่ 3.4 ยูสเคส B ถูกเรียกใช้โดยยูสเคส A แต่ยูสเคส B มีการเรียกใช้ยูสเคส C ซึ่งทำให้เกิดการซ้อนกัน หรือยูสเคสที่ถูกเรียกใช้ซึ่งความสัมพันธ์กันแบบเอ็กซ์เทนต้องไม่มีลำดับการทำงานทางเลือกอื่น และไม่มีการเรียกใช้งานยูสเคสอื่นดังตัวอย่างรูปที่ 3.5 ยูสเคส B ถูกเรียกใช้โดยยูสเคส A ความสัมพันธ์แบบเอ็กซ์เทน แต่ยูสเคส B มีการเรียกใช้งานยูสเคส C ซึ่งทำให้เกิดการซ้อนเช่นกัน



รูปที่ 3.4 ภาพแสดงยูสเคสที่มีความสัมพันธ์ที่ซ้อนกันแบบที่ 1



รูปที่ 3.5 ภาพแสดงยูสเคสที่มีความสัมพันธ์ที่ซ้อนกันแบบที่ 2

ก่อนการสร้างกรณีทดสอบจะต้องรวมยูสเคสที่มีความสัมพันธ์กันให้เป็นหนึ่งยูสเคส แล้วจึงสามารถสร้างกรณีทดสอบได้ ซึ่งการรวมยูสเคสที่มีความสัมพันธ์ได้กล่าวไว้ในหัวข้อ 3.3 ต่อไป

### 3.3 การรวมยูสเคสที่มีความสัมพันธ์กัน

ในขั้นตอนการวิเคราะห์หันทวิเคราะห์ระบบมักเริ่มต้นเขียนแผนภาพยูสเคสแบบง่ายแสดงการทำงานของระบบก่อน กล่าวคือแผนภาพยูสเคสจะประกอบด้วยผู้ใช้งานระบบ ยูสเคสของการทำงานหลัก และเส้นแสดงความสัมพันธ์ระหว่างยูสเคสกับผู้ใช้งานระบบ หลังจากผ่านขั้นตอนการวิเคราะห์ระบบแล้วนักวิเคราะห์ระบบจะปรับให้แผนภาพยูสเคสมีความละเอียดยิ่งขึ้นซึ่งเรียกว่ากระบวนการนี้ว่า Refinement โดยการเพิ่มยูสเคสของการทำงานย่อย และความสัมพันธ์ระหว่างยูสเคสเพื่อให้เข้าใจยิ่งขึ้น ซึ่งเป็นประโยชน์ต่อการพัฒนาระบบ

จากที่ได้กล่าวมาสังเกตได้ว่าแผนภาพยูสเคสที่ทำการปรับแก้ไขแล้วนั้นมีที่มาจากแผนภาพยูสเคสอย่างง่าย ดังนั้นก่อนการสร้างกรณีทดสอบจึงต้องรวมความสัมพันธ์ระหว่างยูสเคสกับยูสเคสเข้าด้วยกันก่อนเพื่อแปลงให้เป็นแผนภาพยูสเคสอย่างง่ายที่มีเฉพาะยูสเคสของการทำงานหลัก

การรวมยูสเคสที่มีความสัมพันธ์กันสำหรับงานวิจัยนี้ แบ่งออกได้เป็น 2 แบบตามความสัมพันธ์ระหว่างยูสเคสได้แก่

#### 3.3.1 การรวมยูสเคสที่มีความสัมพันธ์แบบอินคลูด

ความสัมพันธ์ระหว่างยูสเคสแบบอินคลูดเกิดขึ้นเมื่อยูสเคสหนึ่งเรียกใช้งานอีกยูสเคสหนึ่งในลักษณะเดียวกับการโปรแกรมหลักที่เรียกใช้งานโปรแกรมย่อย จากหัวข้อที่ 3.2.4.1 ยูสเคสที่เรียกใช้ยูสเคสอื่นในลักษณะนี้จะมีจุดเชื่อมโยงระบุในการทำงานของลำดับเหตุการณ์

สำเร็จเท่านั้น ดังตัวอย่างรายละเอียดยูสเคสหมายเลข 1: Add contact การทำงานขั้นตอนที่ 2 ของลำดับเหตุการณ์สำเร็จเรียกใช้งานยูสเคสหมายเลข 2

การรวมยูสเคสที่มีความสัมพันธ์แบบอินคลูซีฟมีขั้นตอนดังนี้

- 1) กำหนดยูสเคสหลักและยูสเคสรอง
 

เริ่มจากกำหนดให้ยูสเคสที่เรียกใช้งานเป็นยูสเคสหลัก และให้ยูสเคสที่ถูกเรียกใช้งานเป็นยูสเคสรอง ดังนั้นจากตารางที่ 3.1 ยูสเคสหมายเลข 1 เป็นยูสเคสหลัก และยูสเคสหมายเลข 2 เป็นยูสเคสรอง
- 2) เพิ่ม pre-condition ของยูสเคสรองให้ยูสเคสหลัก
 

นำ pre-condition ของยูสเคสรองต่อท้าย pre-condition ของยูสเคสหลัก
- 3) เพิ่มข้อมูลนำเข้า required-item ของยูสเคสรองให้กับยูสเคสหลัก
 

นำข้อมูลนำเข้า required-item ของยูสเคสรองเพิ่มให้กับยูสเคสหลักเพื่อเป็นข้อมูลนำเข้าของยูสเคสหลักด้วย จากตารางที่ 3.3 ตัวอย่างรายละเอียดยูสเคสหมายเลข 2: Authenticate client มีข้อมูลนำเข้า 2 ตัวคือ login และ password เพิ่มให้เป็นข้อมูลนำเข้าของยูสเคสหลัก
- 4) รวมประโยคเงื่อนไขของลำดับเหตุการณ์สำเร็จ
 

นำประโยคเงื่อนไขของลำดับเหตุการณ์สำเร็จของยูสเคสรองรวมกับประโยคเงื่อนไขของลำดับเหตุการณ์สำเร็จของยูสเคสหลัก โดยเชื่อมตัวดำเนินการทางตรรกะ “และ (&&)”
- 5) เพิ่มการทำงานของลำดับเหตุการณ์สำเร็จของยูสเคสรองให้ยูสเคสหลัก
 

เพิ่มการทำงานของลำดับเหตุการณ์สำเร็จของยูสเคสรองให้กับการทำงานของลำดับเหตุการณ์สำเร็จของยูสเคสหลัก สำหรับการทำงานใหม่ให้แทรกแทนจุดเชื่อมโยงไปยังยูสเคสรอง โดยหมายเลขลำดับของการทำงานใหม่ให้เริ่มต้นด้วยหมายเลขลำดับของการทำงานของจุดเชื่อม แล้วตามด้วยเครื่องหมายติดลบ (“-”) และหมายเลขลำดับของการทำงานเดิมของลำดับเหตุการณ์สำเร็จของยูสเคสรอง ตัวอย่างในตารางที่ 3.1 และ 3.3 หมายเลขลำดับการทำงานของจุดเชื่อมโยง “{UC2}” คือ 1 เมื่อเพิ่มการทำงานของลำดับเหตุการณ์สำเร็จของยูสเคสหมายเลข 2 ซึ่งมีการทำงาน 3 ขั้นตอน ดังนั้นหมายเลขลำดับการทำงานใหม่คือ 1-1 1-2 และ 1-3
- 6) เพิ่มลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสรองให้ยูสเคสหลัก
 

เพิ่มลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสรองให้ยูสเคสหลักเพื่อเป็นลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสหลัก โดยหมายเลขของประโยคเงื่อนไขใหม่ของลำดับเหตุการณ์ทางเลือกอื่นให้เริ่มต้นด้วยหมายเลขลำดับการทำงานของจุดเชื่อมโยงไปยังยูสเคสรอง แล้วตามด้วย

เครื่องหมาย “-” และหมายเลขลำดับของการทำงานเดิมของลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสรอง และหมายเลขลำดับการทำงานใหม่ของลำดับเหตุการณ์ทางเลือกอื่นมีรูปแบบเช่นเดียวกับหมายเลขของประโยคเงื่อนไขใหม่ จากตัวอย่างตารางที่ 3.1 และ 3.3 ซึ่งหมายเลขลำดับการทำงานของจุดเชื่อมโยง “{UC2}” คือ 1 และยูสเคสหมายเลข 2 มีลำดับเหตุการณ์ทางเลือกอื่น 2 ลำดับเหตุการณ์ซึ่งมีหมายเลขของประโยคเงื่อนไขเป็น 2.1 และ 2.2 ดังนั้นหมายเลขของประโยคเงื่อนไขใหม่ของลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสหลักจะเป็น 1-2.1 และ 1-2.2 ส่วนหมายเลขการทำงานใหม่ของลำดับเหตุการณ์ทางเลือกอื่นจะเป็น 1-2.1.1 เป็นต้นไป และ 1-2.2.1 เป็นต้นไป

7) เพิ่ม post-condition ของลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสรองให้กับยูสเคสหลัก

นำ post-condition ของลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสรองเพิ่มให้กับยูสเคสหลัก โดยระบุหมายเลขของผลลัพธ์เป็นหมายเลขของประโยคเงื่อนไขใหม่ของลำดับเหตุการณ์ทางเลือกอื่น

### 3.3.2 การรวมยูสเคสที่มีความสัมพันธ์แบบเอ็กซ์เทน

ความสัมพันธ์ระหว่างยูสเคสแบบเอ็กซ์เทนเกิดขึ้นเมื่อยูสเคสหนึ่งไม่สามารถทำงานได้ตามปกติ จึงเรียกใช้งานอีกยูสเคสหนึ่งเพื่อทำงานแทน จากหัวข้อที่ 3.2.4.1 ยูสเคสที่เรียกใช้ยูสเคสอื่นในลักษณะนี้จะมีจุดเชื่อมโยงระบุในการทำงานของลำดับเหตุการณ์ทางเลือกเท่านั้น ดังตัวอย่างรายละเอียดยูสเคสหมายเลข 1: Add contact การทำงานขั้นตอนที่ 1 ของลำดับเหตุการณ์ทางเลือกอื่นที่มีหมายเลขของประโยคเงื่อนไข 2.3 เรียกใช้งานยูสเคสหมายเลข 3 การรวมยูสเคสที่มีความสัมพันธ์แบบเอ็กซ์เทนมีขั้นตอนดังนี้

#### 1) กำหนดยูสเคสหลักและยูสเคสรอง

เริ่มจากกำหนดให้ยูสเคสที่เรียกใช้งาน (ยูสเคสที่เอ็กซ์เทน) เป็นยูสเคสหลัก และให้ยูสเคสที่ถูกเรียกใช้งาน (ยูสเคสที่ถูกเอ็กซ์เทน) เป็นยูสเคสรอง ดังนั้นจากตารางที่ 3.1 ยูสเคสหมายเลข 1 เป็นยูสเคสหลัก และยูสเคสหมายเลข 3 เป็นยูสเคสรอง

#### 2) เพิ่ม pre-condition ของยูสเคสรองให้ยูสเคสหลัก

นำ pre-condition ของยูสเคสรองต่อท้าย pre-condition ของยูสเคสหลัก

#### 3) เพิ่มข้อมูลนำเข้า required-item ของยูสเคสรองให้กับยูสเคสหลัก

นำข้อมูลนำเข้า required-item ของยูสเคสรองเพิ่มให้กับยูสเคสหลักเพื่อเป็นข้อมูลนำเข้าของยูสเคสหลักด้วย จากตารางที่ 3.4 ตัวอย่างรายละเอียดยูสเคสหมายเลข 3:

Invalid telephone no มีข้อมูลนำเข้าเพียงหนึ่งตัวคือ telephone เพิ่มให้เป็นข้อมูลนำเข้าของยูสเคสหลัก

4) เพิ่มประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่น

นำประโยคเงื่อนไขของลำดับเหตุการณ์สำเร็จของยูสเคสรองกำหนดให้กับประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่นที่จุดเชื่อมโยงยูสเคส และใช้หมายเลขของประโยคเงื่อนไขของเหตุการณ์ทางเลือกอื่นเป็นหมายเลขของประโยคเงื่อนไขใหม่ จากตารางที่ 3.1 และ 3.4 การทำงานของประโยคเงื่อนไขหมายเลข 2.3 มีจุดเชื่อมโยงไปยังยูสเคสหมายเลข 3 จึงนำประโยคเงื่อนไขของลำดับเหตุการณ์สำเร็จของยูสเคสหมายเลข 3 ไปแทนที่ และมีหมายเลขของประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่นเป็น 2.3

5) นำการทำงานของลำดับเหตุการณ์สำเร็จของยูสเคสรองเป็นลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสหลัก

โดยนำการทำงานของลำดับเหตุการณ์สำเร็จของยูสเคสรองแทนที่การทำงานของลำดับเหตุการณ์ทางเลือกอื่นที่มีจุดเชื่อมโยงยูสเคส โดยหมายเลขของการทำงานใหม่ขึ้นต้นด้วยหมายเลขของประโยคเงื่อนไข แล้วตามด้วยเครื่องหมายจุลภาค และหมายเลขตั้งแต่ 1 เป็นต้นไป

6) เพิ่ม post-condition ของลำดับเหตุการณ์สำเร็จของยูสเคสรองให้กับยูสเคสหลัก

นำ post-condition ของลำดับเหตุการณ์สำเร็จของยูสเคสรองเพิ่มให้กับยูสเคสหลัก โดยนำมาใช้เป็นผลลัพธ์ของลำดับเหตุการณ์ทางเลือกอื่น ดังนั้นหมายเลขของผลลัพธ์เป็นหมายเลขของประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่น

ตัวอย่างรายละเอียดยูสเคสที่มีความสัมพันธ์กันทั้งแบบอินคลูดี และแบบเอ็กซ์เทนชันแสดงในตารางที่ 3.6 ซึ่งเป็นการรวมยูสเคสหมายเลข 1 (ตารางที่ 3.2) ยูสเคสหมายเลข 2 (ตารางที่ 3.4) และยูสเคสหมายเลข 3 (ตารางที่ 3.5)

ตารางที่ 3.4 ตัวอย่างรายละเอียดยูสเคสหมายเลข 2: Authenticate client

Use case no.:	2
Use case name:	Authenticate client
Description:	Authenticate client before using the system.
Actor:	Client

ตารางที่ 3.4 ตัวอย่างรายละเอียดคุณสมบัติหมายเลข 2: Authenticate client (ต่อ)

<b>Pre-condition:</b>		Login name and password are created in system. Client enters login name and password.				
<b>Required-item:</b>		Name	Type	Size	Max	Min
		login	String	8	-	-
		passwd	String	5	-	-
<b>Is abstract</b>		0				
<b>Success scenario:</b>						
Condition no:	0	(login <> "") && (passwd <> "")				
Order		Action				
1		System submits login name and password form client.				
2		System checks login name and password.				
3		Client logs in to the system.				
<b>Alternative scenario:</b>						
Condition no:	2.1	(login == "")				
Step		Action				
2.1.1		System shows an error message "Please enter login".				
Condition no:	2.2	(passwd == "")				
Step		Action				
2.2.1		System shows an error message "Please enter password".				
<b>Post-condition:</b>	0	Client logs into the system.				
	2.1	System shows an error message "Please enter login".				
	2.2	System shows an error message "Please enter password".				

ตารางที่ 3.5 ตัวอย่างรายละเอียดคุณสมบัติหมายเลข 3: Invalid phone no

<b>Use case no.:</b>	3
<b>Use case name:</b>	Invalid phone no
<b>Description:</b>	Telephone number is invalid
<b>Actor:</b>	Client
<b>Pre-condition:</b>	Client enters telephone number.

ตารางที่ 3.5 ตัวอย่างรายละเอียดยูสเคสหมายเลข 3: Invalid phone no (ต่อ)

Required-item:		Name	Type	Size	Max	Min
		telephone	String	7	-	-
Is abstract		0				
Success scenario:						
Condition no:	0	(telephone < "0000000")    (telephone > "9999999")				
Step		Action				
1		System shows error message "telephone number length must be 7"				
Post-condition:	0	System shows error message "telephone number length must be 7"				

ตารางที่ 3.6 ตัวอย่างรายละเอียดยูสเคสที่รวมความสัมพันธ์ยูสเคสหมายเลข 1: Add contact

Use case no.:	1					
Use case name:	Add contact					
Description:	Adding new contact to list					
Actor:	Client					
Pre-condition:	Client enters the contact information: name, surname, telephone number, address and zip code. Login name and password are created in system. Client enters login name and password.					
Required-item:		Name	Type	Size	Min	Max
		login	String	8	-	-
		passwd	String	5	-	-
		name	String	15	-	-
		surname	String	20	-	-
		telephone	String	7	-	-
		address	String	20	-	-
		zipcode	Integer	5	10000	99999
Is abstract		0				



ตารางที่ 3.6 ตัวอย่างรายละเอียดยูสเคสที่รวมความสัมพันธ์ยูสเคสหมายเลข 1: Add contact (ต่อ)

Success scenario:		
Condition no:	0	(name.length>0) && (surname.length>0) && (telephone>="0000000") && (telephone<="9999999") && (login <> "") && (passwd <> "")
Step		Action
1-1		System submits login name and password form client.
1-2		System checks login name and password.
1-3		Client logs in to the system.
2		System submits the contact information form client.
3		System saves the contact information into database.
4		System shows a message "Save new contact complete".
Alternative scenario:		
Condition no:	1-2.1	(login == "")
Step		Action
1-2.1.1		System shows an error message "Please enter login".
Condition no:	1-2.2	(passwd == "")
Step		Action
1-2.2.1		System shows an error message "Please enter password".
Condition no:	2.1	(name.length<=0)
Step		Action
2.1.1		System shows an error message "Please enter name".
Condition no:	2.2	(surname.length<=0)
Step		Action
2.2.1		System shows an error message "Please enter surname".
Condition no:	2.3	(telephone < "0000000")    (telephone > "9999999")
Step		Action
2.3.1		System shows an error message "telephone number length must be 7".

ตารางที่ 3.6 ตัวอย่างรายละเอียดยูสเคสที่รวมความสัมพันธ์ยูสเคสหมายเลข 1: Add contact (ต่อ)

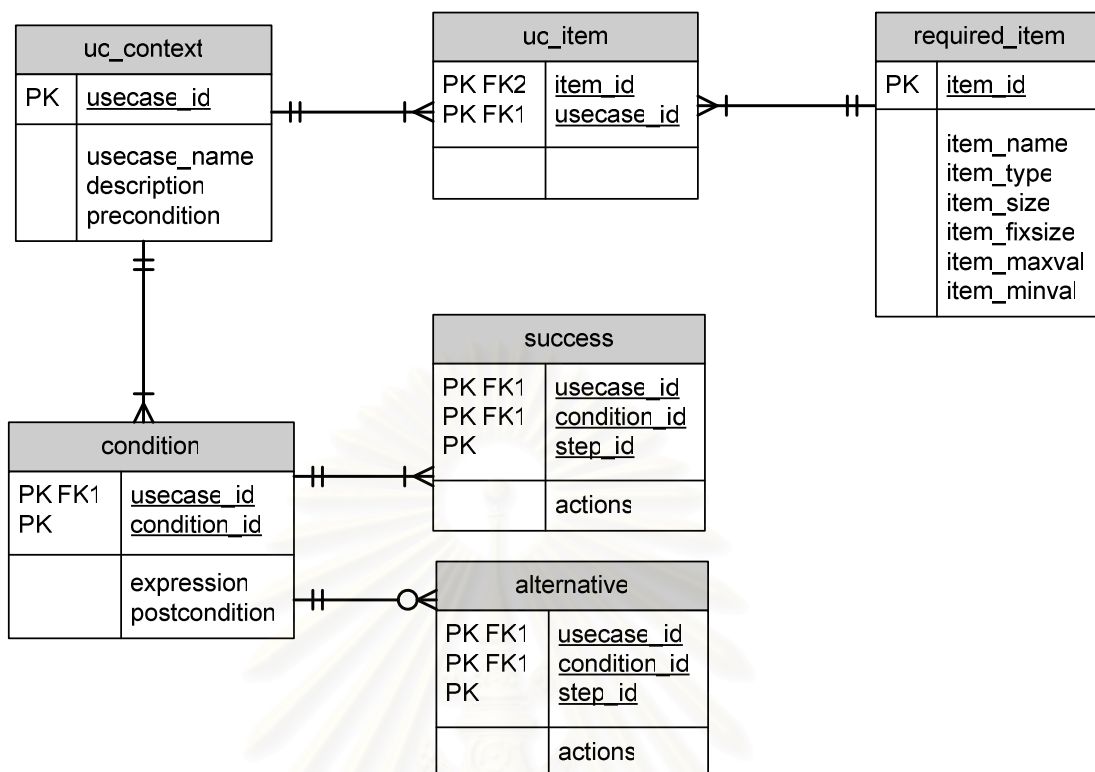
Post-condition:	0	Client logs into the system. System saves new contact and shows a message “Save new contact complete”.
	1-2.1	System shows an error message “Please enter login”.
	1-2.2	System shows an error message “Please enter password”.
	2.1	System shows an error message “Please enter name”.
	2.2	System shows an error message “Please enter sir name”.
	2.3	System shows an error message “telephone number length must be 7”.

### 3.4 การสร้างกรณีทดสอบโดยอัตโนมัติ

หลังจากกำหนดรูปแบบรายละเอียดยูสเคสเพื่อใช้ในการสร้างกรณีทดสอบ และวิธีการรวมยูสเคสที่มีความสัมพันธ์กัน จากนั้นเข้าสู่ขั้นตอนการสร้างกรณีทดสอบจากยูสเคสที่เตรียมไว้ ซึ่งแบ่งขั้นตอนหลักออกเป็น 2 ส่วนได้แก่ การแยกรายละเอียดยูสเคสลงฐานข้อมูล และการสร้างกรณีทดสอบ

#### 3.4.1 การแยกรายละเอียดยูสเคสลงฐานข้อมูล

ขั้นตอนนี้เป็นการแยก และเตรียมข้อมูลรายละเอียดยูสเคสที่จำเป็นสำหรับการสร้างกรณีทดสอบ โดยข้อมูลที่ได้จะถูกจัดเก็บในฐานข้อมูล โครงสร้างของฐานข้อมูลที่ใช้จัดเก็บแสดงได้ดังรูปที่ 3.6



รูปที่ 3.6 แผนภาพความสัมพันธ์ของเอนทิตี (ER-Diagram) ของรายละเอียดยูสเคส

จากรูปที่ 3.6 ซึ่งเป็นแผนภาพแสดงความสัมพันธ์ของข้อมูลรายละเอียดยูสเคสซึ่งแสดงโครงสร้างการจัดเก็บข้อมูลในฐานข้อมูล โดยข้อมูลรายละเอียดยูสเคสต้องแยกออกและจัดเก็บตามโครงสร้างฐานข้อมูลมีรายละเอียดดังนี้

1) พิจารณาที่ละยูสเคสในแผนภาพยูสเคส โดยพิจารณาเฉพาะยูสเคสที่ไม่ได้เป็นยูสเคสต้นแบบสำหรับยูสเคสอื่น กล่าวคือพิจารณายูสเคสที่คุณสมบัติ Is abstract ไม่เท่ากับ 0

2) แยกข้อมูลทั่วไปของยูสเคส และใส่ลงในตาราง uc\_context โดยข้อมูลทั่วไปประกอบด้วย

- usecase\_id คือหมายเลขยูสเคส
- usecase\_name คือชื่อของยูสเคส
- description คือข้อความอธิบายยูสเคส
- precondition คือกิจกรรมที่ต้องปฏิบัติก่อนเริ่มทำงานยูสเคส

3) แยกข้อมูลนำเข้าของยูสเคส แล้วใส่ลงในตาราง required\_item ซึ่งข้อมูลนำเข้าของยูสเคสต้องประกอบด้วย

- `item_id` คือรหัสของข้อมูลนำเข้า ซึ่งต้องตัวเลขที่ไม่ซ้ำกันเพื่อใช้เป็นรหัสประจำข้อมูลนำเข้า

- `item_name` คือชื่อของข้อมูลนำเข้า
- `item_type` คือชนิดของข้อมูลนำเข้า
- `item_size` คือความยาวของข้อมูลนำเข้าชนิดชุดอักขระ หรือจำนวนหลักหลังจุดทศนิยมของข้อมูลนำเข้าชนิดเลขจำนวนจริง หรือจำนวนหลักของข้อมูลนำเข้าชนิดเลขจำนวนเต็ม

- `item_fix` คือคุณสมบัติระบุให้ข้อมูลทดสอบที่สร้างต้องมีขนาดเท่ากับขนาดที่ระบุใน `item_size` มิฉะนั้นข้อมูลทดสอบที่สร้างจะมีขนาดน้อยกว่าหรือเท่ากับขนาดที่ระบุใน `item_size`

- `item_maxval` คือค่าสูงสุดที่เป็นไปได้ของข้อมูลนำเข้า

- `item_minval` คือค่าต่ำสุดที่เป็นไปได้ของข้อมูลนำเข้า

4) ในแต่ละยูสเคสอาจมีข้อมูลนำเข้ามากกว่า 1 ตัว และในทางกลับกันข้อมูลนำเข้าแต่ละตัวอาจเป็นข้อมูลนำเข้าหลายยูสเคส ซึ่งในตาราง `uc_item` เก็บความสัมพันธ์เหล่านี้ด้วยข้อมูลหมายเลขของยูสเคส และรหัสของข้อมูลนำเข้า

5) แยกข้อมูลเกี่ยวกับประโยคเงื่อนไขต่างๆ ของลำดับเหตุการณ์ของยูสเคส และผลลัพธ์หลังจากการทำงานยูสเคสใส่ในตาราง `condition` ซึ่งประกอบด้วย

- `usecase_id` คือหมายเลขของยูสเคส

- `condition_id` คือหมายเลขของประโยคเงื่อนไขการเกิดลำดับเหตุการณ์ของยูสเคส

- `expression` คือประโยคเงื่อนไขการเกิดลำดับเหตุการณ์ของยูสเคส

- `postcondition` คือผลลัพธ์หลังจากการทำงานตามลำดับเหตุการณ์ของยูสเคส

6) แยกข้อมูลลำดับเหตุการณ์สำเร็จของยูสเคสใส่ในตาราง `success` ซึ่งแต่ละยูสเคสมีลำดับเหตุการณ์สำเร็จเพียงหนึ่งลำดับเหตุการณ์เท่านั้น ข้อมูลลำดับเหตุการณ์สำเร็จให้แยกแต่ละขั้นตอนการทำงานของลำดับเหตุการณ์ซึ่งประกอบด้วย

- `usecase_id` คือหมายเลขของยูสเคส

- `condition_id` คือหมายเลขของประโยคเงื่อนไขการเกิดลำดับเหตุการณ์ของยูสเคส

- step\_id คือหมายเลขลำดับการทำงานของลำดับเหตุการณ์สำเร็จ
- actions คือรายละเอียดการทำงานของลำดับเหตุการณ์สำเร็จ

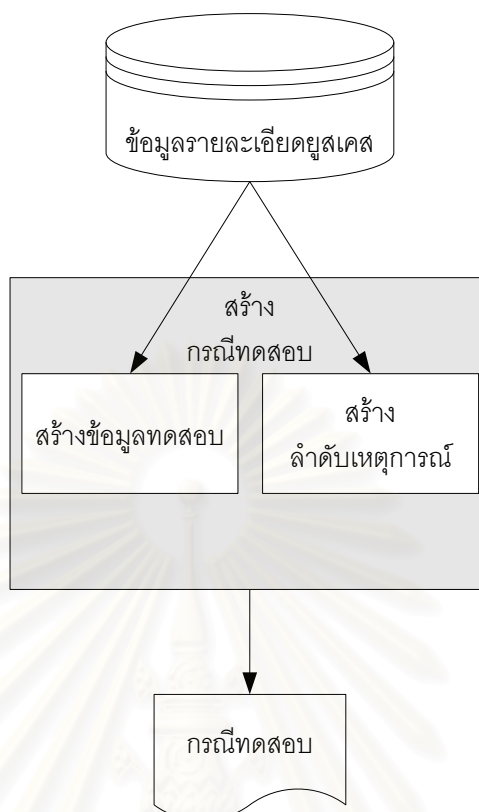
7) ในกรณีที่ยุสเคสมีลำดับเหตุการณ์ทางเลือกอื่น ให้แยกข้อมูลลำดับเหตุการณ์ทางเลือกอื่นของยูสเคสใส่ในตาราง alternative แต่ละยูสเคสอาจลำดับเหตุการณ์ทางเลือกได้มากกว่าหนึ่งลำดับเหตุการณ์ ข้อมูลลำดับเหตุการณ์ทางเลือกอื่นให้แยกแต่ละขั้นตอนการทำงานของลำดับเหตุการณ์เช่นเดียวกับการแยกข้อมูลลำดับเหตุการณ์สำเร็จ

### 3.4.2 การสร้างกรณีทดสอบ

ขั้นตอนนี้เป็นขั้นตอนที่นำข้อมูลรายละเอียดยูสเคสที่แยกลงฐานข้อมูลมาสร้างกรณีทดสอบ ซึ่งการสร้างกรณีทดสอบนั้นพิจารณาจากประโยคเงื่อนไขของแต่ละยูสเคส ซึ่งประโยคเงื่อนไขหนึ่งประโยคทำให้เกิดลำดับเหตุการณ์ได้หนึ่งลำดับเหตุการณ์ และสร้างกรณีทดสอบได้หนึ่งกรณีทดสอบเพื่อใช้ทดสอบลำดับเหตุการณ์ โดยจำนวนกรณีทดสอบที่สร้างได้มีจำนวนเท่ากับผลรวมของประโยคเงื่อนไขของแต่ละยูสเคส

กรณีทดสอบในงานวิจัยนี้ ประกอบด้วย Test case id, Test case name, Description, Pre-condition, Input, Expected output และ Post-condition ส่วนประกอบของกรณีทดสอบมีรายละเอียดดังนี้

- Test case id เป็นหมายเลขของกรณีทดสอบ
  - Test case name เป็นข้อความแสดงชื่อกรณีทดสอบ
  - Description เป็นข้อความแสดงรายละเอียดของกรณีทดสอบ
  - Pre-condition เป็นข้อความแสดงสิ่งที่ต้องปฏิบัติก่อนเริ่มต้นทดสอบ
  - Input เป็นข้อมูลทดสอบซึ่งประกอบด้วย ชื่อของข้อมูลทดสอบ และค่าของข้อมูลทดสอบ
  - Expected output เป็นผลลัพธ์คาดหวังของกรณีทดสอบ โดยระบุเป็นลำดับเหตุการณ์ที่คาดว่าจะเกิดขึ้นเมื่อทดสอบด้วยข้อมูลทดสอบ
  - Post-condition เป็นข้อความแสดงผลหลังจากการทำงานตามเงื่อนไข
- การสร้างกรณีทดสอบแบ่งออกเป็น 2 ส่วนหลักคือ การสุ่มสร้างข้อมูลทดสอบ และการสร้างลำดับเหตุการณ์ที่คาดว่าจะเกิดขึ้น ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 ภาพรวมขั้นตอนการสร้างกรณีทดสอบ

ขั้นตอนการสร้างกรณีทดสอบมีขั้นตอนดังนี้

- 1) พิจารณาสร้างกรณีทดสอบที่ละประโยคเงื่อนไขของแต่ละยูสเคส  
โดยเริ่มต้นพิจารณายูสเคสทีละยูสเคส และพิจารณาทีละประโยคเงื่อนไขของยูสเคส ซึ่งประโยคเงื่อนไขของยูสเคสหนึ่งประโยคสามารถสร้างกรณีทดสอบได้หนึ่งกรณีทดสอบ
- 2) กำหนดหมายเลขของกรณีทดสอบ  
หมายเลขของกรณีทดสอบนำมาจากหมายเลขของยูสเคสที่นำมาสร้างกรณีทดสอบ ตามด้วยเครื่องหมายจุลภาค และตัวเลขที่ไม่ซ้ำกันตั้งแต่ 1 เป็นต้นไป จากตัวอย่างในตารางที่ 3.7 เป็นกรณีทดสอบที่สร้างขึ้นจากยูสเคสหมายเลข 1 ดังนั้นหมายเลขของกรณีทดสอบจึงเป็นหมายเลข 1.1
- 3) ระบุชื่อของกรณีทดสอบ  
ชื่อของกรณีทดสอบนำมาจากชื่อของยูสเคสที่นำมาสร้างกรณีทดสอบ ซึ่งก็คือข้อมูล usecase\_name ในตาราง uc\_context
- 4) ระบุข้อความอธิบายกรณีทดสอบ

ข้อความอธิบายกรณีทดสอบให้ระบุว่า Success Scenario สำหรับกรณีที่เป็นกรณีทดสอบลำดับเหตุการณ์สำเร็จ หรือระบุว่า Alternative Scenario สำหรับกรณีที่เป็นกรณีทดสอบลำดับเหตุการณ์ทางเลือกอื่น และตามด้วยประโยคเงื่อนไขสำหรับลำดับเหตุการณ์ (ข้อมูล expression ในตาราง condition) จากตัวอย่างกรณีทดสอบในตารางที่ 3.6 เป็นกรณีทดสอบลำดับเหตุการณ์สำเร็จ

#### 5) ระบุ Pre-condition ของกรณีทดสอบ

Pre-condition ของกรณีทดสอบนำมาจาก Pre-condition ของยูสเคส ซึ่งก็คือข้อมูล precondition ในตาราง uc\_context

#### 6) สร้างข้อมูลทดสอบของกรณีทดสอบ

สร้างข้อมูลทดสอบสำหรับใช้เป็นข้อมูลนำเข้าของกรณีทดสอบ โดยต้องสร้างข้อมูลทดสอบตามจำนวนข้อมูลนำเข้าของยูสเคสที่นำมาสร้างกรณีทดสอบ การสร้างค่าของข้อมูลทดสอบต้องพิจารณาจากรายละเอียดของข้อมูลนำเข้า (ในตาราง required\_item) และประโยคเงื่อนไข (ข้อมูล expression ในตาราง condition) ซึ่งรูปแบบของพจน์ในประโยคเงื่อนไขมี 2 แบบคือแบบการเปรียบเทียบ และแบบการคำนวณทางคณิตศาสตร์ รายละเอียดของการสร้างค่าของข้อมูลทดสอบมีดังนี้

- สำหรับข้อมูลนำเข้าชนิดชุดอักขระ      สุ่มข้อความที่มีความยาวของตัวอักษรไม่เกินความยาวของตัวอักษรที่ระบุไว้และต้องสอดคล้องกับค่าต่ำสุดที่เป็นไปได้ ค่าสูงสุดที่เป็นไปได้ รวมทั้งประโยคเงื่อนไข
- สำหรับข้อมูลนำเข้าชนิดตรรกะ      สุ่มค่าจริงหรือเท็จซึ่งต้องสอดคล้องกับประโยคเงื่อนไข
- สำหรับข้อมูลนำเข้าชนิดตัวเลขจำนวนเต็มและกำหนดจำนวนหลักของข้อมูลนำเข้า      สุ่มเลขที่มีจำนวนหลักไม่เกินจำนวนหลักที่กำหนดไว้
- สำหรับข้อมูลนำเข้าชนิดตัวเลขจำนวนเต็มหรือจำนวนจริงและรูปแบบของพจน์ในประโยคเงื่อนไขเป็นแบบการเปรียบเทียบ      สุ่มเลขที่มีค่าสอดคล้องกับการเปรียบเทียบนั้น
- สำหรับข้อมูลนำเข้าชนิดตัวเลขจำนวนเต็มหรือจำนวนจริงและรูปแบบของพจน์ในประโยคเงื่อนไขเป็นการคำนวณทางคณิตศาสตร์      สุ่มเลขที่มีค่าสอดคล้องกับการคำนวณทางคณิตศาสตร์

- สำหรับข้อมูลนำเข้าอื่นที่เกี่ยวข้องกับยูสเคส แต่ไม่ได้เป็นส่วนหนึ่งของประโยคเงื่อนไข สุ่มค่าของข้อมูลทดสอบตามชนิดของข้อมูลนำเข้า ขนาดของข้อมูลนำเข้า และมีค่าสอดคล้องกับค่าต่ำสุดที่เป็นไปได้และค่าสูงสุดที่เป็นไปได้ของข้อมูลนำเข้า

7) สร้างลำดับเหตุการณ์ที่คาดว่าจะเกิดขึ้นเพื่อใช้เป็นผลลัพธ์คาดหวังของกรณีทดสอบ

ผลลัพธ์คาดหวังของกรณีทดสอบนำมาจากลำดับเหตุการณ์ของยูสเคสเมื่อประโยคเงื่อนไขเป็นจริง โดยระบุลำดับเหตุการณ์ที่คาดว่าจะเกิดขึ้นเป็นข้อเริ่มต้นตั้งแต่ข้อ 1 เป็นต้นไป ผลลัพธ์คาดหวังของกรณีทดสอบแบ่งได้เป็น 2 แบบคือ

- ผลลัพธ์คาดหวังของกรณีทดสอบสำหรับทดสอบลำดับเหตุการณ์สำเร็จ นำลำดับการทำงานมาจากข้อมูล actions ในตาราง success

- ผลลัพธ์คาดหวังของกรณีทดสอบสำหรับทดสอบลำดับเหตุการณ์ทางเลือกอื่นให้พิจารณาจากหมายเลขของประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่นที่เป็นจริงเมื่อทดสอบด้วยข้อมูลทดสอบที่สร้างขึ้น โดยหมายเลขหน้าเครื่องหมายมหัพภาคจะแสดงหมายเลขลำดับการทำงานในลำดับเหตุการณ์สำเร็จ ให้นำการทำงานของลำดับเหตุการณ์สำเร็จ (ข้อมูล actions ในตาราง success) ตั้งแต่การทำงานลำดับที่ 1 จนถึงการทำงานลำดับหมายเลขหน้าเครื่องหมายมหัพภาค และตามด้วยการทำงานในลำดับเหตุการณ์ทางเลือกอื่น (ข้อมูล actions ในตาราง alternative) ตัวอย่างในตารางที่ 3.6 หมายเลขประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่น 2.1 ผลลัพธ์คาดหวังของกรณีทดสอบสำหรับทดสอบลำดับเหตุการณ์ทางเลือกอื่นนี้ จะนำมาจากการทำงานลำดับที่ 1 และ 2 ในลำดับเหตุการณ์สำเร็จ และตามด้วยการทำงานหมายเลข 2.1.1 ในลำดับเหตุการณ์ทางเลือกอื่น สำหรับกรณีที่การทำงานในลำดับเหตุการณ์ทางเลือกอื่นมีจุดเชื่อมโยงกลับไปสู่ลำดับเหตุการณ์สำเร็จ ต้องตามด้วยลำดับการทำงานที่เกิดขึ้น ณ จุดดังกล่าวจนถึงลำดับการทำงานสุดท้ายของลำดับเหตุการณ์สำเร็จ

8) ระบุ Post-condition

Post-condition ของกรณีทดสอบนำมาจาก Post-condition ของยูสเคสตามลำดับเหตุการณ์ที่เกิดขึ้น ซึ่งก็คือข้อมูล postcondition ในตาราง condition



ตารางที่ 3.7 ตัวอย่างกรณีทดสอบที่สร้างขึ้นจากยูสเคส: Add contact

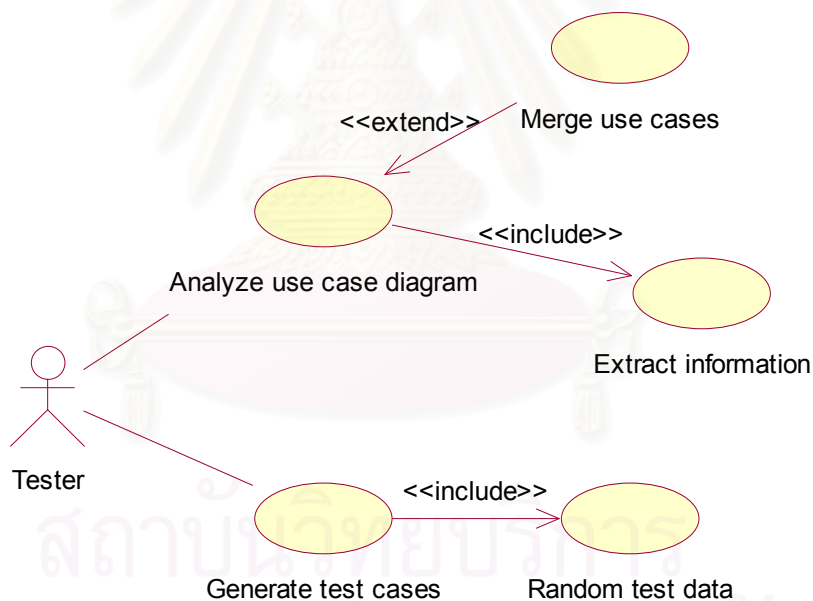
<i>Test case id: 1.1</i>	
<i>Add contact</i>	
Description	Success Scenario:: (name.length>0) && (surname.length>0) && (telephone>="0000000") && (telephone<="9999999") && (login<>"") && (passwd<>"")
Pre-condition	Client enters the contact information: name, surname, telephone number, address and zip code. Login name and password are created in system. Client enters login name and password.
Input	
Name	Values
address	4/30 Moo5
login	yx9
name	mw4e8nopzpwf
passwd	g5ld1
surname	nompwgujdpof9pl92wm
telephone	574256981
zipcode	24813
Expected Output	1. System submits id and password from client.
	2. System checks id and password.
	3. Client logs into the system.
	4. System submits the contact information from client.
	5. System saves the contact information into database.
	6. System shows a message "Save new contact complete".
Post-condition	System saves new contact and shows a message "Save new contact complete". Client logs into the system.

### 3.5 การวิเคราะห์และออกแบบเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติ

การวิเคราะห์และออกแบบเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจะใช้แผนภาพยูเอ็มแอล (UML Diagram) ประกอบด้วย แผนภาพยูสเคส แผนภาพคลาส แผนภาพซีควเอนซ์ และแผนภาพลำดับกิจกรรม ซึ่งมีรายละเอียดดังนี้

#### 3.5.1 แผนภาพยูสเคส

แผนภาพยูสเคสเป็นแผนภาพแสดงเรื่องราวทั้งหมดของขอบเขตของปัญหาว่าต้องประกอบด้วยกิจกรรมใด และแต่ละกิจกรรมมีความสัมพันธ์กันอย่างไรในมุมมองของผู้ใช้ ซึ่งแผนภาพยูสเคสของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสแสดงดังรูปที่ 3.8



รูปที่ 3.8 แผนภาพยูสเคสของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

จากแผนภาพยูสเคสของเครื่องมือสร้างกรณีทดสอบในรูปที่ 3.8 ผู้ทดสอบต้องนำแผนภาพยูสเคสที่มีรายละเอียดแต่ละยูสเคสในรูปแบบเอ็กซ์เอ็มแอลไฟล์มาวิเคราะห์แผนภาพยูสเคส (Analyze use case diagram) ก่อน ซึ่งถ้าในแผนภาพยูสเคสมียูสเคสที่มีความสัมพันธ์กันทั้งแบบอินคลูด หรือแบบเอ็กซ์เทน ต้องทำการรวมยูสเคสที่สัมพันธ์กัน (Merge use cases)

จากนั้นจึงแยกรายละเอียดยูสเคส (Extract information) หลังจากนั้นผู้ทดสอบจึงสร้างกรณีทดสอบ (Generate test cases) ซึ่งสร้างค่าทดสอบสำหรับกรณีทดสอบโดยสุ่ม (Random test data) ด้วย รายละเอียดของแต่ละยูสเคสแสดงในตารางที่ 3.8 ถึง 3.12

ตารางที่ 3.8 รายละเอียดยูสเคสวิเคราะห์แผนภาพยูสเคส

Use case no.:		1	
Use case name:		Analyze use case diagram	
Description:		เพื่อวิเคราะห์แผนภาพยูสเคสสำหรับสร้างกรณีทดสอบ	
Actor:		Tester	
Pre-condition:		ผู้ทดสอบเตรียมแผนภาพยูสเคสที่มีรายละเอียดยูสเคส	
Required-item:		Name	Type
		use case diagram	XML file
		Size	-
Success scenario:			
Condition no:	0	แผนภาพยูสเคสอยู่ในรูปแบบไฟล์ XML	
Step		Action	
1		- ผู้ทดสอบเลือกไฟล์แผนภาพยูสเคส	
2		- เครื่องมือเริ่มอ่านไฟล์แผนภาพยูสเคสของผู้ทดสอบ	
3		- เครื่องมือวิเคราะห์หายูสเคสที่มีความสัมพันธ์กัน	
4		- {UC3}	
5		- เครื่องมือแสดงรายละเอียดยูสเคสจากการวิเคราะห์ทั้งหมดออกทางหน้าจอ	
Alternative scenario:			
Condition no:	3.1	-	
Step		Action	
3.1.1		{UC2}	
Post-condition	0	เครื่องมือแสดงรายละเอียดยูสเคสจากการวิเคราะห์ออกทางหน้าจอ	

ตารางที่ 3.9 รายละเอียดยู่สเคสการรวมยู่สเคส

Use case no.:	2		
Use case name:	Merge use cases		
Description:	เพื่อรวมยู่สเคสที่มีความสัมพันธ์กัน		
Actor:	Tester		
Pre-condition:	ผู้ทดสอบเตรียมแผนภาพยู่สเคสที่มีรายละเอียดยู่สเคส		
Required-item:	Name	Type	Size
	use case diagram	XML file	-
Success scenario:			
Condition no:	0	ยู่สเคสมีความสัมพันธ์กันแบบอินคลูด หรือเอ็กซ์เทน	
Step	Action		
1	- เครื่องมือรวม Pre-condition ของยู่สเคสเข้าด้วยกัน		
2	- เครื่องมือเพิ่มข้อมูลนำเข้าของยู่สเคสรองให้ยู่สเคสหลัก		
3	- เครื่องมือรวมลำดับเหตุการณ์ของยู่สเคสหลักและยู่สเคสรอง พร้อมทั้งแก้ไขหมายเลขลำดับการทำงาน		
4	- เครื่องมือเพิ่ม Post-condition ของยู่สเคสรองให้ยู่สเคสหลัก		
Post-condition	0	-	

ตารางที่ 3.10 รายละเอียดยู่สเคสการแยกรายละเอียดยู่สเคส

Use case no.:	3		
Use case name:	Extract information		
Description:	เพื่อแยกรายละเอียดยู่สเคส และบันทึกลงฐานข้อมูล		
Actor:	Tester		
Pre-condition:	ผู้ทดสอบเตรียมแผนภาพยู่สเคสที่มีรายละเอียดยู่สเคส		
Required-item:	Name	Type	Size
	use case diagram	XML file	-
Success scenario:			
Condition no:	0	แผนภาพยู่สเคสอยู่ในรูปแบบไฟล์ XML	

ตารางที่ 3.10 รายละเอียดยูนิตเคสการแยกรายละเอียดยูนิตเคส (ต่อ)

Step	Action
1	- เครื่องมือแยกรายละเอียดยูนิตเคสออกเป็นส่วนตามโครงสร้างฐานข้อมูล
2	- เครื่องมือบันทึกข้อมูลรายละเอียดยูนิตเคสลงฐานข้อมูล
Post-condition	0 เครื่องมือบันทึกข้อมูลรายละเอียดยูนิตเคสลงฐานข้อมูล

ตารางที่ 3.11 รายละเอียดยูนิตเคสการสร้างกรณีทดสอบ

Use case no.:	4		
Use case name:	Generate test cases		
Description:	เพื่อสร้างกรณีทดสอบ		
Actor:	Tester		
Pre-condition:	แผนภาพยูสเคสและรายละเอียดยูนิตเคสต้องถูกวิเคราะห์ และบันทึกลงฐานข้อมูล		
Required-item:	Name	Type	Size
	use case diagram	XML file	-
Success scenario:			
Condition no:	0	รายละเอียดยูนิตเคสต้องระบุข้อมูลนำเข้า และลำดับเหตุการณ์	
Step	Action		
1	- ผู้ทดสอบกำหนดสถานที่จัดเก็บไฟล์กรณีทดสอบ และเลือกสร้างกรณีทดสอบ		
2	- เครื่องมืออ่านข้อมูลที่ละยูนิตเคสจากฐานข้อมูล		
3	- เครื่องมือสร้างกรณีทดสอบตามประโยคเงื่อนไขของลำดับเหตุการณ์ของยูสเคส		
4	- {UC5}		
5	- เครื่องมือจัดเรียงลำดับการทำงานที่คาดว่าจะเกิดขึ้น		
6	- เครื่องมือส่งออกกรณีทดสอบที่สร้างขึ้นในรูปแบบเอกสารเอชทีเอ็มแอล		
7	- เครื่องมือแสดงรายละเอียดกรณีทดสอบทั้งหมดที่สร้างได้ทางหน้าจอ		

ตารางที่ 3.11 รายละเอียดยูสเคสการสร้างกรณีทดสอบ (ต่อ)

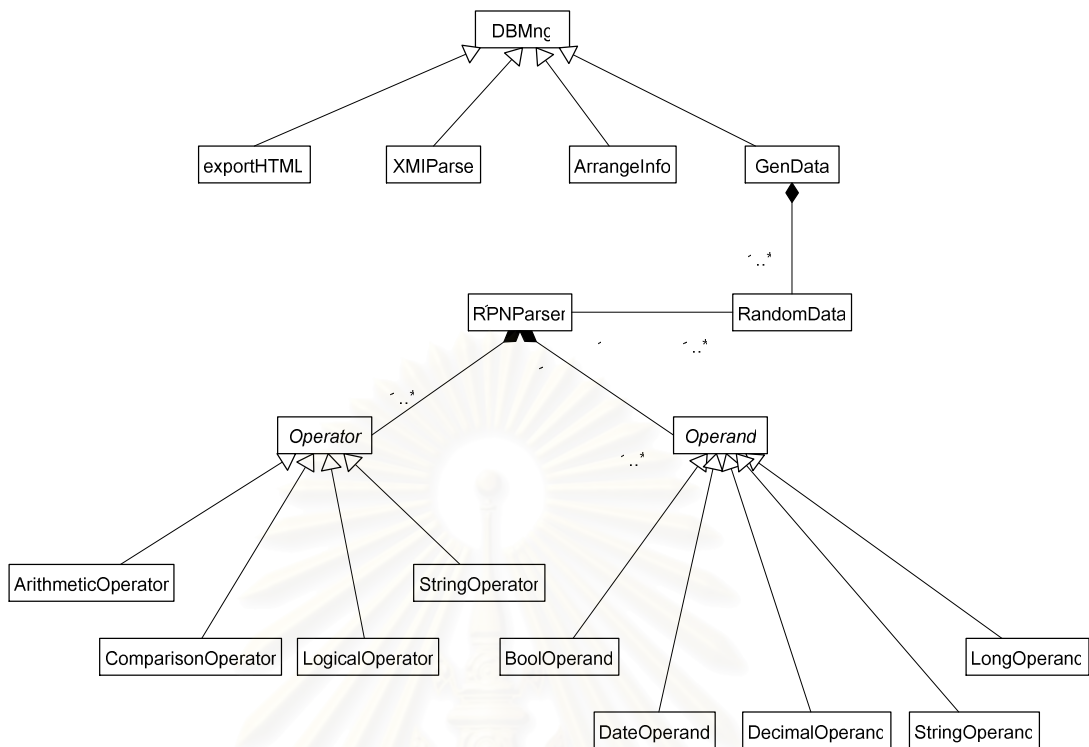
Post-condition	0	เครื่องมือส่งออกกรณีทดสอบทั้งหมดในรูปแบบเอกสารเอชทีเอ็มแอล และแสดงรายละเอียดกรณีทดสอบทางหน้าจอ
----------------	---	--

ตารางที่ 3.12 รายละเอียดยูสเคสการสุ่มข้อมูลทดสอบ

Use case no.:	5		
Use case name:	Random test data		
Description:	เพื่อสุ่มสร้างค่าข้อมูลทดสอบของกรณีทดสอบ		
Actor:	Tester		
Pre-condition:	แผนภาพยูสเคสและรายละเอียดยูสเคสต้องถูกวิเคราะห์ และบันทึกลงฐานข้อมูล		
Required-item:	Name	Type	Size
	use case diagram	XML file	-
Success scenario:			
Condition no:	0	ต้องระบุรายละเอียดข้อมูลนำเข้า และประโยคเงื่อนไขของลำดับเหตุการณ์	
Step		Action	
1		- เครื่องมือพิจารณาชนิด ขนาด ค่าสูงสุด และค่าต่ำสุดของข้อมูลนำเข้าของยูสเคส	
2		- เครื่องมือสุ่มค่าข้อมูลทดสอบที่สอดคล้องกับข้อมูลนำเข้า	
3		- เครื่องมือตรวจสอบค่าข้อมูลทดสอบกับประโยคเงื่อนไขของลำดับเหตุการณ์	
Post-condition	0	-	

### 3.5.2 แผนภาพคลาส

แผนภาพคลาสเป็นแผนภาพที่ใช้แสดงคลาส (Class) และความสัมพันธ์ในแง่ต่างๆ ระหว่างคลาส แผนภาพคลาสของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติแสดงดังรูปที่ 3.9



รูปที่ 3.9 แผนภาพคลาสของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

แผนภาพคลาสในรูปที่ 3.9 มีรายละเอียดของแต่ละคลาสดังนี้

1) คลาส DBMng เป็นคลาสต้นแบบที่ทำงานกับฐานข้อมูลซึ่งเก็บข้อมูลรายละเอียดยูสเคส หน้าที่คลาสนี้ประกอบด้วย การเปิดฐานข้อมูล การอ่านข้อมูลจากฐานข้อมูล การคิวรีฐานข้อมูล และการปิดฐานข้อมูล รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.10

DBMng
+dbconn OleDbConnection
-connstr String
-isConnect Boolean
-DBFolder String
+New()
+IsConnectDB()
+getDataReader()
+getDataTable()
+SqlQuery()
+DeleteAll()
+CloseDB()

รูปที่ 3.10 คลาส DBMng

2) คลาส XMIParse เป็นคลาสที่เก็บรายละเอียด และอ่านไฟล์เอ็กซ์เอ็มแอลของแผนภาพยูสเคส รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.11

<b>XMIParse</b>
-xmi_filename String
+New() +XMIFilename() +ReadXML() -FormatXML()

รูปที่ 3.11 คลาส XMIParse

3) คลาส ArrangeInfo เป็นคลาสที่แยกข้อมูลรายละเอียดยูสเคส รวมทั้งรวมยูสเคสที่มีความสัมพันธ์กัน แล้วบันทึกลงฐานข้อมูล ซึ่งรายละเอียดของคลาสนี้แสดงดังรูปที่ 3.12

<b>ArrangeInfo</b>
+New() +ArrangeAll() -ArrangeUC() -ArrangeItem() -ArrangeAssociation() -ArrangeCondition() -ArrangeSuccess() -ArrangeAlter() +MergeUsecase()

รูปที่ 3.12 คลาส ArrangeInfo

4) คลาส GenData เป็นคลาสหลักที่ใช้สร้างข้อมูลทดสอบทั้งหมดของกรณีทดสอบที่สร้างขึ้น ซึ่งรายละเอียดของคลาสนี้แสดงดังรูปที่ 3.13

<b>GenData</b>
-MaxRnc Long
+New() +Max_Rnd() +ShowExpression() +GenerateValues() +BuildTestData()

รูปที่ 3.13 คลาส GenData

5) คลาส RandomData เป็นคลาสที่มีหน้าที่สุ่มสร้างค่าของข้อมูลทดสอบ โดยค่าของข้อมูลทดสอบที่สร้างขึ้นต้องสอดคล้องกับชนิด ขนาด ค่าสูงสุด และค่าต่ำสุดของข้อมูลนำเข้า ซึ่งรายละเอียดของคลาสนี้แสดงดังรูปที่ 3.14



RandomData
-Seed Integer
-iMoc Integer
+New()
+RandomStringGenerator()
+RandomIntegerGenerator()
+RandomFloatGenerator()
+RandomBooleanGenerator()
+RandomDateGenerator()

รูปที่ 3.14 คลาส RandomData

6) คลาส RPNParser เป็นคลาสที่เก็บรายละเอียดของประโยคเงื่อนไขของลำดับเหตุการณ์ และทำหน้าที่แปลงประโยคเงื่อนไขของลำดับเหตุการณ์ให้อยู่ในรูปแบบ Postfix รวมทั้งตรวจสอบค่าของข้อมูลทดสอบว่าสอดคล้องกับประโยคเงื่อนไขหรือไม่ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.15

RPNParser
+New()
+EvaluateExpression()
+GetPostFixNotation()
+Convert2String()
+EvaluateRPN()

รูปที่ 3.15 คลาส RPNParser

7) คลาส Operator เป็นคลาสต้นแบบที่เก็บรายละเอียดของผลการดำเนินการของประโยคเงื่อนไขในรูปแบบต่างๆ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.16

Operator
#m_szOperator String
+New()
+New()
+ToString()
+Value()

รูปที่ 3.16 คลาส Operator

8) คลาส ArithmeticOperator เป็นคลาสที่สืบทอดมาจากคลาส Operator ซึ่งเก็บรายละเอียดผลการดำเนินการ และทำหน้าที่หาผลลัพธ์ของการดำเนินการทางคณิตศาสตร์ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.17

ArithmeticOperator
+Eval() +New() +New()

รูปที่ 3.17 คลาส ArithmeticOperator

9) คลาส ComparisonOperator เป็นคลาสที่สืบทอดมาจากคลาส Operator ซึ่งเก็บรายละเอียดผลการดำเนินการ และทำหน้าที่หาผลลัพธ์ของการดำเนินการทางการเปรียบเทียบ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.18

ComparisonOperator
+Eval() +New() +New()

รูปที่ 3.18 คลาส ComparisonOperator

10) คลาส LogicalOperator เป็นคลาสที่สืบทอดมาจากคลาส Operator ซึ่งเก็บรายละเอียดผลการดำเนินการ และทำหน้าที่หาผลลัพธ์ของการดำเนินการทางตรรกะ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.19

LogicalOperator
+Eval() +New() +New()

รูปที่ 3.19 คลาส LogicalOperator

11) คลาส StringOperator เป็นคลาสที่สืบทอดมาจากคลาส Operator ซึ่งเก็บรายละเอียดผลการดำเนินการ และทำหน้าที่หาผลลัพธ์ของการดำเนินการของชุดอักขระ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.20

StringOperator
+Eval() +New() +New()

รูปที่ 3.20 คลาส StringOperator

12) คลาส Operand เป็นคลาสต้นแบบที่เก็บรายละเอียดการดำเนินการของตัวดำเนินการชนิดต่างๆ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.21

<b>Operand</b>
#m_szVarName String
#m_VarValue Object
+New()
+New()
+ToString()
+ExtractAndSetValue()
+Name()
+Value()

รูปที่ 3.21 คลาส Operand

13) คลาส BooleanOperand เป็นคลาสที่สืบทอดมาจากคลาส Operand ซึ่งเก็บรายละเอียดการดำเนินการของตัวดำเนินการชนิดตรรกะ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.22

<b>BoolOperand</b>
+ExtractAndSetValue()
+EqualTo()
+NotEqual()
+LessThan()
+LessThanOrEqualTc()
+GreaterThan()
+GreaterThanOrEqualTc()
+New()
+New()
+ANDD()
+ORR()

รูปที่ 3.22 คลาส BooleanOperand

14) คลาส DateOperand เป็นคลาสที่สืบทอดมาจากคลาส Operand ซึ่งเก็บรายละเอียดการดำเนินการของตัวดำเนินการชนิดวันที่ รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.23

<b>DateOperand</b>
+ExtractAndSetValue()
+EqualTo()
+NotEqual()
+LessThan()
+LessThanOrEqualTc()
+GreaterThan()
+GreaterThanOrEqualTc()
+New()
+New()

รูปที่ 3.23 คลาส DateOperand

15) คลาส DecimalOperand เป็นคลาสที่สืบทอดมาจากคลาส Operand ซึ่งเก็บรายละเอียดการดำเนินการของตัวดำเนินการชนิดตัวเลขจำนวนจริง รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.24

DecimalOperand
+ExtractAndSetValue() +EqualTo() +NotEqual() +LessThan() +LessThanOrEqualTc() +GreaterThan() +GreaterThanOrEqualTc() +Plus() +Minus() +Multiply() +Divide() +Modulo() +New() +New() +ToString()

รูปที่ 3.24 คลาส DecimalOperand

16) คลาส StringOperand เป็นคลาสที่สืบทอดมาจากคลาส Operand ซึ่งเก็บรายละเอียดการดำเนินการของตัวดำเนินการชนิดสตริง รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.25

StringOperand
+ExtractAndSetValue() +EqualTc() +NotEqual() +LessThan() +LessThanOrEqualTo() +GreaterThan() +GreaterThanOrEqualTo() +Length() +New() +New()

รูปที่ 3.25 คลาส StringOperand

17) คลาส LongOperand เป็นคลาสที่สืบทอดมาจากคลาส Operand ซึ่งเก็บรายละเอียดการดำเนินการของตัวดำเนินการชนิดตัวเลขจำนวนเต็ม รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.26

LongOperand
+ExtractAndSetValue() +EqualTo() +NotEqual() +LessThan() +LessThanOrEqualTc() +GreaterThan() +GreaterThanOrEqualTc() +Plus() +Minus() +Multiply() +Divide() +Modulo() +New() +New() +ToString()

รูปที่ 3.26 คลาส LongOperand

18) คลาส exportHTML เป็นคลาสที่สร้างกรณีทดสอบในรูปแบบของไฟล์เอชทีเอ็มแอล รายละเอียดของคลาสนี้แสดงดังรูปที่ 3.27

exportHTML
-htm_path String
+New() +htmlPath() +genHTML()

รูปที่ 3.27 คลาส exportHTML

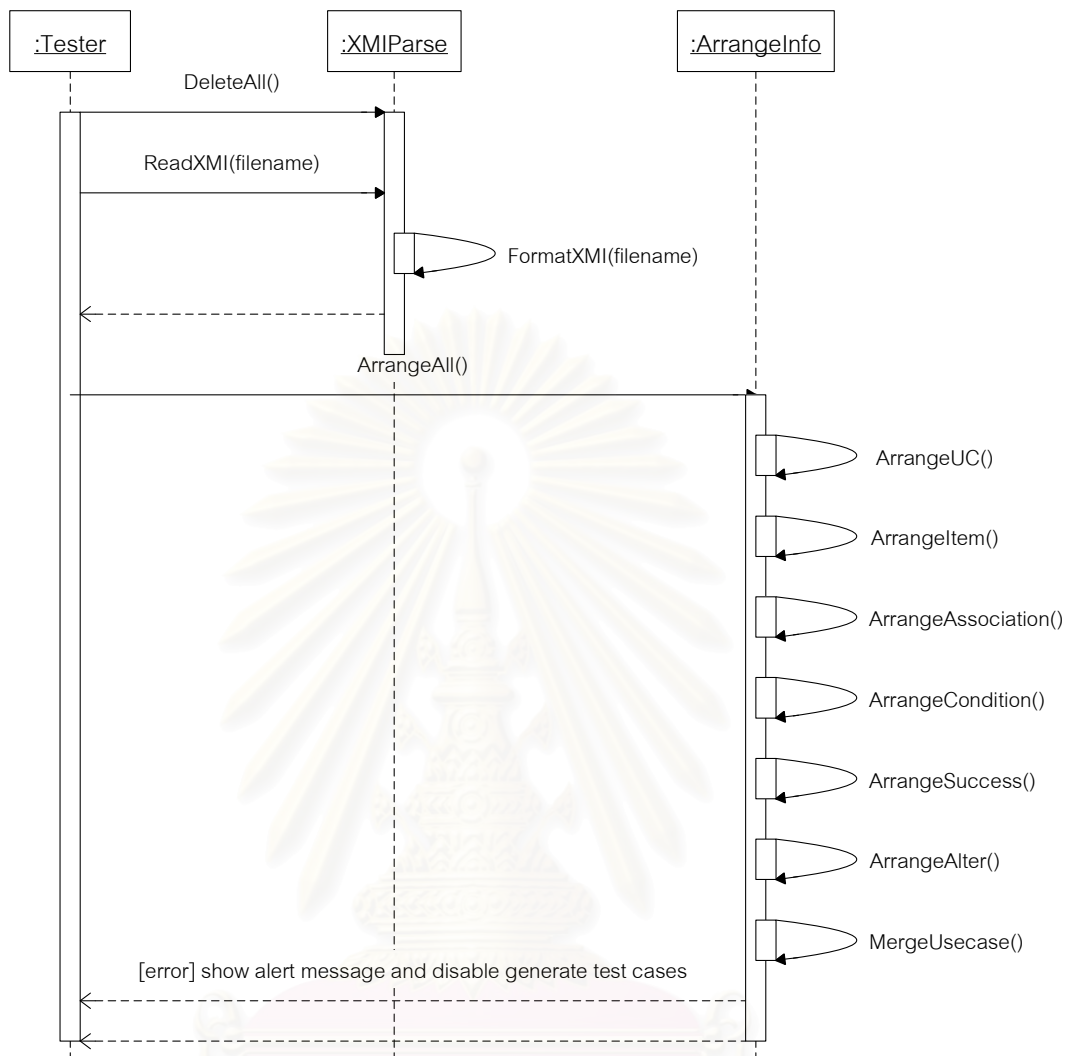
### 3.5.3 แผนภาพซีควเอนซ์

แผนภาพซีควเอนซ์เป็นแผนภาพแสดงกิจกรรมระหว่างวัตถุภายในระบบ ซึ่งแสดงการเรียกใช้และส่งข้อความตามลำดับเวลา และช่วยตรวจสอบความสมบูรณ์ของกิจกรรมในระบบว่าประกอบด้วยกิจกรรมที่เหมาะสม

แผนภาพซีควเอนซ์ของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสมี 2 แผนภาพได้แก่ แผนภาพซีควเอนซ์ของการวิเคราะห์แผนภาพยูสเคส และแผนภาพซีควเอนซ์ของการสร้างกรณีทดสอบ ซึ่งมีรายละเอียดดังนี้

#### 1) แผนภาพซีควเอนซ์ของการวิเคราะห์แผนภาพยูสเคส

แผนภาพซีควเอนซ์ของการวิเคราะห์แผนภาพยูสเคสแสดงดังรูปที่ 3.28



รูปที่ 3.28 แผนภาพซีควเอนซ์ของการวิเคราะห์แผนภาพยูสเคส

จากรูปที่ 3.28 การวิเคราะห์แผนภาพยูสเคสเริ่มต้นจากการลบข้อมูลเก่าในฐานข้อมูล และเมื่อเครื่องมีอรับไฟล์แผนภาพยูสเคสในรูปแบบเอกสารเอ็กซ์เอ็มแอลจากผู้ทดสอบแล้วจะอ่านไฟล์เพื่อตรวจหารายละเอียดยูสเคส และความสัมพันธ์ระหว่างยูสเคส โดยตรวจสอบที่แท็กสำคัญในเอกสารเอ็กซ์เอ็มแอลได้แก่

- uml:association แสดงความสัมพันธ์ระหว่างยูสเคสกับผู้ใช้งานระบบ และความสัมพันธ์ระหว่างยูสเคสด้วยกัน
- uml:usecase คือชื่อยูสเคสในแผนภาพยูสเคส
- uml:actor คือชื่อผู้ใช้งานระบบ

- `uml:associationend` คือจุดปลายของเส้นความสัมพันธ์ ซึ่งระบุเป็นหมายเลขประจำของยูสเคส หรือผู้ใช้งานระบบ

- `uml:stereotype` คือรูปแบบความสัมพันธ์ของยูสเคส

- `uml:taggedvalue` คือรายละเอียดอื่นๆ ของยูสเคส และผู้ใช้งานระบบ เช่น Pre-condition ลำดับเหตุการณ์ เป็นต้น

หลังจากนั้นจะจัดเรียงข้อมูลและบันทึกลงฐานข้อมูล โดยเริ่มต้นจากการจัดเรียงข้อมูลทั่วไปของยูสเคส จัดเรียงข้อมูลนำเข้าของยูสเคส จัดเรียงข้อมูลความสัมพันธ์ระหว่างยูสเคส จัดเรียงประโยคเงื่อนไขของลำดับเหตุการณ์ จัดเรียงลำดับเหตุการณ์สำเร็จ จัดเรียงลำดับเหตุการณ์ทางเลือกอื่น และทำการรวมยูสเคสที่มีความสัมพันธ์กัน เมื่อเครื่องมือทำงานเสร็จทุกขั้นตอนแล้วจะแจ้งเตือนให้ผู้ทดสอบรับทราบ แต่ถ้าเกิดข้อผิดพลาดในขั้นตอนใดขั้นตอนหนึ่งเครื่องมือจะแสดงข้อความผิดพลาด และไม่อนุญาตให้ดำเนินการสร้างกรณีทดสอบ

## 2) แผนภาพซีควเอนซ์ของการสร้างกรณีทดสอบ

แผนภาพซีควเอนซ์ของการสร้างกรณีทดสอบแสดงดังรูปที่ 3.29 ซึ่งจะเริ่มจากการสร้างข้อมูลทดสอบโดยสุ่มที่ละประโยคเงื่อนไขของลำดับเหตุการณ์ของยูสเคส พิจารณาข้อมูลนำเข้าของแต่ละยูสเคสตามชนิดของข้อมูลนำเข้าดังนี้

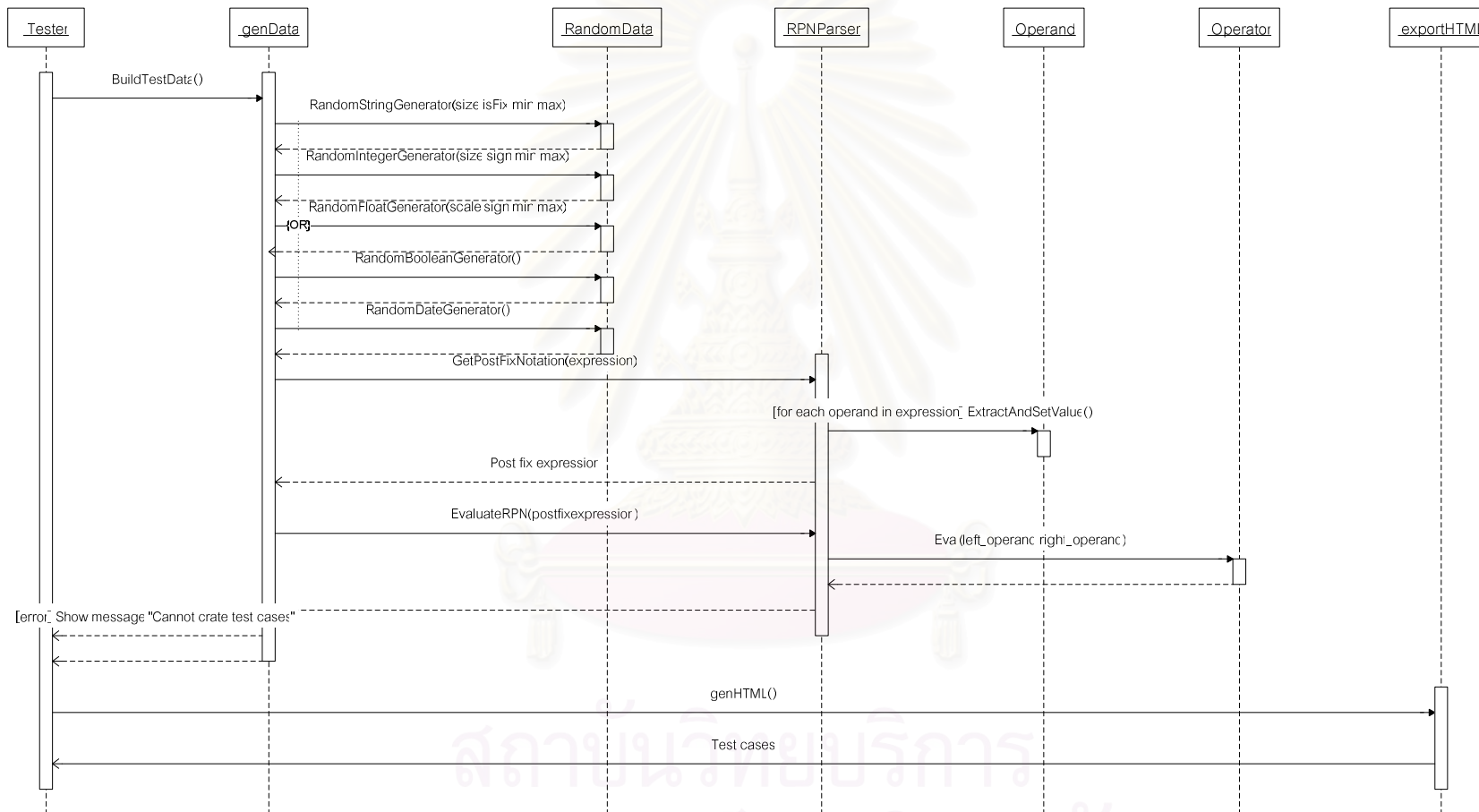
- ข้อมูลนำเข้าเป็นชนิดชุดอักขระ จะสร้างข้อมูลทดสอบโดยเรียกใช้เมธอด `RandomStringGenerator(size,isFix,min,max)`

- ข้อมูลนำเข้าเป็นชนิดเลขจำนวนเต็ม จะสร้างข้อมูลทดสอบโดยเรียกใช้เมธอด `RandomIntegerGenerator(size,sign,min,max)`

- ข้อมูลนำเข้าเป็นชนิดเลขจำนวนจริง จะสร้างข้อมูลทดสอบโดยเรียกใช้เมธอด `RandomFloatGenerator(scale,sign,min,max)`

- ข้อมูลนำเข้าเป็นชนิดตรรกะ จะสร้างข้อมูลทดสอบโดยเรียกใช้เมธอด `RandomBooleanGenerator()`

- ข้อมูลนำเข้าเป็นชนิดวันที่ จะสร้างข้อมูลทดสอบโดยเรียกใช้เมธอด `RandomDateGenerator()`



รูปที่ 3.29 แผนภาพซีควเอนซ์ของการสร้างกรณีทดสอบ



นำค่าของข้อมูลทดสอบที่ได้แทนชื่อข้อมูลนำเข้าในประโยคเงื่อนไขของลำดับเหตุการณ์ให้ได้ประโยคเงื่อนไขใหม่ที่ไม่มีชื่อข้อมูลนำเข้า จากนั้นสร้างวัตถุของคลาส RPNParser แล้วเรียกใช้เมธอด `GetPostFixNotation(expression)` เพื่อแปลงประโยคเงื่อนไขใหม่จากสัญกรณ์เติมกลาง (Infix notation) เป็นสัญกรณ์เติมหลัง (Postfix notation) ในขณะที่แปลงประโยคเงื่อนไขจะสร้างวัตถุของคลาส Operand สำหรับตัวดำเนินการในประโยคเงื่อนไข เมื่อได้ประโยคเงื่อนไขแบบสัญกรณ์เติมหลังแล้วนำไปตรวจสอบว่าข้อมูลทดสอบที่สร้างขึ้นสอดคล้องกับประโยคเงื่อนไขหรือไม่ โดยการเรียกเมธอด `EvaluateRPN(postfixexpression)` ถ้าค่าที่ส่งกลับเป็นจริง แสดงว่าข้อมูลทดสอบสอดคล้องกับประโยคเงื่อนไขแล้ว มิฉะนั้นจะต้องกลับไปสร้างข้อมูลทดสอบใหม่และตรวจสอบกับประโยคเงื่อนไขอีกครั้ง

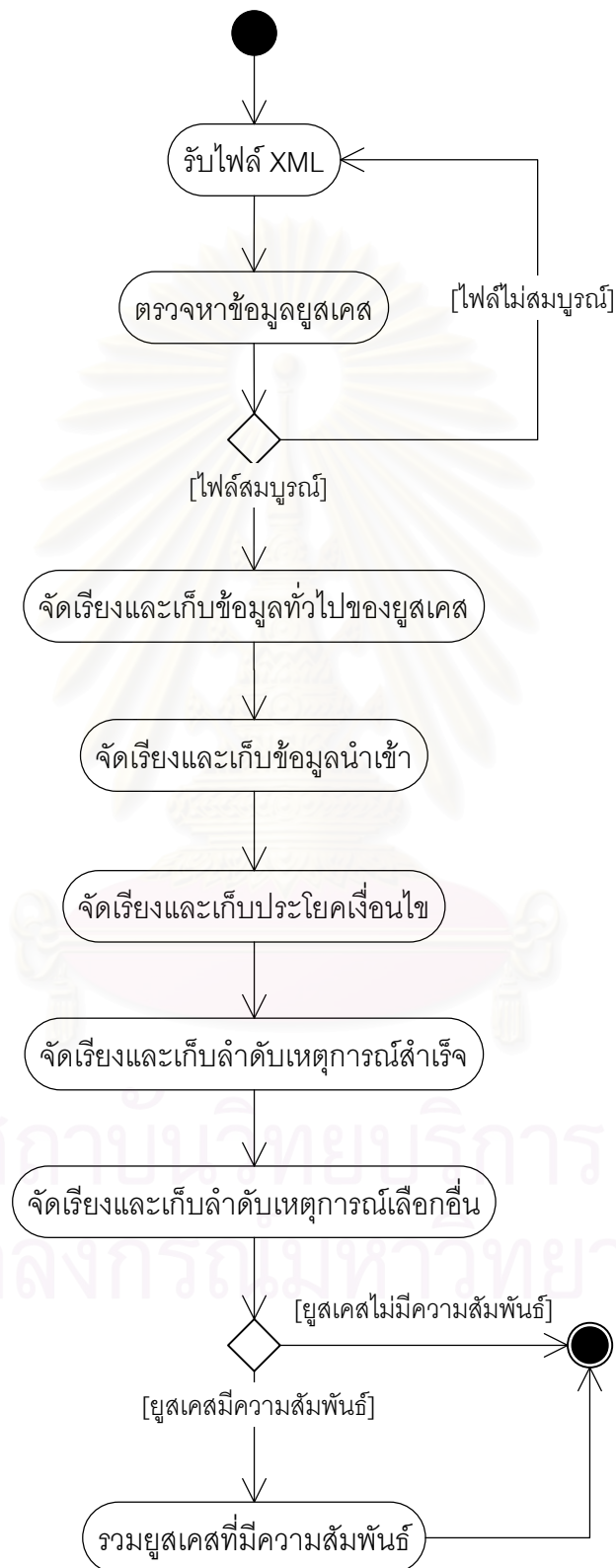
เมื่อสร้างข้อมูลทดสอบแล้วจะนำข้อมูลทดสอบที่สร้างไปใช้ในการสร้างกรณีทดสอบในรูปแบบไฟล์เอชทีเอ็มแอล โดยการสร้างวัตถุของคลาส `exportHTML` แล้วเรียกใช้เมธอด `genHTML()` จะได้กรณีทดสอบทั้งหมดของแผนภาพยูสเคส

#### 3.5.4 แผนภาพลำดับกิจกรรม

แผนภาพลำดับกิจกรรมเป็นแผนภาพแสดงขั้นตอนการทำงานจากกิจกรรมหนึ่งไปอีกกิจกรรมหนึ่ง เพื่อให้สำเร็จการทำงานใดๆ ภายในระบบ แผนภาพลำดับกิจกรรมของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสสามารถแบ่งออกเป็น 2 ส่วนเช่นเดียวกับแผนภาพซีควเอนซ์ได้แก่แผนภาพลำดับกิจกรรมการวิเคราะห์แผนภาพยูสเคส และแผนภาพลำดับกิจกรรมการสร้างกรณีทดสอบ

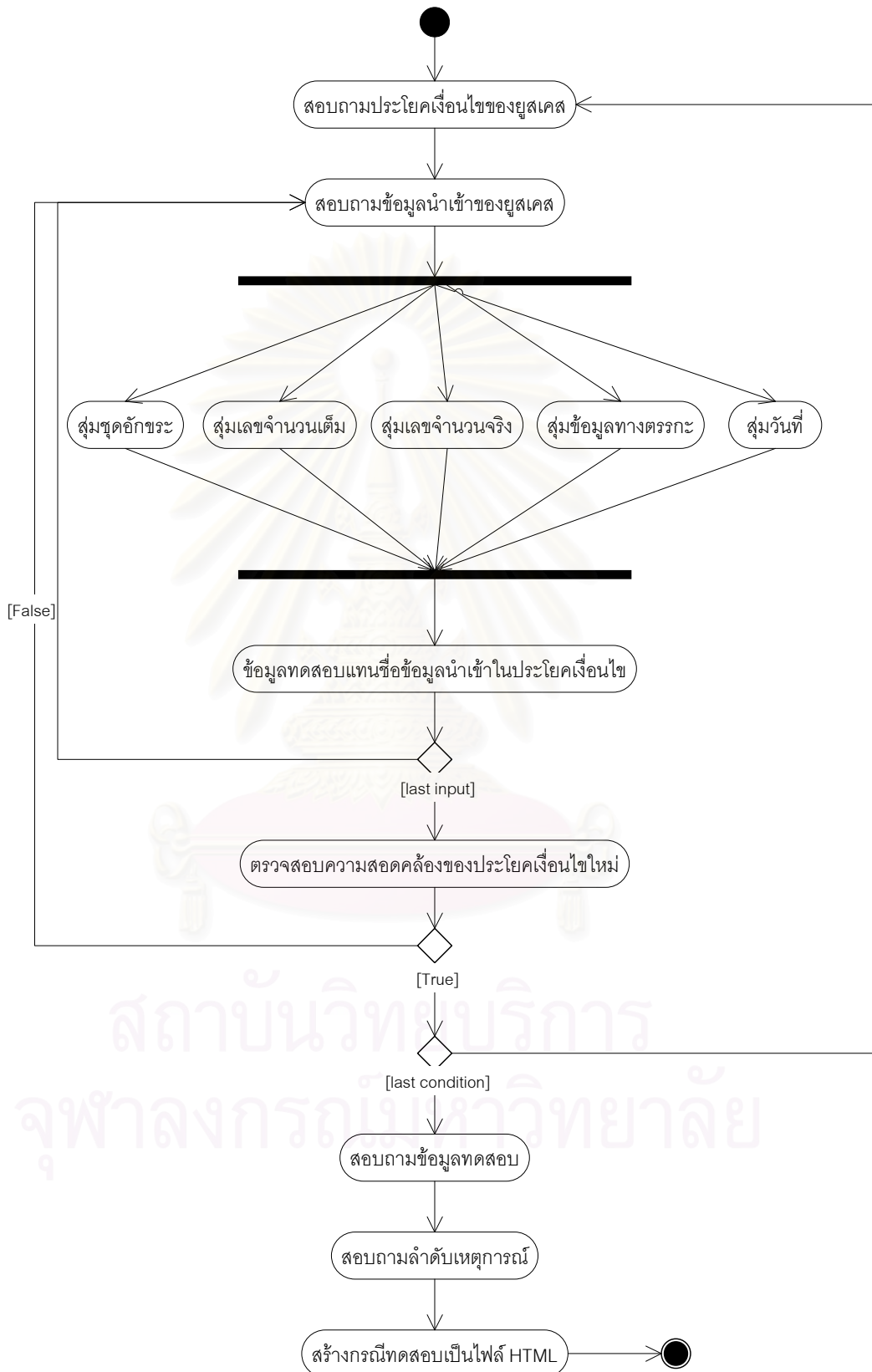
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## 1) แผนภาพลำดับกิจกรรมการวิเคราะห์แผนภาพยูสเคส



รูปที่ 3.30 แผนภาพลำดับกิจกรรมการวิเคราะห์แผนภาพยูสเคส

## 2) แผนภาพลำดับกิจกรรมการสร้างกรณีทดสอบ



รูปที่ 3.31 แผนภาพลำดับกิจกรรมการสร้างกรณีทดสอบ

## บทที่ 4

### การพัฒนาเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

บทนี้กล่าวถึงการพัฒนาเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส โดยกล่าวถึงสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ, ฐานข้อมูลของเครื่องมือ และโครงสร้างของเครื่องมือ ซึ่งมีรายละเอียดดังนี้

#### 4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ

##### 4.1.1 ฮาร์ดแวร์ (Hardware)

ฮาร์ดแวร์ที่ใช้ในการพัฒนาเครื่องมือประกอบด้วย

- เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผลเพนเทียมไฟร์ 1.6 กิกะเฮิร์ตซ์ (Pentium 4 1.6 GHz.)
- หน่วยความจำหลัก (RAM) 256 เมกะไบต์ (256 MB)
- ฮาร์ดดิสก์ (Harddisk) 40 กิกะไบต์ (40 GB)

##### 4.1.2 ซอฟต์แวร์ (Software)

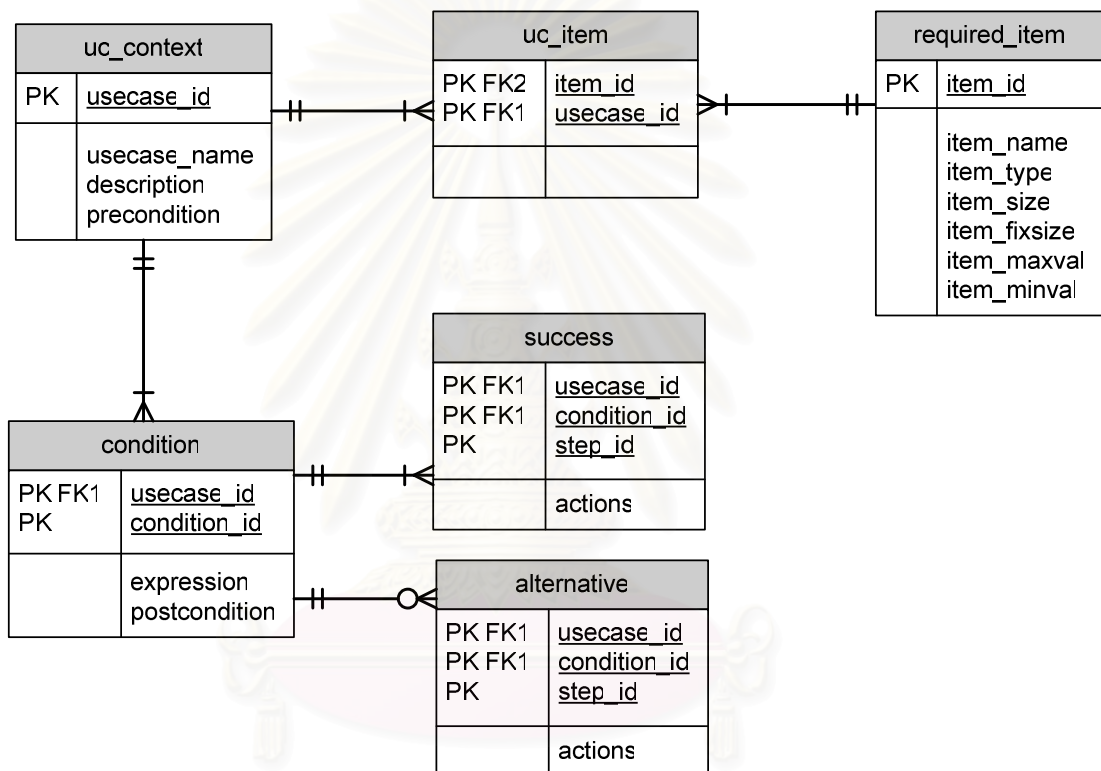
ซอฟต์แวร์ที่ใช้ในการพัฒนาเครื่องมือประกอบด้วย

- ระบบปฏิบัติการ (Operating system) ไมโครซอฟท์วินโดวส์เอ็กซ์พี โพรเฟสชันแนล (Microsoft Windows XP Professional)
- ระบบจัดการฐานข้อมูล (Database management system) ไมโครซอฟท์ออฟฟิศแอ็กเซส 2003 (Microsoft Office Access 2003)
- พัฒนาเครื่องมือด้วยภาษาวิซิวัลเบสิกดอทเน็ตโดยใช้โปรแกรมไมโครซอฟท์วิซิวัลสตูดิโอไดคอตเน็ต 2003 (Microsoft Visual Studio .NET 2003)
- เครื่องมือช่วยสร้างเอกสารแผนภาพยูเอ็มแอล เวอร์ชันรอสเ็นเตอร์ไพร์ส (Rational Rose Enterprise Edition) เวอร์ชัน 2002
- เครื่องมือแปลงเอกสารแผนภาพยูเอ็มแอลเป็นไฟล์เอ็กซ์เอ็มแอล ยูนิซีเอสเอ็กซ์เอ็มไอ 1.3 แอดอิน (Unisys XMI 1.3 add-ins)

## 4.2 ฐานข้อมูลของเครื่องมือ

ฐานข้อมูลของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสสามารถแสดงได้โดยใช้แผนภาพความสัมพันธ์ระหว่างเอนทิตี ซึ่งเป็นแผนภาพที่ใช้แสดงข้อมูลที่เก็บในฐานข้อมูล รวมทั้งแสดงความสัมพันธ์ระหว่างตารางต่างๆ ในฐานข้อมูล

แผนภาพความสัมพันธ์ระหว่างเอนทิตีของเครื่องมือนี้แสดงดังรูปที่ 4.1



รูปที่ 4.1 แผนภาพแสดงความสัมพันธ์ระหว่างเอนทิตีของเครื่องมือ

จากรูปที่ 4.1 ฐานข้อมูลของเครื่องมือประกอบด้วย 6 ตาราง ซึ่งมีรายละเอียดดังนี้

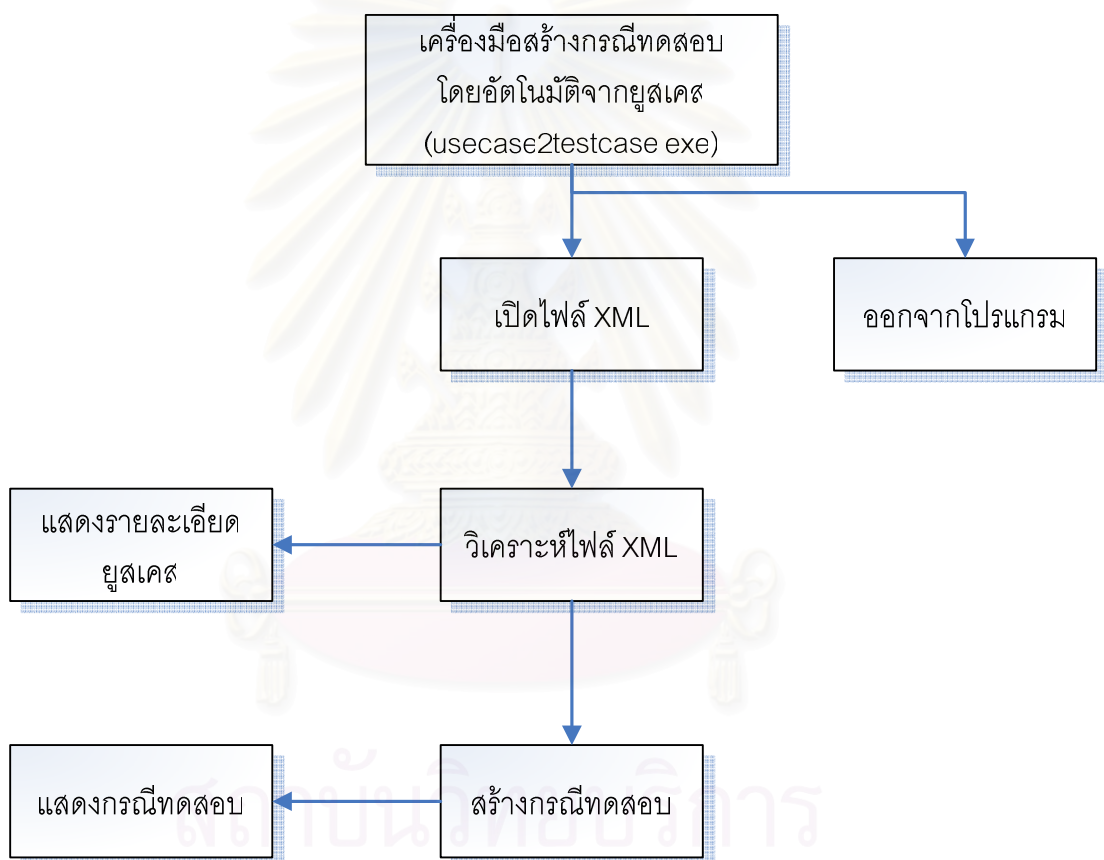
- 1) ตาราง uc\_context เป็นตารางที่เก็บข้อมูลทั่วไปของยูสเคส
- 2) ตาราง required\_item เป็นตารางที่เก็บข้อมูลทั่วไปของข้อมูลนำเข้าของยูสเคส
- 3) ตาราง uc\_item เป็นตารางที่เก็บความสัมพันธ์ระหว่างข้อมูลนำเข้ากับยูสเคส เพื่อระบุว่าเป็นข้อมูลนำเข้าของยูสเคสใดบ้าง
- 4) ตาราง condition เป็นตารางที่เก็บประโยคเงื่อนไขของลำดับเหตุการณ์ที่เกิดขึ้นในยูสเคส
- 5) ตาราง success เป็นตารางที่เก็บลำดับการทำงานของลำดับเหตุการณ์สำเร็จ

6) ตาราง alternative เป็นตารางที่เก็บลำดับการทำงานของลำดับเหตุการณ์ทางเลือกอื่น

หมายเหตุ: พจนานุกรมข้อมูล (Data dictionary) แสดงไว้ในภาคผนวก ก

### 4.3 โครงสร้างของเครื่องมือ

โครงสร้างของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสแสดงได้ดังรูปที่ 4.2

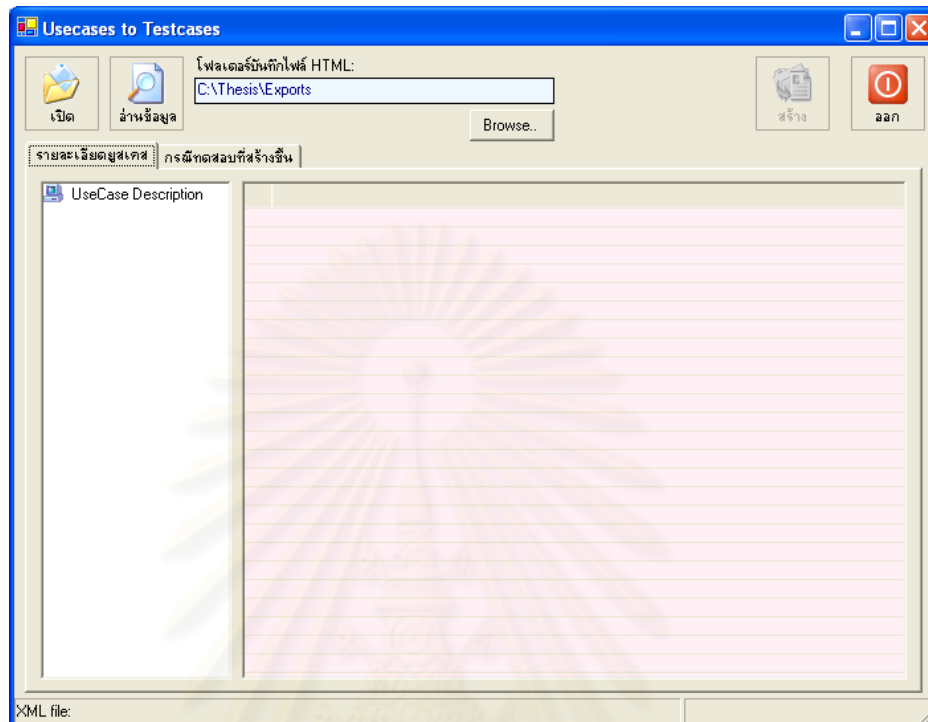


รูปที่ 4.2 โครงสร้างของเครื่องมือ

จากรูปที่ 4.2 โครงสร้างของเครื่องมือเริ่มจากเมื่อเปิดเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสซึ่งเป็นหน้าหลักของเครื่องมือ จากนั้นเลือกไฟล์เอกซ์เอ็มแอลที่ต้องการสร้างกรณีทดสอบ ต่อมาเลือกการอ่านข้อมูลเพื่อวิเคราะห์ไฟล์เอกซ์เอ็มแอล จากนั้นจะแสดงรายละเอียดยูสเคส หลังจากนั้นกำหนดที่เก็บกรณีทดสอบแล้วเลือกสร้างกรณีทดสอบ และขั้นตอนสุดท้ายจะได้กรณีทดสอบที่สร้างขึ้นพร้อมทั้งแสดงรายละเอียดกรณีทดสอบทั้งหมด ซึ่งในแต่ละส่วนมีหน้าจอดังนี้

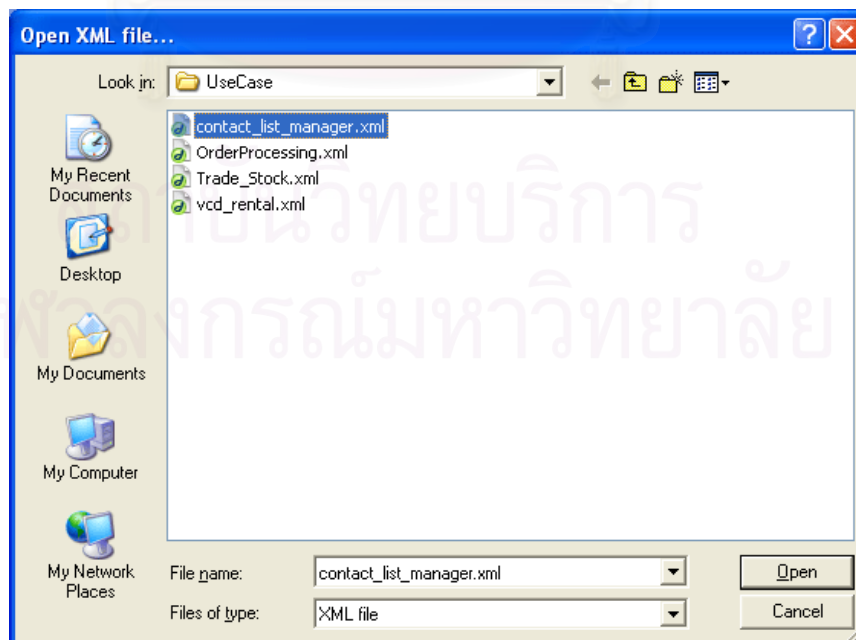
#### 4.3.1 เครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส หน้าจอหลักแสดงดังรูปที่

4.3



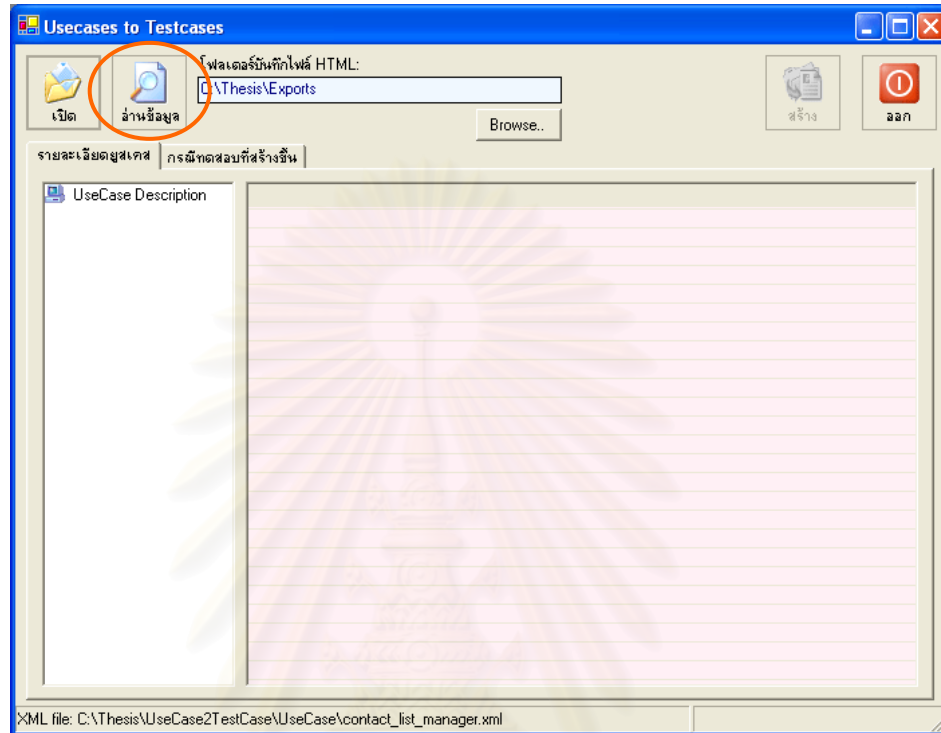
รูปที่ 4.3 หน้าจอหลักของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

#### 4.3.2 เปิดไฟล์เอ็กซ์เอ็มแอล เป็นหน้าจอที่ใช้เลือกไฟล์เอ็กซ์เอ็มแอลสำหรับสร้างกรณีทดสอบ แสดงดังรูปที่ 4.4



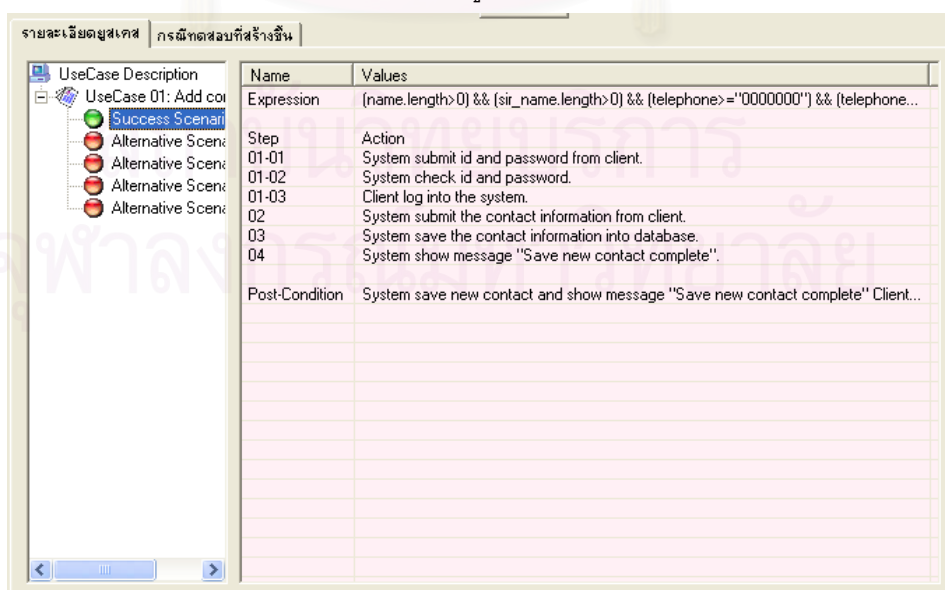
รูปที่ 4.4 หน้าจอสำหรับเลือกไฟล์เอ็กซ์เอ็มแอล

4.3.3 วิเคราะห์ไฟล์เอ็กซ์เอ็มแอล เป็นส่วนที่ใช้สำหรับวิเคราะห์ไฟล์เอ็กซ์เอ็มแอล  
แสดงได้ดังรูปที่ 4.5



รูปที่ 4.5 ปุ่มอ่านข้อมูล (สำหรับวิเคราะห์ไฟล์)

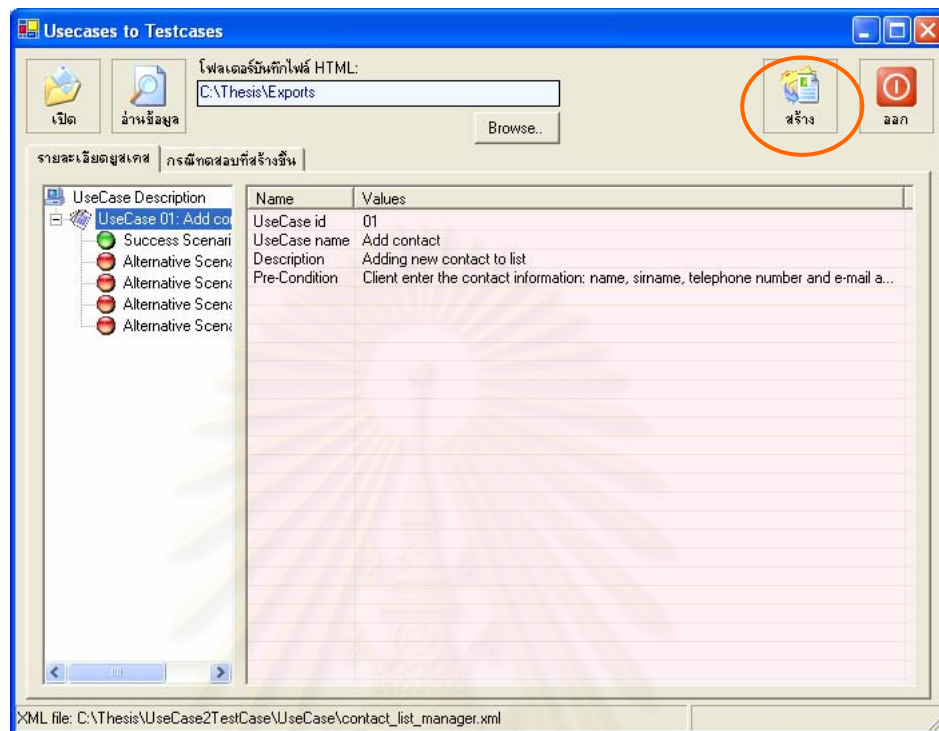
4.3.4 แสดงรายละเอียดยูสเคส เป็นส่วนที่แสดงรายละเอียดของยูสเคสหลังจากที่  
เครื่องมือวิเคราะห์ไฟล์เอ็กซ์เอ็มแอล แสดงได้ดังรูปที่ 4.6



รูปที่ 4.6 หน้าจอแสดงรายละเอียดยูสเคส

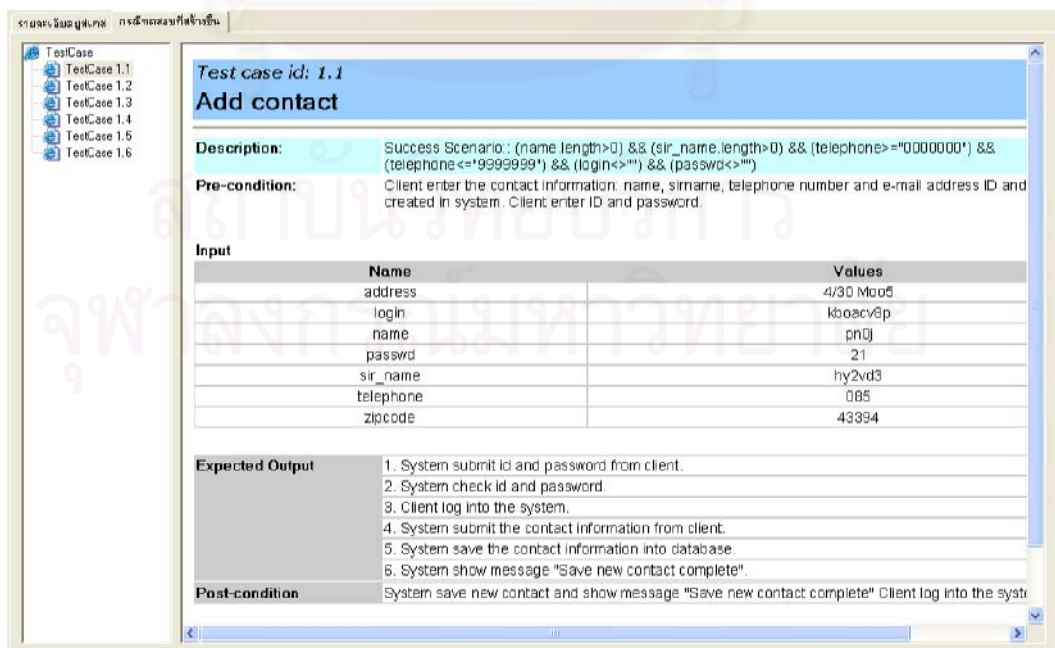


#### 4.3.5 สร้างกรณีทดสอบ เป็นส่วนที่ใช้สำหรับสร้างกรณีทดสอบ แสดงได้ดังรูปที่ 4.7



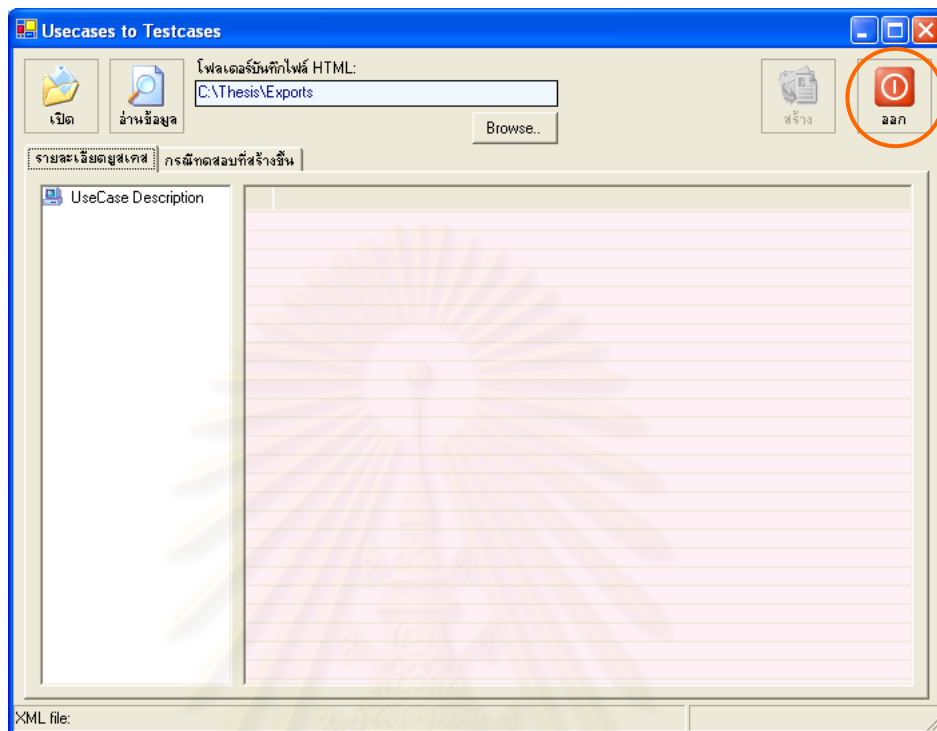
รูปที่ 4.7 ปุ่มสร้างกรณีทดสอบ

#### 4.3.6 แสดงกรณีทดสอบ เป็นส่วนที่ใช้แสดงกรณีทดสอบที่สร้างขึ้นโดยเครื่องมือ แสดงได้ดังรูปที่ 4.8



รูปที่ 4.8 หน้าจอแสดงกรณีทดสอบ

4.3.7 ออกจากโปรแกรม เป็นส่วนที่ใช้ปิดการใช้งานเครื่องมือ แสดงได้ดังรูปที่ 4.9



รูปที่ 4.9 ปุ่มออก

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 5

### การทดสอบเครื่องมือ

บทนี้กล่าวถึงการทดสอบเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสที่พัฒนาขึ้นด้วยแนวทางที่ได้เสนอไว้ในบทที่ 3 สภาพแวดล้อมที่ใช้ในการทดสอบเครื่องมือ และผลที่ได้จากการทดสอบ ซึ่งมีรายละเอียดดังนี้

#### 5.1 ขั้นตอนการทดสอบเครื่องมือที่พัฒนาขึ้น

ขั้นตอนในการทดสอบเครื่องมือที่พัฒนาขึ้นประกอบด้วยขั้นตอนต่างๆ ดังนี้

5.1.1 เลือกแผนภาพยูสเคสที่ประกอบด้วยรายละเอียดแต่ละยูสเคสที่ต้องการทดสอบ

5.1.2 สำหรับกรณีที่แผนภาพยูสเคสและรายละเอียดยูสเคสไม่ตรงตามรูปแบบ ทำการปรับแผนภาพยูสเคสและรายละเอียดยูสเคสให้ตรงกับรูปแบบที่กำหนดไว้ในบทที่ 3

5.1.3 แปลงแผนภาพยูสเคสและรายละเอียดยูสเคสเป็นไฟล์เอกซ์เอ็มแอล

5.1.4 สร้างกรณีทดสอบของไฟล์เอกซ์เอ็มแอลในข้อ 5.1.3 ด้วยเครื่องมือที่พัฒนาขึ้น

5.1.5 พิจารณากรณีทดสอบที่สร้างจากเครื่องมือว่าครอบคลุมทุกลำดับเหตุการณ์ที่เป็นไปได้ซึ่งระบุไว้ในแต่ละยูสเคสหรือไม่ โดยพิจารณาจากจำนวนกรณีทดสอบต้องเท่ากับจำนวนประโยคเงื่อนไขของลำดับเหตุการณ์ในยูสเคส

5.1.6 พิจารณาค่าของข้อมูลทดสอบของกรณีทดสอบที่สร้างขึ้นต้องสอดคล้องกับประโยคเงื่อนไขของลำดับเหตุการณ์

#### 5.2 สภาพแวดล้อมที่ใช้ในการทดสอบเครื่องมือ

เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ มีรายละเอียดดังนี้

- เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผลเพนเทียมโฟร์ 1.6 กิกะเฮิร์ตซ์ (Pentium 4 1.6 GHz.)
- หน่วยความจำหลัก (RAM) 256 เมกะไบต์ (256 MB)

- ฮาร์ดดิสก์ (Harddisk) 40 กิกะไบต์ (40 GB)

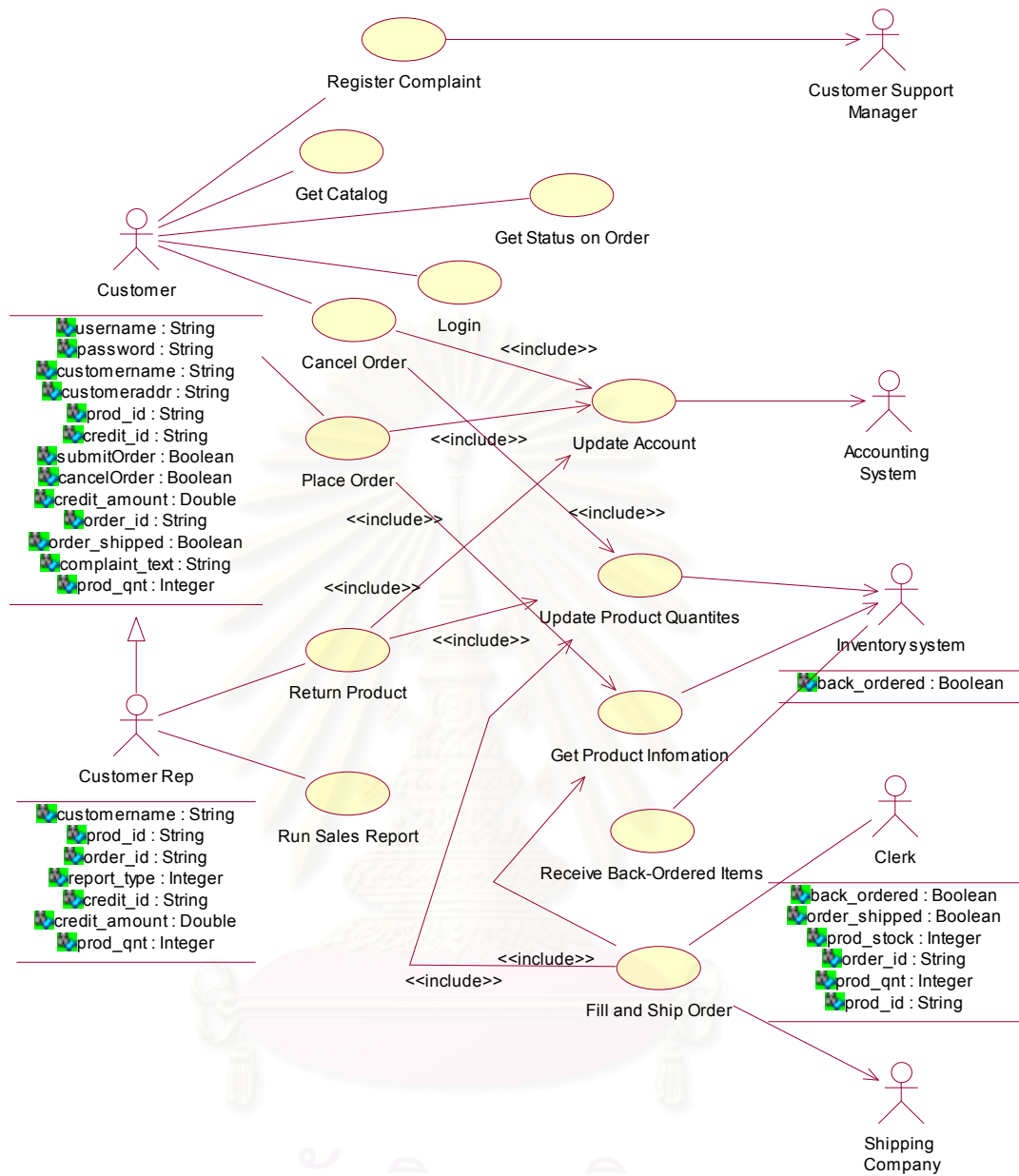
### 5.3 แผนภาพยูสเคสที่ใช้ในการทดสอบเครื่องมือ

แผนภาพยูสเคสที่นำมาใช้ทดสอบเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส ต้องเป็นแผนภาพยูสเคสที่มีรายละเอียดยูสเคส ภายในแผนภาพยูสเคสต้องไม่มียูสเคสที่มีความสัมพันธ์กันแบบซ้ำซ้อนดังที่ได้เสนอไว้ในบทที่ 3 และมีลักษณะตามขอบเขตของงานวิจัยที่กล่าวไว้ในบทที่ 1

แผนภาพยูสเคสและรายละเอียดยูสเคสที่เลือกมาทดสอบเครื่องมือมี 3 แผนภาพคือ แผนภาพยูสเคสของระบบสั่งซื้อสินค้า [10] แผนภาพยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์ [11] และแผนภาพยูสเคสของระบบให้บริการเช่าวีซีดี แต่ละแผนภาพยูสเคสมีรายละเอียดดังนี้

#### 5.3.1 ระบบสั่งซื้อสินค้า

ระบบสั่งซื้อสินค้าเป็นระบบที่ใช้สำหรับจัดการการสั่งซื้อสินค้าของลูกค้า และจัดส่งสินค้าให้กับลูกค้า ซึ่งแผนภาพยูสเคสของระบบสั่งซื้อสินค้านี้แสดงดังรูปที่ 5.1 และตัวอย่างรายละเอียดยูสเคสของระบบสั่งซื้อสินค้าในรูปแบบไฟล์เอ็กซ์เอ็มแอลแสดงในรูปที่ 5.2 โดยระบบสั่งซื้อสินค้านี้ประกอบด้วยยูสเคส 13 ยูสเคส สำหรับรายละเอียดยูสเคสของระบบสั่งซื้อสินค้าแสดงในภาคผนวก ข



รูปที่ 5.1 แผนภาพยูสเคสของระบบสั่งซื้อสินค้า

```

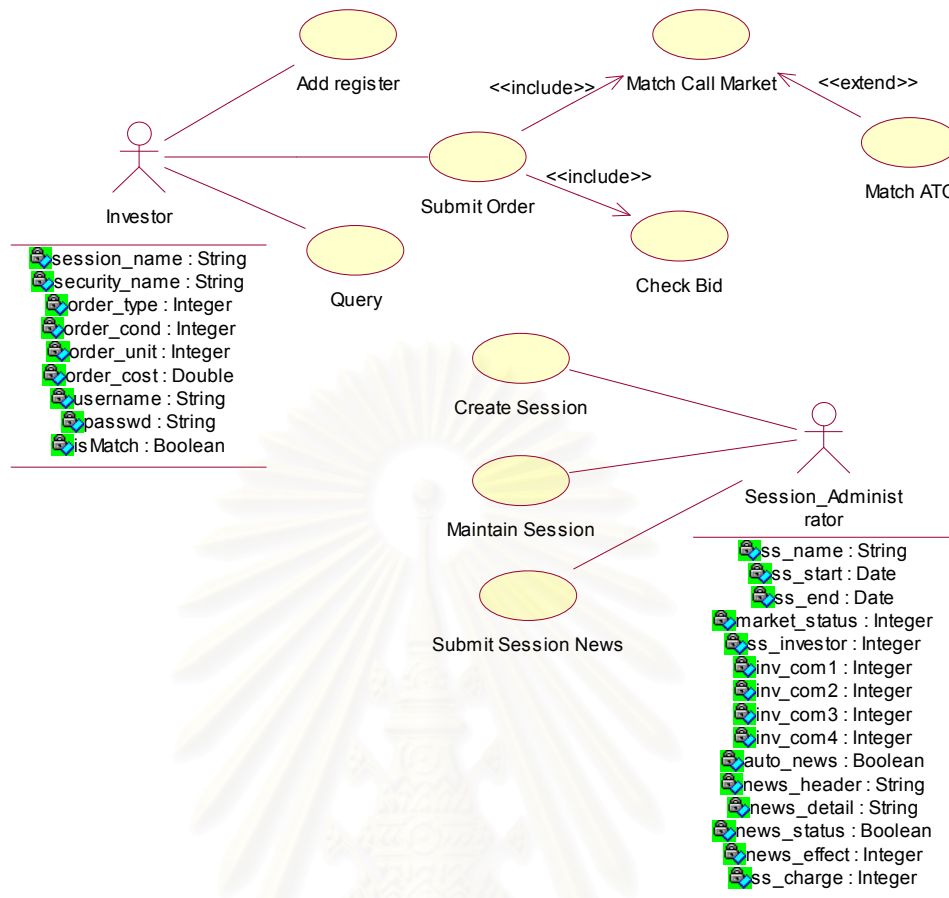
- <XML xmlns:UML="http://org.omg/UML/1.0" timestamp="Sat Sep 11 11:49:28 2004">
+ <XML.header>
- <XML.content>
  <!-- ***** OrderProcessing [node1] ***** -->
+ <UML:Model xml.id="G.0" name="OrderProcessing" visibility="public" isSpecification="false" isRoot="false" isLeaf="false"
  isAbstract="false">
  <UML:TaggedValue xml.id="XX.11.1149.25.2" tag="RationalRose:Tool#0" value="myUseCase" modelElement="G.0" />
  <UML:TaggedValue xml.id="XX.11.1149.25.3" tag="RationalRose$myUseCase:usecase_no" value="10"
  modelElement="S.254.1149.23.37" />
  <UML:TaggedValue xml.id="XX.11.1149.25.4" tag="RationalRose$myUseCase:Pre-condition" value="The user selects Contact
  Customer Service." modelElement="S.254.1149.23.37" />
  <UML:TaggedValue xml.id="XX.11.1149.25.5" tag="RationalRose$myUseCase:Post-condition" value="[0] The system send
  message in an e-mail message to customer support manager." modelElement="S.254.1149.23.37" />
- <UML:TaggedValue xml.id="XX.11.1149.25.6" tag="RationalRose$myUseCase:SuccessScenario"
  modelElement="S.254.1149.23.37">
  <UML:TaggedValue.value>[0] (complaint_text <> "") @ [1] The Message screen is displayed. | [2] The user enters
  complaint text. | [3] The user select Submit. | [4] The system sends the text entered in an e-mail message to the
  customer support manager.</UML:TaggedValue.value>
</UML:TaggedValue>
<UML:TaggedValue xml.id="XX.11.1149.25.7" tag="RationalRose$myUseCase:AlternativeScenario" value=""
  modelElement="S.254.1149.23.37" />
<UML:TaggedValue xml.id="XX.11.1149.25.8" tag="RationalRose$myUseCase:IsAbstract" value="0"
  modelElement="S.254.1149.23.37" />
<UML:TaggedValue xml.id="XX.11.1149.25.9" tag="RationalRose$myUseCase:usecase_no" value="9"
  modelElement="S.254.1149.23.37" />
<UML:TaggedValue xml.id="XX.11.1149.25.10" tag="RationalRose$myUseCase:Pre-condition" value="The customer selects Got
  Catalog." modelElement="S.254.1149.23.38" />
<UML:TaggedValue xml.id="XX.11.1149.25.11" tag="RationalRose$myUseCase:Post-condition" value="[0] The system save the
  order." modelElement="S.254.1149.23.38" />
- <UML:TaggedValue xml.id="XX.11.1149.25.12" tag="RationalRose$myUseCase:SuccessScenario"
  modelElement="S.254.1149.23.38">
  <UML:TaggedValue.value>[0] (customername <> "") &#x26; (customeraddr <> "") @ [1] The Order Catalog screen is
  displayed. | [2] The user enters a name and address. | [3] The user select submit. | [4] The system creates an order
  for a catalog product and a total amount of zero. | [5] The system sets the order status to pending. | [6] The system
  request that the order database store the order. | [7] The order database returns a unique order ID. | [8] The system
  returns the Order ID to the requestor.</UML:TaggedValue.value>
</UML:TaggedValue>
<UML:TaggedValue xml.id="XX.11.1149.25.13" tag="RationalRose$myUseCase:AlternativeScenario" value=""
  modelElement="S.254.1149.23.38" />
<UML:TaggedValue xml.id="XX.11.1149.25.14" tag="RationalRose$myUseCase:IsAbstract" value="0"
  modelElement="S.254.1149.23.38" />

```

รูปที่ 5.2 ตัวอย่างรายละเอียดยูสเคสของระบบสั่งซื้อสินค้า

### 5.3.2 ระบบจำลองการซื้อขายหลักทรัพย์

ระบบจำลองการซื้อขายหลักทรัพย์ เป็นระบบที่ใช้จำลองการซื้อขายหลักทรัพย์ ซึ่งระบบนี้แบ่งการทำงานออกเป็น 2 ส่วนคือส่วนของนักลงทุน และส่วนของตลาดหลักทรัพย์ จากแผนภาพยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์ในรูปที่ 5.3 ประกอบด้วยยูสเคสจำนวน 9 ยูสเคส และตัวอย่างรายละเอียดยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์ในรูปแบบไฟล์เอ็กซ์เอ็มแอลแสดงดังรูปที่ 5.4 สำหรับรายละเอียดยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์แสดงในภาคผนวก ข



รูปที่ 5.3 แผนภาพยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์

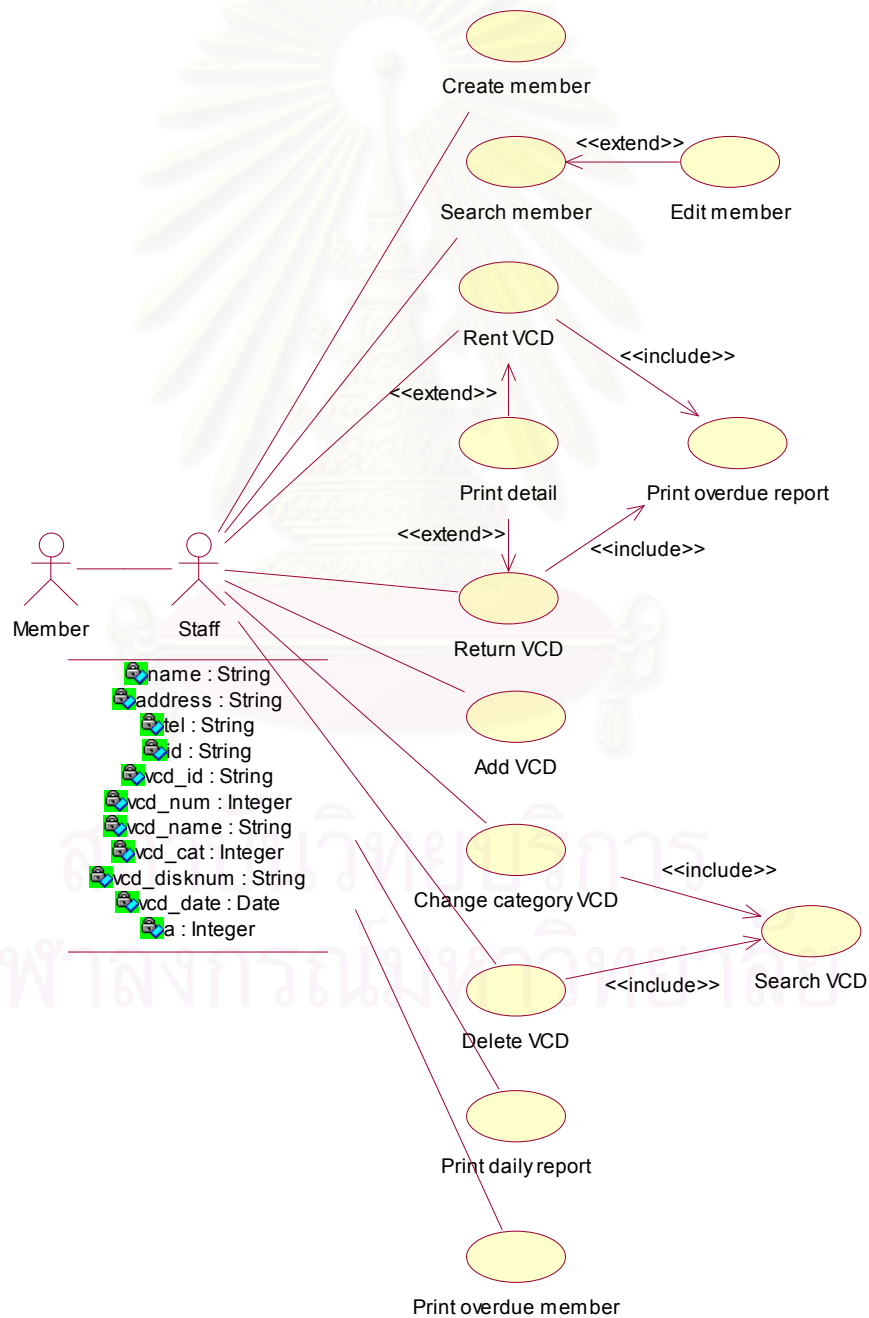
```

- <XML: xmi.version="1.1" xmlns:UML="href://org.omg/UML/1.3" timestamp="Sat Sep 11 13:24:54 2004">
+ <XML:header>
- <XML:content>
  <!-- ----- Trade_Stock [Model] ----- -->
+ <UML:Model xmi.id="G.0" name="Trade_Stock" visibility="public" isSpecification="false" isRoot="false" isLeaf="false"
  isAbstract="false">
  <UML:TaggedValue xmi.id="XX.11.1324.37.2" tag="RationalRose:Tool#0" value="myUseCase" modelElement="G.0" />
  <UML:TaggedValue xmi.id="XX.11.1324.37.3" tag="RationalRose$myUseCase:usecase_no" value="1"
  modelElement="$S.254.1324.36.30" />
  <UML:TaggedValue xmi.id="XX.11.1324.37.4" tag="RationalRose$myUseCase:Pre-condition" value="Session administrator
  register information into system." modelElement="$S.254.1324.36.30" />
- <UML:TaggedValue xmi.id="XX.11.1324.37.5" tag="RationalRose$myUseCase:SuccessScenario"
  modelElement="$S.254.1324.36.30">
  <UML:TaggedValue.value>[0] (ss_name << "" ) && (ss_start < ss_end) && (market_status <= 2) && (ss_charge > 0) @
  [1] Session administrator enter session s information. | [2] System create session id. | [3] System insert new
  session information into database.</UML:TaggedValue.value>
  </UML:TaggedValue>
  <UML:TaggedValue xmi.id="XX.11.1324.37.6" tag="RationalRose$myUseCase:Post-condition" value="[0] New session
  information is crated and saved in database." modelElement="$S.254.1324.36.30" />
  <UML:TaggedValue xmi.id="XX.11.1324.37.7" tag="RationalRose$myUseCase:AlternativeScenario" value=""
  modelElement="$S.254.1324.36.30" />
  <UML:TaggedValue xmi.id="XX.11.1324.37.8" tag="RationalRose$myUseCase:IsAbstract" value="0"
  modelElement="$S.254.1324.36.30" />
  <UML:TaggedValue xmi.id="XX.11.1324.37.9" tag="RationalRose$myUseCase:usecase_no" value="3"
  modelElement="$S.254.1324.36.32" />
  <UML:TaggedValue xmi.id="XX.11.1324.37.10" tag="RationalRose$myUseCase:Pre-condition" value="System start sesion
  news page." modelElement="$S.254.1324.36.32" />
  <UML:TaggedValue xmi.id="XX.11.1324.37.11" tag="RationalRose$myUseCase:Post-condition" value="[0] System save new
  news detail and sent to newsmanager." modelElement="$S.254.1324.36.32" />
- <UML:TaggedValue xmi.id="XX.11.1324.37.12" tag="RationalRose$myUseCase:SuccessScenario"
  modelElement="$S.254.1324.36.32">
  <UML:TaggedValue.value>[0] (news_header << "" ) && (news_detail << "" ) && (news_effect >= 0) && (news_effect <=
  100) @ [1] Session administrator enter news detail. | [2] System submit news and save into database. | [3] System
  send new news to newsmanager.</UML:TaggedValue.value>
  </UML:TaggedValue>
  <UML:TaggedValue xmi.id="XX.11.1324.37.13" tag="RationalRose$myUseCase:AlternativeScenario" value=""
  </UML:TaggedValue>
  </UML:Model>
  </UML:Content>
  </XML:content>
  </XML:header>
</XML:xmi.version="1.1" xmlns:UML="href://org.omg/UML/1.3" timestamp="Sat Sep 11 13:24:54 2004">
  
```

รูปที่ 5.4 ตัวอย่างรายละเอียดยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์

### 5.3.3 ระบบให้บริการเช่าวีซีดี

ระบบให้บริการเช่าวีซีดี เป็นระบบจัดการบริหารร้านเช่าวีซีดีครบวงจร เช่น การรับสมัครสมาชิกร้าน การเพิ่มวีซีดีใหม่ การเช่าวีซีดี และการคืนวีซีดี เป็นต้น จากแผนภาพยูสเคสของระบบให้บริการเช่าวีซีดีในรูปที่ 5.5 ประกอบด้วยยูสเคส 13 ยูสเคส และตัวอย่างรายละเอียดยูสเคสของระบบให้บริการเช่าวีซีดีในรูปแบบไฟล์เอ็กซ์เอ็มแอลแสดงดังรูปที่ 5.6 สำหรับรายละเอียดยูสเคสของระบบให้บริการเช่าวีซีดีแสดงในภาคผนวก ข



รูปที่ 5.5 แผนภาพยูสเคสของระบบให้บริการเช่าวีซีดี



```

<!-- ***** vcd_rental [Model] ***** -->
+ <UML:Model xmi.id="G.0" name="vcd_rental" visibility="public" isSpecification="false" isRoot="false" isLeaf="false"
  isAbstract="false">
  <UML:TaggedValue xmi.id="XX.11.140.23.2" tag="RationalRose:Tool#0" value="myUseCase" modelElement="G.0" />
  <UML:TaggedValue xmi.id="XX.11.140.23.3" tag="RationalRose$myUseCase:usecase_no" value="1"
    modelElement="S.254.1400.21.16" />
  <UML:TaggedValue xmi.id="XX.11.140.23.4" tag="RationalRose$myUseCase:Pre-condition" value="Staff starts sign up new
    member function." modelElement="S.254.1400.21.16" />
  <UML:TaggedValue xmi.id="XX.11.140.23.5" tag="RationalRose$myUseCase:Post-condition" value="[0] New member s
    information is added and print member card." modelElement="S.254.1400.21.16" />
- <UML:TaggedValue xmi.id="XX.11.140.23.6" tag="RationalRose$myUseCase:SuccessScenario"
  modelElement="S.254.1400.21.16">
  <UML:TaggedValue.value>[0] (name <> "" ) && (address <> "" ) && (tel.length == 9) @ [1] Staff entry new member s
    information. | [2] System add new information into database.</UML:TaggedValue.value>
</UML:TaggedValue>
  <UML:TaggedValue xmi.id="XX.11.140.23.7" tag="RationalRose$myUseCase:AlternativeScenario" value=""
    modelElement="S.254.1400.21.16" />
  <UML:TaggedValue xmi.id="XX.11.140.23.8" tag="RationalRose$myUseCase:IsAbstract" value="0"
    modelElement="S.254.1400.21.16" />
  <UML:TaggedValue xmi.id="XX.11.140.23.9" tag="RationalRose$myUseCase:usecase_no" value="2"
    modelElement="S.254.1400.21.17" />
  <UML:TaggedValue xmi.id="XX.11.140.23.10" tag="RationalRose$myUseCase:Pre-condition" value="Staff starts searched
    member function" modelElement="S.254.1400.21.17" />
  <UML:TaggedValue xmi.id="XX.11.140.23.11" tag="RationalRose$myUseCase:Post-condition" value="[0] System display
    member s information." modelElement="S.254.1400.21.17" />
- <UML:TaggedValue xmi.id="XX.11.140.23.12" tag="RationalRose$myUseCase:SuccessScenario"
  modelElement="S.254.1400.21.17">
  <UML:TaggedValue.value>[0] (name <> "" ) || (id <> "" ) @ [1] Staff entry condition to search member s information. |
    [2] System search information from database.</UML:TaggedValue.value>
</UML:TaggedValue>
  <UML:TaggedValue xmi.id="XX.11.140.23.13" tag="RationalRose$myUseCase:IsAbstract" value="0"
    modelElement="S.254.1400.21.17" />
- <UML:TaggedValue xmi.id="XX.11.140.23.14" tag="RationalRose$myUseCase:AlternativeScenario"
  modelElement="S.254.1400.21.17">
  <UML:TaggedValue.value>[2.1] n/a @ [2.1.1] {UC3}</UML:TaggedValue.value>
</UML:TaggedValue>

```

รูปที่ 5.6 ตัวอย่างรายละเอียดยูสเคสของระบบให้บริการเช่าวีซีดี

## 5.4 ผลการทดสอบ

จากการนำแผนภาพยูสเคสและรายละเอียดยูสเคสในรูปแบบไฟล์เอกซ์เอ็มแอลของแต่ละระบบในหัวข้อ 5.3 มาวิเคราะห์ยูสเคสและสร้างกรณีทดสอบด้วยเครื่องมือที่พัฒนาขึ้น ได้ผลการทดสอบดังนี้

### 5.4.1 ระบบสั่งซื้อสินค้า

ผลจากการวิเคราะห์ยูสเคสของระบบสั่งซื้อสินค้าด้วยเครื่องมือที่พัฒนาขึ้น พบว่าหลังจากรวมยูสเคสที่มีความสัมพันธ์กัน มียูสเคสที่สามารถสร้างกรณีทดสอบได้จำนวน 10 ยูสเคส และสร้างกรณีทดสอบได้จำนวน 17 กรณีทดสอบ ซึ่งกรณีทดสอบที่สร้างได้แสดงในภาคผนวก ค

ตารางที่ 5.1 ตารางสรุปยูสเคสของระบบสั่งซื้อสินค้าหลังจากวิเคราะห์ด้วยเครื่องมือ

หมายเลขยูสเคส	ชื่อยูสเคส	จำนวนกรณีทดสอบที่สร้างได้
1	Login	1
2	Place order	2

ตารางที่ 5.1 ตารางสรุปยูสเคสของระบบสั่งซื้อสินค้าหลังจากวิเคราะห์ด้วยเครื่องมือ (ต่อ)

หมายเลขยูสเคส	ชื่อยูสเคส	จำนวนกรณีทดสอบที่สร้างได้
5	Return product	2
7	Cancel order	3
8	Get status on order	2
9	Get catalog	1
10	Register compliant	1
11	Run sales report	1
12	Fill and ship order	3
13	Receive back-ordered item	1

#### 5.4.2 ระบบจำลองการซื้อขายหลักทรัพย์

ผลจากการวิเคราะห์ยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์ด้วยเครื่องมือที่พัฒนาขึ้น พบว่าหลังจากรวมยูสเคสที่มีความสัมพันธ์กัน มียูสเคสที่สามารถสร้างกรณีทดสอบได้จำนวน 4 ยูสเคส และสร้างกรณีทดสอบได้จำนวน 6 กรณีทดสอบ ซึ่งกรณีทดสอบที่สร้างได้แสดงในภาคผนวก ค

ตารางที่ 5.2 ตารางสรุปยูสเคสของระบบจำลองการซื้อขายหลักทรัพย์หลังจากวิเคราะห์ด้วยเครื่องมือ

หมายเลขยูสเคส	ชื่อยูสเคส	จำนวนกรณีทดสอบที่สร้างได้
1	Create session	1
3	Submit news session	1
4	Add register	1
5	Submit order	3

#### 5.4.3 ระบบให้บริการเช่าวีซีดี

ผลจากการวิเคราะห์ยูสเคสของระบบให้บริการเช่าวีซีดีด้วยเครื่องมือที่พัฒนาขึ้น พบว่าหลังจากรวมยูสเคสที่มีความสัมพันธ์กัน มียูสเคสที่สามารถสร้างกรณีทดสอบได้จำนวน 8 ยูสเคส และสร้างกรณีทดสอบได้จำนวน 13 กรณีทดสอบ ซึ่งกรณีทดสอบที่สร้างได้แสดงในภาคผนวก ค

ตารางที่ 5.3 ตารางสรุปยูสเคสของระบบให้บริการเช่าวีซีดีหลังจากวิเคราะห์ด้วยเครื่องมือ

หมายเลขยูสเคส	ชื่อยูสเคส	จำนวนกรณีทดสอบที่สร้างได้
1	Create member	1
2	Search member	2
4	Rent VCD	3
6	Return VCD	3
8	Add VCD	1
9	Change catalog VCD	1
10	Delete VCD	1
12	Print daily report	1

## 5.5 สรุปผลการทดสอบ

จากการทดสอบเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสด้วยแผนภาพยูสเคส และรายละเอียดยูสเคสทั้ง 3 ระบบ พบว่าเครื่องมือสามารถสร้างกรณีทดสอบครอบคลุมลำดับเหตุการณ์ที่เป็นไปได้ทั้งหมด และค่าของข้อมูลทดสอบที่สุ่มสร้างสอดคล้องกับข้อมูลนำเข้าและประโยคเงื่อนไขของลำดับเหตุการณ์

## บทที่ 6

### สรุปผลการวิจัยและข้อเสนอแนะ

#### 6.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้เสนอวิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส โดยขั้นตอนการสร้างกรณีทดสอบเริ่มจากการเตรียมแผนภาพยูสเคสและรายละเอียดยูสเคส แล้วนำมาวิเคราะห์หาความสัมพันธ์ระหว่างยูสเคสเพื่อรวมยูสเคสที่มีความสัมพันธ์ จากนั้นจัดสร้างกรณีทดสอบซึ่งประกอบด้วยข้อมูลทดสอบ และลำดับเหตุการณ์ที่คาดว่าจะเกิดขึ้นเมื่อทดสอบด้วยค่าของข้อมูลทดสอบ

ในงานวิจัยนี้เสนอรูปแบบของรายละเอียดยูสเคสสำหรับใช้สร้างกรณีทดสอบได้โดยอัตโนมัติ ที่ประกอบด้วยหมายเลขยูสเคส ชื่อยูสเคส ผู้ใช้งานระบบ ข้อมูลนำเข้า กิจกรรมที่ต้องปฏิบัติก่อนทำงานยูสเคส ประโยคเงื่อนไขของลำดับเหตุการณ์ ลำดับเหตุการณ์สำเร็จ ลำดับเหตุการณ์ทางเลือกอื่น และผลลัพธ์หลังจากการทำงานยูสเคส

เนื่องจากข้อมูลนำเข้าของยูสเคสในระบบงานทั่วไปมีหลายประเภท งานวิจัยนี้จึงกำหนดให้ชนิดของข้อมูลนำเข้าของยูสเคส 5 ชนิดคือข้อมูลชุดอักขระ ข้อมูลชนิดเลขจำนวนจริง ข้อมูลชนิดเลขจำนวนเต็ม ข้อมูลชนิดตรรกะ และข้อมูลชนิดวันที่ อีกทั้งอนุญาตให้กำหนดขอบเขตของข้อมูลนำเข้าในรูปแบบค่าสูงสุดและค่าต่ำสุดที่เป็นไปได้ หรือระบุประโยคเงื่อนไขแบบการเปรียบเทียบ และแบบการคำนวณทางคณิตศาสตร์

ในการวิเคราะห์ความสัมพันธ์ของยูสเคส เมื่อพบยูสเคสที่มีความสัมพันธ์กันทั้งแบบอินคลูดและแบบเอ็กซ์เทน งานวิจัยนี้เสนอวิธีการรวมยูสเคสที่มีความสัมพันธ์เป็นหนึ่งในยูสเคสทำให้ลำดับเหตุการณ์ของยูสเคสสมบูรณ์ทำให้กรณีทดสอบที่สร้างขึ้นสามารถนำไปใช้ทดสอบได้ถูกต้องยิ่งขึ้น

กรณีทดสอบที่สร้างขึ้นจากวิธีการของงานวิจัยนี้ ประกอบด้วยข้อมูลทดสอบ และลำดับเหตุการณ์ที่คาดว่าจะเกิดขึ้น ซึ่งจากการทดลองสร้างกรณีทดสอบโดยใช้เครื่องมือซึ่งพัฒนาตามวิธีการที่นำเสนอในงานวิจัยนี้ จะได้กรณีทดสอบในรูปแบบเอกสารเอ็กซ์เอ็มแอล ซึ่งผู้ใช้งานสามารถนำไปใช้ในการทดสอบระบบส่วนของการทดสอบฟังก์ชันได้ทันที

ผลจากการทดสอบเครื่องมือที่พัฒนาขึ้นด้วยแผนภาพยูสเคสและรายละเอียดยูสเคสของระบบ 3 ระบบ ผลลัพธ์แสดงว่าสามารถสร้างกรณีทดสอบได้อย่างรวดเร็ว และกรณีทดสอบที่สร้าง

ขึ้นถูกต้องตามวิธีการที่ออกแบบไว้ รวมทั้งครอบคลุมทุกลำดับเหตุการณ์ที่เป็นไปได้ของแผนภาพยูสเคส ดังนั้นเครื่องมือที่พัฒนาขึ้นจึงมีประโยชน์ในขั้นตอนการทดสอบของกระบวนการพัฒนาซอฟต์แวร์ เนื่องจากช่วยลดเวลาของกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุใน ส่วนของการเตรียมกรณีทดสอบสำหรับการทดสอบระบบ

## 6.2 ข้อจำกัดและแนวทางการวิจัยต่อ

6.2.1 ค่าของข้อมูลทดสอบที่สร้างขึ้นเป็นการสร้างโดยสุ่มตามขอบเขตที่กำหนด ซึ่งค่าของข้อมูลทดสอบอาจจะไม่สมจริงในทางปฏิบัติ ดังนั้นจึงควรมีวิธีการสร้างข้อมูลทดสอบที่ทำให้ค่าของข้อมูลทดสอบที่สมจริงยิ่งขึ้นเพื่อให้การทดสอบที่ได้คล้ายกับการปฏิบัติงานจริง

6.2.2 ลำดับเหตุการณ์ทางเลือกอื่นจะทำงานก็ต่อเมื่อไม่สามารถทำงานตามปกติของลำดับเหตุการณ์สำเร็จได้ ซึ่งงานวิจัยนี้ไม่อนุญาตให้ภายในยูสเคสมีการทำงานของลำดับเหตุการณ์ทางเลือกอื่นที่ซ้อนกัน ซึ่งในความเป็นจริงอาจเกิดเหตุการณ์เหล่านี้ ดังนั้นจึงควรเพิ่มให้มีการจัดการกับยูสเคสที่มีลำดับเหตุการณ์ทางเลือกอื่นซ้อนกัน

6.2.3 งานวิจัยนี้ไม่อนุญาตให้ความสัมพันธ์ระหว่างยูสเคสกับยูสเคสมีความสัมพันธ์ที่ซ้อนกันดังที่ได้อธิบายในบทที่ 3 ซึ่งในความเป็นจริงความสัมพันธ์ลักษณะนี้เกิดขึ้นได้ ดังนั้นจึงควรบริหารจัดการกับยูสเคสที่มีความสัมพันธ์กันเช่นนี้

6.2.4 ตามเอกสารข้อกำหนดของยูเอ็มแอล ยูสเคสที่มีความสัมพันธ์กันแบบเอ็กซ์เทน ต้องมีการกำหนดจุดอ้างอิงไว้ที่ยูสเคสหลักเพื่อให้ยูสเคสที่เอ็กซ์เทนอ้างอิงถึง ซึ่งในงานวิจัยนี้ไม่มีการกำหนดจุดอ้างอิงตามเอกสารข้อกำหนดของยูเอ็มแอล เนื่องจากรูปแบบของรายละเอียดยูสเคสและวิธีการสร้างกรณีทดสอบโดยอัตโนมัติที่งานวิจัยนี้นำเสนอไม่รองรับกับการกำหนดจุดอ้างอิง แต่งานวิจัยนี้ออกแบบจุดเชื่อมโยงซึ่งระบุในยูสเคสหลักเพื่ออ้างอิงไปถึงยูสเคสที่เอ็กซ์เทน ดังนั้นจึงควรมีรูปแบบรายละเอียดยูสเคสและวิธีการสร้างกรณีทดสอบที่รองรับกับการกำหนดจุดอ้างอิงตามเอกสารข้อกำหนดของยูเอ็มแอล

6.2.5 งานวิจัยนี้อนุญาตให้การทำงานของลำดับเหตุการณ์ทางเลือกอื่นเรียกยูสเคสอื่นทำงานแทนในลักษณะความสัมพันธ์แบบเอ็กซ์เทน และอนุญาตให้การทำงานของลำดับเหตุการณ์สำเร็จเรียกใช้งานยูสเคสในลักษณะความสัมพันธ์แบบอินคลูเดเท่านั้น ซึ่งในความเป็นจริงอาจมีการทำงานของลำดับเหตุการณ์ทางเลือกอื่นเรียกใช้งานยูสเคสในลักษณะความสัมพันธ์แบบอินคลูเด และการทำงานของลำดับเหตุการณ์สำเร็จเรียกยูสเคสอื่นทำงานแทนในลักษณะความสัมพันธ์แบบเอ็กซ์เทน ดังนั้นจึงควรมีวิธีที่แก้ไขข้อจำกัดที่กล่าวมา

## รายการอ้างอิง

1. Heumann, J., Generating Test Cases From Use Cases, June 2001, Available from: [http://www.therationaledge.com/content/jun\\_01/m\\_cases\\_jh.html](http://www.therationaledge.com/content/jun_01/m_cases_jh.html)
2. Impl Software, Inc., Object-Oriented Behavior Modeling and Software Testing, May 2003, Available from: <http://www.specstudio.com>
3. กิตติ ภัคดีวัฒนะกุล และกิตติพงษ์ กลมกล่อม, UML-วิเคราะห์และออกแบบระบบเชิงวัตถุ, กรุงเทพมหานคร: เคทีพี คอมพ์ แอนด์ คอนซัลท์, 2544.
4. Cockburn, A., Writing Effective Use cases, United States of America: Addison-Wesley, 2000.
5. Jorgensen, P. C., Software Testing: A Craftsman's approach, United States of America: CRC Press, LLC, 2002.
6. Pressman, R. S., Software Engineering: A practitioner's approach, Singapore: McGraw-Hill, 1992.
7. ชาวลิต จีระทีปติสุนทร, XML Step by Step ฉบับภาษาไทย, กรุงเทพมหานคร: DLS กรุงเทพฯ, 2543.
8. Bertolino, A. and Gnesi, S., Use Case-based Testing of Product Lines, ACM SIGSOFT Software Engineering Notes 28(5), September 2003.
9. Object Management Group, OMG Unified Modeling Language Specification, March 2003, Available from: <http://www.omg.org>
10. Schneider, G., and Winters, P.J., Applying Use Cases, Second Edition A Practical Guide. United States of America: Addison-Wesley:189-230, 2001.
11. นางสาว ปิยะดา ชาญยิ่งยงค์. การพัฒนาระบบจำลองการซื้อขายหลักทรัพย์โดยวิธีการเชิงวัตถุ. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2544.



ภาคผนวก

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ก  
พจนานุกรมข้อมูล

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



ตารางที่ ก-1 พจนานุกรมข้อมูลตาราง uc\_context

ชื่อข้อมูล	คำอธิบาย	ชนิดข้อมูล	ขนาดข้อมูล	หมายเหตุ	ตัวอย่างข้อมูล
usecase_id	หมายเลขยูสเคส	Text	50	PK	1
usecase_name	ชื่อยูสเคส	Text	200		Add contact
description	คำอธิบายยูสเคส	Memo	-		Add new contact
precondition	กิจกรรมที่ต้องปฏิบัติก่อน	Memo	-		Login to system

ตารางที่ ก-2 พจนานุกรมข้อมูลตาราง required\_item

ชื่อข้อมูล	คำอธิบาย	ชนิดข้อมูล	ขนาดข้อมูล	หมายเหตุ	ตัวอย่างข้อมูล
item_id	หมายเลขข้อมูลนำเข้า	Text	50	PK	254.1400.21.10
item_name	ชื่อข้อมูลนำเข้า	Text	50		Age
item_type	ชนิดของข้อมูลนำเข้า	Text	50		Integer.unsigned
item_size	ขนาดของข้อมูลนำเข้า	Integer	-		5
item_fixsize	จำกัดขนาดข้อมูลนำเข้า True = จำกัด False = ไม่จำกัด	Text	5		False
item_maxval	ค่าสูงสุดของข้อมูลนำเข้า	Memo	-		10000
item_minval	ค่าต่ำสุดของข้อมูลนำเข้า	Memo	-		0

ตารางที่ ก-3 พจนานุกรมข้อมูลตาราง uc\_item

ชื่อข้อมูล	คำอธิบาย	ชนิดข้อมูล	ขนาดข้อมูล	หมายเหตุ	ตัวอย่างข้อมูล
item_id	หมายเลขข้อมูลนำเข้า	Text	50	PK,FK	254.1400.21.10
usecase_id	หมายเลขยูสเคส	Text	50	PK,FK	1
description	คำอธิบายยูสเคส	Memo	-		Add new contact

ตารางที่ ก-4 พจนานุกรมข้อมูลตาราง condition

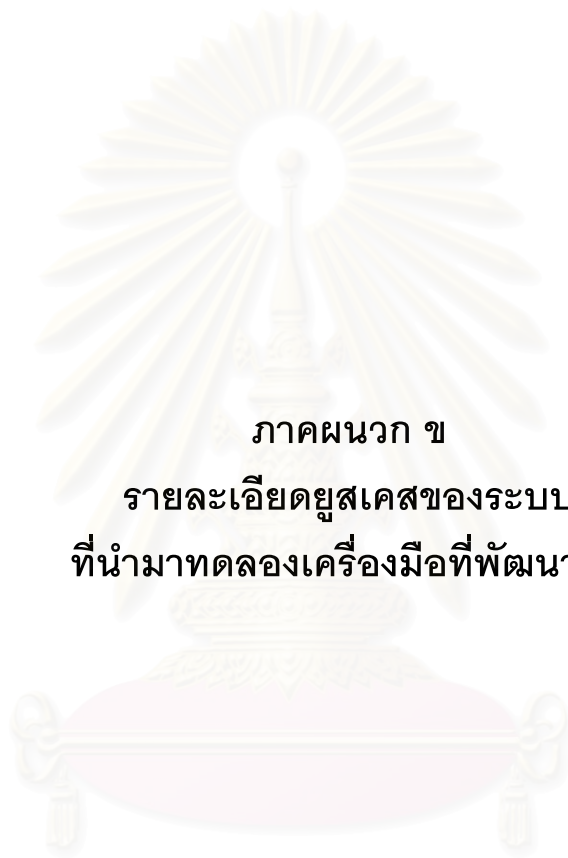
ชื่อข้อมูล	คำอธิบาย	ชนิดข้อมูล	ขนาดข้อมูล	หมายเหตุ	ตัวอย่างข้อมูล
usecase_id	หมายเลขยูสเคส	Text	50	PK,FK	1
condition_id	หมายเลขประโยคเงื่อนไข	Text	50	PK	0
expression	ประโยคเงื่อนไข	Memo	-		(age>10)
postcondition	ผลลัพธ์หลังทำงานยูสเคส	Memo	-		Show message

ตารางที่ ก-5 พจนานุกรมข้อมูลตาราง success

ชื่อข้อมูล	คำอธิบาย	ชนิดข้อมูล	ขนาดข้อมูล	หมายเหตุ	ตัวอย่างข้อมูล
usecase_id	หมายเลขยูสเคส	Text	50	PK,FK	1
condition_id	หมายเลขประโยคเงื่อนไข	Text	50	PK,FK	0
step_id	หมายเลขลำดับการทำงาน	Text	50	PK	2
actions	อธิบายการทำงาน	Memo	-		User enter info.

ตารางที่ ก-6 พจนานุกรมข้อมูลตาราง alternative

ชื่อข้อมูล	คำอธิบาย	ชนิดข้อมูล	ขนาดข้อมูล	หมายเหตุ	ตัวอย่างข้อมูล
usecase_id	หมายเลขยูสเคส	Text	50	PK,FK	1
condition_id	หมายเลขประโยคเงื่อนไข	Text	50	PK,FK	2.1
step_id	หมายเลขลำดับการทำงาน	Text	50	PK	2.1.1
actions	อธิบายการทำงาน	Memo	-		User enter info.



ภาคผนวก ข  
รายละเอียดคุณสมบัติของระบบ  
ที่นำมาทดลองเครื่องมือที่พัฒนาขึ้น

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## 1. ข้อกำหนดการเขียนรายละเอียดยูสเคส

รายละเอียดยูสเคสของระบบที่นำมาทดลองเครื่องมือสร้างกรณีทดสอบต้องเขียนเป็นข้อมูลเพิ่มเติมของแผนภาพยูสเคสที่เขียนด้วยโปรแกรมเรซินนอลโรส ซึ่งรายละเอียดยูสเคสต้องเขียนตามข้อกำหนดของงานวิจัยนี้ มีรายละเอียดดังนี้

- 1.1 usecase\_no ระบุหมายเลขของยูสเคสด้วยตัวเลขไม่ซ้ำกันตั้งแต่ 1
- 1.2 Pre-condition ระบุข้อความแสดงการทำงาน หรือกิจกรรม
- 1.3 IsAbstract ใส่เลข 0 เมื่อไม่เป็นยูสเคสต้นแบบ และใส่ 1 เมื่อเป็นยูสเคสต้นแบบ
- 1.4 SuccessScenario การใส่ลำดับเหตุการณ์สำเร็จมีรายละเอียดดังนี้
  - บรรทัดแรกของลำดับเหตุการณ์สำเร็จต้องเป็นประโยคเงื่อนไขเสมอ โดยระบุหมายเลขของประโยคเงื่อนไขเป็นเลข 0 ซึ่งอยู่ระหว่างเครื่องหมาย “[“ และ “]” ที่หน้าประโยคเงื่อนไข
  - ทำประโยคเงื่อนไขให้ใส่เครื่องหมาย “@” เพื่อแสดงว่าสิ้นสุดประโยคเงื่อนไข
  - หมายเลขลำดับการทำงานเป็นเลขไม่ซ้ำกันตั้งแต่ 1 เป็นต้นไป ซึ่งอยู่ระหว่างเครื่องหมาย “[“ และ “]”
  - ระบุหมายเลขลำดับการทำงานก่อนข้อความแสดงการทำงาน และใส่เครื่องหมาย “|” ที่ท้ายข้อความเพื่อแสดงว่าสิ้นสุดแต่ละการทำงาน
  - เว้นบรรทัดเพื่อแยกการทำงานแต่ละขั้นตอน
- 1.5 AlternativeScenario การใส่ลำดับเหตุการณ์ทางเลือกอื่นมีรายละเอียดดังนี้
  - บรรทัดแรกของแต่ละลำดับเหตุการณ์ทางเลือกอื่นต้องเป็นประโยคเงื่อนไขเสมอ โดยการให้หมายเลขของประโยคเงื่อนไขของลำดับเหตุการณ์ทางเลือกอื่นได้กล่าวไว้ในบทที่ 3 ซึ่งหมายเลขของประโยคเงื่อนไขจะอยู่ระหว่างเครื่องหมาย “[“ และ “]”
  - ทำประโยคเงื่อนไขให้ใส่เครื่องหมาย “@” เพื่อแสดงว่าสิ้นสุดประโยคเงื่อนไข
  - หมายเลขลำดับการทำงานให้ระบุอยู่ระหว่างเครื่องหมาย “[“ และ “]” ซึ่งการให้หมายเลขลำดับการทำงานได้กล่าวไว้ในบทที่ 3
  - ระบุหมายเลขลำดับการทำงานก่อนข้อความแสดงการทำงาน และใส่เครื่องหมาย “|” ที่ท้ายข้อความเพื่อแสดงว่าสิ้นสุดแต่ละการทำงาน
  - เว้นบรรทัดเพื่อแยกการทำงานแต่ละขั้นตอน

- กรณีที่มีการเรียกใช้ยูสเคสอื่น ซึ่งเป็นความสัมพันธ์แบบเอ็กซีเทนให้ระบุประโยคเงื่อนไขเป็น “n/a” เท่านั้น และใส่การทำงานเป็นจุดเชื่อมโยงไปยูสเคสที่ต้องการเรียกใช้
- หลังจากทำงานทางเลือกอื่นแล้วต้องการกลับไปทำงานในลำดับเหตุการณ์สำเร็จให้สร้างจุดเชื่อมโดยระบุ “{SS” แล้วตามด้วยหมายเลขลำดับการทำงานของลำดับเหตุการณ์สำเร็จที่ต้องการอ้างอิง แล้วปิดด้วยเครื่องหมาย “}”

#### 1.6 Post-condition มีรายละเอียดดังนี้

- ระบุผลลัพธ์หลังจากการทำงานของทุกลำดับเหตุการณ์ที่เป็นไปได้ของยูสเคส โดยหนึ่งบรรทัดแทนผลลัพธ์หลังการทำงานยูสเคสหนึ่งลำดับเหตุการณ์
- ระบุหมายเลขของผลลัพธ์หลังจากการทำงานของยูสเคสด้วยหมายเลขของประโยคเงื่อนไขของลำดับเหตุการณ์ โดยหมายเลขของผลลัพธ์ให้ระบุอยู่ระหว่างเครื่องหมาย “[“ และ “]”

#### 1.7 การระบุข้อมูลนำเข้าของยูสเคสมีรายละเอียดดังนี้

- โปรแกรมเรชันนอลโรสกำหนดให้ข้อมูลนำเข้าของยูสเคสต้องระบุที่ผู้ใช้งานระบบ
- Item\_name ให้ใส่ชื่อของข้อมูลนำเข้า
- Item\_type ให้เลือกชนิดข้อมูลนำเข้า ซึ่งมีให้เลือก 9 ชนิดดังนี้
  - String.All เป็นข้อมูลชุดอักขระที่เป็นได้ทั้งตัวอักษรและตัวเลข
  - String.Alphabet เป็นข้อมูลชุดอักขระที่เป็นตัวอักษรเท่านั้น
  - String.Numeric เป็นข้อมูลชุดอักขระที่เป็นตัวเลขเท่านั้น
  - Integer.Unsigned เป็นข้อมูลเลขจำนวนเต็มบวกเสมอ
  - Integer.Signed เป็นข้อมูลเลขจำนวนเต็ม ซึ่งอาจมีค่าเป็นบวกหรือลบ
  - Float.Unsigned เป็นข้อมูลเลขจำนวนจริงบวกเสมอ
  - Float.Signed เป็นข้อมูลเลขจำนวนจริง ซึ่งอาจมีค่าเป็นบวกหรือลบ
  - Boolean เป็นข้อมูลตรรกะ มีค่าเป็นค่าจริงหรือเท็จ
  - Date เป็นข้อมูลชนิดวันที่
- Item\_size ให้ใส่เลขระบุขนาดของข้อมูลนำเข้า ซึ่งมีความหมายดังนี้
  - ขนาดของข้อมูลชุดอักขระ เป็นการกำหนดความยาวสูงสุดข้อความที่สร้าง
  - ขนาดของข้อมูลเลขจำนวนเต็ม เป็นการกำหนดจำนวนหลักสูงสุดของเลขจำนวนเต็มที่สร้าง
  - ขนาดของข้อมูลเลขจำนวนจริง เป็นการกำหนดจำนวนหลักหลังจุดทศนิยมของเลขจำนวนจริงที่สร้าง

- ไม่ต้องระบุขนาดของข้อมูลตรรกะและข้อมูลวันที่
  - Item\_fixsize ให้เลือก True หรือ False โดยถ้าเป็น True คือระบุว่าต้องการสร้างข้อมูลทดสอบให้มีขนาดเท่ากับขนาดที่กำหนด แต่ถ้าเป็น False คือระบุว่าต้องการสร้างข้อมูลทดสอบให้มีขนาดไม่เกินขนาดที่กำหนด
    - Maxvalue ให้ระบุค่าสูงสุดที่เป็นไปได้ของข้อมูลนำเข้า
    - Minvalue ให้ระบุค่าต่ำสุดที่เป็นไปได้ของข้อมูลนำเข้า
    - usecaseno ให้ระบุหมายเลขของยูสเคสที่ต้องการข้อมูลนำเข้านี้ ถ้ามียูสเคสที่ต้องการข้อมูลนำเข้านี้มากกว่าหนึ่งยูสเคสให้คั่นหมายเลขของยูสเคสด้วยเครื่องหมายจุลภาค

หมายเหตุ: 1. ห้ามใส่เครื่องหมายอัฒภาคในข้อความใดๆ

2. การเรียกใช้งานยูสเคสอื่น ซึ่งมีความสัมพันธ์แบบอินคลูดต้องระบุในลำดับเหตุการณ์สำเร็จเท่านั้น

3. การเรียกใช้งานยูสเคสอื่น ซึ่งมีความสัมพันธ์แบบเอ็กซ์เทนต้องระบุในลำดับเหตุการณ์ทางเลือกอื่นเท่านั้น

## 2. รายละเอียดยูสเคสของระบบสั่งซื้อสินค้า

ระบบสั่งซื้อสินค้าประกอบด้วยรายละเอียดยูสเคส 13 ยูสเคส และมีข้อมูลนำเข้า 17 ตัว แสดงได้ดังตารางที่ ข-1 ถึง ข-17

ตารางที่ ข-1 รายละเอียดยูสเคสหมายเลข 1: Login

usecase_no	1
usecase_name	Login
IsAbstract	0
Pre-condition	The user start the application.
SuccessScenario	[0] (username <> "" ) && (password <> "" ) @ [1] The system display the login screen.   [2] The user enters a username and password.   [3] The system verifies the information.   [4] The system sets access permission.   [5] The system displays the Main Screen.
AlternativeScenario	
Post-condition	[0] The system displays the Main screen with the appropriate functions available for selection by the user.

ตารางที่ ข-2 รายละเอียดยูสเคสหมายเลข 2: Place Order

usecase_no	2
usecase_name	Place Order
IsAbstract	0
Pre-condition	Customer selects Place Order.
SuccessScenario	[0] (submitOrder == True) && (cancelOrder == False) && (customername <> "" ) && (prod_id <> "" ) @ [1] The system displays the Place Order screen.   [2] The customer enters his or her name and address.   [3] The customer enters product codes for products to be ordered.   [4] {UC3}   [5] The system adds the price of the item to the total.   [6] The customer enters credit card payment information.

ตารางที่ ข-2 รายละเอียดชุดทดสอบหมายเลข 2: Place Order (ต่อ)

	<p>[7] The customer select Submit.</p> <p>[8] The system verifies the information.  </p> <p>[9] The system sets the order status to pending.  </p> <p>[10] The system request that the order database store the order.  </p> <p>[11] The order database returns a unique order ID.  </p> <p>[12] The system returns the Order ID to the requestor.  </p> <p>[13] {UC4}  </p> <p>[14] The system marks the order confirmed.  </p> <p>[15] The system sends the changed fields to the database with the Order ID and request to update the order records.  </p> <p>[16] The database return a confirmation.  </p> <p>[17] The system returns an order ID to the customer.  </p>
AlternativeScenario	<p>[7.1] (cancelOrder == True) &amp;&amp; (submitOrder == False) @</p> <p>[7.1.1] The system discards any entered information.  </p> <p>[7.1.2] The system returns to the previous display.</p>
Post-condition	<p>[0] The system returns an order ID to the customer.</p> <p>[7.1] The system discards entered information and returns to the previous display.</p>

ตารางที่ ข-3 รายละเอียดชุดทดสอบหมายเลข 3: Get Product Information

usecase_no	3
usecase_name	Get Product Information
IsAbstract	0
Pre-condition	Product code is entered.
SuccessScenario	<p>[0] (prod_id &lt;&gt; "") @</p> <p>[1] The system sends a request to the inventory system for product information based on that product code.  </p> <p>[2] The inventory system returns the product information, which must include at lease product description, price, and stock on hand.  </p>
AlternativeScenario	
Post-condition	[0] The inventory system returns the product information.



ตารางที่ ข-4 รายละเอียดยูสเคสหมายเลข 4: Update Account

usecase_no	4
usecase_name	Update Account
IsAbstract	0
Pre-condition	A request to update an account is received.
SuccessScenario	[0] (credit_id.length == 16) && (credit_amount > 0) @ [1] The system sends credit card information and the amount of credit or debit to the accounting system.   [2] The accounting system sends a status of Okay.
AlternativeScenario	
Post-condition	[0] The accounting system sends a status of Okay.

ตารางที่ ข-5 รายละเอียดยูสเคสหมายเลข 5: Return Product

usecase_no	5
usecase_name	Return Product
IsAbstract	0
Pre-condition	The customer rep selects Return Product.
SuccessScenario	[0] (order_id <> "") && (prod_id <> "") @ [1] The Find Order screen is displayed.   [2] The user enters an order ID.   [3] The user selects search.   [4] The system queries the database to get the orders requested.   [5] The system returns the selected order.   [6] The customer rep selects the products to return.   [7] The customer rep selects Submit.   [8] {UC4}   [9] {UC6}   [10] The system sends the changed fields to the database with the Order ID and request to update the order records.   [11] The database return a confirmation.   [12] The system displays an acknowledgment.
AlternativeScenario	
Post-condition	[0] The system displays an acknowledgment.

ตารางที่ ข-6 รายละเอียดยูสเคสหมายเลข 6: Update Product Quantities

usecase_no	6
usecase_name	Update Product Quantities
IsAbstract	0
Pre-condition	A request to update product quantities is received.
SuccessScenario	[0] (prod_qnt > 0) @ [1] Product quantities is received.   [2] The system will send a request to the inventory system to add that amount to the stock on hand for a product.   [3] The inventory system sends an acknowledgment.
AlternativeScenario	[2.1] (prod_qnt <= 0) @ [2.1.1] The system sends a request to the inventory system to subtract that amount from the stock on hand for a product.   [2.1.2] The inventory system sends an acknowledgment.
Post-condition	[0] The inventory system sends an acknowledgment. [2.1] The inventory system sends an acknowledgment.

ตารางที่ ข-7 รายละเอียดยูสเคสหมายเลข 7: Cancel Order

usecase_no	7
usecase_name	Cancel Order
IsAbstract	0
Pre-condition	The customer selects Cancel Order.
SuccessScenario	[0] (order_id <> "") && (order_shipped == False) @ [1] The Find Order screen is displayed.   [2] The user enters an order ID.   [3] The user selects search.   [4] The system queries the database to get the orders requested.   [5] The system returns the selected order.   [6] The system displays the Cancel Order screen for that order.   [7] The customer select cancel.   [8] The system delete the order from the database.   [9] {UC4}   [10] {UC6}   [11] That order is canceled.

ตารางที่ ข-7 รายละเอียดชุดสเคสหมายเลข 7: Cancel Order (ต่อ)

AlternativeScenario	[7.1] (order_shipped == True) @ [7.1.1] The order has shipped.   [7.1.2] The system displays Return Policies screen.
Post-condition	[0] That order is canceled. [7.1] The system displays Return Policies screen.

ตารางที่ ข-8 รายละเอียดชุดสเคสหมายเลข 8: Get Status on Order

usecase_no	8
usecase_name	Get Status on Order
IsAbstract	0
Pre-condition	The user selects Review Order.
SuccessScenario	[0] (order_id <> "") @ [1] The Find Order screen is displayed.   [2] The user enters an order ID.   [3] The user selects search.   [4] The system queries the database to get the orders requested.   [5] The system returns the selected order.   [6] The system displays Get Status on Order screen.
AlternativeScenario	[2.1] (customername <> "") @ [2.1.1] The system displays a list of orders for that customer, including at least order ID and date of order.   [2.1.2] The user selects one order.   [2.1.3] {SS5}
Post-condition	[0] The system displays Get Status on Order screen. [2.1] The system displays Get Status on Order screen.

ตารางที่ ข-9 รายละเอียดชุดสเคสหมายเลข 9: Get Catalog

usecase_no	9
usecase_name	Get Catalog
IsAbstract	0
Pre-condition	The customer selects Get Catalog.

ตารางที่ ข-9 รายละเอียดยูสเคสหมายเลข 9: Get Catalog (ต่อ)

SuccessScenario	[0] (customername <> "") && (customeraddr <> "") @ [1] The Order Catalog screen is displayed.   [2] The user enters a name and address.   [3] The user select submit.   [4] The system creates an order for a catalog product and a total amount of zero.   [5] The system sets the order status to pending.   [6] The system request that the order database store the order.   [7] The order database returns a unique order ID.   [8] The system returns the Order ID to the requestor.
AlternativeScenario	
Post-condition	[0] The system save the order.

ตารางที่ ข-10 รายละเอียดยูสเคสหมายเลข 10: Register Complaint

usecase_no	10
usecase_name	Register Complaint
IsAbstract	0
Pre-condition	The user selects Contact Customer Service.
SuccessScenario	[0] (complaint_text <> "") @ [1] The Message screen is displayed.   [2] The user enters complaint text.   [3] The user select Submit.   [4] The system sends the text entered in an e-mail message to the customer support manger.
AlternativeScenario	
Post-condition	[0] The system send message in an e-mail message to customer support manager.

ตารางที่ ข-11 รายละเอียดยูสเคสหมายเลข 11: Run Sales Report

usecase_no	11
usecase_name	Run Sales Report
IsAbstract	0

ตารางที่ ข-11 รายละเอียดยูสเคสหมายเลข 11: Run Sales Report (ต่อ)

Pre-condition	The user selects Run Sales Report. (Report Total sales this month or this quarter)
SuccessScenario	[0] (report_type == 0)    (report_type == 1) @ [1] The Choose Report screen is displayed.   [2] The user selects a report.   [3] The user selects Submit.   [4] The system get sales data from database.   [5] The system formats the data into a report.   [6] The system displays the report.
AlternativeScenario	
Post-condition	[0] The system displays the report.

ตารางที่ ข-12 รายละเอียดยูสเคสหมายเลข 12: Fill and Ship Order

usecase_no	12
usecase_name	Fill and Ship Order
IsAbstract	0
Pre-condition	The clerk start the Fill and Ship Order Application
SuccessScenario	[0] (back_ordered == False) && (order_shipped == False) && (prod_stock > 0) @ [1] The system displays the Fill Order screen with a list of all confirmed orders.   [2] The clerk selects an order.   [3] The system displays the order.   [4] {UC3}   [5] The system displays stock on hand information for the item.   [6] {UC6}   [7] The system prints the item name and quantity ordered on a packing slip.   [8] The system marks the item shipped.   [9] The system marks the order complete.   [10] The system calculate postage due.   [11] The system prints a mailing label with the shipping address and postage due.

ตารางที่ ข-12 รายละเอียดของชุดทดสอบหมายเลข 12: Fill and Ship Order (ต่อ)

	[12] The system sends a notice to the shipping company that packages are ready to be picked up.
AlternativeScenario	[5.1] (prod_stock == 0) @ [5.1.1] The system marks the item back-ordered.   [5.1.2] The system sends a back-order request to the inventory system.   [5.1.3] The system marks the order Back-Orders.
Post-condition	[0] The system sends a notice to the shipping company that packages are ready to be picked up. [5.1] The system marks the order Back-Orders.

ตารางที่ ข-13 รายละเอียดของชุดทดสอบหมายเลข 13: Receive Back-Ordered Items

usecase_no	13
usecase_name	Receive Back-Ordered Items
IsAbstract	0
Pre-condition	The inventory system notifies the system that back-ordered items have been received.
SuccessScenario	[0] (back_ordered == True) @ [1] The system finds all order marked Back-Orders.   [2] The system clears the state of the item.   [3] The system changes the order status to confirmed.
AlternativeScenario	
Post-condition	[0] The system changes the order status to confirmed.

ตารางที่ ข-14 รายละเอียดของข้อมูลนำเข้าจาก Customer

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
username	String.All	8	True			1
password	String.All	8	False			1
customername	String.Alphabet	10	False			2,7,8,9
customeraddr	String.All	20	False			2,9
prod_id	String.Numeric	4	True			2,3
credit_id	String.Numeric	16	True			2,4,7

ตารางที่ ข-14 รายละเอียดของข้อมูลนำเข้าจาก Customer (ต่อ)

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
submitOrder	Boolean	-1	False			2
cancelOrder	Boolean	-1	False			2
credit_amount	Float.Unsigned	2	False			2,4,7
order_id	String.Numeric	5	True			7,8
order_shipped	Boolean	-1	False			7
complaint_text	String.All	25	False			10
prod_qnt	Integer.Unsigned	4	False	1000	0	4,6

ตารางที่ ข-15 รายละเอียดของข้อมูลนำเข้าจาก Customer Rep

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
customername	String.Alphabet	10	False			5
prod_id	String.Numeric	4	True			5
order_id	String.Numeric	5	True			5
report_type	Integer.Unsigned	1	True			11
credit_id	String.Numeric	16	True			5
credit_amount	Float.Unsigned	2	False			5
prod_qnt	Integer.Unsigned	4	False	1000	0	4,6

ตารางที่ ข-16 รายละเอียดของข้อมูลนำเข้าจาก Clerk

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
back_ordered	Boolean	-1	False			12
order_shipped	Boolean	-1	False			12
prod_stock	Integer.Unsigned	4	False	500	0	12
order_id	String.Numeric	5	True			12
prod_qnt	Integer.Unsigned	4	False	1000	0	4,6
prod_id	String.Numeric	4	True			12

ตารางที่ ข-17 รายละเอียดของข้อมูลนำเข้าจาก Inventory System

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
back_ordered	Boolean	-1	False			13

### 3. ระบบจำลองการซื้อขายหลักทรัพย์

ระบบจำลองการซื้อขายหลักทรัพย์ประกอบด้วยรายละเอียดยูสเคส 7 ยูสเคส และมีข้อมูลนำเข้า 24 ตัว แสดงได้ดังตารางที่ ข-18 ถึง ข-26

ตารางที่ ข-18 รายละเอียดยูสเคสหมายเลข 1: Create Session

usecase_no	1
usecase_name	Create Session
IsAbstract	0
Pre-condition	Session administrator register information into system.
SuccessScenario	[0] (ss_name <> "") && (ss_start < ss_end) && (market_status <= 2) && (ss_charge > 0) @ [1] Session administrator enter session s information.   [2] System create session id.   [3] System insert new session information into database.
AlternativeScenario	
Post-condition	[0] New session information is crated and saved in database.

ตารางที่ ข-19 รายละเอียดยูสเคสหมายเลข 3: Submit Session News

usecase_no	3
usecase_name	Submit Session News
IsAbstract	0
Pre-condition	System start sesion news page.
SuccessScenario	[0] (news_header <> "") && (news_detail <> "") && (news_effect >= 0) && (news_effect <= 100) @ [1] Session administrator enter news detail.   [2] System submit news and save into database.   [3] System send new news to newsmanager.



ตารางที่ ข-19 รายละเอียดยูสเคสหมายเลข 3: Submit Session News (ต่อ)

AlternativeScenario	
Post-condition	[0] System save new news detail and sent to newsmanager.

ตารางที่ ข-20 รายละเอียดยูสเคสหมายเลข 4: Add Register

usecase_no	4
usecase_name	Add Register
IsAbstract	0
Pre-condition	System start registered session s member page.
SuccessScenario	[0] (session_name <> "" ) @ [1] Investor select session name that want to register.   [2] System submit information from investor.   [3] System save information into database.   [4] System increase number of session s member.
AlternativeScenario	
Post-condition	[0] System update session member and increase number of member.

ตารางที่ ข-21 รายละเอียดยูสเคสหมายเลข 5: Submit Order

usecase_no	5
usecase_name	Submit Order
IsAbstract	0
Pre-condition	System start order page.
SuccessScenario	[0] (username <> "" ) && (passwd <> "" ) @ [1] {UC6} [2] {UC7}
AlternativeScenario	
Post-condition	[0] System submit order.

ตารางที่ ข-22 รายละเอียดยูสเคสหมายเลข 6: Check Bid

usecase_no	6
usecase_name	Check Bid
IsAbstract	0
Pre-condition	System received order.

ตารางที่ ข-22 รายละเอียดยูสเคสหมายเลข 6: Check Bid (ต่อ)

SuccessScenario	[0] (security_name <> "") @ [1] System submit order information from investor.   [2] System check order information.   [3] Oder information is correct.
AlternativeScenario	
Post-condition	[0] Oder information is correct.

ตารางที่ ข-23 รายละเอียดยูสเคสหมายเลข 7: Match Call Market

usecase_no	7
usecase_name	Match Call Market
IsAbstract	0
Pre-condition	Investor enter order information.
SuccessScenario	[0] (order_type == 0) && (order_unit > 0) && (order_cost > 0) && (isMatch == True) @ [1] System submit order type.   [2] System match order.   [3] Order that is matched be deal.   [4] Order is complete.
AlternativeScenario	[1.1] n/a @ [1.1.1] {UC8} [2.1] (isMatch == False) @ [2.1.1] Order is waited or canceled.   [2.1.2] Order is not complete.
Post-condition	[0] Order is submitted complete. [1.1] [2.1] Order is not complete.

ตารางที่ ข-24 รายละเอียดยูสเคสหมายเลข 8: Match ATO

usecase_no	8
usecase_name	Match ATO
IsAbstract	0
Pre-condition	Investor enter order information.

ตารางที่ ข-24 รายละเอียดของชุดทดสอบหมายเลข 8: Match ATO (ต่อ)

SuccessScenario	[0] (order_type == 1) && (order_unit > 0) && (order_cost > 0) @ [1] System match order.   [2] Order that is matched be deal.   [3] Order is complete.
AlternativeScenario	
Post-condition	[0] Order (ATO) is complete.

ตารางที่ ข-25 รายละเอียดของข้อมูลนำเข้าจาก Session\_Administrator

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
ss_name	String.Alphabet	10	False			1
ss_start	Date	-1	False			1
ss_end	Date	-1	False			1
market_status	Integer.Unsigned	1	True			1
ss_investor	Integer.Unsigned	5	False			1
inv_com1	Integer.Unsigned	2	False			1
inv_com2	Integer.Unsigned	2	False			1
inv_com3	Integer.Unsigned	2	False			1
inv_com4	Integer.Unsigned	2	False			1
auto_news	Boolean	-1	False			1
news_header	String.All	20	False			3
news_detail	String.All	100	False			3
news_status	Boolean	-1	False			3
news_effect	Integer.Unsigned	3	False			3
ss_charge	Integer.Unsigned	10	False			1

ตารางที่ ข-26 รายละเอียดของข้อมูลนำเข้าจาก Investor

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
session_name	String.Alphabet	10	False			4
security_name	String.Alphabet	2	True			6
order_type	Integer.Unsigned	1	True	3	0	6,7,8
order_cond	Integer.Unsigned	1	True			6

ตารางที่ ข-26 รายละเอียดของข้อมูลนำเข้าจาก Investor (ต่อ)

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
order_unit	Integer.Unsigned	5	False	9000	0	6,7,8
order_cost	Float.Unsigned	1	False	100000	500	6,7,8
username	Integer.Unsigned	5	True			5
passwd	String.All	8	False			5
isMatch	Boolean	-1	False			7

#### 4. ระบบให้บริการเช่าวีซีดี

ระบบให้บริการเช่าวีซีดีประกอบด้วยรายละเอียดยูสเคส 12 ยูสเคส และมีข้อมูลนำเข้า 10 ตัว แสดงได้ดังตารางที่ ข-27 ถึง ข-39

ตารางที่ ข-27 รายละเอียดยูสเคสหมายเลข 1: Create member

usecase_no	1
usecase_name	Create member
IsAbstract	0
Pre-condition	Staff starts sign up new member function.
SuccessScenario	[0] (name <> "" ) && (address <> "" ) && (tel.length == 9) @ [1] Staff entry new member s information.   [2] System add new information into database.
AlternativeScenario	
Post-condition	[0] New member s information is added and print member card.

ตารางที่ ข-28 รายละเอียดยูสเคสหมายเลข 2: Search member

usecase_no	2
usecase_name	Search member
IsAbstract	0
Pre-condition	Staff starts searched member function
SuccessScenario	[0] (name <> "" )    (id <> "" ) @ [1] Staff entry condition to search member s information.   [2] System search information from database.

ตารางที่ ข-28 รายละเอียดยูสเคสหมายเลข 2: Search member (ต่อ)

AlternativeScenario	[2.1] n/a @ [2.1.1] {UC3}
Post-condition	[0] System display member s information.

ตารางที่ ข-29 รายละเอียดยูสเคสหมายเลข 3: Edit member

usecase_no	3
usecase_name	Edit member
IsAbstract	0
Pre-condition	Staff starts edited member function..
SuccessScenario	[0] (name <> "" ) && (address <> "" ) && (tel.length == 9) @ [1] Staff entry new information.   [2] System update new information into database.
AlternativeScenario	
Post-condition	[0] System update member s information and status.

ตารางที่ ข-30 รายละเอียดยูสเคสหมายเลข 4: Rent VCD

usecase_no	4
usecase_name	Rent VCD
IsAbstract	0
Pre-condition	Staff start rental VCD.
SuccessScenario	[0] (id.length == 5) && (vcd_num < 5) && (vcd_id <> "" ) @ [1] Staff entry member id.   [2] {UC7}   [3] Staff entry rented vcd id into database.   [4] System save rental information into database.
AlternativeScenario	[3.1] (id.length == 5) && (vcd_num >= 5) @ [3.1.1] System show that "Cannot rent VCD because number of rented VCD is more than 5".   [3.2] n/a @ [3.2.1] {UC5}

ตารางที่ ข-30 รายละเอียดยูสเคสหมายเลข 4: Rent VCD (ต่อ)

Post-condition	[0] System save rental information. [3.1] System show that "Cannot rent VCD because number of rented VCD is more than 5".
----------------	--

ตารางที่ ข-31 รายละเอียดยูสเคสหมายเลข 5: Print detail

usecase_no	5
usecase_name	Print detail
IsAbstract	0
Pre-condition	System start printed detail function.
SuccessScenario	[0] (id.length == 5) @ [1] Sytem compute fee.   [2] System print bill.
AlternativeScenario	
Post-condition	[0] Symtem print bill.

ตารางที่ ข-32 รายละเอียดยูสเคสหมายเลข 6: Return VCD

usecase_no	6
usecase_name	Return VCD
IsAbstract	0
Pre-condition	Staff starts returned VCD function.
SuccessScenario	[0] (id.length == 5) && (vcd_id <> "") @ [1] Staff entry member id.   [2] {UC7} [3] Staff entry vcd id to return VCD.   [4] System save vcd id to return.
AlternativeScenario	[3.1] (id.length == 5) && (vcd_num > 0) @ [3.1.1] System show that "Member has VCD that not return".   [3.2] n/a @ [3.2.1] {UC5}
Post-condition	[0] System update returned VCD.

ตารางที่ ข-33 รายละเอียดยูสเคสหมายเลข 7: Print overdue report

usecase_no	7
usecase_name	Print overdue report
IsAbstract	0
Pre-condition	System starts overdue report function.
SuccessScenario	[0] (id.length == 5) @ [1] System search rental information by member id.   [2] System show overdue report.
AlternativeScenario	
Post-condition	[0] System show overdue report.

ตารางที่ ข-34 รายละเอียดยูสเคสหมายเลข 8: Add VCD

usecase_no	8
usecase_name	Add VCD
IsAbstract	0
Pre-condition	System starts add VCD.
SuccessScenario	[0] (vcd_name <> "") && (vcd_cat >= 1) && (vcd_cat <= 3) @ [1] Staff entry VCD s information into system.   [2] Staff select group of movie [3] Staff set category of rental.   [4] Staff set number of disk.   [5] System add new information into database.
AlternativeScenario	
Post-condition	[0] System add new VCD s information and generate VCD id.

ตารางที่ ข-35 รายละเอียดยูสเคสหมายเลข 9: Change category VCD

usecase_no	9
usecase_name	Change category VCD
IsAbstract	0
Pre-condition	Staff starts change category function.
SuccessScenario	[0] (vcd_id <> "") && (vcd_cat >= 1) && (vcd_cat <= 3) @ [1] Staff entry VCD id to search.   [2] {UC11}

ตารางที่ ข-35 รายละเอียดยูสเคสหมายเลข 9: Change category VCD (ต่อ)

	[3] System show VCD category.   [4] Staff set new VCD category.   [5] System update VCD category.
AlternativeScenario	
Post-condition	[0] System update VCD category.

ตารางที่ ข-36 รายละเอียดยูสเคสหมายเลข 10: Delete VCD

usecase_no	10
usecase_name	Delete VCD
IsAbstract	0
Pre-condition	Staff starts deleted VCD function.
SuccessScenario	[0] (vcd_id <> "") @ [1] Staff entry VCD id to delete.   [2] {UC11} [3] System show VCD information.   [4] System show message to confirm to delete VCD.   [5] System delete VCD information.
AlternativeScenario	
Post-condition	[0] System delete VCD information.

ตารางที่ ข-37 รายละเอียดยูสเคสหมายเลข 11: Search VCD

usecase_no	11
usecase_name	Search VCD
IsAbstract	0
Pre-condition	System starts searched VCD function.
SuccessScenario	[0] (vcd_id <> "") @ [1] System search VCD detail by VCD id.   [2] System show VCD detail.
AlternativeScenario	
Post-condition	[0] System show VCD detail.



ตารางที่ ข-38 รายละเอียดของยูสเคสหมายเลข 12: Print daily report

usecase_no	12
usecase_name	Print daily report
IsAbstract	0
Pre-condition	System start print daily report function.
SuccessScenario	[0] (vcd_date <> "") @ [1] Staff enter info into system.   [2] System print daily report.
AlternativeScenario	
Post-condition	[0] System print daily report via printer.

ตารางที่ ข-39 รายละเอียดของข้อมูลนำเข้าจาก Member-Staff

item_name	item_type	item_size	item_fixsize	max value	min value	usecaseno
name	String.Alphabet	30	False			1,2,3
address	String.All	50	False			1,3
tel	String.Numeric	9	True	010000 000	090000 000	1,3
id	String.Numeric	5	True			2,4,5,6,7
vcd_id	String.Numeric	5	True			4,6,9,10,11
vcd_num	Integer.Unsigned	1	False	9	0	4,6
vcd_name	String.All	10	False			8
vcd_cat	Integer.Unsigned	1	False	3	1	8,9
vcd_disknum	Integer.Unsigned	2	False			8
vcd_date	Date	-1	False			12



ภาคผนวก ค  
ตัวอย่างกรณีทดสอบ

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## 1. ระบบสั่งซื้อสินค้า

Test case id: 1.1

## Login

**Description:** Success Scenario: (username <> "" ) && (password <> "" )**Pre-condition:** The user start the application.**Input**

Name	Values
password	eqf7zd
username	6131qcx8

<b>Expected Output</b>	1. The system display the login screen.
	2. The user enters a username and password.
	3. The system verifies the information.
	4. The system sets access permission.
	5. The system displays the Main Screen.

**Post-condition** The system displays the Main screen with the appropriate functions available for selection by the user.

รูปที่ ค-1 กรณีทดสอบหมายเลข 1.1 ของระบบสั่งซื้อสินค้า

Test case id: 2.1

## Place Order

**Description:** Success Scenario: (submitOrder == True) && (cancelOrder == False) && (customername <> "" ) && (prod\_id <> "" ) && (credit\_id.length == 16) && (credit\_amount > 0) && (prod\_id <> "" )**Pre-condition:** Customer selects Place Order. A request to update an account is received. Product code is entered.**Input**

Name	Values
cancelOrder	False
credit_amount	151.45
credit_id	6322684411334921
customeraddr	d50jbfcsiwj7vo
customername	vddezj
prod_id	9971
prod_qnt	744
submitOrder	True

<b>Expected Output</b>	1. The system displays the Place Order screen.
	2. The customer enters his or her name and address.
	3. The customer enters product codes for products to be ordered.
	4. The system sends a request to the inventory system for product information based on that product code.
	5. The inventory system returns the product information, which must include at least product description, price, and stock on hand.
	6. The system adds the price of the item to the total.
	7. The customer enters credit card payment information.
	8. The customer select Submit.
	9. The system verifies the information.
	10. The system sets the order status to pending.
	11. The system request that the order database store the order.
	12. The order database returns a unique order ID.
	13. The system returns the Order ID to the requestor.
	14. The system sends credit card information and the amount of credit or debit to the accounting system.
	15. The accounting system sends a status of Okay.
	16. The system marks the order confirmed.
	17. The system sends the changed fields to the database with the Order ID and request to update the order records.
	18. The database return a confirmation.
	19. The system returns an order ID to the customer.

**Post-condition** The system returns an order ID to the customer. The accounting system sends a status of Okay. The inventory system returns the product information.

รูปที่ ค-2 กรณีทดสอบหมายเลข 2.1 ของระบบสั่งซื้อสินค้า

Test case id: 2.2

**Place Order****Description:** Alternative Scenario: (cancelOrder == True) && (submitOrder == False)**Pre-condition:** Customer selects Place Order. A request to update an account is received. Product code is entered.**Input**

Name	Values
cancelOrder	True
credit_amount	0.13
credit_id	6760642725368441
customeraddr	frdyym
customername	zezi
prod_id	4843
prod_qnt	680
submitOrder	False

**Expected Output**

1. The system displays the Place Order screen.
2. The customer enters his or her name and address.
3. The customer enters product codes for products to be ordered.
4. The system sends a request to the inventory system for product information based on that product code.
5. The inventory system returns the product information, which must include at least product description, price, and stock on hand.
6. The system adds the price of the item to the total.
7. The customer enters credit card payment information.
8. The customer select Submit.
9. The system discards any entered information.
10. The system returns to the previous display.
10. The system returns to the previous display.

**Post-condition**

The system discards entered information and returns to the previous display.

รูปที่ ค-3 กรณีทดสอบหมายเลข 2.2 ของระบบสั่งซื้อสินค้า

Test case id: 5.1

**Return Product****Description:** Success Scenario: (order\_id <> "") && (prod\_id <> "") && (credit\_id.length == 16) && (credit\_amount > 0) && (prod\_qnt > 0)**Pre-condition:** The customer rep selects Return Product. A request to update an account is received. A request to update product quantities is received.**Input**

Name	Values
credit_amount	97.19
credit_id	8927613084750233
customername	gr
order_id	30105
prod_id	6261
prod_qnt	342

**Expected Output**

1. The Find Order screen is displayed.
2. The user enters an order ID.
3. The user selects search.
4. The system queries the database to get the orders requested.
5. The system returns the selected order.
6. The customer rep selects the products to return.
7. The customer rep selects Submit.
8. The system sends credit card information and the amount of credit or debit to the accounting system.
9. The accounting system sends a status of Okay.
10. Product quantities is received.
11. The system will send a request to the inventory system to add that amount to the stock on hand for a product.
12. The inventory system sends an acknowledgment.
13. The system sends the changed fields to the database with the Order ID and request to update the order records.
14. The database return a confirmation.
15. The system displays an acknowledgment.

**Post-condition**

The system displays an acknowledgment. The accounting system sends a status of Okay. The inventory system sends an acknowledgment.

รูปที่ ค-4 กรณีทดสอบหมายเลข 5.1 ของระบบสั่งซื้อสินค้า

Test case id: 5.2

## Return Product

**Description:** Alternative Scenario: (prod\_qty <= 0)

**Pre-condition:** The customer rep selects Return Product. A request to update an account is received. A request to update product quantities is received.

### Input

Name	Values
credit_amount	584933111.72
credit_id	4269587508129312
customername	apsxnpluu
order_id	76829
prod_id	1044
prod_qty	0

### Expected Output

1. The Find Order screen is displayed.
2. The user enters an order ID.
3. The user selects search.
4. The system queries the database to get the orders requested.
5. The system returns the selected order.
6. The customer rep selects the products to return.
7. The customer rep selects Submit.
8. The system sends credit card information and the amount of credit or debit to the accounting system.
9. The accounting system sends a status of Okay.
10. Product quantities is received.
11. The system will send a request to the inventory system to add that amount to the stock on hand for a product.
11. The system will send a request to the inventory system to add that amount to the stock on hand for a product.
12. The system sends a request to the inventory system to subtract that amount from the stock on hand for a product.
13. The inventory system sends an acknowledgment.

**Post-condition** The inventory system sends an acknowledgment.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-5 กรณีทดสอบหมายเลข 5.2 ของระบบสั่งซื้อสินค้า

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Test case id: 7.1

## Cancel Order

**Description:** Success Scenario:: (order\_id <> "") && (order\_shipped == False) && (credit\_id.length == 16) && (credit\_amount > 0) && (prod\_qnt > 0)

**Pre-condition:** The customer selects Cancel Order. A request to update an account is received. A request to update product quantities is received.

### Input

Name	Values
credit_amount	51185.08
credit_id	6196833110400534
customername	bpwuzog
order_id	60038
order_shipped	False
prod_qnt	68

<b>Expected Output</b>	1. The Find Order screen is displayed.
	2. The user enters an order ID.
	3. The user selects search.
	4. The system queries the database to get the orders requested.
	5. The system returns the selected order.
	6. The system displays the Cancel Order screen for that order.
	7. The customer select cancel.
	8. The system delete the order from the database.
	9. The system sends credit card information and the amount of credit or debit to the accounting system.
	10. The accounting system sends a status of Okay.
	11. Product quantities is received.
	12. The system will send a request to the inventory system to add that amount to the stock on hand for a product.
	13. The inventory system sends an acknowledgment.
	14. That order is canceled.
<b>Post-condition</b>	That order is canceled. The accounting system sends a status of Okay. The inventory system sends an acknowledgment.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-6 กรณีทดสอบหมายเลข 7.1 ของระบบสั่งซื้อสินค้า

Test case id: 7.2

## Cancel Order

**Description:** Alternative Scenario:: (order\_shipped == True)

**Pre-condition:** The customer selects Cancel Order. A request to update an account is received. A request to update product quantities is received.

### Input

Name	Values
credit_amount	12.48
credit_id	8427386493750590
customername	lcfarsa
order_id	05650
order_shipped	True
prod_qnt	746

<b>Expected Output</b>	1. The Find Order screen is displayed.
	2. The user enters an order ID.
	3. The user selects search.
	4. The system queries the database to get the orders requested.
	5. The system returns the selected order.
	6. The system displays the Cancel Order screen for that order.
	7. The customer select cancel.
	8. The order has shipped.
	9. The system displays Return Policies screen.
<b>Post-condition</b>	The system displays Return Policies screen.

รูปที่ ค-7 กรณีทดสอบหมายเลข 7.2 ของระบบสั่งซื้อสินค้า

Test case id: 7.3

**Cancel Order****Description:** Alternative Scenario: (prod\_qty <= 0)**Pre-condition:** The customer selects Cancel Order. A request to update an account is received. A request to update product quantities is received.**Input**

Name	Values
credit_amount	0.76
credit_id	7113142570312422
customername	cy
order_id	90385
order_shipped	True
prod_qty	0

**Expected Output**

1. The Find Order screen is displayed.
2. The user enters an order ID.
3. The user selects search.
4. The system queries the database to get the orders requested.
5. The system returns the selected order.
6. The system displays the Cancel Order screen for that order.
7. The customer select cancel.
8. The system delete the order from the database.
9. The system sends credit card information and the amount of credit or debit to the accounting system.
10. The accounting system sends a status of Okay.
11. Product quantities is received.
12. The system will send a request to the Inventory system to add that amount to the stock on hand for a product.
13. The system sends a request to the inventory system to subtract that amount from the stock on hand for a product.
14. The inventory system sends an acknowledgment.

**Post-condition**

The inventory system sends an acknowledgment.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-8 กรณีทดสอบหมายเลข 7.3 ของระบบสั่งซื้อสินค้า

Test case id: 8.1

**Get Status on Order****Description:** Success Scenario: (order\_id <> "")**Pre-condition:** The user selects Review Order.**Input**

Name	Values
customername	h
order_id	99983

**Expected Output**

1. The Find Order screen is displayed.
2. The user enters an order ID.
3. The user selects search.
4. The system queries the database to get the orders requested.
5. The system returns the selected order.
6. The system displays Get Status on Order screen.

**Post-condition**

The system displays Get Status on Order screen.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-9 กรณีทดสอบหมายเลข 8.1 ของระบบสั่งซื้อสินค้า

Test case id: 8.2

### Get Status on Order

**Description:** Alternative Scenario: (customername <> "")

**Pre-condition:** The user selects Review Order.

**Input**

Name	Values
customername	totajtsdw
order_id	34431

<b>Expected Output</b>	1. The Find Order screen is displayed.
	2. The user enters an order ID.
	3. The system displays a list of orders for that customer, including at least order ID and date of order.
	4. The user selects one order.
	5. The system returns the selected order.
	6. The system displays Get Status on Order screen.

**Post-condition** The system displays Get Status on Order screen.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-10 กรณีทดสอบหมายเลข 8.2 ของระบบสั่งซื้อสินค้า

Test case id: 9.1

### Get Catalog

**Description:** Success Scenario: (customername <> "") && (customeraddr <> "")

**Pre-condition:** The customer selects Get Catalog.

**Input**

Name	Values
customeraddr	axcb
customername	f

<b>Expected Output</b>	1. The Order Catalog screen is displayed.
	2. The user enters a name and address.
	3. The user select submit.
	4. The system creates an order for a catalog product and a total amount of zero.
	5. The system sets the order status to pending.
	6. The system request that the order database store the order.
	7. The order database returns a unique order ID.
	8. The system returns the Order ID to the requestor.

**Post-condition** The system save the order.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-11 กรณีทดสอบหมายเลข 9.1 ของระบบสั่งซื้อสินค้า



Test case id: 10.1

## Register Complaint

**Description:** Success Scenario: (complaint\_text <> "")

**Pre-condition:** The user selects Contact Customer Service.

### Input

Name	Values
complaint_text	1167a3

<b>Expected Output</b>	1. The Message screen is displayed.
	2. The user enters complaint text.
	3. The user select Submit.
	4. The system sends the text entered in an e-mail message to the customer support manger.
<b>Post-condition</b>	The system send message in an e-mail message to customer support manager.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-12 กรณีทดสอบหมายเลข 10.1 ของระบบสั่งซื้อสินค้า

Test case id: 11.1

## Run Sales Report

**Description:** Success Scenario: (report\_type == 0) || (report\_type == 1)

**Pre-condition:** The user selects Run Sales Report. (Report Total sales this month or this quarter)

### Input

Name	Values
report_type	1

<b>Expected Output</b>	1. The Choose Report screen is displayed.
	2. The user selects a report.
	3. The user selects Submit.
	4. The system get sales data from database
	5. The system formats the data into a report.
	6. The system displays the report.
<b>Post-condition</b>	The system displays the report.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-13 กรณีทดสอบหมายเลข 11.1 ของระบบสั่งซื้อสินค้า

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Test case id: 12.1

## Fill and Ship Order

**Description:** Success Scenario: (back\_ordered == False) && (order\_shipped == False) && (prod\_stock > 0) && (prod\_qnt > 0) && (prod\_id <> "")

**Pre-condition:** The clerk start the Fill and Ship Order Application A request to update product quantities is received. Product code is entered.

### Input

Name	Values
back_ordered	False
order_id	30492
order_shipped	False
prod_id	9039
prod_qnt	54
prod_stock	311

<b>Expected Output</b>	1. The system displays the Fill Order screen with a list of all confirmed orders.
	2. The clerk selects an order.
	3. The system displays the order.
	4. The system sends a request to the inventory system for product information based on that product code.
	5. The inventory system returns the product information, which must include at lease product description, price, and stock on hand.
	6. The system displays stock on hand information for the item.
	7. Product quantities is received.
	8. The system will send a request to the inventory system to add that amount to the stock on hand for a product.
	9. The inventory system sends an acknowledgment.
	10. The system prints the item name and quantity ordered on a packing slip.
	11. The system marks the item shipped.
	12. The system marks the order complete.
	13. The system calculate postage due.
	14. The system prints a mailing label with the shipping address and postage due.
	15. The system sends a notice to the shipping company that packages are ready to be picked up.
<b>Post-condition</b>	The system sends a notice to the shipping company that packages are ready to be picked up. The inventory system sends an acknowledgment. The inventory system returns the product information.

รูปที่ ค-14 กรณีทดสอบหมายเลข 12.1 ของระบบสั่งซื้อสินค้า

Test case id: 12.2

## Fill and Ship Order

**Description:** Alternative Scenario: (prod\_stock == 0)

**Pre-condition:** The clerk start the Fill and Ship Order Application A request to update product quantities is received. Product code is entered.

### Input

Name	Values
back_ordered	False
order_id	39258
order_shipped	True
prod_id	7140
prod_qnt	645
prod_stock	0

<b>Expected Output</b>	1. The system displays the Fill Order screen with a list of all confirmed orders.
	2. The clerk selects an order.
	3. The system displays the order.
	4. The system sends a request to the inventory system for product information based on that product code.
	5. The inventory system returns the product information, which must include at lease product description, price, and stock on hand.
	6. The system displays stock on hand information for the item.
	7. The system marks the item back-ordered.
	8. The system sends a back-order request to the inventory system.
	9. The system marks the order Back-Orders.
<b>Post-condition</b>	The system marks the order Back-Orders.

รูปที่ ค-15 กรณีทดสอบหมายเลข 12.2 ของระบบสั่งซื้อสินค้า

Test case id: 12.3

## Fill and Ship Order

**Description:** Alternative Scenario: (prod\_qnt <= 0)

**Pre-condition:** The clerk start the Fill and Ship Order Application A request to update product quantities is received. Product code is entered.

### Input

Name	Values
back_ordered	True
order_id	34390
order_shipped	False
prod_id	3591
prod_qnt	0
prod_stock	457

<b>Expected Output</b>	1. The system displays the Fill Order screen with a list of all confirmed orders.
	2. The clerk selects an order.
	3. The system displays the order.
	4. The system sends a request to the inventory system for product information based on that product code.
	5. The inventory system returns the product information, which must include at least product description, price, and stock on hand.
	6. The system displays stock on hand information for the item.
	7. Product quantities is received.
	8. The system will send a request to the inventory system to add that amount to the stock on hand for a product.
	9. The system sends a request to the inventory system to subtract that amount from the stock on hand for a product.
	10. The inventory system sends an acknowledgment.
<b>Post-condition</b>	The inventory system sends an acknowledgment.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-16 กรณีทดสอบหมายเลข 12.3 ของระบบสั่งซื้อสินค้า

Test case id: 13.1

## Receive Back-Ordered Items

**Description:** Success Scenario: (back\_ordered == True)

**Pre-condition:** The inventory system notifies the system that back-ordered items have been received.

### Input

Name	Values
back_ordered	True

<b>Expected Output</b>	1. The system finds all order marked Back-Orders.
	2. The system clears the state of the item.
	3. The system changes the order status to confirmed.
<b>Post-condition</b>	The system changes the order status to confirmed.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-17 กรณีทดสอบหมายเลข 13.1 ของระบบสั่งซื้อสินค้า

## 2. ระบบจำลองการซื้อขายหลักทรัพย์

Test case id: 1.1	
Create Session	
<b>Description:</b>	Success Scenario:: (ss_name <> "") && (ss_start < ss_end) && (market_status <= 2) && (ss_charge > 0)
<b>Pre-condition:</b>	Session administrator register information into system.
<b>Input</b>	
Name	Values
auto_news	False
inv_com1	78
inv_com2	6
inv_com3	6
inv_com4	5
market_status	2
ss_charge	4
ss_end	19 Nov 3470
ss_investor	1
ss_name	djshaey
ss_start	03 Feb 2414
<b>Expected Output</b>	1. Session administrator enter session s information. 2. System create session id. 3. System insert new session information into database.
<b>Post-condition</b>	New session information is crated and saved in database.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-18 กรณีทดสอบหมายเลข 1.1 ของระบบจำลองการซื้อขายหลักทรัพย์

Test case id: 3.1	
Submit Session News	
<b>Description:</b>	Success Scenario:: (news_header <> "") && (news_detail <> "") && (news_effect >= 0) && (news_effect <= 100)
<b>Pre-condition:</b>	System start sesion news page.
<b>Input</b>	
Name	Values
news_detail	8noks39yxg9tyobw9bjsv5bl28mf2j
news_effect	1
news_header	o38ydj0ce3t2h3z8cxuk
news_status	True
<b>Expected Output</b>	1. Session administrator enter news detail. 2. System submit news and save into database. 3. System send new news to newsmanager.
<b>Post-condition</b>	System save new news detail and sent to newsmanager.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-19 กรณีทดสอบหมายเลข 3.1 ของระบบจำลองการซื้อขายหลักทรัพย์

Test case id: 4.1

### Add register

**Description:** Success Scenario: (session\_name <> "")

**Pre-condition:** System start registered session s member page.

#### Input

Name	Values
session_name	g

<b>Expected Output</b>	1. Investor select session name that want to register.
	2. System submit information from investor.
	3. System save information into database.
	4. System increase number of session s member.
<b>Post-condition</b>	System update session member and increase number of member.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-20 กรณีทดสอบหมายเลข 4.1 ของระบบจำลองการซื้อขายหลักทรัพย์

Test case id: 5.1

### Submit Order

**Description:** Success Scenario: (username <> "") && (passwd <> "") && (security\_name <> "") && (order\_type == 0) && (order\_unit > 0) && (order\_cost > 0) && (isMatch == True)

**Pre-condition:** System start order page. System received order. Investor enter order information.

#### Input

Name	Values
isMatch	True
order_cond	8
order_cost	62519.6
order_type	0
order_unit	8484
passwd	J
security_name	jw
username	96854

<b>Expected Output</b>	1. System submit order information from investor.
	2. System check order information.
	3. Oder information is correct.
	4. System submit order type.
	5. System match order.
	6. Order that is matched be deal.
	7. Order is complete.
<b>Post-condition</b>	System submit order. Oder information is correct. Order is submitted complete.

รูปที่ ค-21 กรณีทดสอบหมายเลข 5.1 ของระบบจำลองการซื้อขายหลักทรัพย์

จุฬาลงกรณ์มหาวิทยาลัย

Test case id: 5.2

### Submit Order

**Description:** Alternative Scenario: (order\_type == 1) && (order\_unit > 0) && (order\_cost > 0)

**Pre-condition:** System start order page. System received order. Investor enter order information.

#### Input

Name	Values
isMatch	False
order_cond	8
order_cost	37214.5
order_type	1
order_unit	1688
passwd	x
security_name	nx
username	66546

#### Expected Output

1. System submit order information from investor.
2. System check order information.
3. Oder information is correct.
4. System submit order type.
5. System match order.
6. Order that is matched be deal.
7. Order is complete.

**Post-condition** Order (ATO) is complete.

รูปที่ ค-22 กรณีทดสอบหมายเลข 5.2 ของระบบจำลองการซื้อขายหลักทรัพย์

Test case id: 5.3

### Submit Order

**Description:** Alternative Scenario: (isMatch == False)

**Pre-condition:** System start order page. System received order. Investor enter order information.

#### Input

Name	Values
isMatch	False
order_cond	5
order_cost	28012.6
order_type	1
order_unit	3308
passwd	-
security_name	oq
username	96854

#### Expected Output

1. System submit order information from investor.
2. System check order information.
3. Oder information is correct.
4. System submit order type.
5. System match order.
6. Order is waited or canceled.
7. Order is not complete.

**Post-condition** Order is not complete.

รูปที่ ค-23 กรณีทดสอบหมายเลข 5.3 ของระบบจำลองการซื้อขายหลักทรัพย์

## 3. ระบบให้บริการเช่าวีซีดี

Test case id: 1.1									
Create member									
<b>Description:</b>	Success Scenario: (name <> "") && (address <> "") && (tel.length == 9)								
<b>Pre-condition:</b>	Staff starts sign up new member function.								
<b>Input</b>									
	<table border="1"> <thead> <tr> <th>Name</th> <th>Values</th> </tr> </thead> <tbody> <tr> <td>address</td> <td>xytdakahttkife1azorfwdtbod</td> </tr> <tr> <td>name</td> <td>ztdeoyxrtbfzvjvqcbvhqrciln</td> </tr> <tr> <td>tel</td> <td>013827035</td> </tr> </tbody> </table>	Name	Values	address	xytdakahttkife1azorfwdtbod	name	ztdeoyxrtbfzvjvqcbvhqrciln	tel	013827035
Name	Values								
address	xytdakahttkife1azorfwdtbod								
name	ztdeoyxrtbfzvjvqcbvhqrciln								
tel	013827035								
<b>Expected Output</b>	1. Staff entry new member s information. 2. System add new information into database.								
<b>Post-condition</b>	New member s information is added and print member card.								

รูปที่ ค-24 กรณีสอดสอบหมายเลข 1.1 ของระบบให้บริการเช่าวีซีดี

Test case id: 2.1											
Search member											
<b>Description:</b>	Success Scenario: (name <> "")    (id <> "")										
<b>Pre-condition:</b>	Staff starts searched member function										
<b>Input</b>											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Values</th> </tr> </thead> <tbody> <tr> <td>address</td> <td>47ctwis</td> </tr> <tr> <td>id</td> <td>50415</td> </tr> <tr> <td>name</td> <td>-</td> </tr> <tr> <td>tel</td> <td>072457832</td> </tr> </tbody> </table>	Name	Values	address	47ctwis	id	50415	name	-	tel	072457832
Name	Values										
address	47ctwis										
id	50415										
name	-										
tel	072457832										
<b>Expected Output</b>	1. Staff entry condition to search member s information. 2. System search information from database.										
<b>Post-condition</b>	System display member s information.										

รูปที่ ค-25 กรณีสอดสอบหมายเลข 2.1 ของระบบให้บริการเช่าวีซีดี

Test case id: 2.2											
Search member											
<b>Description:</b>	Alternative Scenario: (name <> "") && (address <> "") && (tel.length == 9)										
<b>Pre-condition:</b>	Staff starts searched member function										
<b>Input</b>											
	<table border="1"> <thead> <tr> <th>Name</th> <th>Values</th> </tr> </thead> <tbody> <tr> <td>address</td> <td>zunyfy6zl2mpkdgxisavazfsuw22lrvhscjond10y0e</td> </tr> <tr> <td>id</td> <td>11201</td> </tr> <tr> <td>name</td> <td>maomfi</td> </tr> <tr> <td>tel</td> <td>075121554</td> </tr> </tbody> </table>	Name	Values	address	zunyfy6zl2mpkdgxisavazfsuw22lrvhscjond10y0e	id	11201	name	maomfi	tel	075121554
Name	Values										
address	zunyfy6zl2mpkdgxisavazfsuw22lrvhscjond10y0e										
id	11201										
name	maomfi										
tel	075121554										
<b>Expected Output</b>	1. Staff entry condition to search member s information. 2. System search information from database. 3. Staff entry new informaton. 4. System update new information into database.										
<b>Post-condition</b>	System update member s information and status.										

รูปที่ ค-26 กรณีสอดสอบหมายเลข 2.2 ของระบบให้บริการเช่าวีซีดี

Test case id: 4.1

**Rent VCD****Description:** Success Scenario: (id.length == 5) && (vcd\_num < 5) && (vcd\_id <> "") && (id.length == 5)**Pre-condition:** Staff start rental VCD. System starts overdue report function.**Input**

Name	Values
id	27400
vcd_id	72925
vcd_num	1

<b>Expected Output</b>	1. Staff entry member id.
	2. System search rental information by member id.
	3. System show overdue report.
	4. Staff entry rented vcd id into database.
	5. System save rental information into database.

**Post-condition** System save rental information. System show overdue report.

รูปที่ ค-27 กรณีทดสอบหมายเลข 4.1 ของระบบให้บริการเช่าวีซีดี

Test case id: 4.2

**Rent VCD****Description:** Alternative Scenario: (id length == 5) && (vcd\_num >= 5)**Pre-condition:** Staff start rental VCD. System starts overdue report function.**Input**

Name	Values
id	89499
vcd_id	26126
vcd_num	8

<b>Expected Output</b>	1. Staff entry member id.
	2. System search rental information by member id.
	3. System show overdue report.
	4. Staff entry rented vcd id into database.
	5. System show that "Cannot rent VCD because number of rented VCD is more than 5".

**Post-condition** System show that "Cannot rent VCD because number of rented VCD is more than 5".

รูปที่ ค-28 กรณีทดสอบหมายเลข 4.2 ของระบบให้บริการเช่าวีซีดี

Test case id: 4.3

**Rent VCD****Description:** Alternative Scenario: (id.length == 5)**Pre-condition:** Staff start rental VCD. System starts overdue report function.**Input**

Name	Values
id	91918
vcd_id	40055
vcd_num	8

<b>Expected Output</b>	1. Staff entry member id.
	2. System search rental information by member id.
	3. System show overdue report.
	4. Staff entry rented vcd id into database.
	5. System compute fee.
	6. System print bill.

**Post-condition** System print bill.

รูปที่ ค-29 กรณีทดสอบหมายเลข 4.3 ของระบบให้บริการเช่าวีซีดี



Test case id: 6.1

### Return VCD

**Description:** Success Scenario: (id.length == 5) && (vcd\_id <> "") && (id.length == 5)

**Pre-condition:** Staff starts returned VCD function. System starts overdue report function.

#### Input

Name	Values
id	59162
vcd_id	75493
vcd_num	7

<b>Expected Output</b>	1. Staff entry member id.
	2. System search rental information by member id.
	3. System show overdue report.
	4. Staff entry vcd id to return VCD.
	5. System save vcd id to return.

**Post-condition** System update returned VCD. System show overdue report.

รูปที่ ค-30 กรณีทดสอบหมายเลข 6.1 ของระบบให้บริการเช่าวีซีดี

Test case id: 6.2

### Return VCD

**Description:** Alternative Scenario: (id.length == 5) && (vcd\_num > 0)

**Pre-condition:** Staff starts returned VCD function. System starts overdue report function.

#### Input

Name	Values
id	94531
vcd_id	36390
vcd_num	6

<b>Expected Output</b>	1. Staff entry member id.
	2. System search rental information by member id.
	3. System show overdue report.
	4. Staff entry vcd id to return VCD.
	5. System show that "Member has VCD that not return".

**Post-condition**

รูปที่ ค-31 กรณีทดสอบหมายเลข 6.2 ของระบบให้บริการเช่าวีซีดี

Test case id: 6.3

### Return VCD

**Description:** Alternative Scenario: (id.length == 5)

**Pre-condition:** Staff starts returned VCD function. System starts overdue report function.

#### Input

Name	Values
id	52705
vcd_id	71748
vcd_num	5

<b>Expected Output</b>	1. Staff entry member id.
	2. System search rental information by member id.
	3. System show overdue report.
	4. Staff entry vcd id to return VCD.
	5. System compute fee.
	6. System print bill.

**Post-condition** System print bill

รูปที่ ค-32 กรณีทดสอบหมายเลข 6.3 ของระบบให้บริการเช่าวีซีดี

Test case id: 8.1

### Add VCD

**Description:** Success Scenario: (vcd\_name <> "") && (vcd\_cat >= 1) && (vcd\_cat <= 3)

**Pre-condition:** System starts add VCD.

#### Input

Name	Values
vcd_cat	2
vcd_disknum	6
vcd_name	ogofmx

**Expected Output**

1. Staff entry VCD s information into system.
2. Staff select group of movie
3. Staff set category of rental.
4. Staff set number of disk.
5. System add new information into database.

**Post-condition** System add new VCD s information and generate VCD id.

รูปที่ ค-33 กรณีทดสอบหมายเลข 8.1 ของระบบให้บริการเช่าวีซีดี

Test case id: 9.1

### Change category VCD

**Description:** Success Scenario: (vcd\_id <> "") && (vcd\_cat >= 1) && (vcd\_cat <= 3) && (vcd\_id <> "")

**Pre-condition:** Staff starts change category function. System starts searched VCD function.

#### Input

Name	Values
vcd_cat	1
vcd_id	57093

**Expected Output**

1. Staff entry VCD id to search.
2. System search VCD detail by VCD id.
3. System show VCD detail.
4. System show VCD category.
5. Staff set new VCD category.
6. System update VCD category.

**Post-condition** System update VCD category. System show VCD detail.

รูปที่ ค-34 กรณีทดสอบหมายเลข 9.1 ของระบบให้บริการเช่าวีซีดี

Test case id: 10.1

### Delete VCD

**Description:** Success Scenario: (vcd\_id <> "") && (vcd\_id <> "")

**Pre-condition:** Staff starts deleted VCD function. System starts searched VCD function.

#### Input

Name	Values
vcd_id	35010

**Expected Output**

1. Staff entry VCD id to delete.
2. System search VCD detail by VCD id.
3. System show VCD detail.
4. System show VCD information.
5. System show message to confirm to delete VCD.
6. System delete VCD information.

**Post-condition** System delete VCD information. System show VCD detail.

รูปที่ ค-35 กรณีทดสอบหมายเลข 10.1 ของระบบให้บริการเช่าวีซีดี

Test case id: 12.1

## Print daily report

**Description:** Success Scenario: (vcd\_date <> "")

**Pre-condition:** System start print daily report function.

### Input

Name	Values
vcd_date	15 Jun 1978

**Expected Output**

1. Staff enter info into system.
2. System print daily report.

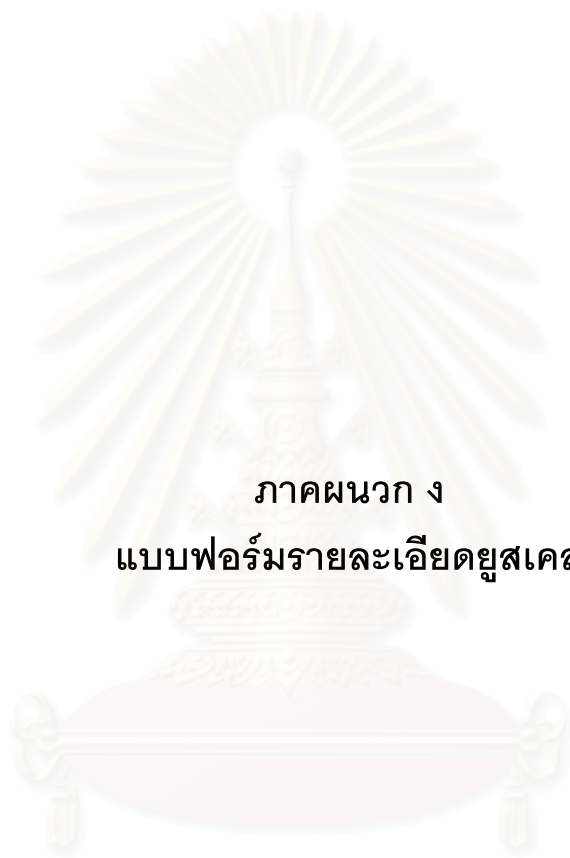
**Post-condition** System print daily report via printer.

Created by UseCase2TestCase (c) 2004 Chulalongkorn University

11/9/2547

รูปที่ ค-36 กรณีทดสอบหมายเลข 12.1 ของระบบให้บริการเช่าวีซีดี

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ง  
แบบฟอร์มรายละเอียดยศยศ

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

แบบฟอร์มรายละเอียดคุณสมบัติของแผนภาพยูสเคสที่นำมาสร้างกรณีทดสอบด้วยเครื่องมือที่พัฒนาขึ้น เป็นไฟล์สคริปต์เพื่อสร้างข้อมูลเพิ่มเติมของโปรแกรมเรขาคณิตแสดงได้ ดังรูปที่ ง-1

```
(object Petal
  version      45
  _written     "MyAddin v1.0"
  charSet      0)

(list Attribute_Set
  (object Attribute
    tool "myUseCase"
    name "propertyID"
    value "809135966")
  #
  # my Use Case Property
  #
  (object Attribute
    tool "myUseCase"
    name "default__UseCase"
    value (list Attribute_Set
      (object Attribute
        tool "myUseCase"
        name "usecase_no"
        value 0
      )
      (object Attribute
        tool "myUseCase"
        name "Pre-condition"
        value ""
      )
    )
  )
)
```

รูปที่ ง-1 สคริปต์ของแบบฟอร์มรายละเอียดคุณสมบัติ

```

        (object Attribute
            tool    "myUseCase"
            name    "Post-condition"
            value   ""
        )
        (object Attribute
            tool    "myUseCase"
            name    "SuccessScenario"
            value   ""
        )
        (object Attribute
            tool    "myUseCase"
            name    "AlternativeScenario"
            value   ""
        )
        (object Attribute
            tool    "myUseCase"
            name    "IsAbstract"
            value   0
        )
    )
)
#name    "default_Attribute"
(object Attribute
    tool    "myUseCase"
    name    "default_Attribute"
    value   (list Attribute_Set

```

รูปที่ ง-1 สคริปต์ของแบบฟอร์มรายละเอียดคุณสมบัติ (ต่อ)

```

(object Attribute
  tool    "myUseCase"
  name    "item_name"
  value   ""
)
(object Attribute
  tool    "myUseCase"
  name    "mytype"
  value   (list Attribute_Set
            (object Attribute
              tool    "myUseCase"
              name    "String.All"
              value   0
            )
            (object Attribute
              tool    "myUseCase"
              name    "String.Alphabet"
              value   1
            )
            (object Attribute
              tool    "myUseCase"
              name    "String.Numeric"
              value   2
            )
            (object Attribute
              tool    "myUseCase"
              name    "Integer.Unsigned"
              value   3
            )
          )
)

```

รูปที่ ง-1 สคริปต์ของแบบฟอร์มรายละเอียดคุณสมบัติ (ต่อ)

```

(object Attribute
    tool    "myUseCase"
    name    "Integer.Signed"
    value   4
)
(object Attribute
    tool    "myUseCase"
    name    "Float.Unsigned"
    value   5
)
(object Attribute
    tool    "myUseCase"
    name    "Float.Signed"
    value   6
)
(object Attribute
    tool    "myUseCase"
    name    "Boolean"
    value   7
)
(object Attribute
    tool    "myUseCase"
    name    "Date"
    value   8
)
)
)
(object Attribute
    tool    "myUseCase"
    name    "item_type"
    value   ("mytype" 0)
)

```

รูปที่ ง-1 สคริปต์ของแบบฟอร์มรายละเอียดคุณสมบัติ (ต่อ)



```

(object Attribute
  tool "myUseCase"
  name "item_size"
  value -1
)
(object Attribute
  tool "myUseCase"
  name "mybool"
  value (list Attribute_Set
    (object Attribute
      tool "myUseCase"
      name "False"
      value 0
    )
    (object Attribute
      tool "myUseCase"
      name "True"
      value 1
    )
  )
)
(object Attribute
  tool "myUseCase"
  name "item_fixsize"
  value ("mybool" 0)
)
(object Attribute
  tool "myUseCase"
  name "maxvalue"
  value ""
)

```

รูปที่ ง-1 สคริปต์ของแบบฟอร์มรายละเอียดคุณสมบัติ (ต่อ)


```

        (object Attribute
            tool    "myUseCase"
            name    "minvalue"
            value   ""
        )
        (object Attribute
            tool    "myUseCase"
            name    "usecaseno"
            value   "-1"
        )
    )
)

```

รูปที่ ง-1 สคริปต์ของแบบฟอร์มรายละเอียดยูสเคส (ต่อ)

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

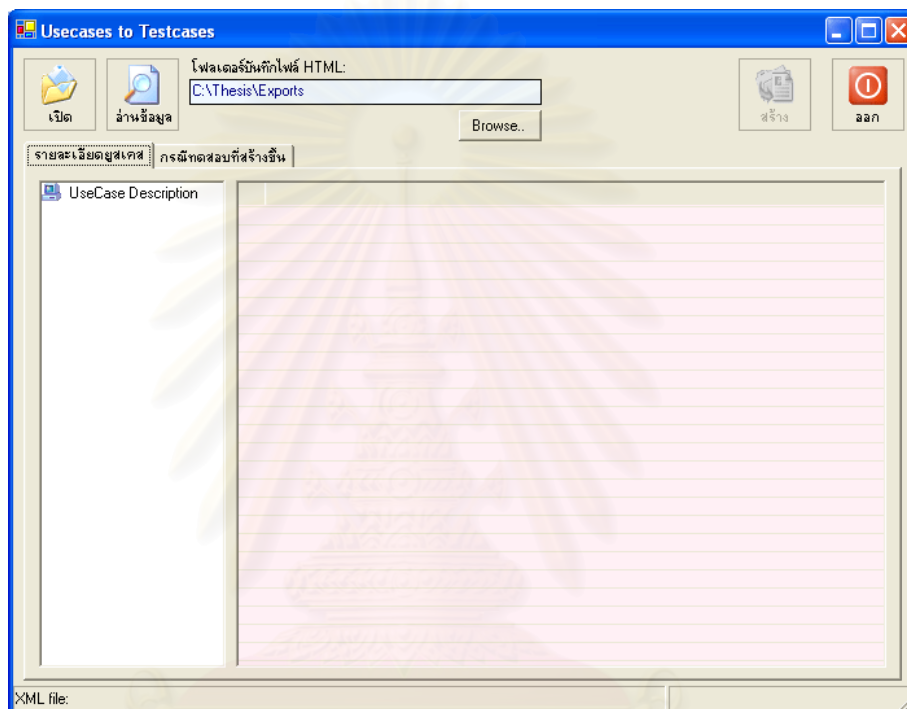


ภาคผนวก จ  
คู่มือการใช้งาน  
เครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส


สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

เครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสที่พัฒนาขึ้นตามวิธีการที่ได้  
นำเสนอในวิทยานิพนธ์นี้มีขั้นตอนการใช้งานดังนี้

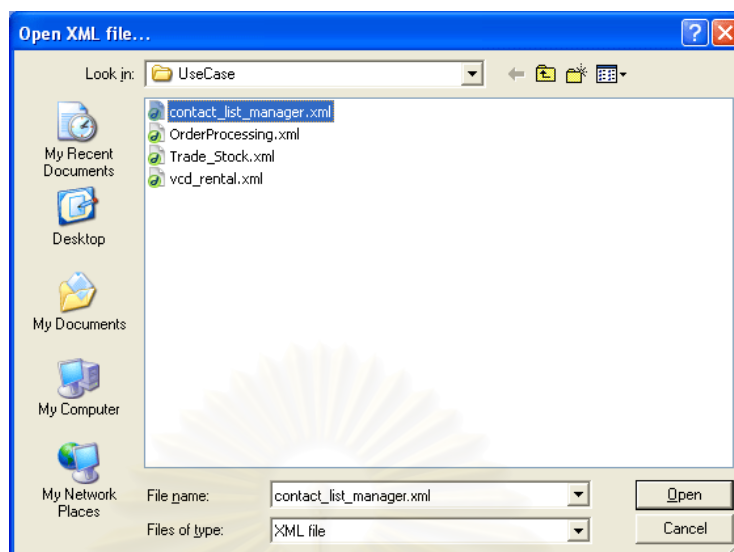
1. เมื่อเปิดเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสจะพบหน้าจอของเครื่องมือดังรูป  
ที่ ๑-1




รูปที่ ๑-1 หน้าจอของเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส

2. กดปุ่ม  เพื่อเลือกไฟล์เอ็กซ์เอ็มแอลซึ่งเก็บข้อมูลแผนภาพยูสเคสและรายละเอียดยูสเคสที่ต้องการสร้างกรณีทดสอบ แล้วจะปรากฏหน้าต่างเลือกไฟล์เอ็กซ์เอ็มแอลดังรูปที่ ๑-2

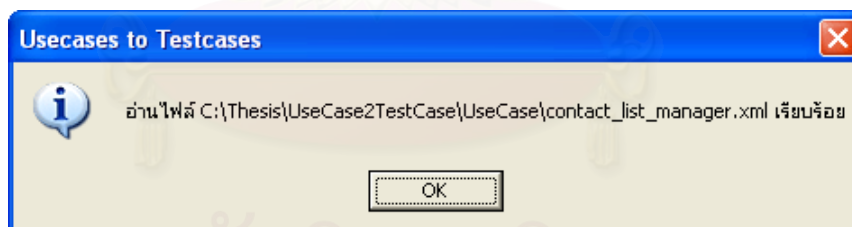
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ๑-2 หน้าต่างเลือกไฟล์เอ็กซ์เอ็มแอล

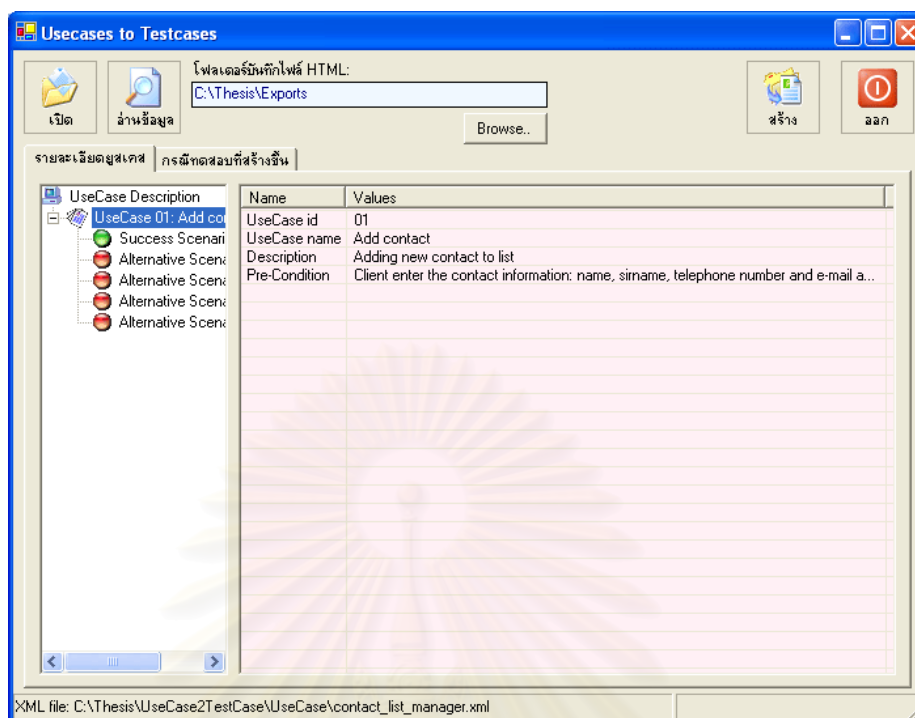
3. เมื่อเลือกไฟล์เอ็กซ์เอ็มแอลที่ต้องการแล้ว จากนั้นกดปุ่ม  เพื่อให้เครื่องมือเริ่มต้นอ่านไฟล์เอ็กซ์เอ็มแอล

4. เครื่องมือจะรายงานผลการอ่านไฟล์เอ็กซ์เอ็มแอลดังรูปที่ ๑-3 พร้อมทั้งแสดงรายละเอียดยูสเคสและลำดับเหตุการณ์ที่เป็นไปได้ทั้งหมดของยูสเคสในแผนภาพยูสเคสดังรูปที่ ๑-4



รูปที่ ๑-3 หน้าต่างรายงานผลการอ่านไฟล์เอ็กซ์เอ็มแอล

จุฬาลงกรณ์มหาวิทยาลัย

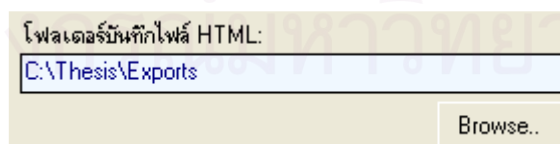


รูปที่ ๑-4 หน้าต่างแสดงรายละเอียดของยูสเคสและลำดับเหตุการณ์ที่เป็นไปได้ของยูสเคส

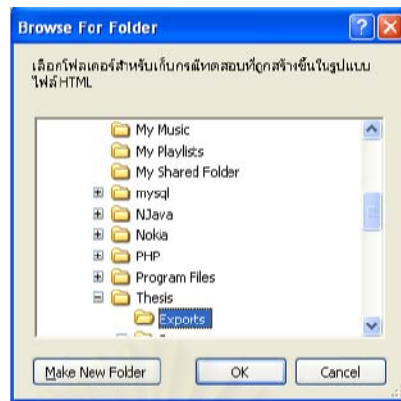
5. ผู้ใช้งานเครื่องมือสามารถดูรายละเอียดของยูสเคส หรือลำดับเหตุการณ์ของยูสเคสโดยดับเบิลคลิกที่ชื่อยูสเคส หรือลำดับเหตุการณ์ที่ต้องการ ซึ่งลำดับเหตุการณ์ของยูสเคสสามารถแบ่งได้เป็น 2 ประเภทคือ

- ลำดับเหตุการณ์สำเร็จของยูสเคส แทนด้วยสัญลักษณ์วงกลมสีเขียว
- ลำดับเหตุการณ์ทางเลือกอื่นของยูสเคส แทนด้วยสัญลักษณ์วงกลมสีแดง


6. จากนั้นจึงเริ่มต้นการสร้างกรณีทดสอบโดยเลือกสถานที่ที่จะเก็บไฟล์เอชทีเอ็มแอลที่จะถูกสร้างขึ้น โดยกดปุ่ม Browse... ซึ่งแสดงได้ดังรูปที่ ๑-5 แล้วจะปรากฏหน้าต่างเลือกสถานที่เก็บไฟล์เอชทีเอ็มแอลดังรูปที่ ๑-6

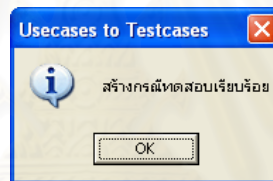


รูปที่ ๑-5 ส่วนที่ใช้กำหนดสถานที่เก็บไฟล์เอชทีเอ็มแอล

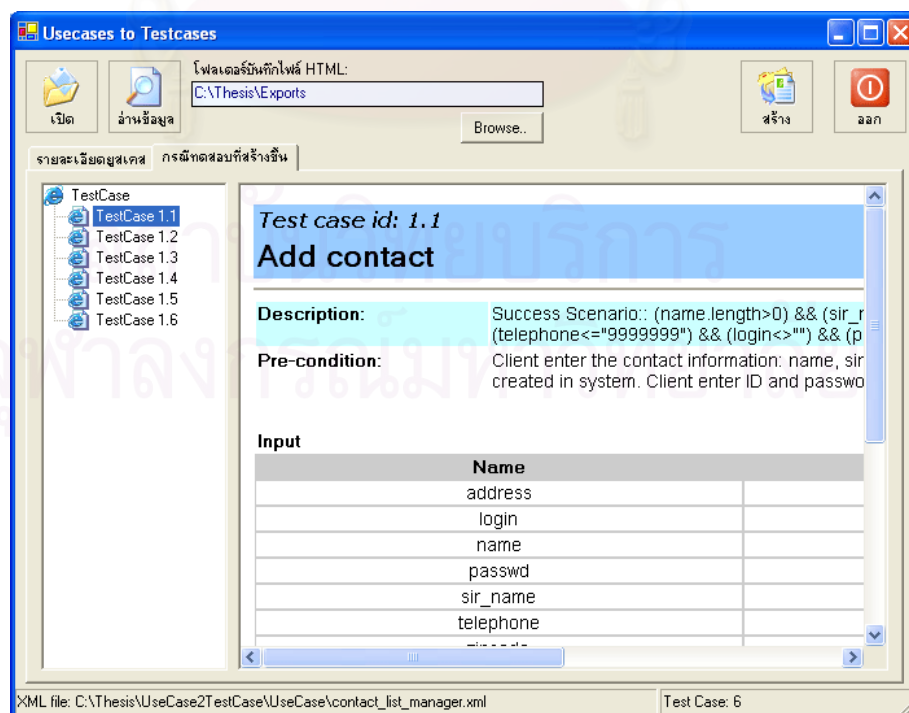


รูปที่ ๑-6 หน้าต่างเลือกสถานที่เก็บไฟล์เอชทีเอ็มแอล

7. หลังจากนั้นกดปุ่ม  เพื่อเริ่มต้นสร้างกรณีทดสอบ เมื่อเครื่องมือสร้างกรณีทดสอบเสร็จแล้วจะรายงานผลการสร้างกรณีทดสอบดังรูปที่ ๑-7 พร้อมทั้งรายชื่อกรณีทดสอบในแถบกรณีทดสอบที่สร้างขึ้นดังรูปที่ ๑-8



รูปที่ ๑-7 หน้าต่างรายงานผลการสร้างกรณีทดสอบ



รูปที่ ๑-8 หน้าต่างแสดงกรณีทดสอบที่สร้างขึ้น

8. ผู้ใช้สามารถดูกรณีทดสอบที่สร้างขึ้นได้โดยดับเบิลคลิกที่ชื่อกรณีทดสอบที่ต้องการ

9. กดปุ่ม  เพื่อออกจากโปรแกรม

หมายเหตุ: กรณีทดสอบที่สร้างขึ้นด้วยเครื่องมือนี้อยู่ในรูปแบบเอกสารเอชทีเอ็มแอลนั้น ถูกเก็บไว้  
สถานที่ที่ผู้ใช้งานเลือก ผู้ทดสอบสามารถนำไปใช้ในการทดสอบระบบได้ทันที



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## ประวัติผู้เขียนวิทยานิพนธ์

นายเศรษฐพงศ์ ลีพหัตถนรักษ์ เกิดวันที่ 15 มิถุนายน พ.ศ. 2521 ที่อำเภอเมือง จังหวัดนครปฐม สำเร็จการศึกษาระดับประถมศึกษาจากโรงเรียนวัดไผ่ล้อม จังหวัดนครปฐม เมื่อปีการศึกษา 2532 สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนสาธิตมหาวิทยาลัยศิลปากร จังหวัดนครปฐม เมื่อปีการศึกษา 2538 สำเร็จการศึกษาในหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาสถิติ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร จังหวัดนครปฐม เมื่อปีการศึกษา 2542 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2545 ที่อยู่ปัจจุบันที่สามารถติดต่อได้คือ บ้านเลขที่ 4/30 หมู่ 5 ตำบลบ่อพลับ อำเภอเมือง จังหวัดนครปฐม 73000 หมายเลขโทรศัพท์ +66 34 258037 อีเมลล์ lsetapong@yahoo.com



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย