การสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

TEST CASE GENERATION FROM ASPECT USE CASE DESCRIPTION

องอาจ กุศรี
ONG-ARJ KUSRI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2552
KMITL-2009-SC-M-002-020

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

TEST CASE GENERATION FROM ASPECT USE CASE DESCRIPTION

องอาจ คูศรี

ONG-ARJ KUSRI

เลขหมู่.....
เลขทะเบียน.....105311
วัน,เดือน,ปี.....18 พ.ย. 2552

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL-2009-SC-M-002-020

TEST CASE GENERATION FROM ASPECT USE CASE DESCRIPTION

ONG-ARJ KUSRI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2009

KMITL-2009-SC-M-002-020

COPYRIGHT 2009

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

Test Case Generation from Aspect Use Case Description

นักศึกษา นายองอาจ คูศรี

รหัสประจำตัว 47063702

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.ดร.ศรัณย์ อินทโกสุม

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.วีระ	บุญจริง	
ผศ.ดร.ศรัณย์	อินทโกสุม	
ผศ.ดร.จีรพร	วีระพันธุ์	
ดร.เฉลิมศักดิ์	เลิศวงศ์เสถียร	

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

วัน / เดือน/ ปี ที่สอบ 18 พฤษภาคม 2552 เวลา 17.00 น. เป็นต้นไป

สถานที่สอบ ณ อาคารจุฬาภรณวลัยลักษณ์ 1 ห้อง 217

คณะวิทยาศาสตร์รับรองแล้ว

(รองศาสตราจารย์ ดร.ธีรวัฒน์ มงคลอัศวรัตน์)

คณบดีคณะวิทยาศาสตร์

วันที่....25.....**เดือน**...พฤษภาคม.....**พ.ศ.**..2552....

หัวข้อวิทยานิพนธ์	การสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส
นักศึกษา	นายองอาจ คูศรี
รหัสประจำตัว	47063702
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ. ดร.ศรัณย์ อินทโกสุม

บทคัดย่อ

งานวิจัยนี้มีจุดมุ่งหมายเพื่อนำเสนอรูปแบบการเขียนยูสเคสสำหรับการพัฒนาซอฟต์แวร์เชิงแอสเปค และขั้นตอนการสร้างกรณีทดสอบจากรูปแบบยูสเคสดังกล่าว โดยรูปแบบคำอธิบายยูสเคสที่พัฒนาขึ้นนี้จะช่วยให้การแปลงจากข้อกำหนดความต้องการไปสู่การออกแบบและพัฒนาโปรแกรมในเชิงแอสเปคทำได้โดยสะดวก ส่วนขั้นตอนวิธีการสร้างกรณีทดสอบจากรูปแบบคำอธิบายยูสเคสใหม่นี้ จะช่วยให้การสร้างกรณีทดสอบสามารถเริ่มต้นทำได้ในระดับข้อกำหนดความต้องการ โดยทำได้อย่างเป็นระบบและง่าย นอกจากนี้ยังได้กรณีทดสอบที่ครอบคลุมทุกเส้นทางการทำงานตามที่เขียนอธิบายไว้ในยูสเคสอีกด้วย

Thesis	Test Case Generation from Aspect Use Case Description
Student	Mr. Ong-arj Kusri
Student ID.	47063702
Degree	Master of Science
Program	Computer Science
Year	2009
Thesis Advisor	Assoc.Prof.Dr.Sarun Intakosum

ABSTRACT

The purposes of this research are to present a new use case description format that supports the idea of aspect oriented software development and develop an algorithm to create test cases from the requirements defined in the proposed use case format. The conversion from requirement specification written in such format to aspect oriented design and implementation are easily performed. Moreover, test cases can be systematically and simply developed at the requirement level and cover all activity paths defined by the use cases.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้เป็นอย่างดี จากการเสียสละเวลาให้คำปรึกษา ชี้แนะแนวทางแก้ไขปัญหา ตลอดจนให้ความรู้ ประสบการณ์ และการดูแลเอาใจใส่อย่างใกล้ชิด จาก ผศ.ดร. ศรัณย์ อินทโกสุม ซึ่งเป็นอาจารย์ที่ปรึกษา ข้าพเจ้าจึงใคร่ขอกราบขอบพระคุณในความกรุณานี้อย่างสูงยิ่ง

ขอกราบขอบพระคุณ รศ.ดร.วีระ บุญจริง ผศ.ดร.จีรพร วีระพันธุ์ และดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร ประธานและคณะกรรมการสอบหัวข้อและโครงร่างวิทยานิพนธ์ที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะ จนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอกราบขอบพระคุณ คณาจารย์ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณาจารย์ภาควิชาคอมพิวเตอร์ มหาวิทยาลัยมหิดล และคณาจารย์ภาควิชาฟิสิกส์อุตสาหกรรมและอุปกรณ์การแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ทุกๆท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่างๆให้กับข้าพเจ้าสำหรับการเรียนในระดับปริญญาโท

ขอกราบขอบพระคุณทุนสนับสนุนการทำวิทยานิพนธ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้การสนับสนุนทุนการศึกษาแก่ข้าพเจ้า

ขอกราบขอบพระคุณ บิดา มารดา และบุคคลภายในครอบครัว ที่ให้การสนับสนุน เป็นกำลังใจในการเรียนเป็นอย่างดีเป็นผลให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

ขอขอบคุณ พี่กิ้ว พี่แก่น พี่จิ๊ก พี่สาว พี่ออย พี่อั๋น ที่ให้ความช่วยเหลือ กระตุ้นเตือนและกำลังใจที่ดีตลอดมา

สุดท้ายนี้ขอขอบคุณเพื่อนร่วมรุ่นอันได้แก่ นายทศนัย ชุ่มวัฒนะ นายพสิษฐ์ สสิกาญจน์ นายมารุต ณ พัทลุง นายโยธิน เทียนดี นายอัศวิน นิ่มกร นายอภินันทน์ เสริมศรี และรุ่นน้อง คือ นายกมล อมรยิ่งเจริญ ที่ให้ความช่วยเหลือ และกำลังใจที่ดีตลอดมา

สำหรับคุณงามความดีและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่บิดา มารดา ผู้มีพระคุณ และคณาจารย์ทุกท่านซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนญาติ พี่ น้อง และเพื่อนทุกท่าน

องอาจ คูศรี

พฤษภาคม 2552

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป IX

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 2

1.3 สมมติฐานของการศึกษา 2

1.4 ขอบเขตการวิจัย 2

1.5 ขั้นตอนการศึกษาและดำเนินงานวิจัย 3

1.6 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 การพัฒนาซอฟต์แวร์ด้วยแนวคิดเชิงแอสเปค 5

2.1.1 ครอสคัตติ้งคอนเซิน 5

2.1.2 แอสเปคเจ 8

2.1.2.1 กลไกการทำงานของแอสเปคเจ 9

2.1.2.2 รูปแบบภาษาของแอสเปคเจ 9

2.1.2.3 ตัวอย่างโปรแกรมที่พัฒนาด้วยแอสเปคเจ 11

2.2 ยูสเคส 13

2.2.1 แผนภาพยูสเคส 13

2.2.1.1 ยูสเคส 13

2.2.1.2 แอคเตอร์ 13

2.2.1.3 ความสัมพันธ์ระหว่างยูสเคส 14

2.2.2 คำอธิบายยูสเคส 16

2.3 ซีเนริโอ 22

2.4 แผนภาพกิจกรรม 23

2.4.1 ส่วนประกอบของแผนภาพกิจกรรม 23

สารบัญ (ต่อ)

หน้า

2.4.2 การใช้แผนภาพกิจกรรมแสดงการทำงานของแผนภาพยูสเคส.....25
2.5 การสร้างกรณีทดสอบ.....29
2.6 การทดสอบความครอบคลุมของกรณีทดสอบ.....38
2.6.1 การครอบคลุมคำสั่ง.....39
2.6.2 การครอบคลุมทางแยก.....40
2.6.3 การครอบคลุมเงื่อนไข.....42
2.6.4 การทดสอบเบซิสพาท.....43

บทที่ 3 การสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส.....49
3.1 รูปแบบคำอธิบายแอสเปคยูสเคส.....49
3.2 การเชื่อมต่อยูสเคสหลักกับแอสเปคยูสเคส.....51
3.3 ขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส.....53
3.3.1 สร้างคำอธิบายยูสเคสหลักและแอสเปคยูสเคส.....55
3.3.2 สร้างแผนภาพกิจกรรมจากข้อกำหนดยูสเคสหลักและแอสเปคยูสเคส...58
3.3.3 การเชื่อมต่อแผนภาพกิจกรรมของยูสเคสหลักและแอสเปคยูสเคส.....59
3.3.4 การค้นหาเส้นทางของกรณีทดสอบจากแผนภาพกิจกรรม.....61
3.3.5 สร้างซีแนริโอจากเส้นทางที่หาได้.....66
3.3.6 เขียนกรณีทดสอบ.....66

บทที่ 4 การประเมินผลงานวิจัย.....68
4.1 การออกแบบการประเมินผล.....68
4.2 การประเมินผล.....68
4.2.1 การประเมินความสามารถของรูปแบบคำอธิบายแอสเปคยูสเคส.....69
4.2.2 การประเมินผลขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส.....70
4.3 สรุปการประเมินผล.....75

บทที่ 5 สรุปและข้อเสนอแนะ.....77
5.1 สรุป.....77
5.2 ข้อเสนอแนะ.....78

สารบัญ (ต่อ)

หน้า

เอกสารอ้างอิง 79

ประวัติผู้เขียน 82

สารบัญตาราง

ตารางที่ หน้า

2.1 ตารางแสดงรูปแบบยูสเคสสำหรับการนำเสนอข้อมูลความต้องการของ Cockburn 16
2.2 ตารางแสดงรูปแบบยูสเคสแบบสองสดมภ์ของ Rebecca J. Wirfs-Brock 18
2.3 ตารางแสดงรูปแบบยูสเคสของ Geri Schneider และ Jason P. Winters 19
2.4 ตารางแสดงรูปแบบยูสเคสของ Javier J. Gutierrez และคณะ 22
2.5 ตารางแสดงคำอธิบายยูสเคสการตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน 22
2.6 คำอธิบายยูสเคสการโอนเงิน 25
2.7 คำอธิบายยูสเคสการตรวจสอบรหัสผ่าน 26
2.8 คำอธิบายยูสเคสการเรียกเก็บภาษี 27
2.9 คำอธิบายยูสเคสการแสดงใบเสร็จ 27
2.10 ซีแนริโอที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Dave Wood และ Jim Reis ... 30
2.11 กรณีทดสอบที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Dave Wood และ Jim Reis 30
2.12 ซีแนริโอที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Jim Heumann 31
2.13 กรณีทดสอบที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Jim Heumann 32
2.14 กรณีทดสอบที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Jim Heumann ซึ่งได้กำหนดข้อมูลจริงแล้ว 32
2.15 ซีแนริโอที่ได้จากแผนภาพกิจกรรมของยูสเคสการโอนเงินด้วยวิธีการของ Peter Zielezynski 33
2.16 กำหนดตัวแปรข้อมูลให้กับซีแนริโอที่ได้จากแผนภาพกิจกรรมของยูสเคสการโอนเงินด้วยวิธีการของ Peter Zielezynski 34
2.17 กรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีการของ Peter Zielezynski 34
2.18 ซีแนริโอที่ได้จากแผนภาพกิจกรรมของยูสเคสการโอนเงินด้วยวิธีการของ Javier J. Gutierrez และคณะ 36
2.19 กำหนดข้อมูลสำหรับซีแนริโอของยูสเคสการโอนเงินด้วยวิธีการของ Javier J. Gutierrez และคณะ 36
2.20 กรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีการของ Javier J. Gutierrez และคณะ 37
2.21 กำหนดผลลัพธ์ที่คาดว่าจะได้รับของแต่ละกรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีการของ Javier J. Gutierrez และคณะ 38
3.1 รูปแบบคำอธิบายแอสเปคยูสเคส 49
3.2 ตัวอย่างข้อกำหนดแอสเปคด้วยคำอธิบายแอสเปคยูสเคส 50

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

3.3 คำอธิบายยูสเคสการโอนเงิน 56
3.4 คำอธิบายยูสเคสการแสดงใบเสร็จ 56
3.5 คำอธิบายแอสเปคยูสเคสการตรวจสอบรหัสผ่าน 57
3.6 คำอธิบายแอสเปคยูสเคสการเรียกเก็บภาษี 57
3.7 แสดงคู่โหนดขั้นตอนการทำงานที่มีความเกี่ยวข้องกันจากแผนภาพกิจกรรม รูปที่ 3.9 61
3.8 แสดงเส้นทางที่สมบูรณ์ของคู่โหนดขั้นตอนการทำงานที่มีความเกี่ยวข้องกันจากตารางที่ 3.7 62
3.9 ซีแนริโอของยูสเคสการโอนเงินที่เชื่อมต่อกับแอสเปคยูสเคสการตรวจสอบรหัสผ่าน และแอสเปคยูสเคสการเรียกเก็บภาษี 66
3.10 แสดงการกำหนดกรณีทดสอบสำหรับยูสเคสการโอนเงินที่เชื่อมต่อกับแอสเปคยูสเคสการตรวจสอบรหัสผ่าน และแอสเปคยูสเคสการเรียกเก็บภาษี 67
3.11 การกำหนดข้อมูลสำหรับกรณีทดสอบ 67

สารบัญรูป

รูปที่ หน้า

2.1 แผนภาพคลาสของซอฟต์แวร์สำหรับวาดรูป 6
2.2 แสดงโค้ดของคลาส Line และคลาส Point 7
2.3 แสดงโค้ดของคลาส Line และคลาส Point ที่ได้เพิ่มการทำงานสำหรับปรับการแสดงของรูป 8
2.4 แสดงกลไกการทำงานของแอสเปคเจ 9
2.5 แสดงคลาส Line และคลาส Point 12
2.6 แสดงแอสเปค DisplayUpdating 12
2.7 แผนภาพยูสเคสระบบธนาคารอินเทอร์เน็ต 14
2.8 แผนภาพยูสเคสแสดงความสัมพันธ์แบบ Extension 15
2.9 แผนภาพยูสเคสแสดงความสัมพันธ์แบบ Generalization 16
2.10 สัญลักษณ์แสดงแทนกิจกรรม 23
2.11 สัญลักษณ์แสดงแทนจุดเริ่มต้น 24
2.12 สัญลักษณ์แสดงแทนจุดสิ้นสุด 24
2.13 สัญลักษณ์แสดงการเปลี่ยนสถานะ 24
2.14 สัญลักษณ์แสดงแทนทางเลือก 25
2.15 แผนภาพกิจกรรมของยูสเคสการโอนเงิน 28
2.16 แผนภาพกิจกรรมแสดงการทำงานของประโยคเงื่อนไข (If Statement) 39
2.17 แผนภาพกิจกรรมแสดงการทำงานของประโยคเงื่อนไข (If Statement) 40
2.18 แผนภาพกิจกรรมแสดงการทำงานของเมธอด maxPositive() 42
2.19 แผนภาพกิจกรรมแสดงการทำงานของเมธอด maxPositive2() 43
2.20 เส้นทางอิสระเชิงเส้น 1 ของแผนภาพกิจกรรมยูสเคสการโอนเงิน 45
2.21 เส้นทางอิสระเชิงเส้น 2 ของแผนภาพกิจกรรมยูสเคสการโอนเงิน 46
2.22 เส้นทางอิสระเชิงเส้น 3 ของแผนภาพกิจกรรมยูสเคสการโอนเงิน 47
3.1 แสดงการเชื่อมต่อแบบ Before 52
3.2 แสดงการเชื่อมต่อแบบ After 52
3.3 แสดงการเชื่อมต่อแบบ Around 53
3.4 ภาพรวมขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส 54
3.5 แผนภาพยูสเคสแสดงการเกิดครอสคัตติ้งคอนเซินในระบบธนาคารอินเทอร์เน็ต 55
3.6 แผนภาพกิจกรรมของยูสเคสการโอนเงิน 58
3.7 แผนภาพกิจกรรมของแอสเปคยูสเคสการตรวจสอบรหัสผ่าน 59

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.8 แผนภาพกิจกรรมแอสเปคยูสเคสการเรียกเก็บภาษี 59
3.9 แผนภาพกิจกรรมการเชื่อมต่อระหว่างยูสเคสการโอนเงิน แอสเปคยูสเคสการตรวจสอบรหัสผ่านและแอสเปคยูสเคสการเรียกเก็บภาษี 60
3.10 เส้นทางเชิงอิสระ 1 63
3.11 เส้นทางเชิงอิสระ 2 64
3.12 เส้นทางเชิงอิสระ 3 65
4.1 แสดงการระบุจุดเพื่อการเรียกใช้ยูสเคสย่อยของคำอธิบายยูสเคสการโอนเงินในกรณีศึกษาข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายยูสเคส 71
4.2 แสดงส่วนสำหรับการระบุถึงยูสเคสหลักของคำอธิบายแอสเปคยูสเคส 72
4.3 แสดงคำอธิบายยูสเคสการโอนเงินในกรณีศึกษาข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายยูสเคส 73

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทดสอบซอฟต์แวร์ (Software Testing) เป็นขั้นตอนหนึ่งในวงจรการพัฒนาซอฟต์แวร์ (Software Development Life Cycle) เพื่อช่วยให้ซอฟต์แวร์ที่พัฒนามีความถูกต้อง สมบูรณ์ ปลอดภัย และมีคุณภาพที่ดี ตลอดจนเป็นเครื่องมือในการรับประกันคุณภาพให้กับซอฟต์แวร์ที่พัฒนาด้วย [5][19][35] นอกจากนี้การทดสอบยังเป็นกลไกสำคัญในการป้องกันความล้มเหลวที่อาจเกิดขึ้นกับซอฟต์แวร์ได้ จากการศึกษาพบว่าเครื่องมือตลอดจนขั้นตอนวิธีการทดสอบโดยส่วนใหญ่ให้ความสนใจกับการทดสอบในขั้นตอนการโปรแกรมเป็นส่วนใหญ่ [4][10] ซึ่งแนวทางนี้อาจก่อให้เกิดความเสียหายอย่างยิ่งในโครงการซอฟต์แวร์ที่พัฒนาได้ โดยเฉพาะหากข้อผิดพลาดนั้นเกิดขึ้นมาก่อนแล้วตั้งแต่ขั้นตอนการวิเคราะห์และออกแบบซอฟต์แวร์ ข้อผิดพลาดดังกล่าวจะนำมาซึ่งระยะเวลา และค่าใช้จ่ายที่เพิ่มขึ้นจากการเริ่มต้นพัฒนาซอฟต์แวร์ใหม่ทั้งหมด แนวทางหนึ่งที่สามารถช่วยแก้ปัญหาได้คือ การเริ่มทำการทดสอบตั้งแต่ขั้นตอนการออกแบบและวิเคราะห์ข้อกำหนดความต้องการซึ่งจะลดความเสี่ยงต่อการเกิดข้อผิดพลาดในขั้นตอนอื่นๆ ของกระบวนการพัฒนาซอฟต์แวร์ได้ การทดสอบซอฟต์แวร์ในขั้นตอนการวิเคราะห์และออกแบบข้อกำหนดความต้องการนี้ประกอบด้วย 2 ส่วนหลัก คือ ข้อมูล (Data) และกรณีทดสอบ (Test Case) โดยกรณีทดสอบคือหลักสำคัญที่จะบ่งชี้ถึงความสามารถในการทำงานต่างๆ ที่เป็นไปได้ของซอฟต์แวร์ งานวิจัยจำนวนมากได้ถูกพัฒนาขึ้นเพื่อรองรับการสร้างกรณีทดสอบที่มีประสิทธิภาพ โดยเฉพาะการสร้างกรณีทดสอบจากความต้องการในรูปแบบของยูสเคส (Use Case) [8][17][19][20][23] ถือได้ว่าเป็นกลุ่มงานวิจัยที่ได้รับความสนใจและยอมรับมากที่สุด

สำหรับกระบวนการพัฒนาซอฟต์แวร์เชิงแอสเปค (Aspect-Oriented Software Development) ซึ่งเป็นกระบวนการพัฒนาซอฟต์แวร์ที่พัฒนาขึ้นเพื่อแก้ไขปัญหาการกระจาย (Scatter) และความเกี่ยวพัน (Tangle) ของเหตุการณ์ต่างๆ ที่เราสนใจในการทำงานของซอฟต์แวร์ หรือคอนเซิน (Concern) และทำการแยกส่วนคอนเซิน (Separation of Concern) ที่มีลักษณะปัญหาดังกล่าวออกเป็นมอดูล (Module) ใหม่ซึ่งเรียกว่าแอสเปค (Aspect) [32][33] กระบวนการทดสอบก็ยังคงเป็นกระบวนการหลักที่ใช้เพื่อรับประกันความสำเร็จเช่นเดียวกับวิธีการพัฒนาซอฟต์แวร์อื่นๆ ที่ผ่านมา ในปัจจุบันนี้มีงานวิจัยจำนวนมาก [7][9][16][28] ที่นำเสนอวิธีการทดสอบสำหรับซอฟต์แวร์เชิงแอสเปค อย่างไรก็ตามงานวิจัยโดยส่วนใหญ่จะเป็นการทดสอบในขั้นตอนหลังจาก

การพัฒนาซอฟต์แวร์เสร็จสิ้นแล้วมากกว่าการทดสอบในขั้นตอนอื่นๆ โดยเฉพาะขั้นตอนการวิเคราะห์และออกแบบข้อกำหนดความต้องการ

งานวิจัยนี้จึงมีวัตถุประสงค์เพื่อนำเสนอแนวทางที่เป็นประโยชน์ต่อกระบวนการทดสอบซอฟต์แวร์เชิงแอสเปคที่ขั้นตอนการวิเคราะห์และออกแบบข้อกำหนดความต้องการ โดยนำเสนอแนวทางการออกแบบวิธีการสร้างกรณีทดสอบที่มีความครอบคลุมจากข้อกำหนดความต้องการซึ่งถูกอธิบายในรูปแบบของคำอธิบายยูสเคส อนึ่ง เนื่องจากรูปแบบยูสเคสสำหรับการอธิบายข้อกำหนดความต้องการเชิงแอสเปคที่มีการนำเสนอ [11] นั้นยังมีรูปแบบที่ไม่สามารถนำเสนอรายละเอียดได้เพียงพอที่จะนำไปใช้สร้างกรณีทดสอบที่มีประสิทธิภาพได้ งานวิจัยนี้จึงได้ออกแบบรูปแบบการเขียนคำอธิบายยูสเคสใหม่ที่เหมาะสมกับคุณลักษณะของการพัฒนาซอฟต์แวร์เชิงแอสเปคและเพียงพอต่อการนำไปใช้สร้างกรณีทดสอบซึ่งเรียกว่า แอสเปคยูสเคส (Aspect Use Case) ไว้เป็นอีกส่วนประกอบหนึ่งของงานวิจัยอีกด้วย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบรูปแบบคำอธิบายยูสเคสสำหรับอธิบายข้อกำหนดความต้องการของแอสเปค และทำการออกแบบขั้นตอนวิธีสำหรับการสร้างกรณีทดสอบจากข้อกำหนดความต้องการของแอสเปคในรูปแบบของคำอธิบายยูสเคส

1.3 สมมติฐานของการศึกษา

แอสเปคยูสเคสที่ออกแบบขึ้นมีความสามารถอธิบายความต้องการในส่วนของแอสเปคได้ รวมทั้งขั้นตอนสำหรับการสร้างกรณีทดสอบจากแอสเปคยูสเคสที่ได้จากงานวิจัยนี้สามารถนำไปใช้เพื่อทดสอบความถูกต้องของการทำงานร่วมกันระหว่างแอสเปคกับฟังก์ชันการทำงานหลักได้อย่างครอบคลุมทุกเส้นทางการทำงาน

1.4 ขอบเขตการวิจัย

ในวิทยานิพนธ์นี้มีขอบเขตของการวิจัย ดังต่อไปนี้

1.4.1 งานวิจัยนี้จะเน้นในสองส่วนหลัก คือ การออกแบบรูปแบบการเขียนคำอธิบายยูสเคสให้มีความสามารถในการอธิบายข้อมูลความต้องการของแอสเปคได้ตามมาตรฐานของแอสเปคเจ (AspectJ) และการออกแบบขั้นตอนวิธีการสำหรับการสร้างกรณีทดสอบจากแอสเปคยูสเคส

โดยกรณีทดสอบที่ได้จะครอบคลุมในทุกการทำงานร่วมกันระหว่างฟังก์ชันการทำงานหลักกับแอสเปคเท่านั้น โดยไม่ครอบคลุมการทดสอบในรูปแบบอื่น อาทิ การทดสอบการยอมรับจากผู้ใช้ (Acceptation Test) เป็นต้น

1.4.2 การประเมินผลงานวิจัยจะแบ่งการประเมินออกเป็น 2 ส่วน คือ ส่วนแรกทำการประเมินผลรูปแบบการเขียนคำอธิบายยูสเคสสำหรับแอสเปค โดยการนำไปเปรียบเทียบกับกลไกการทำงานของแอสเปคตามมาตรฐานของแอสเปคเจ เพื่อตรวจสอบความสามารถในการอธิบายการทำงานของแอสเปค ส่วนที่สองทำการประเมินผลขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส โดยใช้ข้อกำหนดความต้องการที่อธิบายด้วยคำอธิบายแอสเปคยูสเคสมาสร้างกรณีทดสอบด้วยขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส นำกรณีทดสอบที่ถูกสร้างขึ้นมาเปรียบเทียบความครอบคลุมเส้นการทำงานกับกรณีทดสอบที่ถูกสร้างจากข้อกำหนดความต้องการเดียวกันที่อธิบายด้วยคำอธิบายยูสเคสด้วยขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคสด้วยการทดสอบเบซิสพาท และพิจารณาเปรียบเทียบข้อดีข้อเสียระหว่างขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคสกับขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

1.5 ขั้นตอนการศึกษาและดำเนินงานวิจัย

วิทยานิพนธ์นี้มีขั้นตอนการศึกษาและดำเนินงานวิจัย ดังนี้

1.5.1 ศึกษาแนวคิดวิธีการของกระบวนการพัฒนาซอฟต์แวร์เชิงแอสเปค

1.5.2 ศึกษาการสร้างกรณีทดสอบรูปแบบต่างๆ อาทิ การสร้างกรณีทดสอบจากคำอธิบายยูสเคส การสร้างกรณีทดสอบจากแผนภาพกิจกรรม (Activity Diagram)

1.5.3 ออกแบบรูปแบบวิธีการเขียนคำอธิบายยูสเคสในเชิงแอสเปค เพื่ออธิบายหน้าที่การทำงานของคอนเซินที่กระจายและเกี่ยวพันกันในซอฟต์แวร์ที่พัฒนาด้วยกระบวนการพัฒนาซอฟต์แวร์เชิงแอสเปค

1.5.4 ออกแบบขั้นตอนการสร้างกรณีทดสอบจากแอสเปคยูสเคส สำหรับสร้างการทดสอบเพื่อทดสอบซอฟต์แวร์ที่พัฒนาด้วยกระบวนการพัฒนาซอฟต์แวร์เชิงแอสเปค

1.5.5 พัฒนาโปรแกรมประยุกต์เพื่อทดสอบการทำงานสำหรับขั้นตอนการสร้างกรณีทดสอบจากแอสเปคยูสเคสที่ได้ออกแบบไว้

1.5.6 ประเมิน และสรุปผลการประเมิน

1.5.7 เขียนวิทยานิพนธ์

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎี และงานวิจัยที่เกี่ยวข้องกับการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส เนื้อหาที่นำเสนอในบทนี้จะประกอบด้วย ทฤษฎีและงานวิจัยเกี่ยวกับการพัฒนาซอฟต์แวร์ด้วยแนวคิดเชิงแอสเปค ทฤษฎีและงานวิจัยออกแบบรูปแบบการเขียนคำอธิบายยูสเคสซอฟต์แวร์ ทฤษฎีและงานวิจัยที่เกี่ยวกับซีแนริโอสำหรับแสดงเส้นทางการทำงานที่เป็นไปได้ของยูสเคส ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับแผนภาพกิจกรรมในการอธิบายถึงขั้นตอนการทำงานของแต่ละยูสเคส ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการสร้างกรณีทดสอบ และส่วนสุดท้ายเป็นทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการทดสอบความครอบคลุมของกรณีทดสอบ

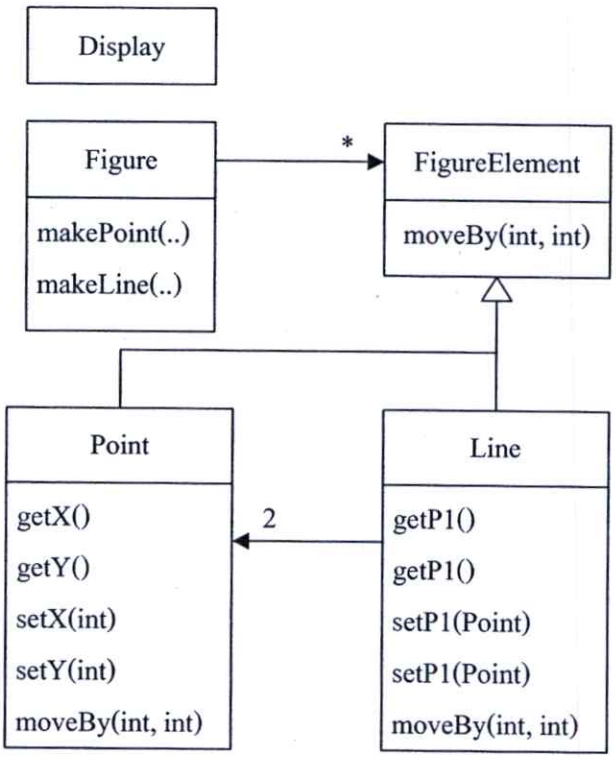
2.1 การพัฒนาซอฟต์แวร์ด้วยแนวคิดเชิงแอสเปค

กระบวนการพัฒนาซอฟต์แวร์เชิงแอสเปคเป็นแนวความคิดในการพัฒนาซอฟต์แวร์สำหรับปรับปรุงการแยกส่วนคอนเซิน ในการพัฒนาซอฟต์แวร์ทั่วไป เช่น การพัฒนาซอฟต์แวร์เชิงวัตถุ ทำให้ซอฟต์แวร์ที่พัฒนาขึ้นด้วยแนวความคิดนี้ง่ายต่อการบำรุงรักษา (Maintain) และนำกลับมาใช้ใหม่ (Reuse) กระบวนการพัฒนาซอฟต์แวร์เชิงแอสเปคนี้เป็นกระบวนการพัฒนาซอฟต์แวร์ที่สนับสนุนการแยกส่วนครอสคัตติ้งคอนเซิน (Crosscutting Concern) โดยจัดเตรียมกลไกการแยกส่วนและการรวมครอสคัตติ้งคอนเซินในกระบวนการพัฒนาซอฟต์แวร์ และได้กำหนดมอดูลใหม่ที่เรียกว่า แอสเปค ให้เป็นมอดูลสำหรับครอสคัตติ้งคอนเซินที่ได้ถูกแยกส่วนออกมาและกำหนดให้ จอยพอยท์ (Join Point) เป็นจุดสำหรับรวมแอสเปคเข้ากับส่วนประกอบของซอฟต์แวร์ [1][31][32][33]

2.1.1 ครอสคัตติ้งคอนเซิน

คอนเซิน คือ ความต้องการฟังก์ชันการทำงานหรือขอบเขตความสนใจที่ต้องการจัดการเพื่อตอบสนองเป้าหมายของซอฟต์แวร์ [24][25][29][31][32] การแยกส่วนคอนเซินเป็นหลักการของวิศวกรรมซอฟต์แวร์ สำหรับการพัฒนาซอฟต์แวร์ให้มีความเป็นมอดูล โดยการระบุและกำหนดให้แต่ละคอนเซินเป็นมอดูลเดี่ยว เพื่อให้ง่ายต่อการเปลี่ยนแปลงและแก้ไข [1][31][32] กระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุเป็นแนวความคิดหนึ่งในการพัฒนาซอฟต์แวร์ที่สนับสนุนการแยกส่วนคอนเซิน โดยกำหนดให้ออบเจกต์ (Object) เป็นมอดูลสำหรับคอนเซิน แต่มีบางคอนเซินที่ยากต่อการทำเป็นมอดูลโดยแนวความคิดในการพัฒนาซอฟต์แวร์ที่มีอยู่ คอนเซิน

เหล่านั้นจะมีความสัมพันธ์กับหลายส่วนประกอบของซอฟต์แวร์ ส่งผลให้คอนเซินนั้นกระจัดกระจาย (Scatter) และเกิดความเกี่ยวพัน (Tangle) ทำให้ยากต่อการเปลี่ยนแปลงแก้ไข ซึ่งการเปลี่ยนแปลงดังกล่าวอาจมีขอบเขตและมีผลกระทบต่อส่วนประกอบของซอฟต์แวร์ที่กว้างขวาง คอนเซินเหล่านี้เรียกว่า ครอสคัตติ้งคอนเซิน [1][24][25][27][31][32]



รูปที่ 2.1 แผนภาพคลาสของซอฟต์แวร์สำหรับวาดรูป

จากรูปที่ 2.1 เป็นแผนภาพคลาส (Class Diagram) ของซอฟต์แวร์สำหรับวาดรูปซึ่งเป็นผลมาจากการวิเคราะห์และออกแบบด้วยกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ โดยคลาส Figure จะประกอบด้วยคลาส FigureElement ต่างๆ คือ คลาส Point ซึ่งจะประกอบด้วยพิกัดเอกซ์ (x) กับพิกัดวาย (y) และคลาส Line ซึ่งจะถูกกำหนดด้วยจุด 2 จุด การแสดงผลของรูปจะถูกแสดงผลด้วยคลาส Display สามารถเขียนโค้ดของคลาส Point และคลาส Line ได้ดังรูปที่ 2.2

```
class Point {
        private int x = 0, y = 0;

        int getX() { return x; }
        int getY() { return y; }

        void setX(int x) {
                this.x = x;
        }
        void setY(int y) {
                this.y = y;
        }
        void moveBy(int dx, int dy) {
                x += dx;
                y += dy;
        }
}
```

```
class Line {
        private Point p1, p2;

        Point getP1() { return p1; }
        Point getP2() { return p2; }

        void setP1(Point p1) {
                this.p1 = p1;
        }
        void setP2(Point p2) {
                this.p2 = p2;
        }
        void moveBy(int dx, int dy) {
                x += dx;
                y += dy;
        }
}
```

รูปที่ 2.2 โค้ดของคลาส Line และคลาส Point

ส่วนประกอบของรูปอาจมีการเปลี่ยนแปลงตำแหน่ง จึงต้องมีการปรับการแสดงของรูปบนส่วนแสดงผล โดยเพิ่มเติมการทำงานสำหรับปรับการแสดงของรูปซึ่งเป็นการทำงานในส่วนของการแสดงผลที่คลาส Point และคลาส Line ซึ่งแสดงดังรูปที่ 2.3

```
class Point {
        private int x = 0, y = 0;

        int getX() { return x; }
        int getY() { return y; }

        void setX(int x) {
                this.x = x;
                Display.update();
        }
        void setY(int y) {
                this.y = y;
                Display.update();
        }
        void moveBy(int dx, int dy) {
                x += dx;
                y += dy;
                Display.update();
        }
}
```

```
class Line {
        private Point p1, p2;

        Point getP1() { return p1; }
        Point getP2() { return p2; }

        void setP1(Point p1) {
                this.p1 = p1;
                Display.update();
        }
        void setP2(Point p2) {
                this.p2 = p2;
                Display.update();
        }
        void moveBy(int dx, int dy) {
                x += dx;
                y += dy;
                Display.update();
        }
}
```

รูปที่ 2.3 โค้ดของคลาส Line และคลาส Point ที่ได้เพิ่มการทำงานสำหรับปรับการแสดงของรูป

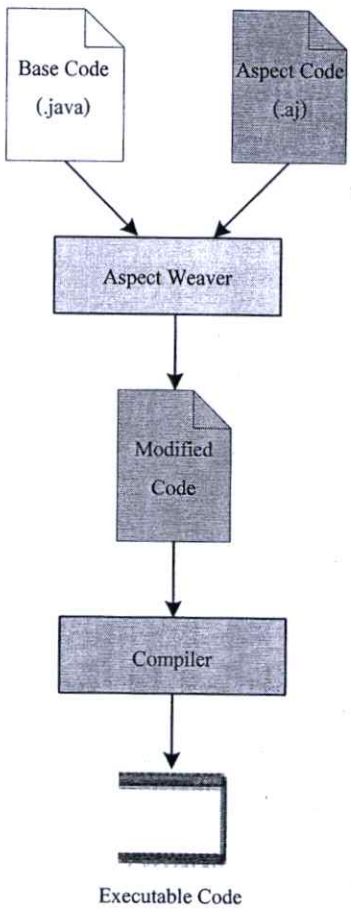
หลังจากที่ได้เพิ่มเติมการทำงานสำหรับปรับการแสดงของรูป ทำให้เกิดการครอสคัตติ้งคอนเซินของคลาส Display ที่ได้เข้าไปมีความเกี่ยวข้องในคลาส Line และคลาส Point ทำให้โค้ดในคลาส Line และคลาส Point มีความเกี่ยวพันกันมากขึ้น และเกิดการกระจัดกระจายของคอนเซิน Display ไปยังหลายๆ คลาส ดังรูปที่ 2.3 ด้วยเหตุนี้ทำให้ยากต่อการทำความเข้าใจ (Understandability) เนื่องจากความซ้ำซ้อนของโค้ดในหลายๆ มอดูลและทำให้ยากต่อการปรับปรุงแก้ไข (Maintainability) เนื่องจากโค้ดมีความเกี่ยวข้องกับหลายๆ มอดูล

2.1.2 แอสเปคเจ

แอสเปคเจ เป็น ภาษาโปรแกรมที่เป็นส่วนขยายในการพัฒนาโปรแกรมเชิงแอสเปคสำหรับภาษาโปรแกรมจาวา (Java Programming Language) ซึ่งเป็นภาษาโปรแกรมสำหรับการพัฒนาโปรแกรมเชิงวัตถุ เพื่อขจัดปัญหาครอสคัตติ้งคอนเซินที่เกิดขึ้นในการพัฒนาโปรแกรมเชิงวัตถุ โดยแบ่งแยกคอนเซินที่เกิดการครอสคัต (Crosscut) กับมอดูลอื่นๆ ออกมาเป็นมอดูลใหม่ที่เรียกว่าแอสเปค

2.1.2.1 กลไกการทำงานของแอสเปคเจ

แอสเปคเจมีกลไกที่ทำหน้าที่ในการเชื่อมความสัมพันธ์ระหว่างโค้ดฐาน (Base Code) กับแอสเปคให้สามารถทำงานร่วมกันได้ คือ แอสเปควีเวอร์ (Aspect Weaver) โดยการทำงานของแอสเปคเจจะเริ่มจากการนำโค้ดฐานซึ่งเป็นไฟล์ .java กับแอสเปคซึ่งเป็นไฟล์ .aj มาทำการเชื่อมความสัมพันธ์กันด้วยแอสเปควีเวอร์ ซึ่งจะทำให้ได้โค้ดที่ถูกเชื่อมความสัมพันธ์กันแล้วหรือโค้ดที่ถูกดัดแปลงแล้ว (Modified Code) จากนั้นนำโค้ดที่ได้มาทำการคอมไพล์ (Compile) ด้วยคอมไพล์เลอร์ (Compiler) ซึ่งจะได้โค้ดที่สามารถทำงานได้ (Executable Code) ดังรูปที่ 2.4



รูปที่ 2.4 แสดงกลไกการทำงานของแอสเปคเจ

2.1.2.2 รูปแบบภาษาของแอสเปคเจ

แอสเปคเจประกอบไปด้วยโครงสร้างหลัก 5 ส่วน คือ จอยพอยท์ พอยท์คัต แอดไวซ์ (Advice) การประกาศประเภทข้อมูลร่วม (Inter-Type Declaration) และแอสเปค ซึ่งแต่ละส่วนมีรายละเอียดดังต่อไปนี้

(1) จอยพอยท์ คือ จุดของเหตุการณ์ในขั้นตอนการทำงานของโปรแกรม โดยแอสเปคเจได้แบ่งเหตุการณ์ที่เป็นจอยพอยท์ ดังต่อไปนี้

- การเรียกใช้เมธอด (Method Call) หรือ คอนสทรัคเตอร์ (Constructor Call) เป็นจอยพอยท์ที่เกิดขึ้นเมื่อเมธอดหรือคอนสทรัคเตอร์ถูกเรียกใช้งาน
- การทำงานของเมธอด (Method Execution) หรือ คอนสทรัคเตอร์ (Constructor Execution) เป็นจอยพอยท์ที่เกิดขึ้นเมื่อเมธอดหรือคอนสทรัคเตอร์มีการทำงาน
- การเข้าถึงแอททริบิวต์ (Attribute Access) เป็นจอยพอยท์ที่เกิดขึ้นเมื่อมีการเรียกดูค่าหรือกำหนดค่าของแอททริบิวต์
- การกำหนดค่าเริ่มต้นของคลาส (Class Initialization) และออบเจกต์ (Object Initialization) เป็นจอยพอยท์ที่เกิดขึ้นเมื่อมีการกำหนดค่าเริ่มต้นให้กับคลาส และออบเจกต์
- การทำงานของการจัดการข้อผิดพลาด (Exception Handler Execution) เป็นจอยพอยท์ที่เกิดขึ้นเมื่อมีการทำงานในส่วนของการจัดการข้อผิดพลาดที่เกิดขึ้นในโปรแกรม

(2) พอยท์คัต ถูกใช้สำหรับเลือกเหตุการณ์ของจอยพอยท์ เพื่อกำหนดการทำงานของโปรแกรมเมื่อเกิดเหตุการณ์ดังที่ได้เลือกไว้ โดยมีโครงสร้างสำหรับการประกาศพอยท์คัตดังนี้

[Modifier] pointcut Pointcut-Name ([args]) : Pointcut-Definition ;

ซึ่งมีรายละเอียดดังนี้

- Modifier เป็นการกำหนดการเข้าถึงหรือการมองเห็นของพอยท์คัต ซึ่งมีลักษณะเหมือนกันในโปรแกรมภาษาจาวา โดยสามารถแบ่งออกได้ดังนี้ คือ public, private และ protected
- pointcut คือ คีย์เวิร์ด (Key word) ที่ใช้สำหรับประกาศพอยท์คัต
- Pointcut-Name คือ ชื่อของพอยท์คัต
- args คือ พารามิเตอร์ที่รับเข้ามาเพื่อทำงานในพอยท์คัต
- Pointcut-Definition คือ ส่วนที่กำหนดเหตุการณ์ของจอยพอยท์ ที่ให้พอยท์คัตทำงาน

(3) แอดไวซ์ คือ ส่วนของเนื้อหาในการทำงานของโปรแกรม เมื่อพอยท์คัตที่ได้กำหนดไว้ทำงาน โดยสามารถแบ่งแอดไวซ์ออกเป็น 3 ประเภท ได้แก่

- Before Advice เป็นแอดไวซ์ที่จะทำงานก่อนเหตุการณ์ที่ได้กำหนดไว้ในพอยท์คัตนั้นจะทำงาน
- After Advice เป็นแอดไวซ์ที่จะทำงานหลังจากเหตุการณ์ที่ได้กำหนดไว้ในพอยท์คัตนั้นทำงานเสร็จแล้ว

- Around Advice เป็นแอดไวซ์ที่สามารถข้ามการทำงาน ทำการทำงานเดิมต่อไป หรือทำการเปลี่ยนแปลงการทำงานของเหตุการณ์ที่ได้กำหนดไว้ในพอยท์คัต

โดยมีโครงสร้างสำหรับการประกาศแอดไวซ์ ดังนี้

AdviceType ([args]) : Pointcut-Name {Body} ;

ซึ่งมีรายละเอียดดังนี้

- AdviceType คือ ประเภทของแอดไวซ์
- args คือ พารามิเตอร์ที่รับเข้ามาเพื่อทำงานในแอดไวซ์
- Pointcut-Name คือ ชื่อของพอยท์คัตที่ทำงานร่วมกับแอดไวซ์
- Body คือ ส่วนการทำงานของแอดไวซ์

(4) การประกาศประเภทข้อมูลร่วม คือ การประกาศตัวแปรและเมธอดภายในแอสเปค ทำให้สามารถเรียกใช้งานตัวแปรและเมธอดเหล่านั้นเพื่อทำงานภายในภายแอสเปคได้ ซึ่งการประกาศจะมีลักษณะการประกาศเช่นเดียวกับโปรแกรมภาษาจาวา

(5) แอสเปค คือ มอดูลสำหรับครอสคัตติ้งคอนเซิน ซึ่งจะประกอบด้วย พอยท์คัต และแอดไวซ์ โดยมีโครงสร้างสำหรับการประกาศแอสเปค ดังนี้

[Modifier] Aspect-Name {Body}

ซึ่งมีรายละเอียดดังนี้

- Modifier เป็นการกำหนดการเข้าถึงหรือการมองเห็นของแอสเปค ซึ่งมีลักษณะเหมือนกันในโปรแกรมภาษาจาวา โดยสามารถแบ่งออกได้ดังนี้ คือ public, private และ protected
- Aspect-Name คือ ชื่อของแอสเปค
- Body คือ ส่วนการทำงานของแอสเปค

2.1.2.3 ตัวอย่างโปรแกรมที่พัฒนาด้วยแอสเปคเจ

จากรูปที่ 2.3 ซึ่งแสดงการเกิดครอสคัตติ้งคอนเซินของคลาส Display ภายในคลาส Line และคลาส Point สามารถแก้ปัญหาครอสคัตติ้งคอนเซินด้วยการพัฒนาโปรแกรมเชิงแอสเปคด้วยแอสเปคเจ ได้ดังรูปที่ 2.5 และรูปที่ 2.6 ตามลำดับ

```
class Point {
        private int x = 0, y = 0;

        int getX() { return x; }
        int getY() { return y; }

        void setX(int x) {
                this.x = x;
        }
        void setY(int y) {
                this.y = y;
        }
        void moveBy(int dx, int dy) {
                x += dx;
                y += dy;
        }
}
```

```
class Line {
        private Point p1, p2;

        Point getP1() { return p1; }
        Point getP2() { return p2; }

        void setP1(Point p1) {
                this.p1 = p1;
        }
        void setP2(Point p2) {
                this.p2 = p2;
        }
        void moveBy(int dx, int dy) {
                x += dx;
                y += dy;
        }
}
```

รูปที่ 2.5 คลาส Line และคลาส Point

```
aspect DisplayUpdating {

        pointcut move():
                call(void Line.setP1(Point))        ||
                call(void Line.setP2(Point))        ||
                call(void Line.moveBy(int, int))    ||
                call(void Point.setX(int))          ||
                call(void Point.setY(int))          ||
                call(void Point.moveBy(int, int));

        after() : move() {
                Display.update();
        }
}
```

รูปที่ 2.6 แอสเปค DisplayUpdating

รูปที่ 2.6 แสดงแอสเปค DisplayUpdating เป็นมอดูลการทำงานสำหรับปรับการแสดงของรูปที่เกิดการครอสคัตในคลาส Line และคลาส Point ดังรูปที่ 2.3 ซึ่งถูกแยกส่วนออกมาเป็นมอดูลนี้ เมื่อมีการเรียกใช้เมธอดที่ได้กำหนดไว้ในพอยท์คัต move การทำงานสำหรับปรับการแสดงของรูปที่ได้กำหนดไว้ในแอดไวซ์จะทำงานหลังจากการเรียกใช้เมธอดตามที่ได้กำหนดไว้ในพอยท์คัต จากการพัฒนาโปรแกรมเชิงแอสเปคด้วยแอสเปคเจดังกล่าวทำให้ซอฟต์แวร์ง่ายต่อการทำความเข้าใจ การปรับปรุงแก้ไขและซอฟต์แวร์มีความเป็นมอดูลมากขึ้น

2.2 ยูสเคส

ยูสเคส เป็นเครื่องมือที่มีประสิทธิภาพในการนำเสนอความต้องการ และหน้าที่การทำงานต่างๆในระบบ ซึ่งจะส่งผลต่อการทำแผนการทดสอบ การสร้างคู่มือการใช้งานระบบตลอดจนการสร้างและการตรวจสอบความถูกต้องของการออกแบบ การอธิบายยูสเคสในปัจจุบันประกอบด้วย แผนภาพยูสเคส (Use Case Diagram) และ คำอธิบายยูสเคส (Use Case Description)

2.2.1 แผนภาพยูสเคส

แผนภาพยูสเคสเป็นแผนภาพสำหรับอธิบายความต้องการ และกำหนดฟังก์ชันการทำงานของระบบจากมุมมองของผู้ใช้งาน โดยผู้ใช้งานก็คือสิ่งใดก็ตามที่อยู่ภายนอกระบบและมีการติดต่อกับระบบ แผนภาพยูสเคสช่วยทำให้ผู้ใช้งานสามารถเข้าใจการทำงานของระบบได้ง่ายยิ่งขึ้นและช่วยตรวจสอบระบบที่ได้ถูกออกแบบมานั้นว่าสามารถตอบสนองทุกๆ ความต้องการหรือไม่ โดยปกติแผนภาพยูสเคสจะถูกนำไปใช้ในขั้นตอนการวิเคราะห์ความต้องการ (Requirement Analysis) ของกระบวนการพัฒนาระบบ ช่วยเหลือในการวางแผนการทดสอบ (Test Plan) และช่วยเหลือในการทำคู่มือผู้ใช้งาน (User Guide) [4][12][13][22] ซึ่งส่วนประกอบหลักของแผนภาพยูสเคสจะประกอบด้วย

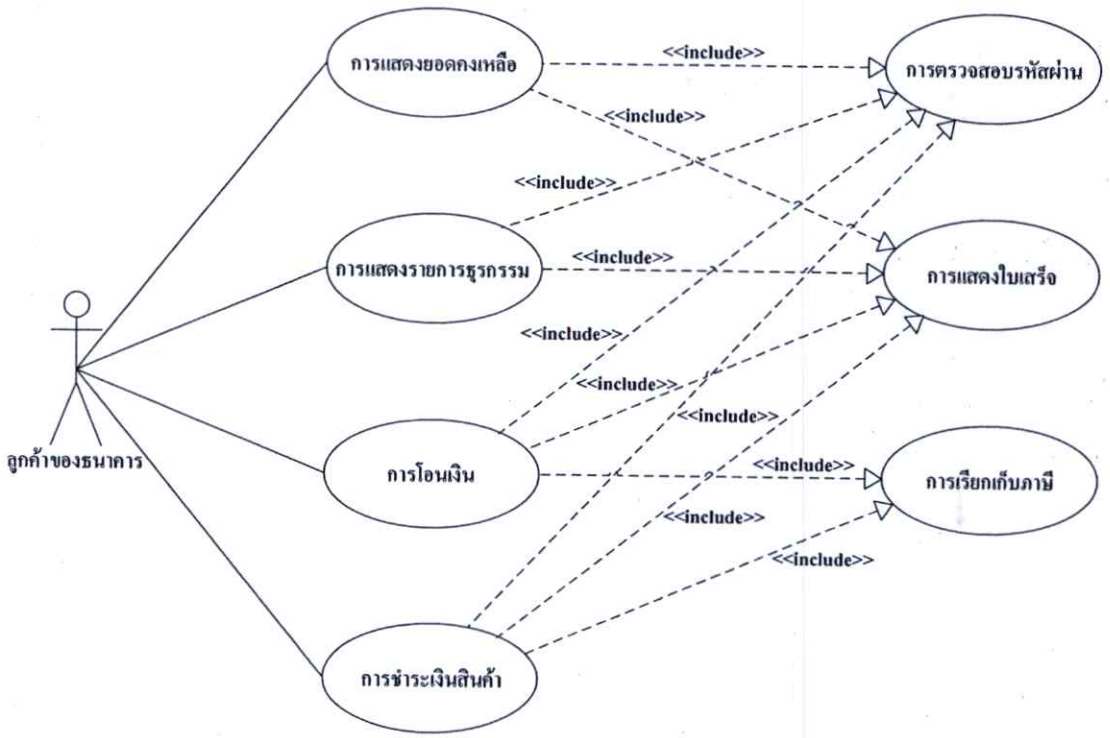
2.2.1.1 ยูสเคส

ยูสเคสจะกำหนดและอธิบายฟังก์ชันการทำงานของระบบ ซึ่งระบบได้จัดเตรียมไว้จากมุมมองของผู้ใช้งาน โดยอธิบายถึงการทำงานที่ถูกกระทำจากระบบเพื่อตอบสนองเหตุการณ์ที่เกิดขึ้นภายนอกระบบโดยผู้ใช้งาน เพื่อให้บรรลุเป้าหมายของฟังก์ชันการทำงานของยูสเคสนั้นๆ ยูสเคสสามารถแสดงแทนได้ด้วยรูปวงรี ดังรูปที่ 2.7 แผนภาพยูสเคสระบบธนาคารอินเทอร์เน็ต (Internet Banking System) ซึ่งจะประกอบไปด้วย ยูสเคสการแสดงยอดคงเหลือ ยูสเคสการแสดงรายการธุรกรรม ยูสเคสการโอนเงิน ยูสเคสการชำระเงินสินค้า เป็นต้น [4][12][13][22]

2.2.1.2 แอคเตอร์

แอคเตอร์คือสิ่งใดก็ตามที่ติดต่อกับระบบทางใดทางหนึ่งและเกี่ยวข้องกับยูสเคสอย่างน้อยหนึ่งยูสเคส เช่น คน ซอฟต์แวร์ อุปกรณ์ฮาร์ดแวร์หรือระบบอื่นๆ โดยแต่ละแอคเตอร์จะ

ถูกกำหนดบทบาทหน้าที่เฉพาะที่มีต่อระบบโดยมุมมองของผู้ใช้งาน แอคเตอร์สามารถแสดงแทนได้ด้วยรูปคน ดังรูปที่ 2.7 แผนภาพยูสเคสระบบธนาคารอินเทอร์เน็ต แอคเตอร์ คือ ลูกค้าของธนาคาร [4][12][13][22]



รูปที่ 2.7 แผนภาพยูสเคสระบบธนาคารอินเทอร์เน็ต

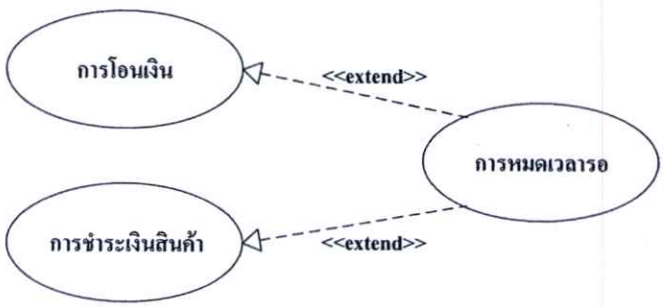
2.2.1.3 ความสัมพันธ์ระหว่างยูสเคส

สามารถแบ่งประเภทของความสัมพันธ์ออกได้เป็น 3 ประเภทหลัก [13] [22] [37] ดังนี้คือ

(1) Inclusion คือการนำขั้นตอนการทำงานที่ซ้ำซ้อนกันมาสร้างเป็นยูสเคสแยกต่างหากเพื่อให้ยูสเคสอื่นได้เรียกใช้ เป็นลักษณะเดียวกันกับการเขียนโปรแกรมเป็นมอดูลย่อยๆ เพื่อให้มอดูลอื่นๆ เรียกใช้ และสามารถแสดงแทนด้วยการเชื่อมโยงระหว่างยูสเคสที่เรียกใช้กับยูสเคสที่ถูกเรียกใช้ด้วยเส้นประที่มีหัวลูกศรชี้ไปทางยูสเคสที่ถูกเรียกใช้ และมีเครื่องหมาย <<include>> กำกับไว้ที่เส้นประนั้นด้วย ตัวอย่างเช่น ในระบบธนาคารอินเทอร์เน็ต ยูสเคสการแสดงยอดคงเหลือ ยูสเคสการแสดงรายการธุรกรรม ยูสเคสการโอนเงิน และยูสเคสการชำระเงินสินค้ามีขั้นตอนการทำงานที่เหมือนกันอยู่ 2 อย่าง คือ การตรวจสอบรหัสผ่านกับการแสดงใบเสร็จ ดังนั้น เราสามารถสร้างยูสเคสสำหรับการตรวจสอบรหัสผ่าน และยูสเคสการแสดงใบเสร็จขึ้นมา

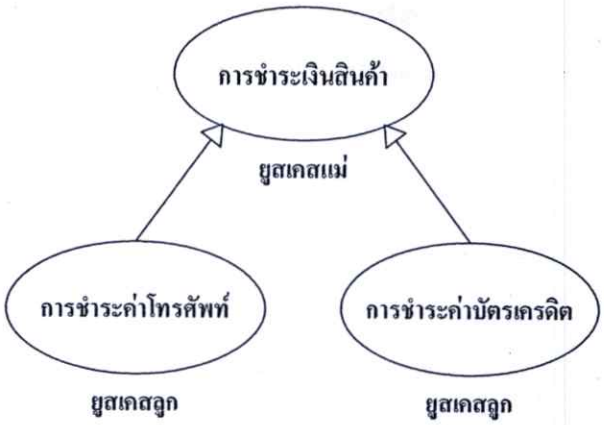
เพื่อให้ยูสเคสการแสดงยอดคงเหลือ ยูสเคสการแสดงรายการธุรกรรม ยูสเคสการโอนเงิน และยูสเคสการชำระเงินสินค้าเรียกใช้ ดังรูปที่ 2.7

(2) Extension คือการนำเอายูสเคสเดิมที่มีอยู่แล้วมาเพิ่มการทำงานบางอย่างเข้าไป สามารถแสดงแทนด้วยการเชื่อมโยงระหว่างยูสเคสที่ต้องการเพิ่มการทำงานกับยูสเคสที่ถูกเรียกใช้เพื่อเพิ่มการทำงานด้วยเส้นประที่มีหัวลูกศรชี้ไปทางยูสเคสที่ต้องการเพิ่มการทำงาน และมีเครื่องหมาย <<extend>> กำกับไว้ที่เส้นประนั้นด้วย ตัวอย่างเช่น ยูสเคสการโอนเงินและยูสเคสการชำระเงินสินค้าในระบบธนาคารอินเทอร์เน็ต เพื่อความปลอดภัยในการใช้งานระบบเพื่อโอนเงินและชำระเงินสินค้าเนื่องจากอาจเกิดกรณีที่ลูกค้าของธนาคารไม่ได้ทำการลงบันทึกออกจากระบบหลังจากใช้งานระบบเสร็จแล้ว ซึ่งมีความเสี่ยงที่อาจจะก่อให้เกิดความเสียหายต่อข้อมูล และเงินในบัญชีของลูกค้าของธนาคารได้ ระบบจะทำการออกจากระบบให้โดยอัตโนมัติเมื่อไม่มีการกระทำการใดๆ กับระบบตามระยะเวลาที่ระบบได้กำหนดไว้ ในกรณีดังกล่าวเราสามารถสร้างยูสเคสการหมดเวลาคอยเพื่อนำมาขยายหรือเพิ่มเติมการทำงานให้กับยูสเคสการโอนเงินและยูสเคสการชำระสินค้า ดังรูป 2.8



รูปที่ 2.8 แผนภาพยูสเคสแสดงความสัมพันธ์แบบ Extension

(3) Generalization คือการถ่ายทอดคุณสมบัติหรือพฤติกรรมบางอย่างจากยูสเคสหนึ่งไปยังอีกยูสเคสหนึ่ง หรือจากแอคเตอร์หนึ่งไปยังอีกแอคเตอร์หนึ่ง โดยยูสเคสที่เป็นผู้ถ่ายทอดพฤติกรรมจะเรียกว่า ยูสเคสแม่ (Parent Use Case) และยูสเคสที่รับการถ่ายทอดพฤติกรรมจะเรียกว่า ยูสเคสลูก (Child Use Case) ซึ่งยูสเคสลูกจะมีการเพิ่มเติมพฤติกรรมบางอย่างของตัวเองเข้าไปด้วย ซึ่งทำให้ยูสเคสลูกมีคุณสมบัติหรือพฤติกรรมที่มีความเฉพาะตัวหรือพิเศษมากกว่ายูสเคสแม่ ตัวอย่างเช่น จากระบบธนาคารอินเทอร์เน็ต สามารถนำยูสเคสการชำระเงินสินค้ามาถ่ายทอดพฤติกรรมให้แก่ยูสเคสการชำระค่าโทรศัพท์ และยูสเคสการชำระค่าบัตรเครดิต ซึ่งยูสเคสลูกดังกล่าวได้มีการเพิ่มพฤติกรรม เช่น การแสดงรายการผู้ให้บริการเครือข่ายโทรศัพท์ในยูสเคสการชำระค่าโทรศัพท์ และการแสดงรายการผู้ให้บริการบัตรเครดิตในยูสเคสการชำระค่าบัตรเครดิต เป็นต้น ตัวอย่างดังกล่าวสามารถแสดงได้ดังรูปที่ 2.9



รูปที่ 2.9 แผนภาพยูสเคสแสดงความสัมพันธ์แบบ Generalization

2.2.2 คำอธิบายยูสเคส

คำอธิบายยูสเคสเป็นการอธิบายถึงรายละเอียดสำหรับการทำงานของยูสเคส ทำให้สามารถเพิ่มความเข้าใจถึงขั้นตอนการทำงานและเงื่อนไขในการทำงานของแต่ละยูสเคสได้ดียิ่งขึ้น ซึ่งองค์ประกอบของยูสเคสนั้นจะมีจำนวนที่แตกต่างกันตามลักษณะและวัตถุประสงค์ในการใช้งาน อาทิ การนิยามรูปแบบการเขียนยูสเคสอย่างมีประสิทธิภาพของ Cockburn [6] การนำเสนอรูปแบบยูสเคสแบบสองสดมภ์ (Column) ของ Rebecca Wirfs-Brock [26] รูปแบบยูสเคสของ Geri Schneider และ Jason P. Winters [14] หรือรูปแบบยูสเคสของ Javier J. Gutierrez และคณะ [12] เป็นต้น ดังตัวอย่างคำอธิบายยูสเคสซึ่งแสดงในตารางที่ 2.1, 2.2, 2.3 และ2.4 ตามลำดับ

ตารางที่ 2.1 ตารางแสดงรูปแบบยูสเคสสำหรับการนำเสนอข้อมูลความต้องการของ Cockburn [6]

ชื่อยูสเคส #	ยืม
เป้าหมายในเนื้อหา	จัดการระบบการยืมทรัพยากรของระบบ
ขอบเขตและระดับ	งานหลัก
เงื่อนไขก่อน	ผู้ใช้ระบบต้องผ่านการตรวจสอบอำนาจการเข้าใช้ระบบ
เงื่อนไขเสร็จสิ้น	รายการยืมคืนเสร็จสิ้น สถานะผู้ใช้บริการได้รับการปรับปรุง
เงื่อนไขล้มเหลว	ผู้ใช้ระบบไม่ได้รับอนุญาตให้เข้าใช้ระบบ
แอคเตอร์ปฐมภูมิ **แอคเตอร์ทุติยภูมิ**	ผู้ใช้ระบบ ผู้ใช้บริการ
ทริกเกอร์	ผู้ใช้บริการต้องการยืมทรัพยากร

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่ 2.1 (ต่อ)

คำอธิบาย	ลำดับ	การกระทำ
	1	ผู้ใช้บริการต้องการยืมทรัพยากร
	2	ผู้ใช้ระบบคีย์รหัสผู้ใช้บริการ
	3	ระบบทำการค้นหาและตรวจสอบสถานะของผู้ใช้บริการ
	4	ระบบแสดงผลให้ผู้ใช้ระบบทราบ
	5	ถ้าสถานะของผู้ใช้บริการสามารถยืมได้ผู้ใช้ระบบจะเริ่มทำรายการยืมทรัพยากร
	6	ผู้ใช้ระบบใส่หมายเลขเรียกทรัพยากร <หนังสือ,วีดิโอ,รถเช่า>
	7	ระบบทำการค้นหาและตรวจสอบสถานะของทรัพยากร
	8	ถ้าสถานะของทรัพยากรสามารถยืมได้ผู้ใช้ระบบจะเพิ่มรายการทรัพยากรที่ยืมเข้าสู่รายการยืม
	9	ผู้ใช้ระบบสามารถเพิ่มลดปรับปรุงรายการยืมได้
	10	ผู้ใช้ระบบบันทึกรายการยืม ระบบทำการปรับปรุงสถานะของผู้ใช้บริการและทรัพยากรที่ถูกยืม
	11	ออกจากการทำรายการ
ส่วนต่อขยาย	**ลำดับ**	**การกระทำสาขา**
	3.a	สถานะผู้ใช้บริการไม่สามารถยืมได้ระบบจะแสดงสถานะของปัญหาแต่ละแบบให้ผู้ใช้ระบบทราบ
	7.a	สถานะของทรัพยากรไม่สามารถยืมได้ระบบจะแสดงสถานะของปัญหาแต่ละแบบให้ผู้ใช้ระบบทราบ
	10.a	หากระบบไม่สามารถบันทึกรายการยืมได้ในครั้งแรก ระบบจะอนุญาติให้ทำการบันทึกใหม่อีกครั้ง ถ้ายังไม่ได้ระบบจะแจ้งให้ผู้ใช้ระบบทราบเพื่อเริ่มต้นทำรายการยืมใหม่
	10.b	ระบบคำนวนค่าใช้บริการ
ความแปรผันย่อย		**การกระทำสาขา**
	1	สถานะที่ผู้ใช้บริการไม่สามารถยืมได้ ประกอบด้วย หมดอายุสมาชิก ยืมทรัพยากรเต็มอัตราแล้ว ยังไม่ส่งคืนทรัพยากรตามกำหนด
	2	สถานะของทรัพยากรที่ไม่สามารถยืมได้ประกอบด้วย มีคนจอง ส่งซ่อม สูญหาย

105311

ตารางที่ 2.1 (ต่อ)

ข้อมูลที่เกี่ยวข้อง	ประเภททรัพยากร
ความสำคัญ	สูง
สมรรถนะ	2 นาที
ความถี่	ตลอดวันทำการ
ช่องทางของแอคเตอร์	ฐานข้อมูล
หัวเรื่องเปิด	จะเกิดอะไรขึ้นเมื่อบัตรสมาชิกของผู้ใช้บริการสูญหาย
วันครบกำหนด	31 มีนาคม 2549 รุ่น 1.0
ข้อมูลอื่นๆ ที่จำเป็นต่อการพัฒนา	การตรวจสอบอำนาจในการใช้งานของผู้กี่ยวข้องกับระบบควรมีการตอบสนองต่อผู้ใช้ภายใน 1 นาที
ยูสเคสระดับบน	-
อนุยูสเคส	ยืม

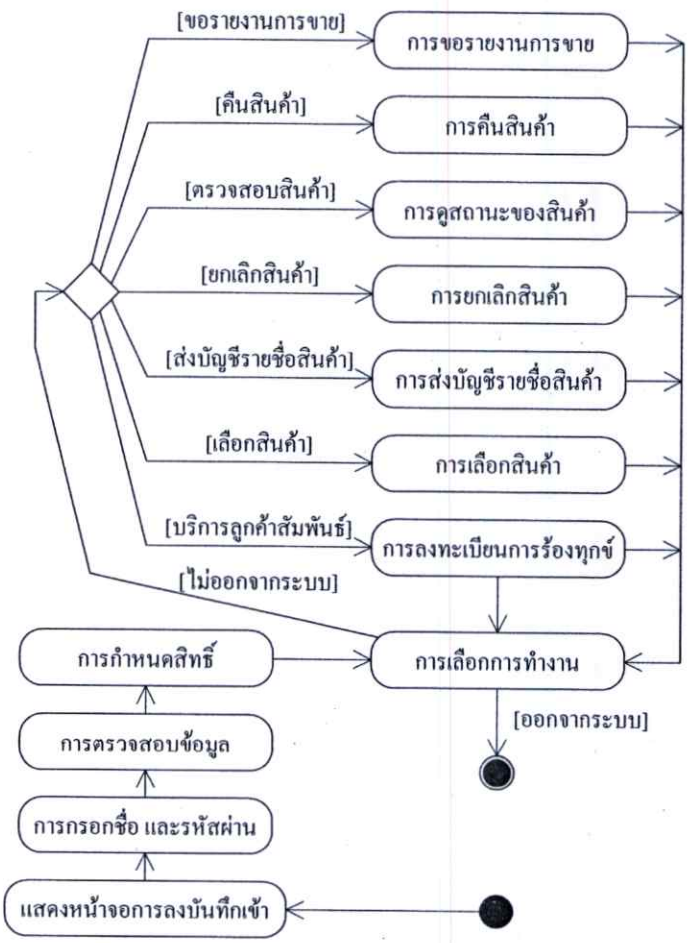
ตารางที่ 2.2 ตารางแสดงรูปแบบยูสเคสแบบสองสดมภ์ของ Rebecca J. Wirfs-Brock [26]

ยูสเคส	ประมวลผลรายการถอน
แอคเตอร์	ลูกค้าธนาคาร
รายละเอียดเบื้องต้น	ลูกค้าจะเป็นผู้ทำรายการทางการเงินด้วยตนเอง
แอคเตอร์กระทำ (Actor Action)	**ระบบตอบสนอง(System Response)**
ผู้ใช้แจ้งความจำนงในการถอน	
	แสดงรายการบัญชีของลูกค้า
เลือกบัญชีที่ต้องการ	
	พร้อมรับการจ่ายเงิน
...	
	...
ทางเลือก (Alternative)	1. ไม่มีเงินเพียงพอ 2. บัญชีลูกค้ามีเงินไม่เพียงพอ

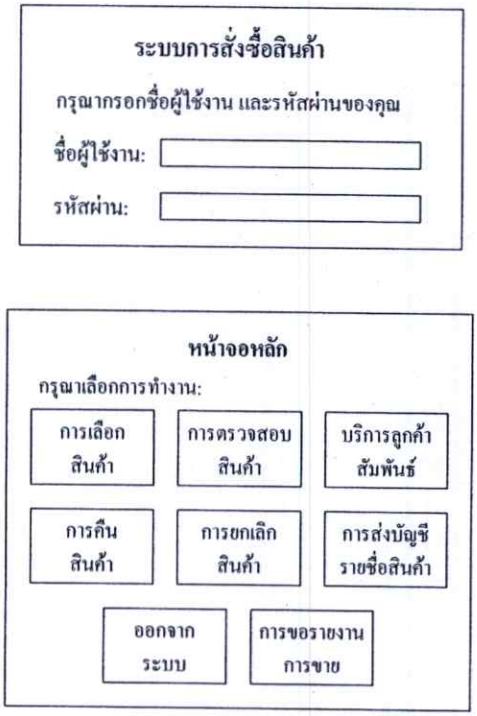
ตารางที่ 2.3 ตารางแสดงรูปแบบยูสเคสของ Geri Schneider และ Jason P. Winters [12]

ชื่อของยูสเคส	การลงบันทึกเข้า เป็นกระบวนการที่ผู้ใช้งานลงบันทึกเพื่อเข้าระบบการสั่งซื้อ และจัดเตรียมการกำหนดสิทธิ์ของผู้ใช้งานประเภทต่างๆ
แอคเตอร์	ลูกค้า
ลำดับความสำคัญ	สูง
สถานะ	-
เงื่อนไขก่อน	-
เงื่อนไขหลัง	-
จุดต่อขยาย (Extension Points)	-
ยูสเคสที่ถูกใช้งาน ("Used" Use Cases)	1. การเลือกสินค้า 2. การคืนสินค้า 3. การยกเลิกสินค้า 4. การดูสถานะของสินค้า 5. การส่งบัญชีรายชื่อสินค้า 6. การลงทะเบียนการร้องทุกข์ 7. การขอรายงานการขาย
ลำดับเหตุการณ์ (Flow of Events)	1. ยูสเคสจะเริ่มทำงานเมื่อผู้ใช้งานเริ่มใช้งานโปรแกรม ประยุกต์ 2. ระบบแสดงหน้าจอการลงบันทึกเข้า 3. ผู้ใช้งานกรอกชื่อผู้ใช้งาน และรหัสผ่าน 4. ระบบจะตรวจสอบข้อมูล 5. ระบบจะกำหนดสิทธิ์ของผู้ใช้งาน 6. ระบบจะแสดงหน้าจอหลัก 7. ผู้ใช้งานจะเลือกการทำงาน 8. ผู้ใช้งานไม่สามารถเลือกออกจากระบบชั่วขณะหนึ่ง 9. ถ้าผู้ใช้งานเลือก "การเลือกสินค้า" แล้ว 9.1 ใช้งาน "การเลือกสินค้า" 10. ไม่เช่นนั้น ถ้าผู้ใช้งานเลือก "การคืนสินค้า" แล้ว 10.1 ใช้งาน "การคืนสินค้า"

ตารางที่ 2.3 (ต่อ)

	11. ไม่เช่นนั้น ถ้าผู้ใช้งานเลือก “การยกเลิกสินค้า” แล้ว 11.1 ใช้งาน “การยกเลิกสินค้า” 12. ไม่เช่นนั้น ถ้าผู้ใช้งานเลือก “การตรวจสอบสินค้า” แล้ว 12.1 ใช้งาน “การดูสถานะของสินค้า” 13. ไม่เช่นนั้น ถ้าผู้ใช้งานเลือก “การส่งบัญชีรายชื่อสินค้า” แล้ว 13.1 ใช้งาน “การส่งบัญชีรายชื่อสินค้า” 14. ไม่เช่นนั้น ถ้าผู้ใช้งานเลือก “บริการลูกค้าสัมพันธ์” แล้ว 14.1 ใช้งาน “การลงทะเบียนการร้องทุกข์” 15. ไม่เช่นนั้น ถ้าผู้ใช้งานเลือก “การขอรายงานการขาย” แล้ว 15.1 ใช้งาน “การขอรายงานการขาย” 16. ผู้ใช้งานจะเลือกการออกจากระบบ 17. สิ้นสุดการทำงานของยูสเคส
แผนภาพกิจกรรม **(Activity Diagram)**	

ตารางที่ 2.3 (ต่อ)

ส่วนติดต่อกับผู้ใช้ **(User Interface)**	
ซีแนริโอ **(Scenarios)**	1. ชื่อผู้ใช้งานไม่ถูกต้อง 2. รหัสผ่านไม่ถูกต้อง 3. ผู้ใช้งานไม่มีผู้ใช้งาน และรหัสผ่านของระบบ 4. ผู้ใช้งานเลือกใช้งานการทำงานที่ไม่มีสิทธิ์ใช้งาน
แผนภาพลำดับ **(Sequence Diagram)**	-
ยูสเคสย่อย **(Subordinate Use Cases)**	1. การเข้าถึงระบบ - การลงบันทึกเข้า 2. การจัดการคำสั่ง - แสดงหน้าจอหลัก
คลาสที่เกี่ยวข้อง **(View of Participate Classes)**	-
อาร์ทิแฟค **(Other Artifacts)**	-
ข้อมูลความต้องการอื่นที่เกี่ยวข้อง **(Other Requirements)**	-

ตารางที่ 2.4 ตารางแสดงรูปแบบยูสเคสของ Javier J. Gutierrez และคณะ[14]

ชื่อของยูสเคส	เพิ่มการเชื่อมโยงใหม่
ลำดับเหตุการณ์หลัก	1. ผู้ใช้งานเลือกทางเลือก: add a new link 2. ระบบแสดงแบบฟอร์มสำหรับกรอกข้อมูลของการเชื่อมโยง 3. ผู้ใช้งานกรอกข้อมูลการเชื่อมโยงใหม่ และกดปุ่ม insert 4. ระบบจัดเก็บข้อมูลการเชื่อมโยงใหม่
ข้อผิดพลาด/ทางเลือก	1. ถ้าชื่อการเชื่องโยงหรือ URL ของการเชื่อมโยงไม่ได้กรอก ระบบจะแสดงข้อความข้อผิดพลาด และขอให้กรอกข้อมูลอีกครั้ง
เงื่อนไขก่อน	การเชื่อมโยงใหม่ถูกจัดเก็บลงในระบบ

2.3 ซีแนริโอ (Scenario)

ซีแนริโอเป็นกรณีหรือตัวอย่างของเส้นทางการทำงานของยูสเคสที่เป็นไปได้ แต่ละซีแนริโอจะแสดงแทนหนึ่งกรณีของเส้นทางการทำงานของยูสเคส

ตารางที่ 2.5 ตารางแสดงคำอธิบายยูสเคสการตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน

ชื่อของยูสเคส	การตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน
ลำดับเหตุการณ์หลัก	1. ทำการใส่ชื่อผู้ใช้งานและรหัสผ่าน 2. ระบบจะทำการตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน 3. ผู้ใช้งานสามารถใส่ชื่อผู้ใช้งานและรหัสผ่านได้ถูกต้องผู้ใช้งานสามารถเข้าใช้งานระบบได้
ลำดับเหตุการณ์ทางเลือก	2.1 ถ้าผู้ใช้งานใส่ชื่อผู้ใช้งานหรือรหัสผ่านไม่ถูกต้อง ระบบจะแจ้งว่าผู้ใช้งานใส่ชื่อผู้ใช้งานหรือรหัสผ่านไม่ถูกต้อง และให้ผู้ใช้งานทำการใส่ชื่อหรือรหัสผ่านใหม่อีกครั้ง

ซีแนริโอสามารถแบ่งออกเป็น 2 ประเภท [2][4][12] ดังต่อไปนี้

1. ซีแนริโอปฐมภูมิ (Primary Scenario) เป็นเส้นทางการทำงานปกติของยูสเคส แต่ละยูสเคสจะมีเพียงหนึ่งซีแนริโอปฐมภูมิ สามารถเรียกได้อีกชื่อว่า happy day scenario ซึ่งหมายถึง

เส้นทางการทำงานของยูสเคสที่ไม่มีข้อผิดพลาดเลย จากตารางที่ 2.5 เส้นทางการทำงานปกติของยูสเคสการตรวจสอบชื่อผู้ใช้งานและรหัสผ่านก็คือ ขั้นตอนที่ 1, 2, 3

2. ซีแนริโอทุติยภูมิ (Secondary Scenario) สามารถแบ่งออกเป็น 2 ประเภท ดังนี้

2.1 ซีแนริโอทางเลือก (Alternative Scenario) เป็นเส้นทางการทำงานทางเลือกหรือเป็นเส้นทางที่แตกแขนงจากซีแนริโอปฐมภูมิ ซึ่งอธิบายทางเลือกและเงื่อนไขของเส้นทางการทำงานสำหรับยูสเคสที่แยกจากเส้นทางการทำงานปกติ จากตารางที่ 2.5 เส้นทางการทำงานทางเลือกของยูสเคสการตรวจสอบชื่อผู้ใช้งานและรหัสผ่านคือ ขั้นตอนที่ 1, 2, 2.1, 1, 2, 3 และ 1, 2, 2.1, 1, 2, 2.1, 1, 2, 3 เป็นต้น

2.2 ซีแนริโอพิเศษ (Exceptional Scenario) เป็นเส้นทางการทำงานที่เกิดข้อผิดพลาดหรือเงื่อนไขที่ไม่คาดคิดอื่นๆ ที่เกิดขึ้นระหว่างเส้นทางการทำงานปกติหรือเส้นทางการทำงานที่เป็นทางเลือก ข้อผิดพลาดเหล่านี้อาจประกอบด้วย ข้อมูลนำเข้าไม่ถูกต้อง และข้อผิดพลาดในการติดต่อ เป็นต้น

2.4 แผนภาพกิจกรรม

แผนภาพกิจกรรมเป็นแผนภูมิการไหล (Flow Chart) ชนิดหนึ่งที่ใช้สำหรับแสดงขั้นตอนการทำงาน และอธิบายกระบวนการทำงานต่างๆ รวมทั้งแสดงการตัดสินใจของระบบ แผนภาพกิจกรรมถูกนำไปใช้อธิบายแผนภาพยูสเคสเพื่อแสดงขั้นตอนการทำงานของแต่ละยูสเคส ทำให้สามารถเข้าใจถึงการทำงานของแต่ละยูสเคสนั้นได้ดียิ่งขึ้น และยังช่วยในการสื่อสารถึงขั้นตอนการทำงานของระบบระหว่างผู้มีส่วนในการกระบวนการพัฒนาระบบ [12] [14] [23] จากการที่แผนภาพกิจกรรมมีความสามารถในการอธิบายและแสดงขั้นตอนการทำงานของยูสเคสได้อย่างมีประสิทธิภาพ ทำให้แผนภาพกิจกรรมมีส่วนช่วยในการระบุทุกๆ ซีแนริโอที่เป็นไปได้สำหรับแต่ละยูสเคสได้อีกด้วย [14][22][23]

2.4.1 ส่วนประกอบของแผนภาพกิจกรรม

(1) กิจกรรม คือ ขั้นตอนในการทำงาน เช่น การคำนวนผลลัพธ์บางอย่าง การเปลี่ยนแปลงสถานะของระบบ และการเรียกให้ฟังก์ชันการทำงานอื่นๆ เป็นต้น ซึ่งสามารถแสดงแทนด้วยรูปสี่เหลี่ยมมุมมน ดังรูปที่ 2.10

รูปที่ 2.10 สัญลักษณ์แสดงแทนกิจกรรม

(2) จุดเริ่มต้น (Start State) เป็นจุดเริ่มต้นในการทำงานของแผนภาพกิจกรรม ซึ่งสามารถแสดงแทนด้วยรูปวงกลมทึบ ดังรูปที่ 2.11

รูปที่ 2.11 สัญลักษณ์แสดงแทนจุดเริ่มต้น

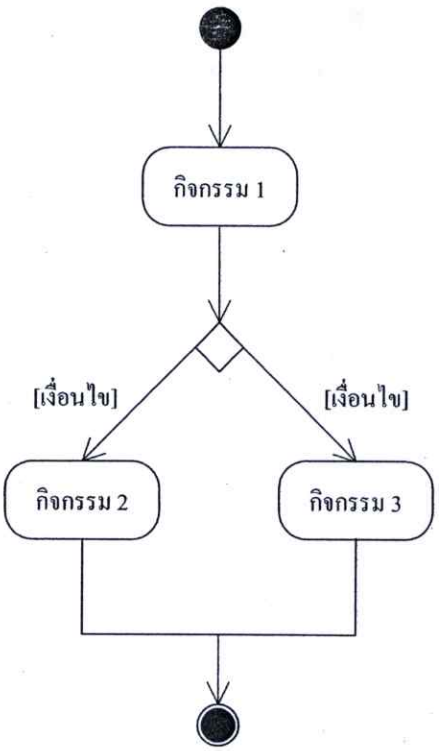
(3) จุดสิ้นสุด (Stop State) เป็นจุดสิ้นสุดในการทำงานของแผนภาพกิจกรรม ซึ่งสามารถแสดงแทนด้วยรูปวงกลมทึบที่อยู่ภายในวงกลมโปร่ง ดังรูปที่ 2.12

รูปที่ 2.12 สัญลักษณ์แสดงแทนจุดสิ้นสุด

(4) การเปลี่ยนสถานะ (Transition) เป็นการเปลี่ยนจากสถานะหนึ่งไปเป็นอีกสถานะหนึ่ง หรือเป็นการเปลี่ยนจากขั้นตอนการทำงานหนึ่งไปเป็นอีกขั้นตอนการทำงานหนึ่ง ซึ่งสามารถแสดงแทนด้วยลูกศร ดังรูปที่ 2.13

รูปที่ 2.13 สัญลักษณ์แสดงการเปลี่ยนสถานะ

(5) ทางเลือก (Decision Point) เป็นการสร้างทางเลือกสำหรับเปลี่ยนจากขั้นตอนการทำงานหนึ่งไปเป็นอีกขั้นตอนการทำงานหนึ่ง ซึ่งสามารถแสดงด้วยรูปสี่เหลี่ยมขนมเปียกปูน โดยใช้ลูกศรที่แสดงการเปลี่ยนสถานะชี้ไปที่รูปสี่เหลี่ยมขนมเปียกปูน และใช้ลูกศรที่แสดงการเปลี่ยนสถานะที่กำกับด้วยประโยคแสดงเงื่อนไขที่อยู่ในวงเล็บชี้ออกจากรูปสี่เหลี่ยมขนมเปียกปูนไปยังกิจกรรมต่อไป ดังรูปที่ 2.14



รูปที่ 2.14 สัญลักษณ์แสดงแทนทางเลือก

2.4.2 การใช้แผนภาพกิจกรรมแสดงการทำงานของแผนภาพยูสเคส

จากรูปที่ 2.7 แผนภาพยูสเคสระบบธนาคารอินเทอร์เน็ต และตารางที่ 2.6 ตารางที่ 2.7 ตารางที่ 2.8 และตารางที่ 2.9 ซึ่งแสดงคำอธิบายยูสเคสการโอนเงิน คำอธิบายยูสเคสการตรวจสอบรหัสผ่าน คำอธิบายยูสเคสการเรียกเก็บภาษี และคำอธิบายยูสเคสการแสดงใบเสร็จตามลำดับ สามารถแสดงขั้นตอนการทำงานของแผนภาพยูสเคสการโอนเงินด้วยแผนภาพกิจกรรมดังรูปที่ 2.15

ตารางที่ 2.6 คำอธิบายยูสเคสการโอนเงิน

ชื่อยูสเคส	การโอนเงิน
คำอธิบาย	การโอนเงินทางอินเทอร์เน็ต
เงื่อนไขก่อน	-
จุดต่อขยาย	-
ยูสเคสที่ถูกใช้งาน	1. การตรวจสอบรหัสผ่าน 2. การเรียกเก็บภาษี 3. แสดงใบเสร็จ

ตารางที่ 2.6 (ต่อ)

ลำดับเหตุการณ์	1. ลูกค้าเลือกเมนูการโอนเงิน 2. ลูกค้ากำหนดจำนวนเงินที่ต้องการโอน, หมายเลขบัญชีปลายทาง และหมายเลขสาขาของธนาคาร 3. ระบบทำการตรวจสอบรหัสผ่าน 3.1 ใช้งาน "การตรวจสอบรหัสผ่าน" 4. ระบบตรวจสอบความถูกต้องของข้อมูลการโอนที่ได้กำหนดไปในขั้นตอนที่แล้ว 5. ระบบหักเงินจากบัญชีของลูกค้าและเพิ่มเงินให้กับบัญชีปลายทาง 6. ระบบทำการเรียกเก็บภาษี 6.1 ใช้งาน "การเรียกเก็บภาษี" 7. ระบบแสดงใบเสร็จ 7.1 ใช้งาน "แสดงใบเสร็จ"
ลำดับเหตุการณ์ทางเลือก	4.1 ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง 4.1.1 ระบบแสดงข้อความเพื่อบอกถึงข้อมูลที่ไม่ถูกต้อง 4.1.2 กลับไปที่ขั้นตอนที่ 2
เงื่อนไขหลัง	-

ตารางที่ 2.7 คำอธิบายยูสเคสการตรวจสอบรหัสผ่าน

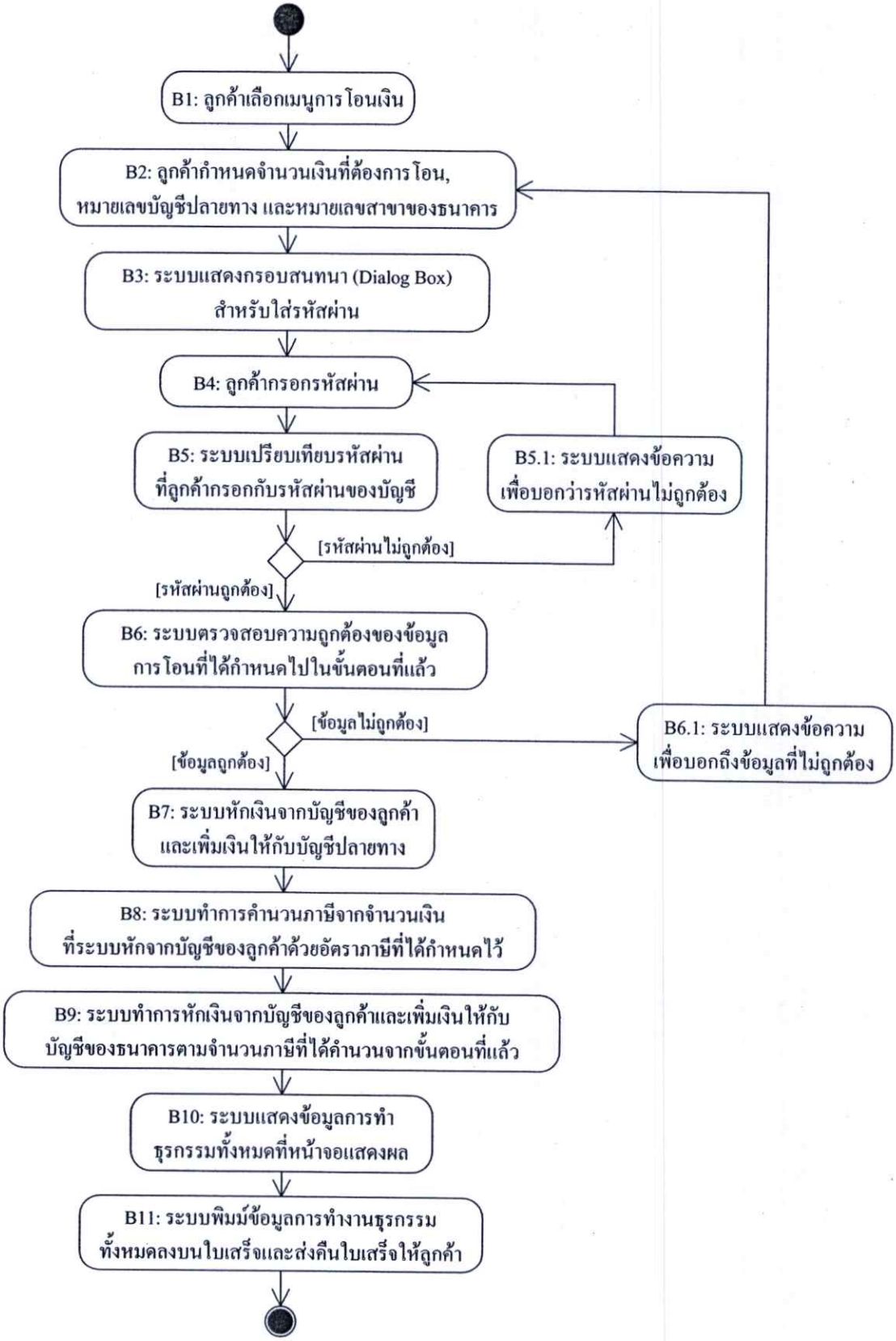
ชื่อยูสเคส	การตรวจสอบรหัสผ่าน
คำอธิบาย	ตรวจสอบรหัสผ่านของผู้มีสิทธิ์ใช้งาน
เงื่อนไขก่อน	-
จุดต่อขยาย	-
ยูสเคสที่ถูกใช้งาน	-
ลำดับเหตุการณ์	1. ระบบแสดงกรอบสนทนา (Dialog Box) สำหรับใส่รหัสผ่าน 2. ลูกค้ากรอกรหัสผ่าน 3. ระบบเปรียบเทียบรหัสผ่านที่ลูกค้ากรอกกับรหัสผ่านของบัญชี
ลำดับเหตุการณ์ทางเลือก	3.1 ลูกค้าใส่รหัสผ่านไม่ถูกต้อง 3.1.1 ระบบแสดงข้อความเพื่อบอกว่ารหัสผ่านไม่ถูกต้อง 3.1.2 กลับไปยังขั้นตอนที่ 2
เงื่อนไขหลัง	-

ตารางที่ 2.8 คำอธิบายยูสเคสการเรียกเก็บภาษี

ชื่อยูสเคส	การเรียกเก็บภาษี
คำอธิบาย	เรียกเก็บภาษีจากการทำธุรกรรมทางการเงิน
เงื่อนไขก่อน	-
จุดต่อขยาย	-
ยูสเคสที่ถูกใช้งาน	-
ลำดับเหตุการณ์	1. ระบบทำการคำนวนภาษีจากจำนวนเงินที่ระบบหักจากบัญชีของลูกค้าด้วยอัตราภาษีที่ได้กำหนดไว้ 2. ระบบทำการหักเงินจากบัญชีของลูกค้าและเพิ่มเงินให้กับบัญชีของธนาคารตามจำนวนภาษีที่ได้คำนวนจากขั้นตอนที่แล้ว
ลำดับเหตุการณ์ทางเลือก	-
เงื่อนไขหลัง	-

ตารางที่ 2.9 คำอธิบายยูสเคสการแสดงใบเสร็จ

ชื่อยูสเคส	การแสดงใบเสร็จ
คำอธิบาย	แสดงข้อมูลการทำธุรกรรมที่ลูกค้าได้ดำเนินการ
เงื่อนไขก่อน	-
จุดต่อขยาย	-
ยูสเคสที่ถูกใช้งาน	-
ลำดับเหตุการณ์	1. ระบบแสดงข้อมูลการทำธุรกรรมทั้งหมดที่หน้าจอแสดงผล 2. ระบบพิมม์ข้อมูลการทำงานธุรกรรมทั้งหมดลงบนใบเสร็จและส่งคืนใบเสร็จให้ลูกค้า
ลำดับเหตุการณ์ทางเลือก	-
เงื่อนไขหลัง	-



รูปที่ 2.15 แผนภาพกิจกรรมของยูสเคสการโอนเงิน

2.5 การสร้างกรณีทดสอบ

การทดสอบซอฟต์แวร์เป็นเครื่องมือที่มีความสำคัญในการแสดงข้อผิดพลาดในซอฟต์แวร์ และยังใช้เพื่อรับประกันว่าซอฟต์แวร์ที่ได้ถูกพัฒนาขึ้นนั้นสามารถทำงานได้บรรลุผลตามความต้องการของซอฟต์แวร์ [14][20]

กระบวนการพัฒนาซอฟต์แวร์โดยทั่วไปมักให้ความสนใจในการทดสอบซอฟต์แวร์ที่ขั้นตอนหลังจากที่ได้พัฒนาซอฟต์แวร์เสร็จสิ้นแล้ว ซึ่งจะเป็นความเสี่ยงที่ทำให้เกิดข้อผิดพลาดที่มากขึ้นและทำให้ขั้นตอนในการแก้ไขข้อผิดพลาดดังกล่าวมีความเกี่ยวพันมากขึ้น อีกทั้งเป็นการเพิ่มงบประมาณในการพัฒนาซอฟต์แวร์และทำให้ตารางการทำงานต้องถูกเลื่อนออกไปอีกด้วย [17][20] โดยปกติข้อกำหนดความต้องการของซอฟต์แวร์จะถูกแสดงผ่านแผนภาพยูสเคสและคำอธิบายยูสเคส ซึ่งยูสเคสจะถูกใช้กำหนดแต่ละฟังก์ชันการทำงานในซอฟต์แวร์ จากการที่แผนภาพยูสเคสถูกสร้างขึ้นก่อนกระบวนการพัฒนาซอฟต์แวร์และยังเป็นเครื่องมือที่ใช้สำหรับรับประกันว่าฟังก์ชันการทำงานของซอฟต์แวร์นั้นสนับสนุนทุกๆ ความต้องการ ทำให้ยูสเคสเหมาะสมที่จะเป็นจุดเริ่มต้นในขั้นตอนการสร้างกรณีทดสอบ ซึ่งการออกแบบการทดสอบในขั้นตอนก่อนกระบวนการพัฒนาเป็นการค้นหาข้อบกพร่องจากข้อกำหนดความต้องการที่เป็นพื้นฐานของการพัฒนาซอฟต์แวร์และทำให้ง่ายต่อการแก้ไขข้อบกพร่องนั้นด้วย [8][14][17][19][20]

ดังนั้นในส่วนนี้จึงนำเสนองานวิจัยที่เกี่ยวข้องกับการสร้างกรณีทดสอบซอฟต์แวร์ โดยให้ความสำคัญกับการสร้างกรณีทดสอบจากยูสเคส ซึ่งประกอบด้วยงานวิจัยต่างๆ ดังต่อไปนี้

1. Dave Wood และ Jim Reis [8] นำเสนอแนวทางการสร้างกรณีทดสอบด้วยยูสเคสขึ้นในปี ค.ศ.1999

1.1 ขั้นตอนการสร้างกรณีทดสอบ

ขั้นตอนที่ 1: สร้างแผนภาพยูสเคสจากข้อกำหนดความต้องการของซอฟต์แวร์ (Software Requirement Specification: SRS) และเขียนคำอธิบายยูสเคส (Use Case Description)

ขั้นตอนที่ 2: กำหนดซีแนริโอจากคำอธิบายยูสเคส โดยมีข้อกำหนดสำหรับการกำหนดซีแนริโอ คือ

ก) เลือกสายลำดับเหตุการณ์หลักที่ไม่ผ่านสายลำดับเหตุการณ์ทางเลือกเลย

ข) เลือกสายลำดับเหตุการณ์หลักที่ผ่านแต่ละสายลำดับเหตุการณ์ทางเลือก จนครบทุกสายลำดับเหตุการณ์ทางเลือก

ขั้นตอนที่ 3: สร้างกรณีทดสอบ (Test Case) จากแต่ละซีแนริโอที่ได้จากขั้นตอนก่อนหน้านี้

ขั้นตอนที่ 4: กรณีที่มียูสเคสจำนวนมาก ทำให้ซีแนริโอและกรณีทดสอบที่ได้จากยูสเคสเหล่านั้นมีจำนวนมากเช่นกัน ซึ่งเป็นไปได้ยากที่จะสามารถทดสอบได้ทุกกรณีทดสอบ จึงต้องมีการจัดลำดับความสำคัญของงาน (Prioritization) ให้กับกรณีทดสอบ โดยกรณีทดสอบที่จะต้องถูกทดสอบนั้นต้องสามารถทำการทดสอบได้จริงตามสายลำดับเหตุการณ์ของยูสเคส

1.2 ตัวอย่างการสร้างกรณีทดสอบ

ขั้นตอนที่ 1: กำหนดให้รูปที่ 2.7 ตารางที่ 2.6 ตารางที่ 2.7 ตารางที่ 2.8 และตารางที่ 2.9 เป็นแผนภาพยูสเคส และคำอธิบายยูสเคสสำหรับสร้างกรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีของ Dave Wood และ Jim Reis ในตัวอย่างนี้

ขั้นตอนที่ 2: จากคำอธิบายยูสเคสที่ได้กำหนดไว้ในขั้นตอนที่ 1 สามารถนำมากำหนดซีแนริโอของยูสเคสการโอนเงินด้วยข้อกำหนดของวิธีการนี้ที่ได้กำหนดไว้ได้ดังตารางที่ 2.10

ตารางที่ 2.10 ซีแนริโอที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Dave Wood และ Jim Reis

ชื่อซีแนริโอ	เส้นทางการทำงานเริ่มต้น	เส้นทางการทำงานทางเลือก
1. การโอนเงินประสบผลสำเร็จ	เส้นทางการทำงานหลัก	
2. รหัสผ่านไม่ถูกต้อง	เส้นทางการทำงานหลัก	ลูกค้าใส่รหัสผ่านไม่ถูกต้อง
3. ข้อมูลการโอนเงินไม่ถูกต้อง	เส้นทางการทำงานหลัก	ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง

ขั้นตอนที่ 3: จากซีแนริโอที่กำหนดได้จากขั้นตอนที่ 2 สามารถนำมาสร้างเป็นกรณีทดสอบของยูสเคสการโอนเงินได้ดังตารางที่ 2.11

ตารางที่ 2.11 กรณีทดสอบที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Dave Wood และ Jim Reis

ชื่อกรณีทดสอบ	รหัสผ่าน	ข้อมูลการโอนเงิน	ผลที่คาดว่าจะได้รับ
1. การโอนเงินประสบผลสำเร็จ	ถูกต้อง	ถูกต้อง	โอนเงินได้สำเร็จ
2. รหัสผ่านไม่ถูกต้อง	ไม่ถูกต้อง	ถูกต้อง	ลูกค้าต้องใส่รหัสผ่านอีกครั้ง
3. ข้อมูลการโอนเงินไม่ถูกต้อง	ถูกต้อง	ไม่ถูกต้อง	ลูกค้าต้องกรอกข้อมูลการโอนเงินอีกครั้ง

ขั้นตอนที่ 4: เนื่องจากกรณีทดสอบที่ได้จากขั้นตอนที่ 3 มีจำนวนน้อย จึงไม่จำเป็นต้องมีการจัดลำดับความสำคัญให้กรณีทดสอบ

2. Jim Heumann [17] นำเสนอการสร้างกรณีทดสอบจากยูสเคสขึ้นในปี ค.ศ. 2001

2.1 ขั้นตอนการสร้างกรณีทดสอบ

ขั้นตอนที่ 1: สร้างแผนภาพยูสเคสและคำอธิบายยูสเคสให้ถูกต้อง

ขั้นตอนที่ 2: สร้างซีแนริโอของยูสเคสจากคำอธิบายยูสเคส โดยมีข้อกำหนดสำหรับกำหนดซีแนริโอ คือ

ก) เลือกสายลำดับเหตุการณ์หลักที่ไม่ผ่านสายลำดับเหตุการณ์ทางเลือกเลย

ข) เลือกสายลำดับเหตุการณ์หลักที่ผ่านแต่ละสายลำดับเหตุการณ์ทางเลือก จนครบทุกสายลำดับเหตุการณ์ทางเลือก

ขั้นตอนที่ 3: กำหนดกรณีทดสอบจากซีแนริโอที่ได้จากขั้นตอนก่อนหน้านี้ และให้พิจารณาแต่ละซีแนริโอว่าสามารถแยกเป็นหลายกรณีทดสอบได้อีกหรือไม่ แล้วจึงทำการกำหนดตัวแปรข้อมูลที่ใช้สำหรับเป็นข้อมูลนำเข้าของกรณีทดสอบ ซึ่งสามารถพิจารณาได้จากคำอธิบายยูสเคส

ขั้นตอนที่ 4: กำหนดข้อมูลจริงที่จะทดสอบให้กับแต่ละกรณีทดสอบ

2.2 ตัวอย่างการสร้างกรณีทดสอบ

ขั้นตอนที่ 1: กำหนดให้รูปที่ 2.7 ตารางที่ 2.6 ตารางที่ 2.7 ตารางที่ 2.8 และตารางที่ 2.9 เป็นแผนภาพยูสเคส และคำอธิบายยูสเคสสำหรับสร้างกรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีของ Jim Heumann ในตัวอย่างนี้

ขั้นตอนที่ 2: จากคำอธิบายยูสเคสที่ได้กำหนดไว้ในขั้นตอนที่ 1 สามารถนำมากำหนดซีแนริโอของยูสเคสการโอนเงินด้วยข้อกำหนดของวิธีการนี้ที่ได้กำหนดไว้ได้ดังตารางที่ 2.12

ตารางที่ 2.12 ซีแนริโอที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Jim Heumann

ชื่อซีแนริโอ	เส้นทางการทำงานเริ่มต้น	เส้นทางการทำงานทางเลือก
1. การโอนเงินประสบผลสำเร็จ	เส้นทางการทำงานหลัก	
2. รหัสผ่านไม่ถูกต้อง	เส้นทางการทำงานหลัก	ลูกค้าใส่รหัสผ่านไม่ถูกต้อง
3. ข้อมูลการโอนเงินไม่ถูกต้อง	เส้นทางการทำงานหลัก	ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง

ขั้นตอนที่ 3: จากซีแนริโอที่กำหนดได้จากขั้นตอนที่ 2 สามารถนำมาสร้างเป็นกรณีทดสอบของยูสเคสการโอนเงิน และกำหนดตัวแปรข้อมูลที่ใช้สำหรับเป็นข้อมูลนำเข้าของกรณีทดสอบได้ดังตารางที่ 2.13

ตารางที่ 2.13 กรณีทดสอบที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Jim Heumann

ชื่อกรณีทดสอบ	รหัสผ่าน	ข้อมูลการโอนเงิน	ผลที่คาดว่าจะได้รับ
1. การโอนเงินประสบผลสำเร็จ	ถูกต้อง	ถูกต้อง	โอนเงินได้สำเร็จ
2. รหัสผ่านไม่ถูกต้อง	ไม่ถูกต้อง	ถูกต้อง	ลูกค้าต้องใส่รหัสผ่านอีกครั้ง
3. ข้อมูลการโอนเงินไม่ถูกต้อง	ถูกต้อง	ไม่ถูกต้อง	ลูกค้าต้องกรอกข้อมูลการโอนเงินอีกครั้ง

ขั้นตอนที่ 4: จากกรณีทดสอบที่ได้จากขั้นตอนที่ 3 สามารถนำมากำหนดข้อมูลจริงที่จะทดสอบสำหรับแต่ละกรณีทดสอบของยูสเคสการโอนเงินได้ดังตารางที่ 2.14

ตารางที่ 2.14 กรณีทดสอบที่ได้จากคำอธิบายยูสเคสการโอนเงินด้วยวิธีการของ Jim Heumann ซึ่งได้กำหนดข้อมูลจริงแล้ว

ชื่อกรณีทดสอบ	รหัสผ่าน	ข้อมูลการโอนเงิน	ผลที่คาดว่าจะได้รับ
1. การโอนเงินประสบผลสำเร็จ	abc123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003	โอนเงินได้สำเร็จ
2. รหัสผ่านไม่ถูกต้อง	123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003	ลูกค้าต้องใส่รหัสผ่านอีกครั้ง
3. ข้อมูลการโอนเงินไม่ถูกต้อง	abc123	- จำนวนเงิน: -2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: -	ลูกค้าต้องกรอกข้อมูลการโอนเงินอีกครั้ง

3. Perter Zielezynski [23] ได้นำเสนองานวิจัยนี้ในปี ค.ศ. 2006

3.1 ขั้นตอนการสร้างกรณีทดสอบ

ขั้นตอนที่ 1: สร้างแผนภาพยูสเคสและคำอธิบายยูสเคสให้ถูกต้อง

ขั้นตอนที่ 2: สร้างแผนภาพกิจกรรมจากคำอธิบายยูสเคส

ขั้นตอนที่ 3: กำหนดซีแนริโอจากแผนภาพกิจกรรมที่ได้จากขั้นตอนก่อนหน้านี้ โดยข้อกำหนดสำหรับกำหนดซีแนริโอ คือ

ก) ซีแนริโอที่เป็นสายลำดับเหตุการณ์หลัก

ข) ซีแนริโอที่ประกอบด้วยสายลำดับเหตุการณ์หลักกับแต่ละสายลำดับการทำงานเหตุการณ์ทางเลือก

ค) ซีแนริโอที่เกิดจากการรวมกันของสายลำดับเหตุการณ์หลักกับหลายสายลำดับเหตุการณ์ทางเลือกอย่างสมเหตุสมผล

ขั้นตอนที่ 4: กำหนดตัวแปรข้อมูลของทุกๆ ขั้นตอนในซีแนริโอ

ขั้นตอนที่ 5: กำหนดข้อจำกัดให้กับแต่ละข้อมูลที่จะใช้งาน

ขั้นตอนที่ 6: กำหนดข้อมูลจริงสำหรับทดสอบ

3.2 ตัวอย่างการสร้างกรณีทดสอบ

ขั้นตอนที่ 1: กำหนดให้รูปที่ 2.7 ตารางที่ 2.6 ตารางที่ 2.7 ตารางที่ 2.8 และตารางที่ 2.9 เป็นแผนภาพยูสเคส และคำอธิบายยูสเคสสำหรับสร้างกรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีของ Perter Zielezynski ในตัวอย่างนี้

ขั้นตอนที่ 2: สามารถสร้างแผนภาพกิจกรรมของยูสเคสการโอนเงินจากคำอธิบายยูสเคสการโอนเงินที่ได้กำหนดไว้ในขั้นตอนที่ 1 ได้ดังรูปที่ 2.15

ขั้นตอนที่ 3: กำหนดซีแนริโอของยูสเคสการโอนเงินจากแผนภาพกิจกรรมที่ได้จากขั้นตอนที่ 2 ด้วยข้อกำหนดของวิธีการนี้ที่ได้กำหนดไว้ได้ ดังตารางที่ 2.15

ตารางที่ 2.15 ซีแนริโอที่ได้จากแผนภาพกิจกรรมของยูสเคสการโอนเงินด้วยวิธีการของ Peter Zielezynski

ชื่อซีแนริโอ	เส้นทางการทำงานเริ่มต้น	เส้นทางการทำงานทางเลือก
1. การโอนเงินประสบผลสำเร็จ	เส้นทางการทำงานหลัก	
2. รหัสผ่านไม่ถูกต้อง	เส้นทางการทำงานหลัก	ลูกค้าใส่รหัสผ่านไม่ถูกต้อง
3. ข้อมูลการโอนเงินไม่ถูกต้อง	เส้นทางการทำงานหลัก	ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง
4. รหัสผ่านไม่ถูกต้อง และ ข้อมูลการโอนเงินไม่ถูกต้อง	เส้นทางการทำงานหลัก	ลูกค้าใส่รหัสผ่านไม่ถูกต้อง และ ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง

ขั้นตอนที่ 4: กำหนดตัวแปรข้อมูลของทุกๆ ขั้นตอนในซีแนริโอที่ได้จากขั้นตอนที่ 3 ได้ดังตารางที่ 2.16

ตารางที่ 2.16 กำหนดตัวแปรข้อมูลให้กับซีแนริโอที่ได้จากแผนภาพกิจกรรมของยูสเคสการโอนเงินด้วยวิธีการของ Peter Zielezynski

ชื่อซีแนริโอ	รหัสผ่าน	ข้อมูลการโอนเงิน
1. การโอนเงินประสบผลสำเร็จ	ถูกต้อง	ถูกต้อง
2. รหัสผ่านไม่ถูกต้อง	ไม่ถูกต้อง	ถูกต้อง
3. ข้อมูลการโอนเงินไม่ถูกต้อง	ถูกต้อง	ไม่ถูกต้อง
4. รหัสผ่านไม่ถูกต้อง และข้อมูลการโอนเงินไม่ถูกต้อง	ไม่ถูกต้อง	ไม่ถูกต้อง

ขั้นตอนที่ 5: กำหนดข้อจำกัดให้กับแต่ละข้อมูลที่จะใช้งานได้ดังนี้ คือ

1. รหัสผ่าน: กำหนดให้เป็นสายอักขระของตัวอักษรหรือตัวเลขที่มีความยาวไม่เกิน 6 ตัวอักษร และไม่น้อยกว่า 3 ตัวอักษร

2. ข้อมูลการโอนเงิน

- จำนวนเงินที่ต้องการโอน: กำหนดให้เป็นตัวเลข และจำนวนเงินที่ต้องการโอนจะต้องไม่มีค่าเป็นลบ

- หมายเลขบัญชีปลายทาง: กำหนดให้เป็นสายอักขระของตัวเลขที่มีรูปแบบดังนี้ คือ xxx-xxxx-xx

- หมายเลขสาขาของธนาคาร: กำหนดให้เป็นสายอักขระของตัวเลขที่มีความยาว 3 ตัวอักษร

ขั้นตอนที่ 6: กำหนดข้อมูลจริงสำหรับทดสอบของกรณีทดสอบของยูสเคสการโอนเงินได้ดังตารางที่ 2.17

ตารางที่ 2.17 กรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีการของ Peter Zielezynski

ชื่อกรณีทดสอบ	รหัสผ่าน	ข้อมูลการโอนเงิน	ผลที่คาดว่าจะได้รับ
1. การโอนเงินประสบผลสำเร็จ	abc123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003	โอนเงินได้สำเร็จ
2. รหัสผ่านไม่ถูกต้อง	123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003	ลูกค้าต้องใส่รหัสผ่านอีกครั้ง

ตารางที่ 2.17 (ต่อ)

3. ข้อมูลการโอนเงินไม่ถูกต้อง	abc123	- จำนวนเงิน: -2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: -	ลูกค้าต้องกรอกข้อมูลการโอนเงินอีกครั้ง
4. รหัสผ่านไม่ถูกต้อง และข้อมูลการโอนเงินไม่ถูกต้อง	123	- จำนวนเงิน: -2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: -	ลูกค้าต้องใส่รหัสผ่านอีกครั้งและลูกค้าต้องกรอกข้อมูลการโอนเงินอีกครั้ง

4. Javier J. Gutierrez และคณะ [14] นำเสนอแนวทางการสร้างกรณีทดสอบด้วยยูสเคสขึ้นในปี ค.ศ. 2006

4.1 ขั้นตอนการสร้างกรณีทดสอบ

ขั้นตอนที่ 1: สร้างแผนภาพยูสเคสและคำอธิบายยูสเคสให้ถูกต้อง

ขั้นตอนที่ 2: สร้างแผนภาพกิจกรรมจากคำอธิบายยูสเคส

ขั้นตอนที่ 3: กำหนดซีแนริโอของยูสเคสจากแผนภาพกิจกรรมที่ได้จากขั้นตอนก่อนหน้านี้ด้วย ราวนด์ทริป แพทเทิร์น (Roundtrip Pattern) ซึ่งมีข้อกำหนด คือ

ก) เลือกสายลำดับเหตุการณ์หลัก โดยไม่ผ่านสายลำดับเหตุการณ์ที่เกิดการวนซ้ำ

ข) เลือกสายลำดับเหตุการณ์หลักที่ผ่านแต่ละสายลำดับเหตุการณ์ทางเลือก โดยไม่ผ่านสายลำดับเหตุการณ์ที่เกิดการวนซ้ำ จนครบทุกสายลำดับเหตุการณ์ทางเลือก

ค) เลือกสายลำดับเหตุการณ์ที่ผ่านแต่ละสายลำดับเหตุการณ์ที่เกิดการวนซ้ำ 1 เหตุการณ์ และกำหนดให้แต่ละสายลำดับเหตุการณ์ที่ถูกเลือกเกิดการวนซ้ำ 1 ครั้ง และเกิดการวนซ้ำด้วยจำนวนครั้งที่มากที่สุด

ขั้นตอนที่ 4: สร้างข้อมูลที่ใช้ทดสอบ

ขั้นตอนที่ 5: สร้างกรณีทดสอบ

ขั้นตอนที่ 6: สร้างผลลัพธ์ที่คาดว่าจะได้รับ

4.2 ตัวอย่างการสร้างกรณีทดสอบ

ขั้นตอนที่ 1: กำหนดให้รูปที่ 2.7 ตารางที่ 2.6 ตารางที่ 2.7 ตารางที่ 2.8 และตารางที่ 2.9 เป็นแผนภาพยูสเคส และคำอธิบายยูสเคสสำหรับสร้างกรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีของ Javier J. Gutierrez และคณะ ในตัวอย่างนี้

ขั้นตอนที่ 2: สามารถสร้างแผนภาพกิจกรรมของยูสเคสการโอนเงินจากคำอธิบายยูสเคสการโอนเงินที่ได้กำหนดไว้ในขั้นตอนที่ 1 ได้ดังรูปที่ 2.15

ขั้นตอนที่ 3: กำหนดซีแนริโอของยูสเคสการโอนเงินจากแผนภาพกิจกรรมที่ได้จากขั้นตอนที่ 2 ด้วยข้อกำหนดของวิธีการนี้ที่ได้กำหนดไว้ได้ ดังตารางที่ 2.18

ตารางที่ 2.18 ซีแนริโอที่ได้จากแผนภาพกิจกรรมของยูสเคสการโอนเงินด้วยวิธีการของ Javier J. Gutierrez และคณะ

ชื่อซีแนริโอ	เส้นทางการทำงานเริ่มต้น	เส้นทางการทำงานทางเลือก
1. การโอนเงินประสบผลสำเร็จ	เส้นทางการทำงานหลัก	
2. รหัสผ่านไม่ถูกต้อง, ผ่านการวนซ้ำ 1 ครั้ง	เส้นทางการทำงานหลัก	ลูกค้าใส่รหัสผ่านไม่ถูกต้อง
3. รหัสผ่านไม่ถูกต้อง, ผ่านการวนซ้ำด้วยจำนวนครั้งมากที่สุด	เส้นทางการทำงานหลัก	ลูกค้าใส่รหัสผ่านไม่ถูกต้อง
4. ข้อมูลการโอนเงินไม่ถูกต้อง, ผ่านการวนซ้ำ 1 ครั้ง	เส้นทางการทำงานหลัก	ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง
5. ข้อมูลการโอนเงินไม่ถูกต้อง, ผ่านการวนซ้ำด้วยจำนวนครั้งมากที่สุด	เส้นทางการทำงานหลัก	ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง

ขั้นตอนที่ 4: สามารถกำหนดข้อมูลที่ใช้ทดสอบได้ดังตารางที่ 2.19

ตารางที่ 2.19 กำหนดข้อมูลสำหรับซีแนริโอของยูสเคสการโอนเงินด้วยวิธีการของ Javier J. Gutierrez และคณะ

ชื่อซีแนริโอ	รหัสผ่าน	ข้อมูลการโอนเงิน
1. การโอนเงินประสบผลสำเร็จ	ถูกต้อง	ถูกต้อง
2. รหัสผ่านไม่ถูกต้อง, ผ่านการวนซ้ำ 1 ครั้ง	ไม่ถูกต้อง	ถูกต้อง
3. รหัสผ่านไม่ถูกต้อง, ผ่านการวนซ้ำด้วยจำนวนครั้งมากที่สุด	ไม่ถูกต้อง	ถูกต้อง
4. ข้อมูลการโอนเงินไม่ถูกต้อง, ผ่านการวนซ้ำ 1 ครั้ง	ถูกต้อง	ไม่ถูกต้อง
5. ข้อมูลการโอนเงินไม่ถูกต้อง, ผ่านการวนซ้ำด้วยจำนวนครั้งมากที่สุด	ถูกต้อง	ไม่ถูกต้อง

ขั้นตอนที่ 5: จากซีแนริโอที่ได้จากขั้นตอนก่อนหน้านี้สามารถนำมาสร้างกรณีทดสอบของยูสเคสการโอนเงินได้ดังตารางที่ 2.20

ตารางที่ 2.20 กรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีการของ Javier J. Gutierrez และคณะ

ชื่อกรณีทดสอบ	รหัสผ่าน	ข้อมูลการโอนเงิน
1. การโอนเงินประสบผลสำเร็จ	abc123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003
2. รหัสผ่านไม่ถูกต้อง, ผ่านการวนซ้ำ 1 ครั้ง	123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003
3. รหัสผ่านไม่ถูกต้อง, ผ่านการวนซ้ำด้วยจำนวนครั้งมากที่สุด	123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003
4. ข้อมูลการโอนเงินไม่ถูกต้อง, ผ่านการวนซ้ำ 1 ครั้ง	abc123	- จำนวนเงิน: -2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: -
5. ข้อมูลการโอนเงินไม่ถูกต้อง, ผ่านการวนซ้ำด้วยจำนวนครั้งมากที่สุด	abc123	- จำนวนเงิน: -2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: -

ขั้นตอนที่ 6: สร้างผลลัพธ์ที่คาดว่าจะได้รับของแต่ละกรณีทดสอบของยูสเคสการโอนเงินที่ได้จากขั้นตอนที่ 5 ได้ดังตารางที่ 2.21

ตารางที่ 2.21 กำหนดผลลัพธ์ที่คาดว่าจะได้รับของแต่ละกรณีทดสอบของยูสเคสการโอนเงินด้วยวิธีการของ Javier J. Gutierrez และคณะ

ชื่อกรณีทดสอบ	รหัสผ่าน	ข้อมูลการโอนเงิน	ผลที่คาดว่าจะได้รับ
1. การโอนเงินประสบผลสำเร็จ	abc123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003	โอนเงินได้สำเร็จ
2. รหัสผ่านไม่ถูกต้อง, ผ่านการวนซ้ำ 1 ครั้ง	123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003	ลูกค้าต้องใส่รหัสผ่านอีกครั้ง
3. รหัสผ่านไม่ถูกต้อง, ผ่านการวนซ้ำด้วยจำนวนครั้งมากที่สุด	123	- จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: 003	ลูกค้าต้องใส่รหัสผ่านอีกครั้ง
4. ข้อมูลการโอนเงินไม่ถูกต้อง, ผ่านการวนซ้ำ 1 ครั้ง	abc123	- จำนวนเงิน: -2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: -	ลูกค้าต้องกรอกข้อมูลการโอนเงินอีกครั้ง
5. ข้อมูลการโอนเงินไม่ถูกต้อง, ผ่านการวนซ้ำด้วยจำนวนครั้งมากที่สุด	abc123	- จำนวนเงิน: -2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขาของธนาคาร: -	ลูกค้าต้องกรอกข้อมูลการโอนเงินอีกครั้ง

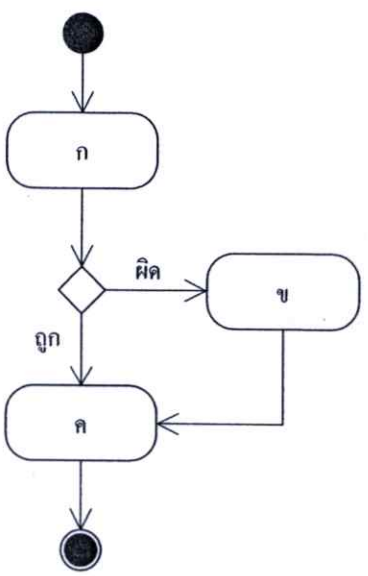
2.6 การทดสอบความครอบคลุมของกรณีทดสอบ

วิธีการทดสอบเส้นทางการทำงานเป็นการทดสอบโครงสร้างของซอฟต์แวร์ เพื่อให้แน่ใจว่าทุกๆ ส่วนในการทำงานของซอฟต์แวร์ได้ถูกทดสอบ การทำงานของซอฟต์แวร์สามารถถูกอธิบายได้ด้วยแผนภาพที่แสดงถึงขั้นตอนการทำงานของซอฟต์แวร์ เช่น กราฟการไหล (Flow Graph) และแผนภาพกิจกรรม เป็นต้น ซึ่งการอธิบายการทำงานของซอฟต์แวร์ด้วยวิธีดังกล่าวจะ

ทำให้ง่ายต่อการกำหนดเส้นทางการทำงานของซอฟต์แวร์ และด้วยแต่ละเส้นทางการทำงานของซอฟต์แวร์เป็นการแสดงถึงขั้นตอนการทำงานของซอฟแวร์ เราจึงสามารถนำเส้นทางการทำงานเหล่านี้มาสร้างเป็นกรณีทดสอบ เพื่อทดสอบการทำงานของซอฟต์แวร์สำหรับเส้นทางการทำงานนั้นๆ ได้ [15][21][34] การทดสอบเส้นทางการทำงานที่สมบูรณ์คือเส้นทางการทำงานของซอฟต์แวร์ทั้งหมดต้องถูกทดสอบ ซอฟต์แวร์ขนาดเล็กจะมีจำนวนเส้นทางการทำงานจำนวนน้อย ซึ่งสามารถนำทุกๆ เส้นทางการทำงานมาทำการทดสอบได้ แต่สำหรับซอฟต์แวร์ที่มีขนาดใหญ่ซึ่งจะมีจำนวนเส้นทางการทำงานจำนวนมาก ทำให้เป็นการยากและเป็นการสูญเสียทรัพยากรจำนวนมากในการทดสอบเส้นทางการทำงานทั้งหมด ซึ่งจะเป็นประโยชน์อย่างมากถ้าสามารถเลือกเส้นทางการทำงานของซอฟต์แวร์จำนวนน้อยแต่สามารถทำการทดสอบการทำงานของซอฟต์แวร์ได้อย่างครอบคลุมทุกๆ ส่วน การทดสอบเส้นทางการทำงานสามารถแบ่งออกได้เป็นหลายวิธีดังต่อไปนี้

2.6.1 การครอบคลุมคำสั่ง (Statement Coverage)

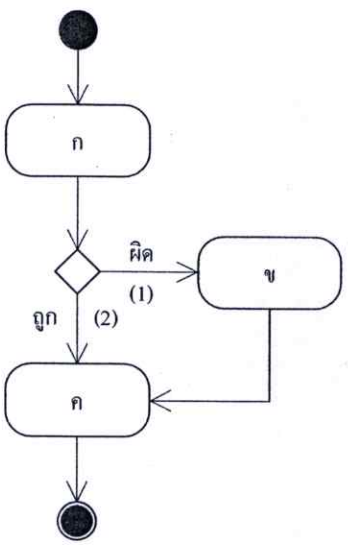
การครอบคลุมคำสั่ง [15][18][30][34][36] เป็นการทดสอบเส้นทางการทำงานของซอฟต์แวร์เพื่อให้ครอบคลุมทุกคำสั่ง โดยที่ทุกๆ คำสั่งจะต้องถูกทำงานอย่างน้อยหนึ่งครั้ง แต่ข้อบกพร่องของวิธีการนี้คือสามารถก่อให้เกิดความผิดพลาดในการทดสอบเส้นทางการทำงานของประโยคเงื่อนไข (If Statement) จากรูปที่ 2.16 เส้นทางการทำงาน ก, ข, ค เป็นเส้นทางการทำงานที่ครอบคลุมทุกคำสั่ง และเมื่อนำผลลัพธ์ดังกล่าวมาสร้างกรณีทดสอบจะทำให้เป็นการทดสอบเพียงเส้นทางเดียว อีกหนึ่งเส้นทางที่เป็นเส้นทางการทำงาน ก, ค ไม่ได้นำมาสร้างเป็นกรณีทดสอบ ซึ่งถ้าหากมีข้อผิดพลาดในเส้นทางการทำงานนี้ก็จะทำให้ไม่สามารถหาข้อผิดพลาดนี้พบ



รูปที่ 2.16 แผนภาพกิจกรรมแสดงการทำงานของประโยคเงื่อนไข (If Statement)

2.6.2 การครอบคลุมทางแยก (Branch/Decision Coverage)

การครอบคลุมทางแยก [15][18][30][34][36] เป็นการทดสอบเส้นทางการทำงานของซอฟต์แวร์เพื่อให้ครอบคลุมทุกทางแยก (Branch) ที่เกิดจากการตัดสินใจ (Decision) ซึ่งทุกทางแยกของการติดสินใจจะต้องถูกทำงานอย่างน้อยหนึ่งครั้ง การตัดสินใจเป็นผลมาจากหนึ่งเงื่อนไขหรือกลุ่มเงื่อนไขที่เกิดจากหลายๆ เงื่อนไขรวมกัน โดยวิธีการนี้จะพิจารณาว่ากลุ่มเงื่อนไขเป็นหนึ่งเงื่อนไข ซึ่งจะทำให้ผลลัพธ์ของการตัดสินใจจะมีแค่สองทางแยกเหมือนผลลัพธ์ของการตัดสินใจที่มีหนึ่งเงื่อนไข สำหรับทางแยกจะหมายถึงเส้นเชื่อมขาออก (Outgoing Edge) ซึ่งเป็นการส่งผ่านเงื่อนไขของการควบคุมการทำงานในกราฟแสดงการไหล หรือ แผนภาพกิจกรรม สามารถแสดงแทนด้วยลูกศรชี้ออกจากโหนด (Node) การตัดสินใจ วิธีการนี้สามารถช่วยแก้ไขข้อผิดพลาดของการครอบคลุมคำสั่งได้ โดยกำหนดทางแยกที่ 1 ให้กับเส้นเชื่อมขาออกที่ชี้ไปที่การทำงาน ข และกำหนดทางแยกที่ 2 ให้กับเส้นเชื่อมขาออกที่ใช้ไปที่การทำงาน ก ทั้งสองทางแยกเป็นผลของการตัดสินใจในประโยคเงื่อนไข ทำให้สามารถกำหนดเส้นทางการทำงานได้เป็นสองเส้นทางคือ ก, ข, ค และ ก, ค จึงทำให้สามารถสร้างเป็นกรณีทดสอบได้สองเส้นทางซึ่งครอบคลุมทุกทางแยก ดังรูปที่ 2.17



รูปที่ 2.17 แผนภาพกิจกรรมแสดงการทำงานของประโยคเงื่อนไข (If Statement)

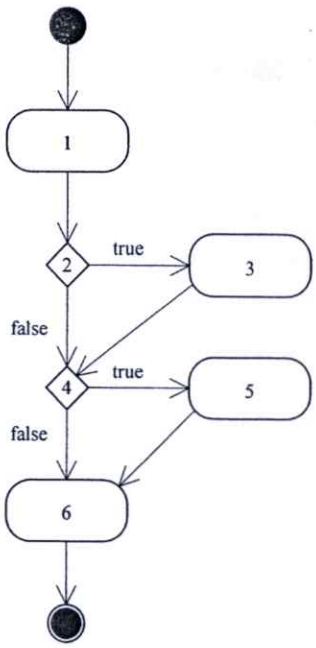
```
1       Int foo(int x) {
2               if (a = = b || (x = = y && isEmpty())) {
3                       ++x;
4               }
```

```
5               else {
6                       --x;
7               }
8               Return x;
9       }
```

จากเมธอด foo ทางด้านบน ++x จะทำงานเมื่อ a = = b โดยค่าของ x, y และ เมธอด this.isEmpty() จะเป็นอะไรก็ได้ และ --x จะทำงานเมื่อค่าของ a ไม่เท่ากับค่าของ b และค่าของ x ไม่เท่ากับค่าของ y โดยค่าของเมธอด this.isEmpty() จะเป็นอะไรก็ได้ ทำให้สามารถกำหนดเส้นทางการทำงานที่ครอบคลุมทุกๆ ทางแยกของการตัดสินใจได้โดยเมธอด this.isEmpty() ไม่ได้ถูกทำงานเลย กรณีดังกล่าวเป็นข้อบกพร่องหนึ่งของวิธีการนี้ที่อาจเกิดขึ้นได้ถ้ามีการใช้การประเมินแบบทางลัด (Short Circuit Evaluation)

ด้วยวิธีการนี้ได้พิจารณาให้กลุ่มเงื่อนไขเป็นหนึ่งเงื่อนไข ซึ่งอาจจะทำให้บางผลลัพธ์ของการตัดสินใจไม่ได้ถูกนำมาพิจารณา ดังเมธอด maxPositive() และรูปที่ 2.18 ซึ่งแสดงแผนภาพกิจกรรมของเมธอด maxPositive() เส้นทางการทำงาน 1, 2, 3, 4, 5, 6 ที่เป็นผลมาจาก n1 = n2 = 1 และเส้นทางการทำงาน 1, 2, 4, 6 ที่เป็นผลมาจาก n1 = 2, n2 = 0 ทำให้ทั้งสองเส้นทางนี้ครอบคลุมทุกทางแยก แต่บางผลลัพธ์ของการตัดสินใจไม่ได้ถูกนำมาพิจารณา เช่น กรณีที่ n1 = 0, n2 = 2

```
1       int maxPositive (int n1, int n2) {
2               if (n1 <= 1 && n2 <= 1)
3                       n1 = 1;
4               if (n1 <= n2)
5                       n1 = n2;
6               return n1;              }
```

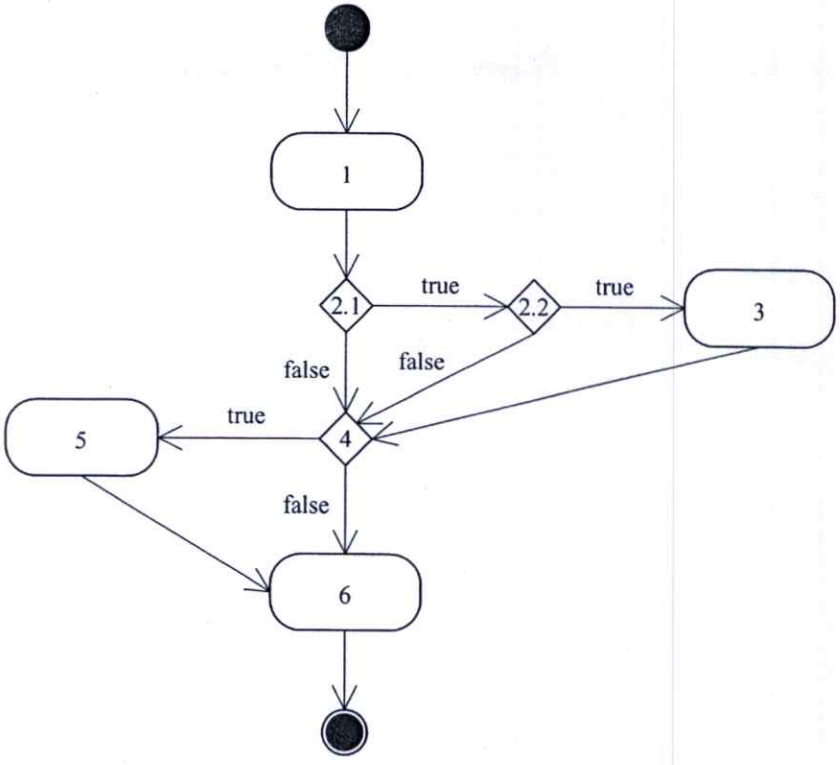


รูปที่ 2.18 แผนภาพกิจกรรมแสดงการทำงานของเมธอด maxPositive()

2.6.3 การครอบคลุมเงื่อนไข (Condition Coverage)

การครอบคลุมเงื่อนไข [15][30][34][36] เป็นการทดสอบเส้นทางการทำงานของซอฟต์แวร์เพื่อให้ครอบคลุมทุกเงื่อนไขในซอฟต์แวร์ โดยที่ทุกๆ เงื่อนไขจะต้องถูกทำงานอย่างน้อยหนึ่งครั้ง ซึ่งจะเป็นการแก้ไขข้อบกพร่องของ Branch/Decision Coverage ที่บางเงื่อนไขในกลุ่มเงื่อนไขไม่ได้ถูกนำมาพิจารณา จากเมธอด maxPositive() ก่อนหน้านี้สามารถเขียนใหม่เป็น maxPositive2() โดยแยกเงื่อนไขที่อยู่ในกลุ่มเงื่อนไขออกมาเป็นเงื่อนไขเดี่ยวๆ และทำการเชื่อมโยงเงื่อนไขที่ได้ถูกแยกออกมาด้วยประโยคเงื่อนไข และสามารถเขียนเป็นแผนภาพกิจกรรม ดังรูปที่ 2.19

```
1       int maxPositive2 (int n1, int n2) {
2.1             if (n1 <= 1) {
2.2                     if (n2 <= 1)
3                               n1 = 1;  }
4               if (n1 <= n2)
5                       n1 = n2;
6               return n1;              }
```



รูปที่ 2.19 แผนภาพกิจกรรมแสดงการทำงานของเมธอด maxPositive2()

จากเมธอด maxPositive2() และรูปที่ 2.19 สามารถกำหมดเส้นทางการทำงานที่ครอบคลุมทุกเงื่อนไขคือ เส้นทางการทำงาน 1, 2.1, 2.2, 3, 4, 5, 6 ที่เป็นผลมาจาก n1 = n2 = 1, เส้นทางการทำงาน 1, 2.1, 4, 6 ที่เป็นผลมาจาก n1 = 2, n2 = 0 และเส้นทางการทำงาน 1, 2.1, 2.2, 4, 5, 6 ที่เป็นผลมาจาก n1 = 0, n2 = 2

2.6.4 การทดสอบเบซิสพาท (Basis Path Testing)

การทดสอบเส้นทางการทำงาน 3 วิธีก่อนหน้านี้ให้ความสนใจกับส่วนประกอบพื้นฐานในแผนภาพแสดงการทำงานของซอฟต์แวร์ คือ คำสั่ง ทางแยก และเงื่อนไข เส้นทางการทำงานที่เป็นผลมาจากวิธีการดังกล่าวไม่สามารถทำให้ครอบคลุมทุกเส้นทางการทำงาน และด้วยซอฟต์แวร์หนึ่งๆ สามารถมีเส้นทางการทำงานจำนวนมากหรือมีจำนวนไม่จำกัด ซึ่งยากลำบากและสิ้นเปลืองทรัพยากรในการนำมาทดสอบให้ครบทุกเส้นทาง วิธีการทดสอบเบซิสพาท [15][18][30] จะทำการเลือกตัวแทนของเส้นทางการทำงานทั้งหมดเพื่อนำเส้นทางการทำงานดังกล่าวไปสร้างเป็นกรณีทดสอบ ตัวแทนของเส้นทางการทำงานแต่ละเส้นทางจะถูกเรียกว่า เส้นทางอิสระเชิงเส้น (Linearly Independent Path) และกลุ่มของเส้นทางอิสระเชิงเส้นจะถูกเรียกว่า เบซิสเซต (Basis Set) โดยเส้นทางอิสระเชิงเส้น คือ เส้นทางการทำงานของซอฟต์แวร์ที่มีเส้นเชื่อม (Edge) ใหม่ที่ไม่มีในเส้นทางอิสระเชิงเส้นอื่นๆ ซึ่งเส้นทางอิสระเชิงเส้นที่อยู่ในเบซิสเซตสามารถนำมาแสดงแทน

เส้นทางการทำงานอื่นๆ ด้วยผลรวมเชิงเส้น (Linear Combination) ของเส้นทางอิสระเชิงเส้นนั้นเอง และแต่ละเส้นทางอิสระเชิงเส้นภายในเบซิสเซตจะไม่สามารถถูกแสดงแทนได้ด้วยผลรวมเชิงเส้นจากเส้นทางอิสระเชิงเส้นที่เหลือได้ จำนวนของเส้นทางอิสระเชิงเส้นสามารถคำนวนได้จากความซับซ้อนไซโคลเมติก (Cyclomatic Complexity) ของกราฟซึ่งเป็นมาตรวัดความซับซ้อนของซอฟต์แวร์และมีสูตรในการคำนวนดังนี้

$$V(G) = E - V + 2$$

กำหนดให้

E = จำนวนเส้นเชื่อม

V = จำนวนโหนด

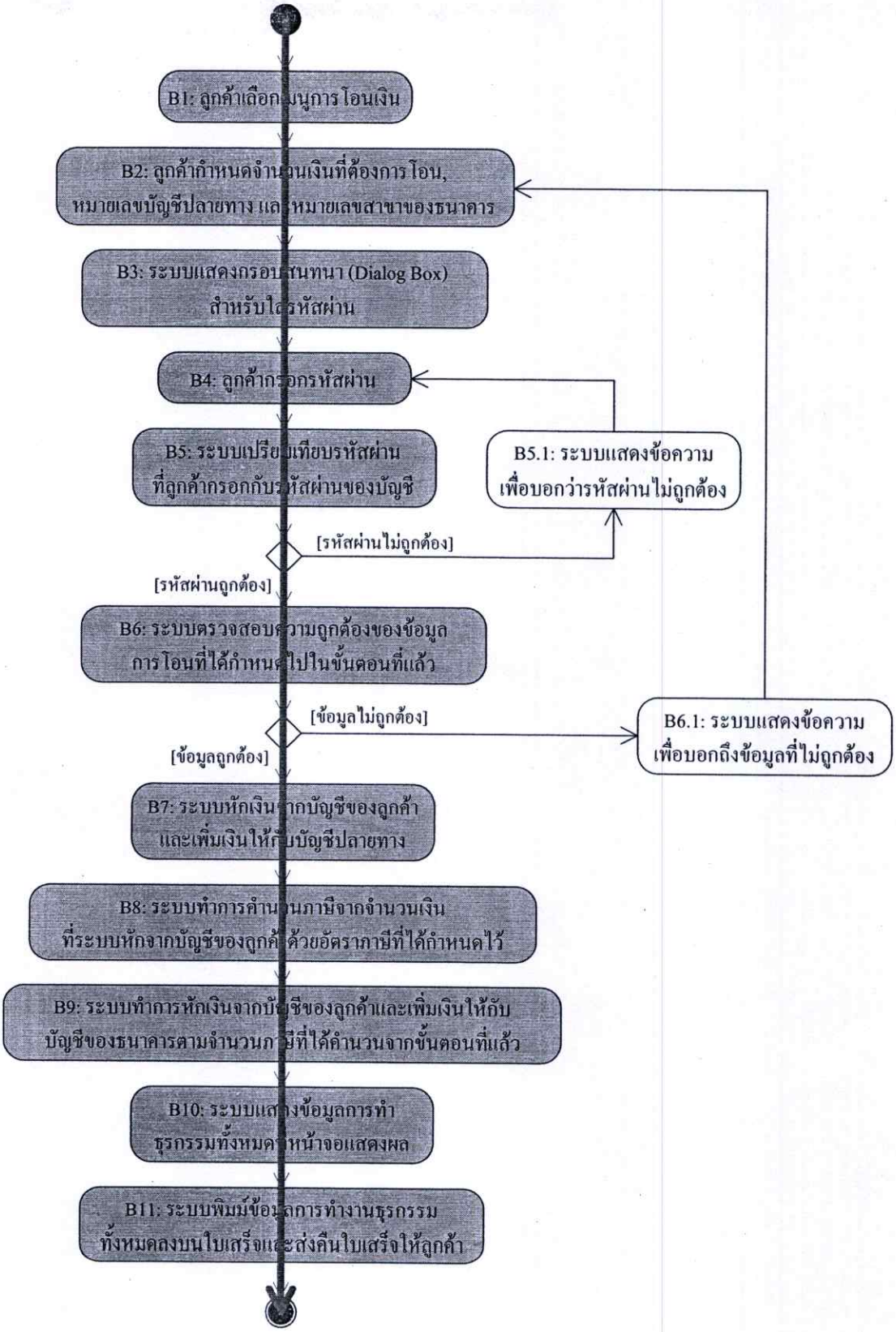
จากแผนภาพกิจกรรม ดังรูปที่ 2.15 เป็นแผนภาพกิจกรรมของยูสเคสการโอนเงินที่สร้างขึ้นจากคำอธิบายยูสเคสการโอนเงิน ดังตารางที่ 2.6 สามารถนำมาคำนวนความซับซ้อนไซโคลเมติกเพื่อหาจำนวนเส้นทางอิสระเชิงเส้นของแผนภาพกิจกรรมนี้ได้ คือ

$$\begin{aligned} V(G) &= E - V + 2 \\ &= 14 - 13 + 2 \\ &= 3 \end{aligned}$$

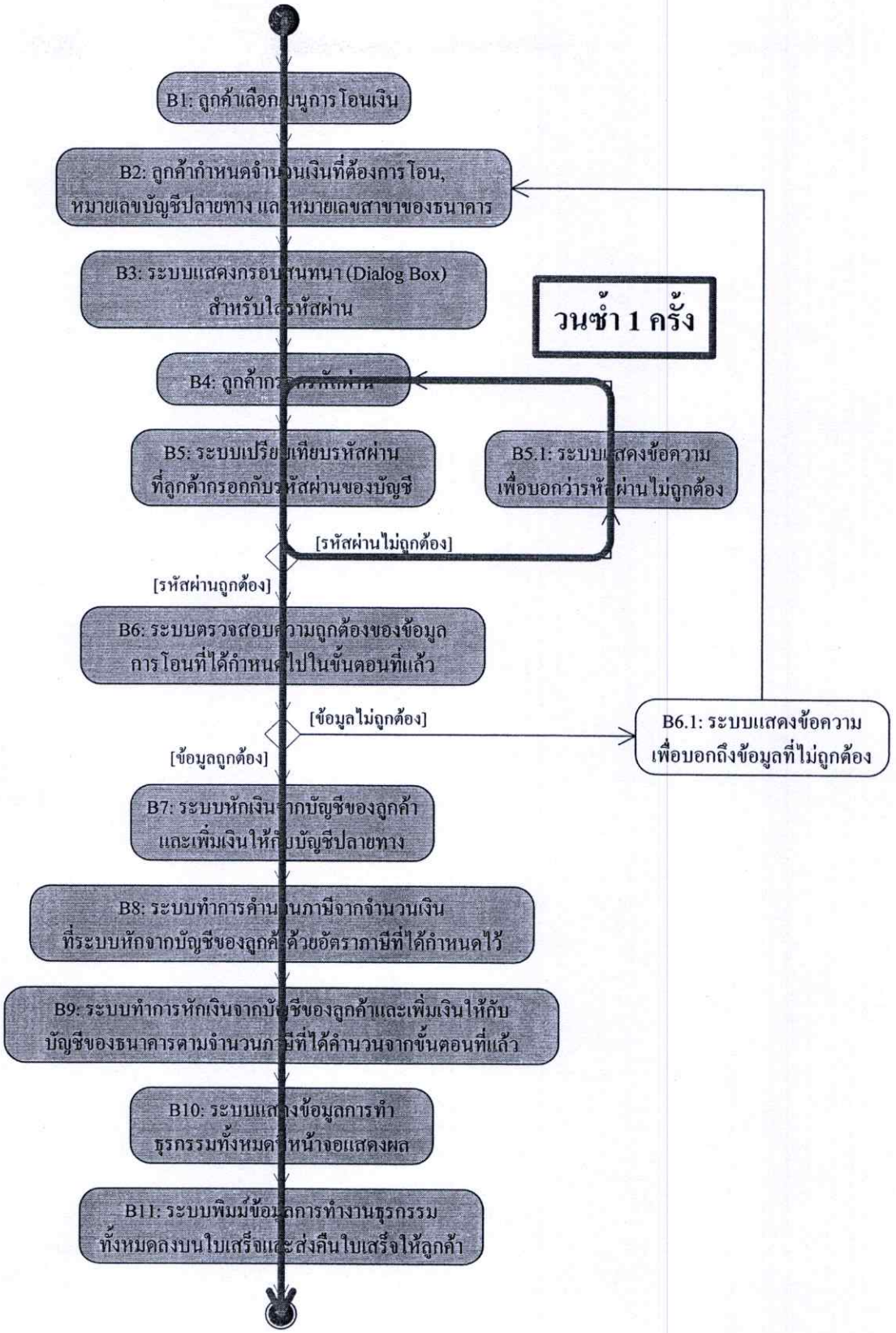
ผลของการคำนวนได้จำนวนเส้นทางอิสระเชิงเส้นของแผนภาพกิจกรรมยูสเคสการโอนเงินจำนวน 3 เส้นทาง และจากแผนภาพกิจกรรมยูสเคสการโอนเงินสามารถนำมาค้นหาเส้นทางอิสระเชิงเส้นได้ คือ

- เส้นทางเชิงอิสระ 1: B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11
- เส้นทางเชิงอิสระ 2: B1, B2, B3, B4, B5, B5.1, B4, B5, B6, B7, B8, B9, B10, B11
- เส้นทางเชิงอิสระ 3: B1, B2, B3, B4, B5, B6, B6.1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11

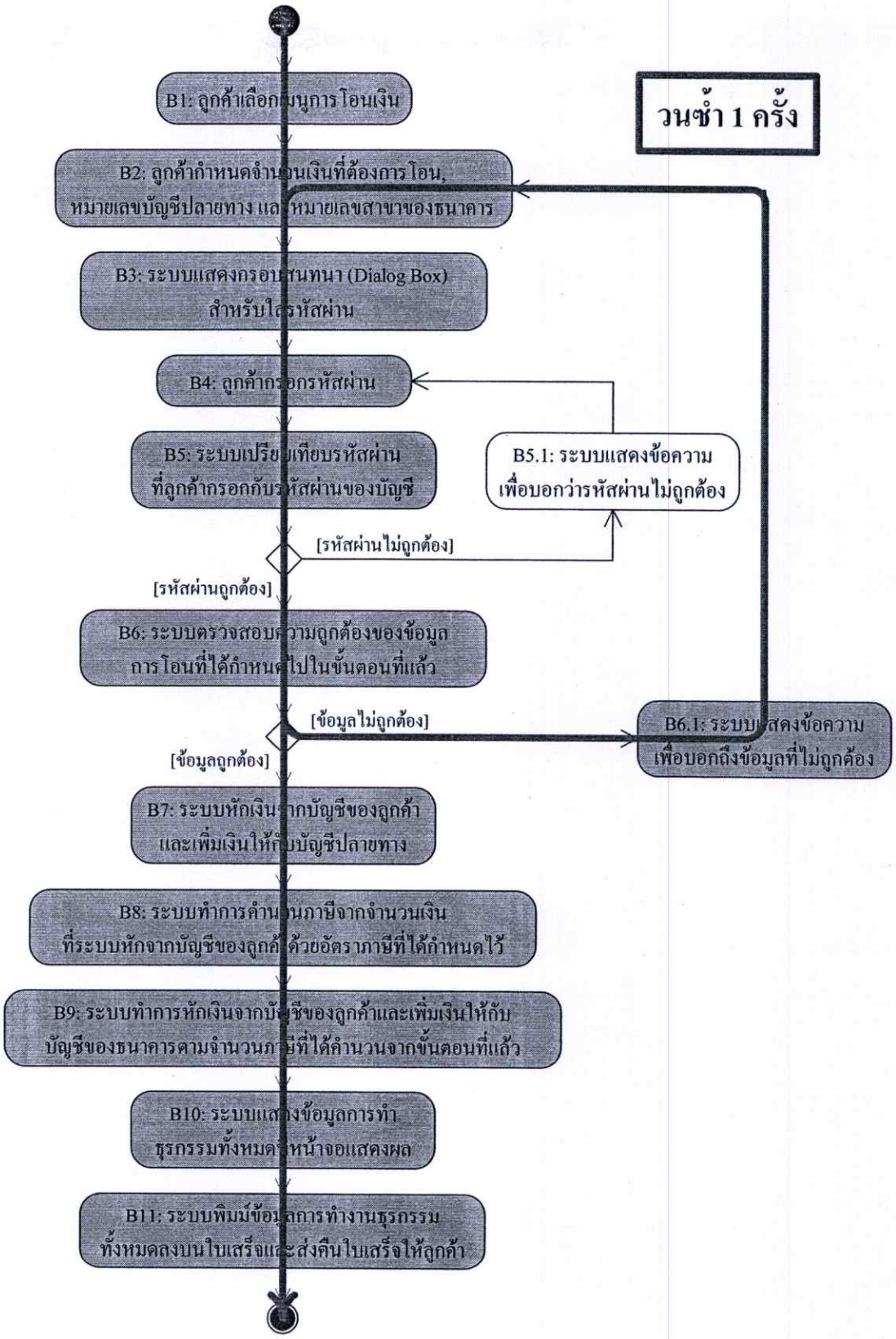
และสามารถแสดงเส้นทางอิสระเชิงเส้นทั้ง 3 เส้นทางด้วยแผนภาพกิจกรรมยูสเคสการโอนเงินได้ดังรูปที่ 2.20 รูปที่ 2.21 และรูปที่ 2.22 ตามลำดับ



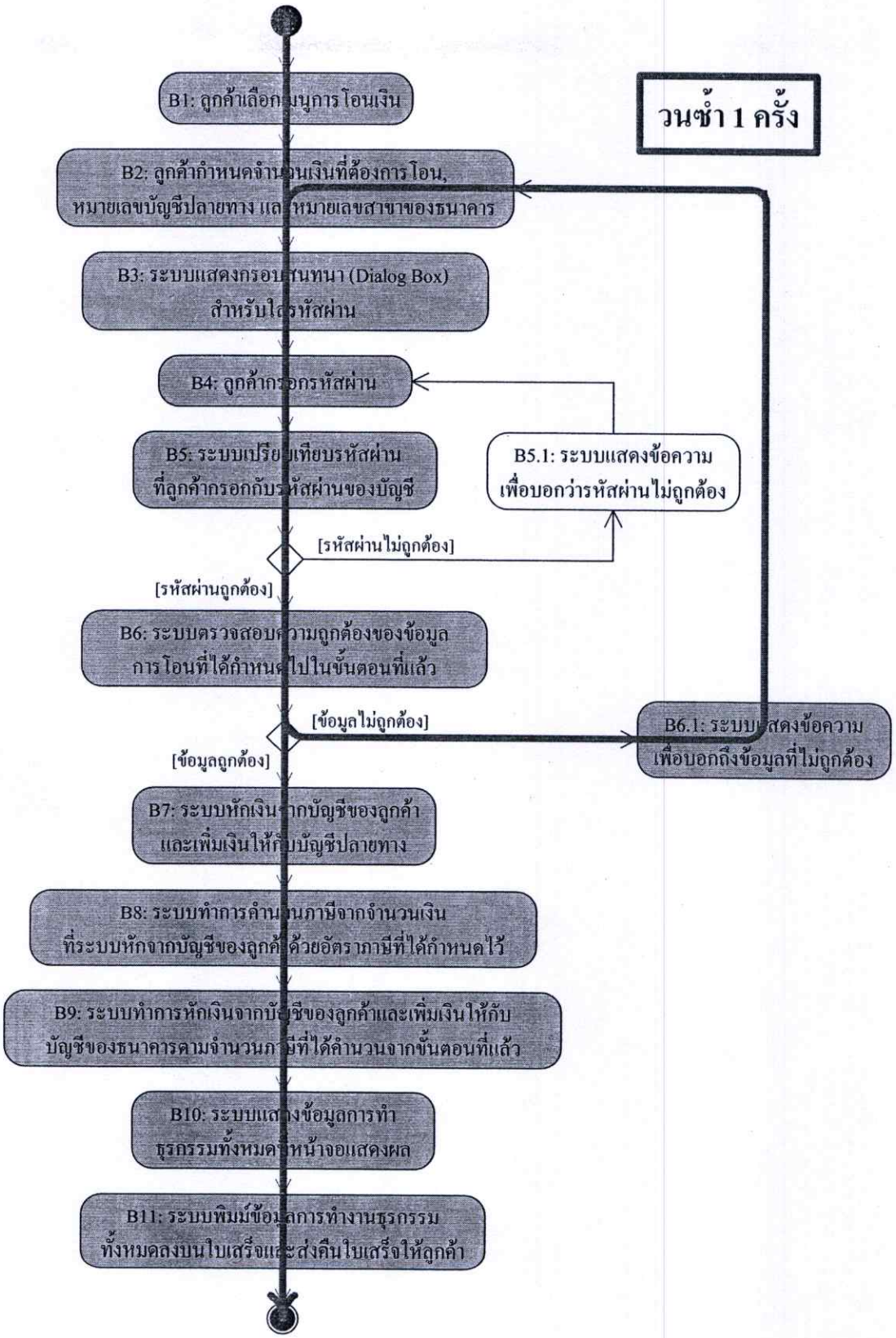
รูปที่ 2.20 เส้นทางอิสระเชิงเส้น 1 ของแผนภาพกิจกรรมยูสเคสการโอนเงิน



รูปที่ 2.21 เส้นทางอิสระเชิงเส้น 2 ของแผนภาพกิจกรรมยูสเคสการโอนเงิน



รูปที่ 2.22 เส้นทางอิสระเชิงเส้น 3 ของแผนภาพกิจกรรมยูสเคสการโอนเงิน



รูปที่ 2.22 เส้นทางอิสระเชิงเส้น 3 ของแผนภาพกิจกรรมยูสเคสการโอนเงิน

จากเส้นทางอิสระเชิงเส้นที่ค้นหาได้ทั้ง 3 เส้นทาง เป็นกลุ่มเส้นทางที่เป็นตัวแทนของเส้นทางทั้งหมดซึ่งสามารถนำมาแสดงแทนเส้นทางทั้งหมดของรูปที่ 2.15 แผนภาพกิจกรรมยูสเคสการโอนเงินได้ เส้นทางอิสระเชิงเส้นทั้ง 3 เส้นทางจึงถือได้ว่ามีความครอบคลุมทุกเส้นทาง ในงานวิจัยนี้จึงได้นำการทดสอบเบซิสพาทไปใช้สำหรับค้นหาเส้นทางการทำงานสำหรับขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส เพื่อที่จะสามารถสร้างกรณีทดสอบที่มีความครอบคลุมทุกเส้นทางการทำงาน ซึ่งจะได้นำเสนอในบทต่อไป

บทที่ 3
การสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

ในบทนี้จะกล่าวถึงขั้นตอนวิธีการสำหรับการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส เนื้อหาที่นำเสนอในบทนี้จะแบ่งการนำเสนอออกเป็น 3 ส่วน คือ การนำเสนอรายละเอียดของรูปแบบคำอธิบายยูสเคสสำหรับข้อกำหนดเชิงแอสเปค ซึ่งงานวิจัยนี้ได้ปรับปรุงเพื่อให้เหมาะสมกับการสร้างข้อกำหนดของการพัฒนาซอฟต์แวร์เชิงแอสเปค นำเสนอรูปแบบการเชื่อมต่อการทำงานระหว่างยูสเคสหลักกับแอสเปคยูสเคสโดยใช้แผนภาพกิจกรรม และการนำเสนอขั้นตอนวิธีการสำหรับการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

3.1 รูปแบบคำอธิบายแอสเปคยูสเคส

ในส่วนนี้เป็นการนำเสนอรูปแบบคำอธิบายยูสเคสซึ่งได้รับการปรับปรุงให้มีความสามารถในการนำเสนอข้อกำหนดความต้องการสำหรับการพัฒนาซอฟต์แวร์เชิงแอสเปค ซึ่งจะอธิบายในรูปแบบของคำอธิบายยูสเคสสำหรับแอสเปคที่ได้ถูกออกแบบขึ้นโดยเฉพาะ ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 รูปแบบคำอธิบายแอสเปคยูสเคส

ชื่อแอสเปคยูสเคส	<แสดงชื่อของแอสเปคยูสเคส>	
คำอธิบาย	<แสดงคำอธิบายแอสเปคยูสเคส>	
ลำดับเหตุการณ์	<แสดงลำดับเหตุการณ์ของแอสเปคยูสเคส>	
ลำดับเหตุการณ์ทางเลือก	<แสดงลำดับเหตุการณ์ทางเลือกของแอสเปคยูสเคส>	
ยูสเคสที่ได้รับผลกระทบ	**เงื่อนไข**	**การเชื่อมต่อ**
<แสดงชื่อของยูสเคสที่เกี่ยวข้องกับแอสเปคยูสเคส>	<เงื่อนไขของขั้นตอนการทำงาน>	<แสดงชนิดของการเชื่อมต่อ>

จากตารางที่ 3.1 แสดงให้เห็นถึงภาพรวมของรูปแบบยูสเคสที่ได้รับการปรับปรุงใหม่ ซึ่งประกอบด้วย 7 องค์ประกอบ คือ

1. ชื่อแอสยูสเคส (Aspect Use Case Name) ส่วนนี้จะแสดงชื่อของแอสเปคยูสเคส
2. คำอธิบายยูสเคส (Description) แสดงคำอธิบายแอสเปคยูสเคสเพิ่มเติม
3. ลำดับเหตุการณ์ (Event Flow) แสดงลำดับเหตุการณ์ของแอสเปคยูสเคสนั้น

4. ลำดับเหตุการณ์ทางเลือก (Alternate Flow of Events) แสดงลำดับเหตุการณ์ทางเลือกของแอสเปคยูสเคสนั้น
5. ยูสเคสที่ได้รับผลกระทบ (Affected Use Case) แสดงรายชื่อยูสเคสจากยูสเคสหลักที่เกี่ยวข้องกับแอสเปคยูสเคส
6. เงื่อนไข (Conditions) แสดงรายการของขั้นตอนการทำงานในยูสเคสหลัก เพื่อนำมาเป็นเงื่อนไขสำหรับเชื่อมต่อยูสเคสหลักกับแอสเปคยูสเคส ซึ่งสามารถมีจำนวนได้มากกว่า 1 เงื่อนไข
7. การเชื่อมต่อ (Composition Operator) แสดงรูปแบบการเชื่อมต่อระหว่างยูสเคสหลักกับแอสเปคยูสเคสที่ขั้นตอนการทำงานที่ได้กำหนดไว้ในเงื่อนไข ซึ่งอ้างอิงจากรูปแบบการเชื่อมต่อตามมาตรฐานของ AspectJ ประกอบด้วย
   ก. Before คือ การเชื่อมต่อก่อนขั้นตอนการทำงานของยูสเคสหลักที่ได้กำหนดไว้ในเงื่อนไข
   ข. After คือ การเชื่อมต่อหลังขั้นตอนการทำงานของยูสเคสหลักที่ได้กำหนดไว้ในเงื่อนไข
   ค. Around คือ การเชื่อมต่อโดยครอบขั้นตอนการทำงานของยูสเคสหลักที่ได้กำหนดไว้ในเงื่อนไข

จากรูปแบบคำอธิบายข้อกำหนดความต้องการของยูสเคสที่ปรับปรุงขึ้นเพื่อให้มีความเหมาะสมกับแอสเปคยูสเคสนี้สามารถแสดงตัวอย่างการใช้งานได้ดังตารางที่ 3.2

ตารางที่ 3.2 ตัวอย่างข้อกำหนดแอสเปคด้วยคำอธิบายแอสเปคยูสเคส

ชื่อแอสเปคยูสเคส	การตรวจสอบรหัสผ่าน	
คำอธิบายยูสเคส	ตรวจสอบรหัสผ่านของผู้มีสิทธิใช้งาน	
ลำดับเหตุการณ์	1. ระบบแสดงกรอบสนทนา(Dialog Box) สำหรับใส่รหัสผ่าน 2. ลูกค้าใส่รหัสผ่าน 3. ระบบเปรียบเทียบรหัสผ่านที่ลูกค้าใส่กับรหัสผ่านของบัญชี	
ลำดับเหตุการณ์ทางเลือก	3.1 ลูกค้าใส่รหัสผ่านไม่ถูกต้อง 3.1.1 ระบบแสดงข้อความเพื่อบอกว่ารหัสผ่านไม่ถูกต้อง 3.1.2 กลับไปยังขั้นตอนที่ 2	
ยูสเคสที่ได้รับผลกระทบ	**เงื่อนไข**	**การเชื่อมต่อ**
การแสดงยอดคงเหลือ	ระบบแสดงยอดเงินในบัญชี	before

ตารางที่ 3.2 (ต่อ)

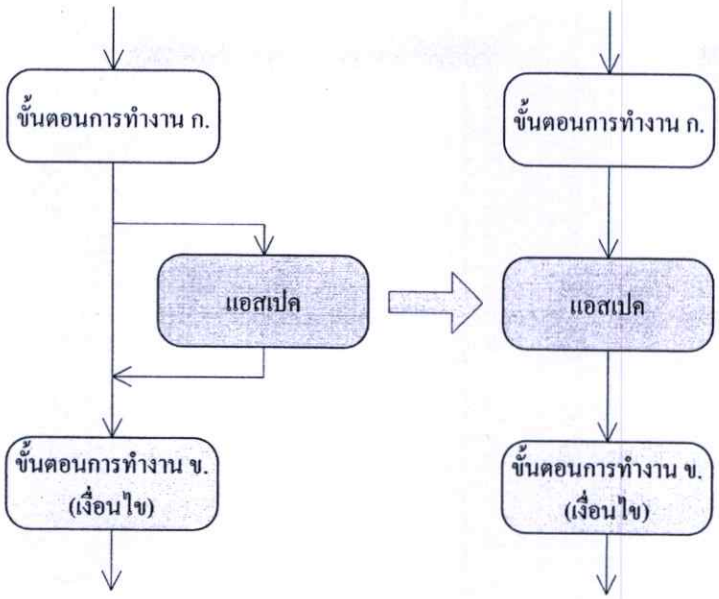
การแสดงรายการธุรกรรม	ระบบแสดงรายการของการถอนเงิน, การฝากเงินและคำอธิบาย ซึ่งเรียงตามลำดับวันที่	before
การโอนเงิน	ลูกค้ากำหนดจำนวนเงินที่ต้องการโอน, หมายเลขบัญชีปลายทางและหมายเลขสาขาของธนาคาร	after
การชำระเงินสินค้า	ลูกค้ากำหนดจำนวนเงินที่ต้องการชำระ, หมายเลขบัญชีสินค้าที่ต้องการชำระ	after

3.2 การเชื่อมต่อยูสเคสหลักกับแอสเปคยูสเคส

แผนภาพกิจกรรมเป็นแผนภาพสำหรับแสดงขั้นตอนการทำงานของยูสเคส และสามารถแสดงขั้นตอนการทำงานของแอสเปคยูสเคสได้ด้วย ด้วยยูสเคสหลักกับแอสเปคยูสเคสมีการรูปแบบการเชื่อมต่อการทำงานระหว่างกันอยู่ 3 แบบ ประกอบด้วย การเชื่อมต่อแบบ Before การเชื่อมต่อแบบ After และ การเชื่อมต่อแบบ Around ตามที่ได้กำหนดไว้ในหัวข้อที่ 3.1 เพื่อให้แผนภาพกิจกรรมสามารถแสดงขั้นตอนการทำงานที่เชื่อมโยงกันระหว่างยูสเคสหลักและแอสเปคยูสเคส จึงได้เสนอวิธีการเชื่อมต่อขั้นตอนการทำงานระหว่างยูสเคสหลักกับแอสเปคยูสเคสตามรูปแบบการเชื่อมต่อที่ได้กำหนดไว้ ดังนี้

1. การเชื่อมต่อแบบ Before ในแผนภาพกิจกรรม

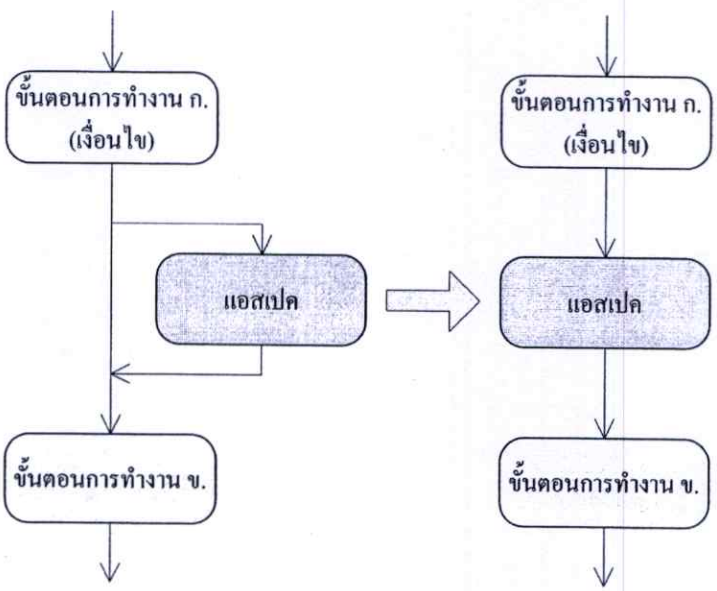
ทำการเชื่อมต่อโดยนำขั้นตอนการทำงานก่อนหน้าขั้นตอนการทำงานที่เป็นเงื่อนไขของยูสเคสหลักเชื่อมต่อกับขั้นตอนการทำงานแรกของแอสเปคยูสเคส และนำขั้นตอนการทำงานสุดท้ายของแอสเปคยูสเคสเชื่อมต่อเข้ากับขั้นตอนการทำงานที่เป็นเงื่อนไขของยูสเคสหลัก ดังรูปที่ 3.1



รูปที่ 3.1 แสดงการเชื่อมต่อแบบ Before

2. การเชื่อมต่อแบบ After ในแผนภาพกิจกรรม

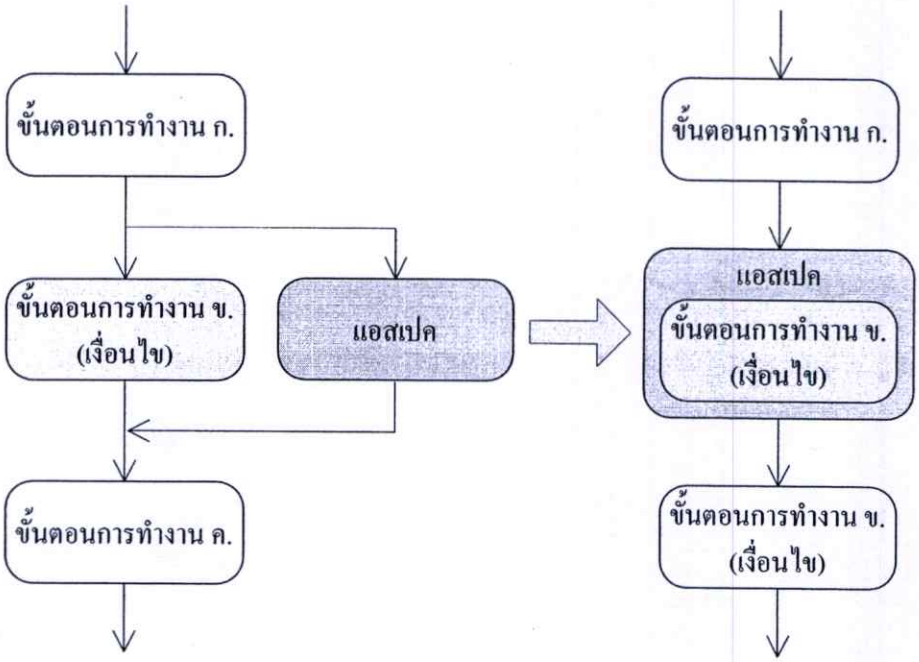
ทำการเชื่อมต่อ โดยนำขั้นตอนการทำงานที่เป็นเงื่อนไขของยูสเคสหลักเชื่อมต่อเข้ากับขั้นตอนการทำงานแรกของแอสเปคยูสเคส และนำขั้นตอนการทำงานสุดท้ายของแอสเปคยูสเคสเชื่อมต่อเข้ากับขั้นตอนการทำงานถัดจากขั้นตอนการทำงานที่เป็นเงื่อนไขของยูสเคสหลัก ดังรูปที่ 3.2



รูปที่ 3.2 แสดงการเชื่อมต่อแบบ After

3. การเชื่อมต่อแบบ Around ในแผนภาพกิจกรรม

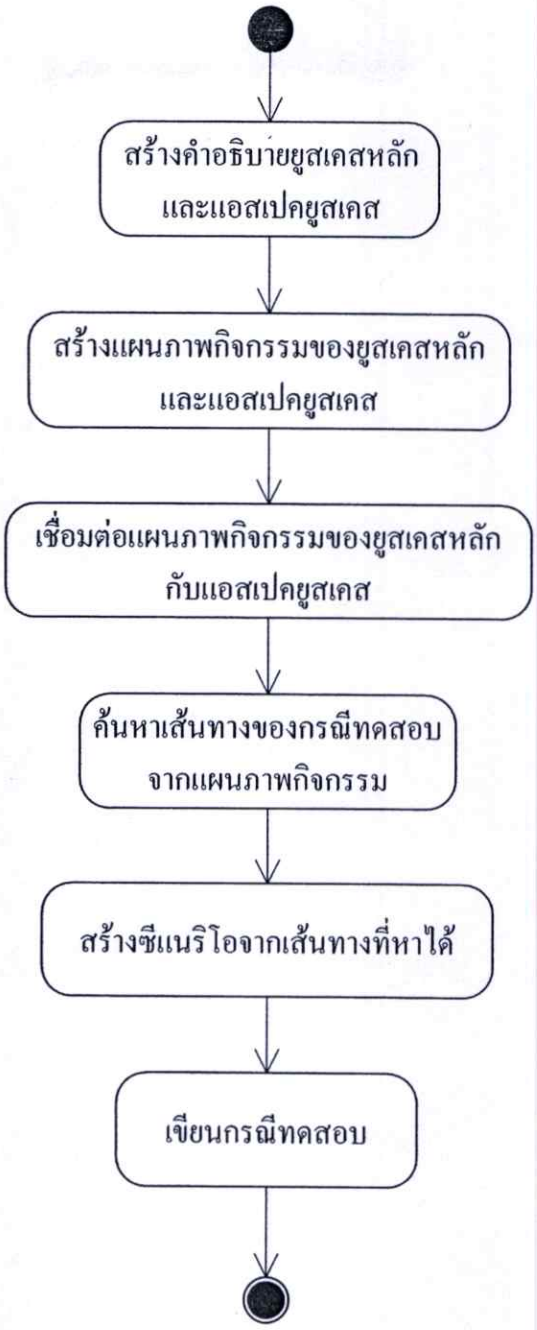
ทำการเชื่อมต่อโดยนำขั้นตอนการทำงานก่อนหน้าขั้นตอนการทำงานที่เป็นเงื่อนไขของยูสเคสหลักเชื่อมต่อเข้ากับขั้นตอนการทำงานแรกของแอสเปคยูสเคส และนำขั้นตอนการทำงานสุดท้ายของแอสเปคยูสเคสเชื่อมต่อเข้ากับขั้นตอนการทำงานถัดจากขั้นตอนการทำงานที่เป็นเงื่อนไขของยูสเคสหลัก ดังรูปที่ 3.3



รูปที่ 3.3 แสดงการเชื่อมต่อแบบ Around

3.3 ขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

ในส่วนนี้เป็นการนำเสนอขั้นตอนวิธีการสำหรับการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส ดังแสดงให้เห็นภาพรวมของกระบวนการ ดังรูปที่ 3.4

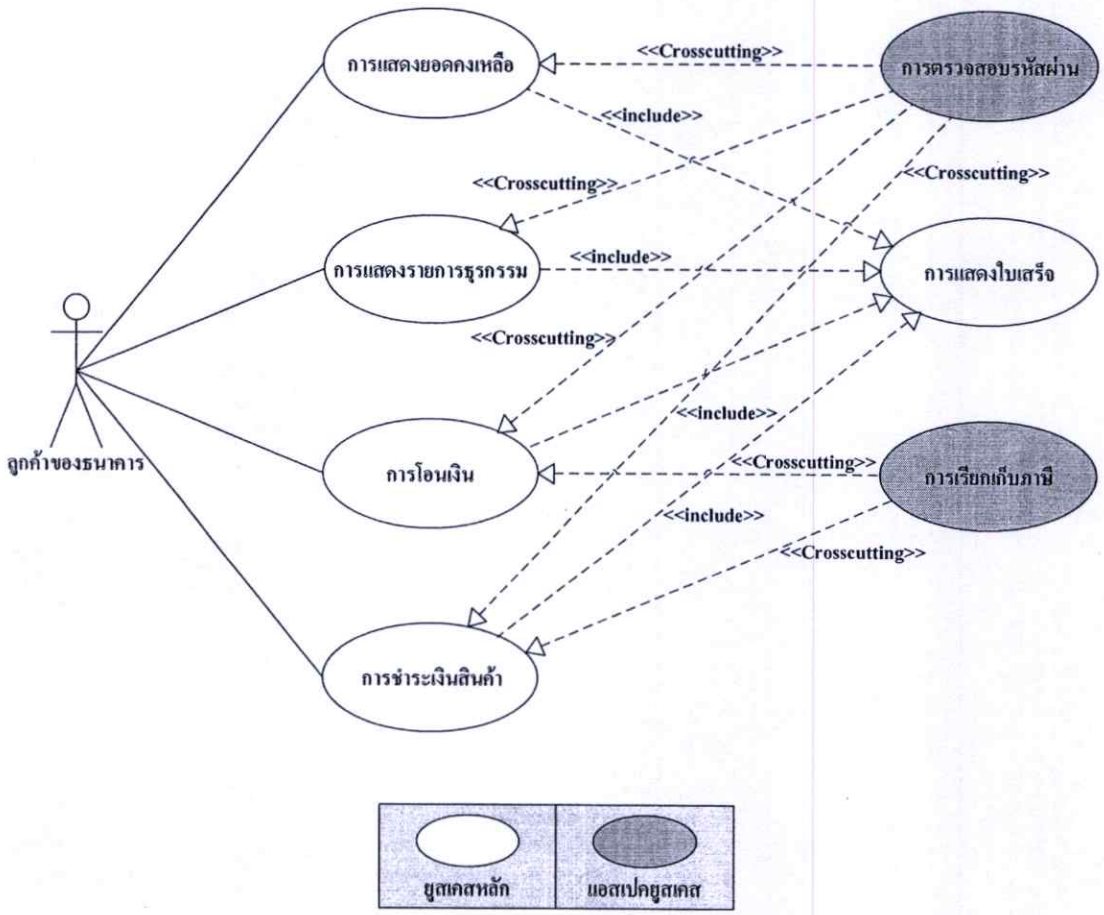


รูปที่ 3.4 ภาพรวมขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

จากรูปที่ 3.4 ขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสประกอบด้วยขั้นตอนสำคัญ ซึ่งสามารถอธิบายขั้นตอนต่างๆ ได้ดังนี้

3.3.1 สร้างคำอธิบายยูสเคสหลักและแอสเปคยูสเคส

จากรูปที่ 2.7 แผนภาพยูสเคสระบบธนาคารอินเทอร์เน็ต เมื่อพิจารณาแผนภาพยูสเคสดังกล่าวแสดงให้เห็นว่ามีการครอสคัตติ้งคอนเซินเกิดขึ้น ซึ่งยูสเคสการตรวจสอบรหัสผ่านและยูสเคสการเรียกเก็บภาษีได้กระจายเข้าไปทำงานและเกี่ยวพันในยูสเคสการแสดงยอดเงินคงเหลือ ยูสเคสแสดงรายการธุรกรรม ยูสเคสการโอนเงิน และยูสเคสการชำระเงินสินค้า สามารถสร้างแผนภาพยูสเคสที่แสดงการเกิดครอสคัตติ้งคอนเซินของยูสเคสดังกล่าวได้รูปที่ 3.5



รูปที่ 3.5 แผนภาพยูสเคสแสดงการเกิดครอสคัตติ้งคอนเซินในระบบธนาคารอินเทอร์เน็ต

จากแผนภาพยูสเคสข้างต้น และจากตารางที่ 2.6 ตารางที่ 2.7 ตารางที่ 2.8 และตารางที่ 2.9 ซึ่งแสดงคำอธิบายยูสเคสการโอนเงิน คำอธิบายยูสเคสการตรวจสอบรหัสผ่าน คำอธิบายยูสเคสการเรียกเก็บภาษี และคำอธิบายยูสเคสการแสดงใบเสร็จของแผนภาพยูสเคสระบบธนาคารอินเทอร์เน็ต รูปที่ 2.7 สามารถนำมาสร้างเป็นคำอธิบายยูสเคสและคำอธิบายแอสเปคยูสเคสเพื่ออธิบายขั้นตอนการทำงานของยูสเคสการโอนเงิน ยูสเคสการแสดงใบเสร็จ แอสเปคยูสเคสการตรวจสอบรหัสผ่าน และแอสเปคยูสเคสการเรียกเก็บภาษีในรูปที่ 3.5 ได้ดังตารางที่ 3.3 ตารางที่ 3.4 ตารางที่ 3.5 และตารางที่ 3.6 ตามลำดับ

ตารางที่ 3.3 คำอธิบายยูสเคสการโอนเงิน

ชื่อยูสเคส	การโอนเงิน
คำอธิบาย	การโอนเงินทางอินเทอร์เน็ต
เงื่อนไขก่อน	-
จุดต่อขยาย	-
ยูสเคสที่ถูกใช้งาน	1. แสดงใบเสร็จ
ลำดับเหตุการณ์	1. ลูกค้าเลือกเมนูการโอนเงิน 2. ลูกค้ากำหนดจำนวนเงินที่ต้องการโอน, หมายเลขบัญชีปลายทาง และหมายเลขสาขาของธนาคาร 3. ระบบตรวจสอบความถูกต้องของข้อมูลการโอนที่ได้กำหนดไปในขั้นตอนที่แล้ว 4. ระบบหักเงินจากบัญชีของลูกค้าและเพิ่มเงินให้กับบัญชีปลายทาง 5. ระบบแสดงใบเสร็จ 5.1 ใช้งาน "แสดงใบเสร็จ"
ลำดับเหตุการณ์ทางเลือก	3.1 ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง 3.1.1 ระบบแสดงข้อความเพื่อบอกถึงข้อมูลที่ไม่ถูกต้อง 3.1.2 กลับไปที่ขั้นตอนที่ 2
เงื่อนไขหลัง	-

ตารางที่ 3.4 คำอธิบายยูสเคสการแสดงใบเสร็จ

ชื่อยูสเคส	การแสดงใบเสร็จ
คำอธิบาย	แสดงข้อมูลการทำธุรกรรมที่ลูกค้าได้ดำเนินการ
เงื่อนไขก่อน	-
จุดต่อขยาย	-
ยูสเคสที่ถูกใช้งาน	-
ลำดับเหตุการณ์	1. ระบบแสดงข้อมูลการทำธุรกรรมทั้งหมดที่หน้าจอแสดงผล 2. ระบบพิมพ์ข้อมูลการทำงานธุรกรรมทั้งหมดลงบนใบเสร็จและส่งคืนใบเสร็จให้ลูกค้า
ลำดับเหตุการณ์ทางเลือก	-
เงื่อนไขหลัง	-

ตารางที่ 3.5 คำอธิบายแอสเปคยูสเคสการตรวจสอบรหัสผ่าน

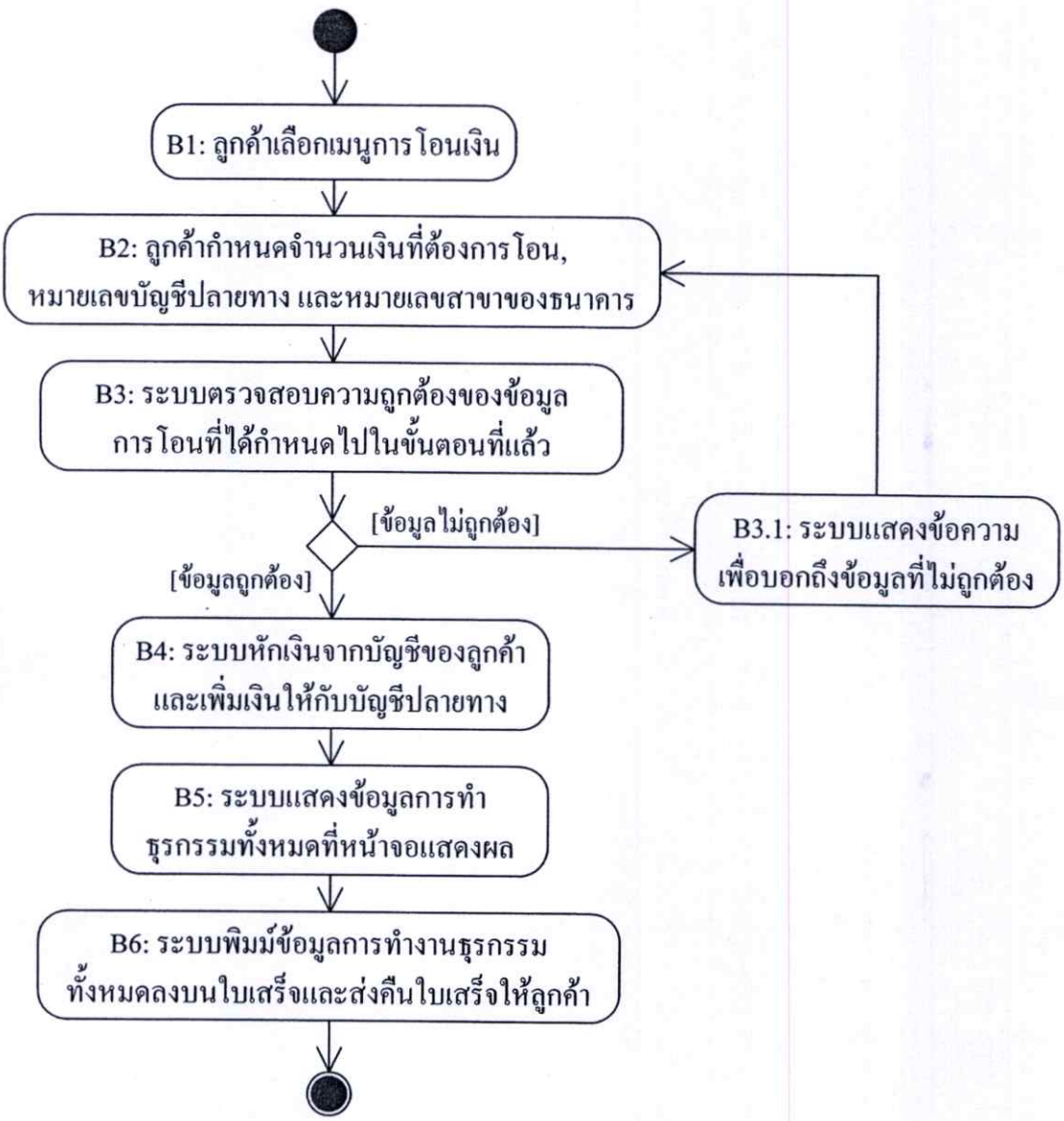
ชื่อแอสเปคยูสเคส	การตรวจสอบรหัสผ่าน	
คำอธิบายยูสเคส	ตรวจสอบรหัสผ่านของผู้มีสิทธิใช้งาน	
ลำดับเหตุการณ์	1. ระบบแสดงกรอบสนทนา (Dialog Box) สำหรับใส่รหัสผ่าน 2. ลูกค้ากรอกรหัสผ่าน 3. ระบบเปรียบเทียบรหัสผ่านที่ลูกค้ากรอกกับรหัสผ่านของบัญชี	
ลำดับเหตุการณ์ทางเลือก	3.1 ลูกค้าใส่รหัสผ่านไม่ถูกต้อง 3.1.1 ระบบแสดงข้อความเพื่อบอกว่ารหัสผ่านไม่ถูกต้อง 3.1.2 กลับไปยังขั้นตอนที่ 2	
ยูสเคสที่ได้รับผลกระทบ	**เงื่อนไข**	**การเชื่อมต่อ**
การแสดงยอดคงเหลือ	ระบบแสดงยอดเงินในบัญชี	before
การแสดงรายการธุรกรรม	ระบบแสดงรายการของการถอนเงิน, การฝากเงินและคำอธิบาย ซึ่งเรียงตามลำดับวันที่	before
การโอนเงิน	ลูกค้ากำหนดจำนวนเงินที่ต้องการโอน, หมายเลขบัญชีปลายทางและหมายเลขสาขาของธนาคาร	after
การชำระเงินสินค้า	ลูกค้ากำหนดจำนวนเงินที่ต้องการชำระ, หมายเลขบัญชีสินค้าที่ต้องการชำระ	after

ตารางที่ 3.6 คำอธิบายแอสเปคยูสเคสการเรียกเก็บภาษี

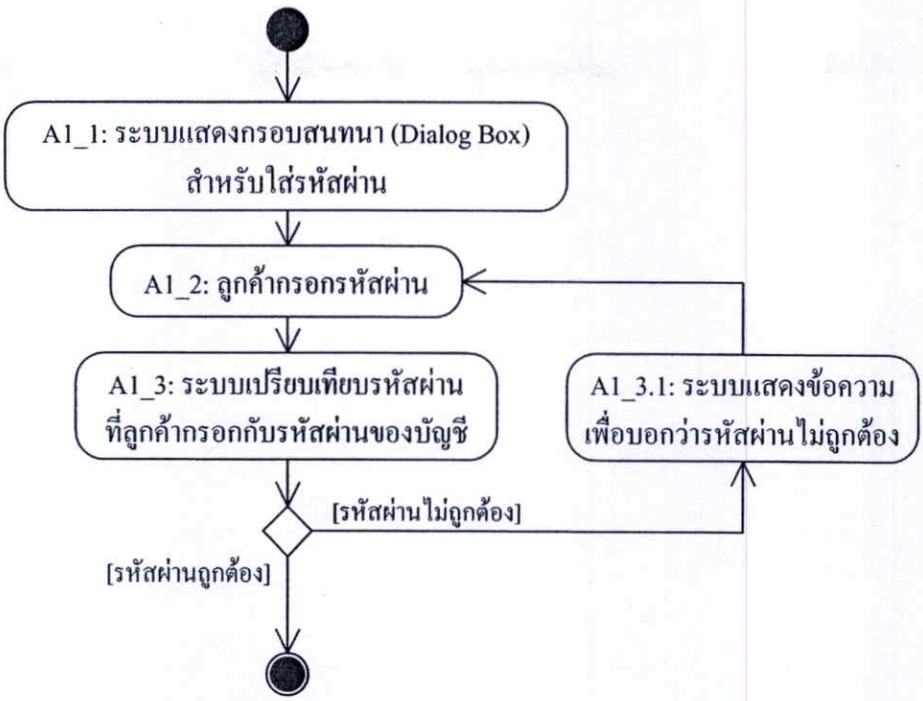
ชื่อแอสเปคยูสเคส	การเรียกเก็บภาษี	
คำอธิบายยูสเคส	เรียกเก็บภาษีจากการทำธุรกรรมทางการเงิน	
ลำดับเหตุการณ์	1. ระบบทำการคำนวนภาษีจากจำนวนเงินที่ระบบหักจากบัญชีของลูกค้าด้วยอัตราภาษีที่ได้กำหนดไว้ 2. ระบบทำการหักเงินจากบัญชีของลูกค้าและเพิ่มเงินให้กับบัญชีของธนาคารตามจำนวนภาษีที่ได้คำนวนจากขั้นตอนที่แล้ว	
ลำดับเหตุการณ์ทางเลือก	-	
ยูสเคสที่ได้รับผลกระทบ	**เงื่อนไข**	**การเชื่อมต่อ**
การโอนเงิน	ระบบหักเงินจากบัญชีของลูกค้าและเพิ่มเงินให้กับบัญชีปลายทางของธนาคาร	after
การชำระเงินสินค้า	ระบบหักเงินจากบัญชีของลูกค้าและเพิ่มเงินให้กับบัญชีสินค้าที่ต้องการชำระ	after

3.3.2 สร้างแผนภาพกิจกรรมจากข้อกำหนดยูสเคสหลัก และแอสเปคยูสเคส

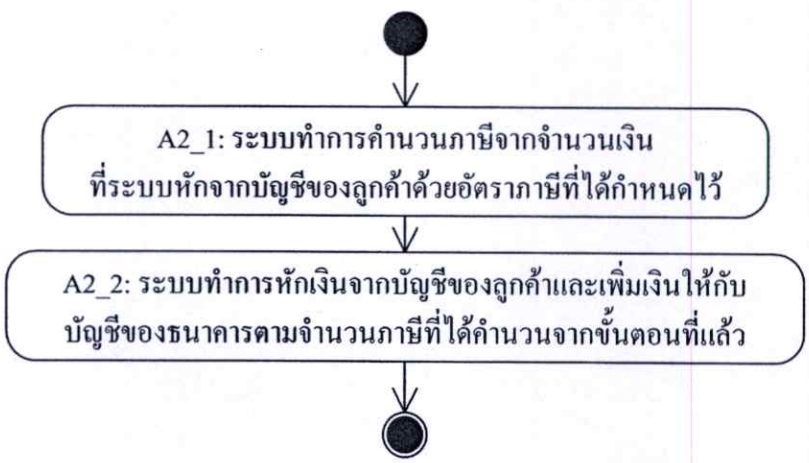
ในขั้นตอนนี้เป็นการนำเอาคำอธิบายของยูสเคสหลักและยูสเคสแอสเปคจากตารางที่ 3.3 ตารางที่ 3.4 ตารางที่ 3.5 และตารางที่ 3.6 มาสร้างเป็นแผนภาพกิจกรรม ดังรูปที่ 3.6 รูปที่ 3.7 และรูปที่ 3.8



รูปที่ 3.6 แผนภาพกิจกรรมยูสเคสการโอนเงิน



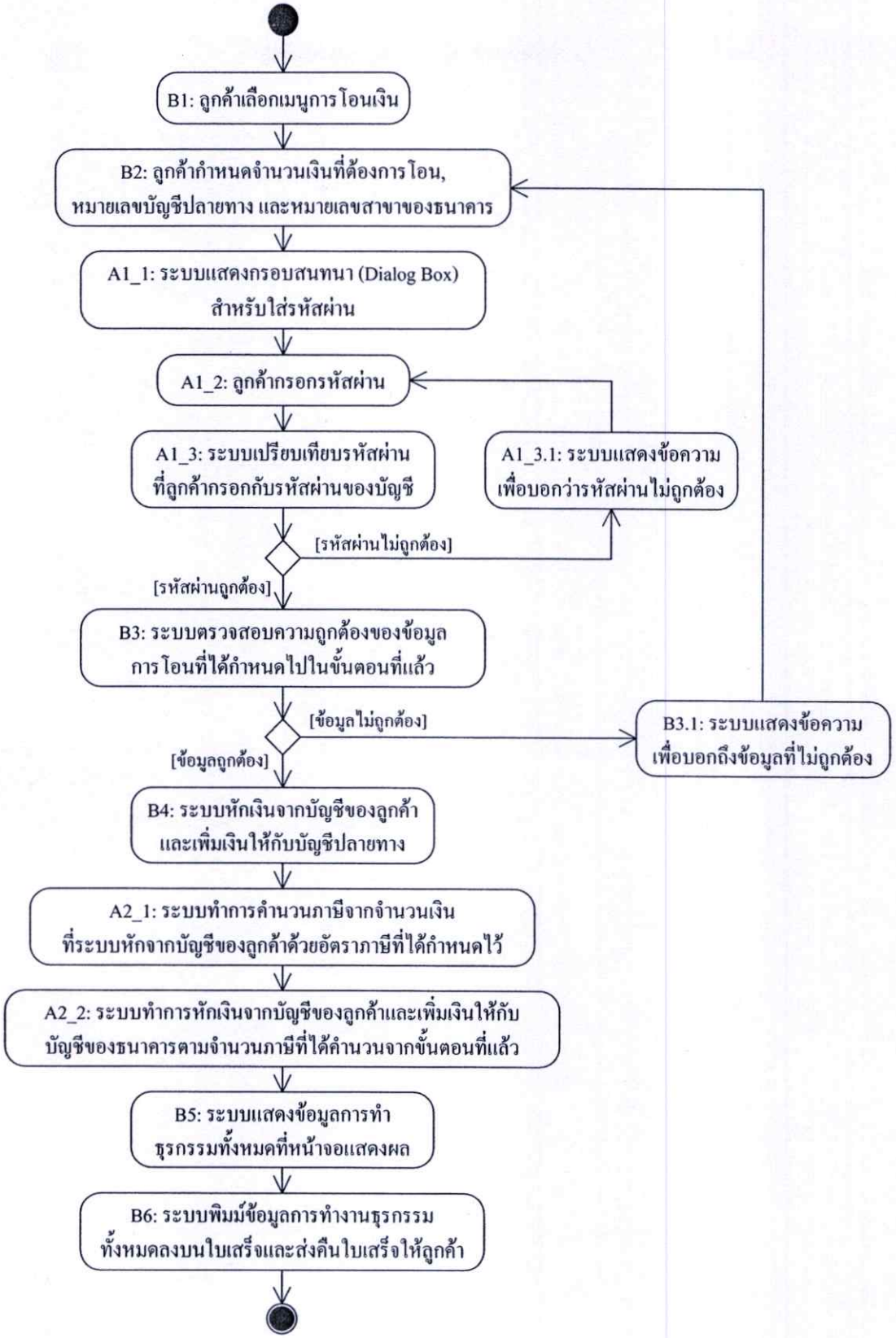
รูปที่ 3.7 แผนภาพกิจกรรมแอสเปคยูสเคสการตรวจสอบรหัสผ่าน



รูปที่ 3.8 แผนภาพกิจกรรมแอสเปคยูสเคสการเรียกเก็บภาษี

3.3.3 การเชื่อมต่อแผนภาพกิจกรรมของยูสเคสหลักและแอสเปคยูสเคส

ขั้นตอนนี้เป็นขั้นตอนการเชื่อมต่อยูสเคสหลักและแอสเปคยูสเคสเข้าด้วยกัน เพื่อใช้ในการค้นหาเส้นทางของกรณีทดสอบต่อไป ซึ่งรูปแบบการเชื่อมต่อระหว่างยูสเคสหลักกับแอสเปคยูสเคสเป็นไปตามหัวข้อ 3.2 โดยผลการเชื่อมต่อระหว่างแผนภาพกิจกรรมของยูสเคสการโอนเงินกับแผนภาพกิจกรรมแอสเปคยูสเคสการตรวจสอบรหัสผ่าน และแผนภาพกิจกรรมแอสเปคยูสเคสการเรียกเก็บภาษี สามารถแสดงได้ดังรูปที่ 3.9



รูปที่ 3.9 แผนภาพกิจกรรมการเชื่อมต่อระหว่างยูสเคสการโอนเงิน แอสเปคยูสเคสการตรวจสอบรหัสผ่านและแอสเปคยูสเคสการเรียกเก็บภาษี

3.3.4 การค้นหาเส้นทางของกรณีทดสอบจากแผนภาพกิจกรรม

เมื่อได้แผนภาพกิจกรรมที่เกิดจากการเชื่อมต่อการทำงานระหว่างยูสเคสหลักและแอสเปคยูสเคสแล้ว แผนภาพกิจกรรมที่ได้จะถูกนำมาค้นหาเส้นทางของกรณีทดสอบ โดยจะประกอบด้วย 2 ขั้นตอนย่อย ดังนี้คือ

1. คำนวนหาความซับซ้อนไซโคลเมติกของแผนภาพกิจกรรม เป็นการคำนวนหาจำนวนของเส้นทางอิสระเชิงเส้นและเบซิสเซตจากแผนภาพกิจกรรมที่ได้ทำการเชื่อมต่อกันระหว่างยูสเคสหลักและแอสเปคยูสเคสแล้ว ซึ่งมีสูตรสำหรับการคำนวนดังแสดงในหัวข้อ 2.6.4 และจากรูปที่ 3.9 สามารถคำนวนหาจำนวนเส้นทางอิสระเชิงเส้นได้ 3 เส้นทาง ซึ่งสามารถแสดงการคำนวนได้ดังนี้คือ

$$V(G) = E - V + 2$$
$$= 14 - 13 + 2$$
$$= 3$$

2. ค้นหาเส้นทางอิสระเชิงเส้น ในขั้นตอนนี้เป็นการกำหนดเส้นทางอิสระเชิงเส้นจากแผนภาพกิจกรรม โดยกำหนดคู่ของโหนดขั้นตอนการทำงานที่มีความเกี่ยวข้องกันสำหรับแต่ละโหนดขั้นตอนการทำงานที่มีเส้นเชื่อมขาออกมากกว่า 1 เส้น จากรูปที่ 3.9 สามารถแสดงคู่ของโหนดขั้นตอนการทำงานที่มีความเกี่ยวข้องกัน ดังตารางที่ 3.7

ตารางที่ 3.7 แสดงคู่โหนดขั้นตอนการทำงานที่มีความเกี่ยวข้องกันจากแผนภาพกิจกรรม รูปที่ 3.9

หมายเลขคู่โหนด	โหนดปัจจุบัน	โหนดถัดไป
1	A1_3	B3
2	A1_3	A1_3.1
3	B3	B4
4	B3	B3.1

จากนั้นนำแต่ละคู่ของโหนดขั้นตอนการทำงานที่กำหนดได้มากำหนดเส้นทางที่สมบูรณ์ของแต่ละคู่โหนด โดยกำหนดเส้นทางที่สั้นที่สุดจากโหนดขั้นตอนการทำงานเริ่มต้นไปสู่โหนดปัจจุบัน และเส้นทางที่สั้นที่สุดจากโหนดถัดไปไปสู่โหนดขั้นตอนการทำงานสุดท้ายของแต่ละคู่โหนด และนำทั้ง 2 เส้นทางมาเชื่อมต่อกันเป็นเส้นทางที่สมบูรณ์ จากตารางที่ 3.7 สามารถกำหนดเส้นทางที่สมบูรณ์ของแต่ละคู่ของโหนดขั้นตอนการทำงานที่มีความเกี่ยวข้องกันได้ดังตารางที่ 3.8

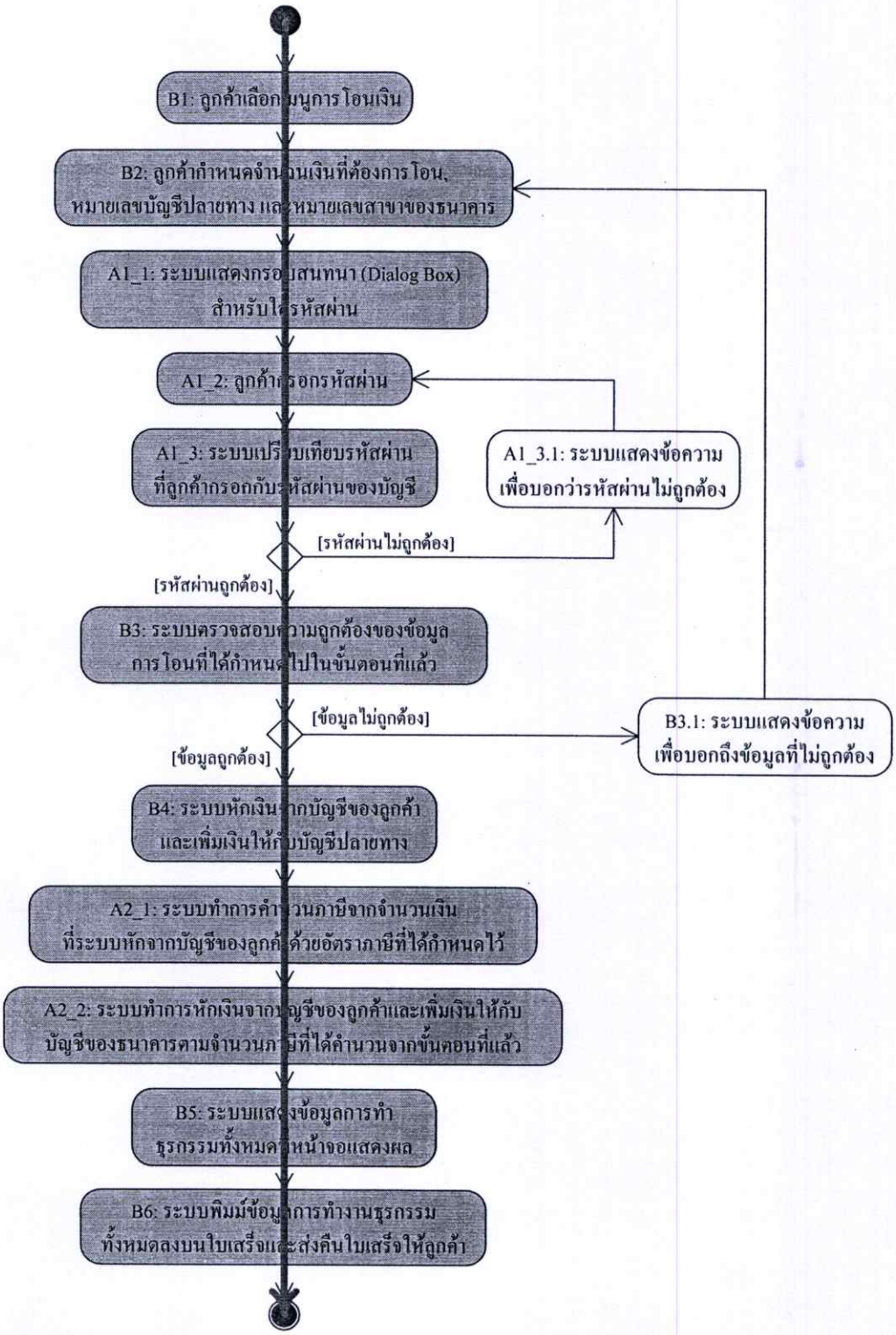
ตารางที่ 3.8 แสดงเส้นทางที่สมบูรณ์ของคู่โหนดขั้นตอนการทำงานที่มีความเกี่ยวข้องกันจากตารางที่ 3.7

หมายเลขคู่โหนด	เส้นทางไปสู่โหนดปัจจุบัน	เส้นทางจากโหนดถัดไป	เส้นทางที่สมบูรณ์
1	B1, B2, A1_1, A1_2, A1_3	B3, B4, A2_1, A2_2, B5, B6	B1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6
2	B1, B2, A1_1, A1_2, A1_3	A1_3.1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6	B1, B2, A1_1, A1_2, A1_3, A1_3.1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6
3	B1, B2, A1_1, A1_2, A1_3, B3	B4, A2_1, A2_2, B5, B6	B1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6
4	B1, B2, A1_1, A1_2, A1_3, B3	B3.1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6	B1, B2, A1_1, A1_2, A1_3, B3, B3.1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6

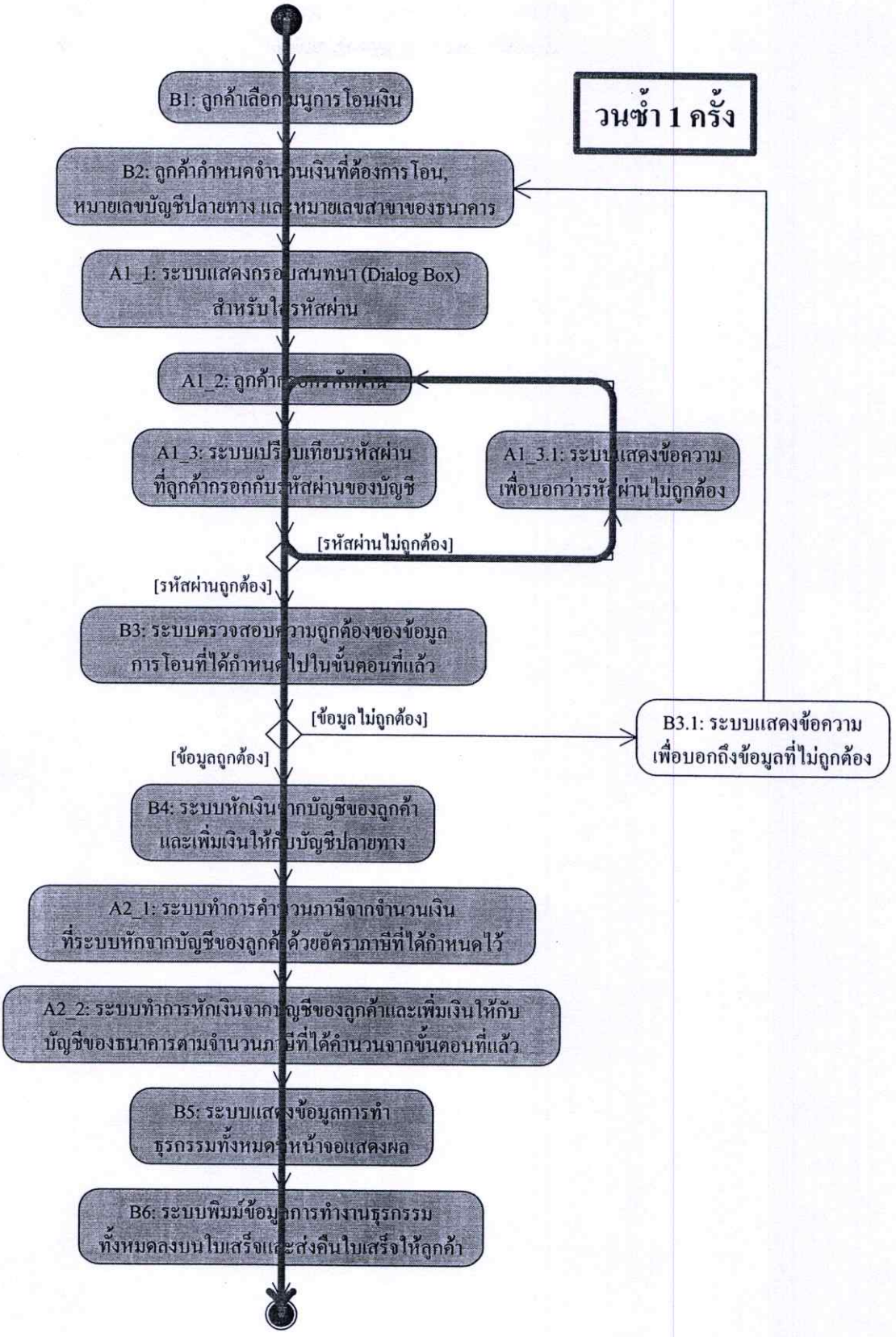
จากนั้นให้ทำการลบเส้นทางที่สมบูรณ์ที่มีความซ้ำซ้อนกับเส้นทางที่สมบูรณ์อื่นออก จากตารางที่ 3.8 หมายเลขคู่โหนดที่ 3 มีเส้นทางที่สมบูรณ์ซ้ำซ้อนกับเส้นทางที่สมบูรณ์ของหลายเลขคู่โหนดที่ 1 จึงทำการลบหมายเลขคู่โหนดที่ 3 ออก ทำให้สามารถค้นหาเส้นทางเส้นทางอิสระเชิงและเบซิสเซตของรูปที่ 3.9 ได้จำนวน 3 เส้นทาง ซึ่งเท่ากับจำนวนเส้นทางที่ได้จากการคำนวนความซับซ้อนไซโคลเมติกของแผนภาพกิจกรรมในขั้นตอนก่อนหน้านี้ ดังนั้นเส้นทางอิสระเชิงและเบซิสเซตที่ค้นหาได้ คือ

- เส้นทางเชิงอิสระ 1: B1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6
- เส้นทางเชิงอิสระ 2: B1, B2, A1_1, A1_2, A1_3, A1_3.1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6
- เส้นทางเชิงอิสระ 3: B1, B2, A1_1, A1_2, A1_3, B3, B3.1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6

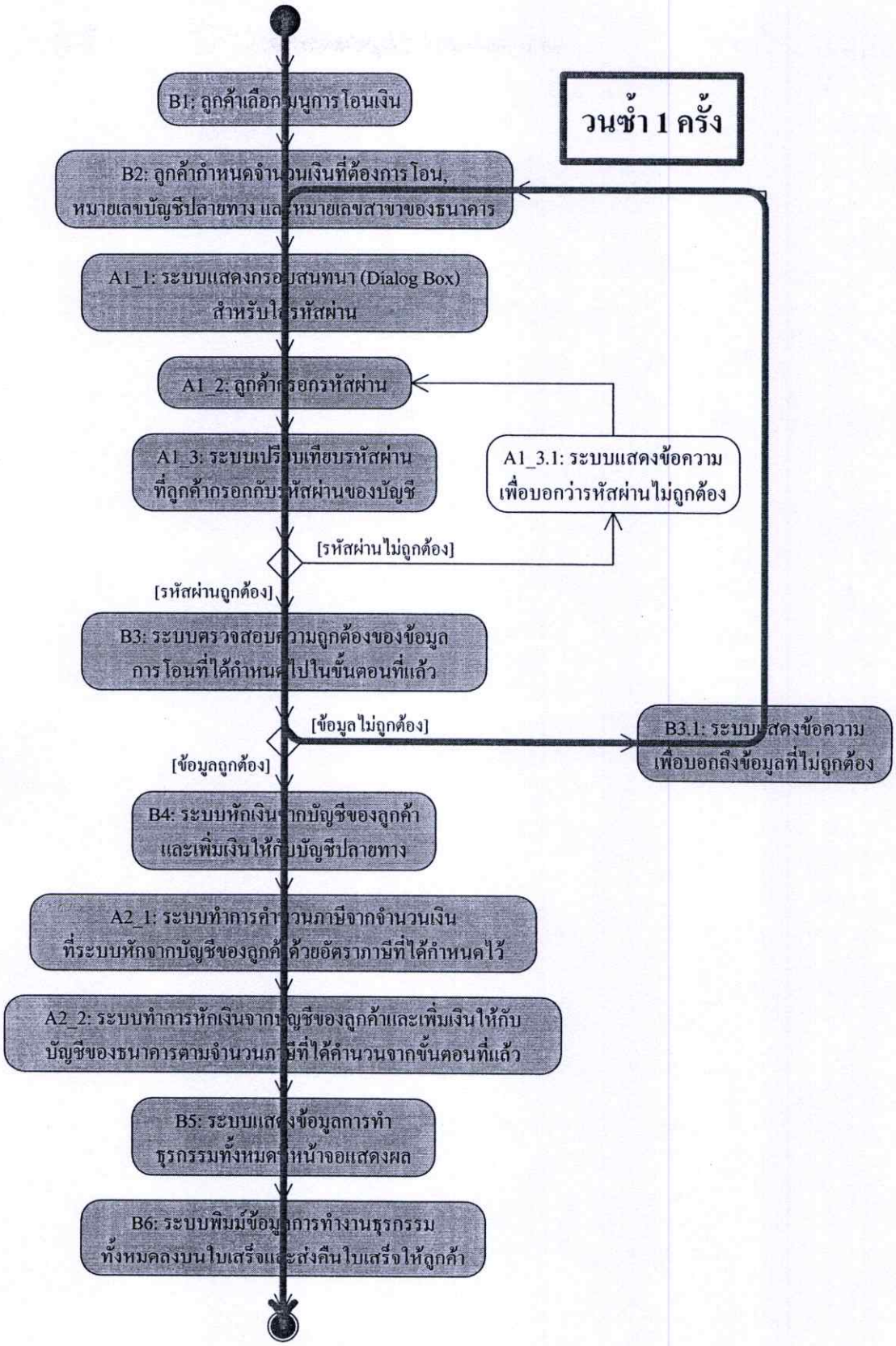
สามารถแสดงเส้นทางอิสระเชิงเส้นทั้ง 3 เส้นทางด้วยแผนภาพกิจกรรมแผนภาพกิจกรรมการเชื่อมต่อระหว่างยูสเคสการโอนเงิน แอสเปคยูสเคสการตรวจสอบรหัสผ่านและแอสเปคยูสเคสการเรียกเก็บภาษี รูปที่ 3.9 ได้ดังรูปที่ 3.10 รูปที่ 3. 11 และรูปที่ 3.12 ตามลำดับ



รูปที่ 3.10 เส้นทางเชิงอิสระ 1



รูปที่ 3.11 เส้นทางเชิงอิสระ 2



รูปที่ 3.12 เส้นทางเชิงอิสระ 3

3.3.5 สร้างซีแนริโอจากเส้นทางที่หาได้

จากเส้นทางเชิงอิสระหาที่ค้นหาได้ เส้นทางทั้งหมดจะถูกนำมาสร้างเป็นซีแนริโอของยูสเคสหลักที่เชื่อมต่อการทำงานกับแอสเปคยูสเคส โดยพิจารณาร่วมกันระหว่างเส้นทางเชิงอิสระในแต่ละเส้นทางกับแผนภาพกิจกรรมแผนภาพกิจกรรมการเชื่อมต่อระหว่างยูสเคสการโอนเงินแอสเปคยูสเคสการตรวจสอบรหัสผ่านและแอสเปคยูสเคสการเรียกเก็บภาษี รูปที่ 3.9 เช่น ในเส้นทางเชิงอิสระ 1 ประกอบด้วยขั้นตอนการทำงาน B1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6 ซึ่งถูกสร้างเป็นซีแนริโอสำหรับการโอนเงินในกรณีที่รหัสผ่าน และข้อมูลสำหรับการโอนเงินมีความถูกต้อง เป็นต้น จากเส้นทางเชิงอิสระที่ค้นหาได้จากขั้นตอนที่แล้ว สามารถสร้างเป็นซีแนริโอได้ ดังตารางที่ 3.9

ตารางที่ 3.9 ซีแนริโอของยูสเคสการโอนเงินที่เชื่อมต่อกับแอสเปคยูสเคสการตรวจสอบรหัสผ่านและแอสเปคยูสเคสการเรียกเก็บภาษี

ซีแนริโอ	เส้นทาง
ซีแนริโอ 1: การโอนเงินประสบผลสำเร็จ	B1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6
ซีแนริโอ 2: รหัสผ่านไม่ถูกต้อง	B1, B2, A1_1, A1_2, A1_3, A1_3.1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6
ซีแนริโอ 3: ข้อมูลการโอนเงินไม่ถูกต้อง	B1, B2, A1_1, A1_2, A1_3, B3, B3.1, B2, A1_1, A1_2, A1_3, B3, B4, A2_1, A2_2, B5, B6, B5, B6, B7, B8, B9, B10, B11

3.3.6 เขียนกรณีทดสอบ

ขั้นตอนนี้เป็นขั้นตอนสุดท้ายของขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส โดยประกอบด้วย 2 ขั้นตอนย่อย คือ

1. กำหนดกรณีทดสอบ ขั้นตอนนี้เป็นการนำซีแนริโอที่ได้กำหนดไว้มาสร้างเป็นกรณีทดสอบ โดยเพิ่มรายละเอียดในการกำหนดกำหนดชนิดข้อมูลที่จะนำมาทดสอบและผลที่คาดว่าจะได้รับในแต่ละกรณีทดสอบ ดังนั้นสามารถแสดงการกำหนดกรณีทดสอบจากซีแนริโอที่ได้จากขั้นตอนที่แล้ว ได้ดังตารางที่ 3.10

ตารางที่ 3.10 แสดงการกำหนดกรณีทดสอบสำหรับยูสเคสการโอนเงินที่เชื่อมต่อกับแอสเปคยูสเคสการตรวจสอบรหัสผ่าน และแอสเปคยูสเคสการเรียกเก็บภาษี

รหัสกรณีทดสอบ	ซีแนริโอ	ข้อมูลที่ใช้ทดสอบ	ผลที่คาดว่าจะได้รับ
TC1	ซีแนริโอ 1: รหัสผ่านถูกต้อง, ข้อมูลถูกต้อง	รหัสผ่าน, ข้อมูลการโอนเงิน	สามารถโอนเงินทางอินเทอร์เน็ตได้สำเร็จ
TC2	ซีแนริโอ 2: รหัสผ่านไม่ถูกต้อง, ข้อมูลถูกต้อง	รหัสผ่าน, ข้อมูลการโอนเงิน	ระบบแสดงข้อความเพื่อบอกว่ารหัสผ่านไม่ถูกต้อง
TC3	ซีแนริโอ 3: รหัสผ่านถูกต้อง, ข้อมูลไม่ถูกต้อง	รหัสผ่าน, ข้อมูลการโอนเงิน	ระบบแสดงข้อความเพื่อบอกถึงข้อมูลที่ไม่ถูกต้อง

2. กำหนดข้อมูลสำหรับกรณีทดสอบ ขั้นตอนนี้จะทำการกำหนดข้อมูลจริงสำหรับทดสอบในแต่ละกรณีทดสอบตามที่ได้กำหนดไว้ในขั้นตอนที่แล้ว โดยข้อมูลต่างๆ ผู้ทดสอบสามารถกำหนดได้ตามเงื่อนไขของการทดสอบนั้น ดังแสดงดังตารางที่ 3.11

ตารางที่ 3.11 การกำหนดข้อมูลสำหรับกรณีทดสอบ

รหัสกรณีทดสอบ	ซีแนริโอ	ข้อมูลที่ใช้ทดสอบ	ผลที่คาดว่าจะได้รับ
TC1	ซีแนริโอ 1: รหัสผ่านถูกต้อง, ข้อมูลถูกต้อง	- รหัสผ่าน: abc123 - จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขา: 003	สามารถโอนเงินทางอินเทอร์เน็ตได้สำเร็จ
TC2	ซีแนริโอ 2: รหัสผ่านไม่ถูกต้อง, ข้อมูลถูกต้อง	- รหัสผ่าน: 123 - จำนวนเงิน: 2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขา: 003	ระบบแสดงข้อความเพื่อบอกว่ารหัสผ่านไม่ถูกต้อง
TC3	ซีแนริโอ 3: รหัสผ่านถูกต้อง, ข้อมูลไม่ถูกต้อง	- รหัสผ่าน: abc123 - จำนวนเงิน: -2000 - หมายเลขบัญชีปลายทาง: 123-4567-89 - หมายเลขสาขา: -	ระบบแสดงข้อความเพื่อบอกถึงข้อมูลที่ไม่ถูกต้อง

บทที่ 4
การประเมินผลงานวิจัย

ในบทนี้จะนำเสนอแนวทางการประเมินผลรูปแบบคำอธิบายแอสเปคยูสเคสและขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส ซึ่งมีการจัดแบ่งโครงสร้างการนำเสนอออกเป็น 3 ส่วน โดยส่วนแรกจะกล่าวถึงการออกแบบการประเมินผลแนวคิดที่ได้นำเสนอไว้ในงานวิจัย ส่วนที่สองเป็นการนำเสนอผลการประเมินและการอภิปรายผลการประเมิน และส่วนสุดท้ายเป็นการสรุปผลการประเมิน

4.1 การออกแบบการประเมินผล

สำหรับแนวทางการออกแบบการประเมินผลนี้สามารถจัดแบ่งการประเมินออกเป็น 2 ส่วน ดังนี้ คือ

4.1.1 ส่วนการประเมินผลความสามารถของรูปแบบคำอธิบายแอสเปคยูสเคสสำหรับอธิบายข้อกำหนดความต้องการของแอสเปคตามมาตรฐานของแอสเปคเจ โดยเปรียบเทียบโครงสร้างหลักต่างๆ ของแอสเปคตามมาตรฐานของแอสเปคเจกับส่วนประกอบของแอสเปคยูสเคสที่ทำหน้าที่อธิบายโครงสร้างหลักนั้นๆ

4.1.2 ส่วนการประเมินผลขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส ทำการเปรียบเทียบความครอบคลุมเส้นการทำงานของกรณีทดสอบที่ถูกสร้างจากข้อกำหนดความต้องการซึ่งมีการครอสคัตติ้งคอนเซินที่อธิบายด้วยคำอธิบายแอสเปคยูสเคสโดยขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสที่ได้นำเสนอในงานวิจัยนี้กับกรณีทดสอบที่ถูกสร้างจากข้อกำหนดความต้องการเดียวกันซึ่งมีการครอสคัตติ้งคอนเซินที่อธิบายด้วยคำอธิบายยูสเคสโดยขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคสด้วยการทดสอบเบซิสพาท และพิจารณาเปรียบเทียบข้อดีข้อเสียระหว่างขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคสกับขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

4.2 การประเมินผล

การอธิบายผลการประเมินรูปแบบคำอธิบายแอสเปคยูสเคสและขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสที่นำเสนอในงานวิจัยนี้ จะแบ่งส่วนการอธิบายออกเป็น 2 ส่วน คือ การประเมินความสามารถของรูปแบบคำอธิบายแอสเปคยูสเคส และการการประเมินขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส ดังรายละเอียดต่อไปนี้

4.2.1 การประเมินความสามารถของรูปแบบคำอธิบายแอสเปคยูสเคส

ในส่วนนี้จะนำเสนอผลการประเมินที่ได้จากการเปรียบเทียบโครงสร้างหลักต่างๆ ของแอสเปคตามมาตรฐานแอสเปคเจกับส่วนประกอบของแอสเปคยูสเคสซึ่งทำหน้าที่อธิบายโครงสร้างหลักนั้นๆ ซึ่งแบ่งการอธิบายตามโครงสร้างหลักของแอสเปคตามมาตรฐานแอสเปคเจดังนี้คือ

1. จอยพอยท์ และพอยท์คัต

จอยพอยท์ คือ จุดของเหตุการณ์ในขั้นตอนการทำงานของโปรแกรม และพอยท์คัต เป็นส่วนที่ใช้สำหรับเลือกเหตุการณ์ของจอยพอยท์ เพื่อนำมากำหนดการทำงานของโปรแกรมเมื่อเกิดเหตุการณ์ของจอยพอยท์ดังที่ได้เลือกไว้ ซึ่งสามารถถูกอธิบายได้ด้วยส่วนของ “เงื่อนไข” ในรูปแบบคำอธิบายแอสเปคยูสเคส ดังตารางที่ 3.1 และสามารถแสดงตัวอย่างการอธิบายจอยพอยท์ และพอยท์คัตด้วยส่วนของ“เงื่อนไข” ดังตารางที่ 3.2 ซึ่งเป็นคำอธิบายแอสเปคยูสเคสการตรวจสอบรหัสผ่าน โดยในส่วนของ“เงื่อนไข” ได้กำหนดพอยท์คัตสำหรับกำหนดการทำงานของแอสเปคยูสเคสต่อยูสเคสที่เกี่ยวข้องซึ่งแสดงในส่วนของ “ยูสเคสที่ได้รับผลกระทบ” ตัวอย่างของพอยท์คัตที่ถูกกำหนดในส่วนของ“เงื่อนไข” ของตารางที่ 3.2 เช่น เหตุการณ์ระบบแสดงยอดเงินในบัญชีของยูสเคสการแสดงยอดคงเหลือ และเหตุการณ์ลูกค้ากำหนดจำนวนเงินที่ต้องการชำระหมายเลขบัญชีสินค้าที่ต้องการชำระของยูสเคสการชำระเงินสินค้า เป็นต้น

2. แอดไวซ์

แอดไวซ์ เป็นส่วนการทำงานของโปรแกรม เมื่อพอยท์คัตที่ได้กำหนดไว้ทำงาน และสามารถแบ่งแอดไวซ์ออกได้เป็น 3 ประเภท คือ Before Advice After Advice และ Around Advice ซึ่งสามารถอธิบายส่วนการทำงานของโปรแกรมได้ด้วยส่วนของ “ลำดับเหตุการณ์” กับ “ลำดับเหตุการณ์ทางเลือก” และการทำงานของแอดไวซ์ประเภทต่างๆ สามารถอธิบายได้ด้วยส่วนของ “การเชื่อมต่อ” ในรูปแบบต่างๆ คือ Before After และAround ดังตารางที่ 3.1 สามารถแสดงตัวอย่างการอธิบายแอดไวซ์ด้วยส่วนของ “ลำดับเหตุการณ์” กับ “ลำดับเหตุการณ์ทางเลือก” และส่วนของ “การเชื่อมต่อ” ดังตารางที่ 3.2 ซึ่งเป็นคำอธิบายแอสเปคยูสเคสการตรวจสอบรหัสผ่าน โดยในส่วนการทำงานหลักของแอสเปคยูสเคสนี้ที่เกี่ยวกับขั้นตอนการตรวจสอบรหัสผ่านจะถูกอธิบายอยู่ในส่วนของ “ลำดับเหตุการณ์” และการทำงานซึ่งเป็นทางเลือกจากการทำงานหลักของแอสเปคยูสเคสนี้ เช่น กรณีที่ผู้ใช้งานกรอกชื่อผู้ใช้งานและรหัสผ่านไม่ถูกต้อง จะถูกอธิบายอยู่ในส่วนของ “ลำดับเหตุการณ์ทางเลือก” ส่วนการทำงานของแอดไวซ์ประเภทต่างๆ ซึ่งเป็นรูปแบบการเชื่อมต่อของการทำงานของแอสเปคยูสเคสการตรวจสอบรหัสผ่าน จะถูกอธิบายไว้ในส่วนของ “การเชื่อมต่อ” เช่น แอสเปคยูสเคสนี้จะเริ่มต้นทำงานก่อนเหตุการณ์ระบบแสดงยอดเงินในบัญชีของยูสเคสการแสดงยอดคงเหลือ และจะเริ่มต้นทำงานหลักจากเหตุการณ์ลูกค้ากำหนด

จำนวนเงินที่ต้องการชำระ, หมายเลขบัญชีสินค้าที่ต้องการชำระของยูสเคสการชำระเงินสินค้า เป็นต้น

3. การประกาศประเภทข้อมูลร่วม

เป็นการประกาศตัวแปรและเมธอดภายในแอสเปค ซึ่งจะถูกกำหนดในขั้นตอนการโปรแกรม แต่ในงานวิจัยนี้เป็นทดสอบซอฟต์แวร์ในขั้นตอนการวิเคราะห์และการออกแบบข้อกำหนดความต้องการ จึงไม่นำการประกาศประเภทข้อมูลร่วมมาพิจารณา

4. แอสเปค

คือ มอดูลสำหรับครอสคัตติ้งคอนเซิน ซึ่งประกอบด้วย พอยท์คัต และแอดไวซ์ ซึ่งสามารถอธิบายด้วยรูปแบบคำอธิบายแอสเปคยูสเคสที่ได้นำเสนอในงานวิจัยนี้ โดยรูปแบบคำอธิบายแอสเปคยูสเคสจะประกอบด้วยส่วนประกอบต่างๆ ซึ่งสามารถทำหน้าที่อธิบายพอยท์คัตและแอดไวซ์ ตามที่ได้อธิบายไว้ที่หัวข้อที่ 1 และหัวข้อที่ 2

4.2.2 การประเมินผลขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

ในขั้นตอนนี้จะทำการประเมินผลขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส โดยการนำผลกรณีทดสอบของยูสเคสการโอนเงินในข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายแอสเปคยูสเคสในหัวข้อที่ 3.3 มาเปรียบเทียบกับผลของกรณีทดสอบที่ได้จากขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคสด้วยเบซิสพาทในหัวข้อที่ 2.6.4 และพิจารณาข้อดีข้อเสียของขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสกับขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคส ดังนี้

1. การประเมินผลการเขียนคำอธิบายยูสเคสเปรียบเทียบกับการเขียนคำอธิบายแอสเปคยูสเคส

การเขียนข้อกำหนดความต้องการด้วยคำอธิบายแอสเปคยูสเคส ทำให้การเขียนข้อกำหนดความต้องการของยูสเคสหลักกับยูสเคสย่อยที่ทำงานร่วมกันสามารถทำได้ง่ายขึ้น และลดความยุ่งยากเมื่อมีการเปลี่ยนแปลงการทำงานระหว่างกัน จากการเขียนข้อกำหนดความต้องการด้วยคำอธิบายยูสเคส ถ้าหากยูสเคสหลักมีการเรียกใช้ยูสเคสย่อยเพื่อทำงานร่วมกันภายในยูสเคสหลัก การเขียนคำอธิบายยูสเคสของยูสเคสหลักจะต้องมีการระบุการเรียกใช้งานยูสเคสย่อยภายในลำดับเหตุการณ์ของคำอธิบายยูสเคสหลัก ยกตัวอย่างเช่น คำอธิบายยูสเคสการโอนเงินในข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายยูสเคส ที่มีการระบุเพื่อการเรียกใช้ยูสเคสย่อยที่ลำดับเหตุการณ์ที่ 3 ลำดับเหตุการณ์ที่ 6 และลำดับเหตุการณ์ที่ 7 เพื่อเรียกใช้ยูสเคสการตรวจสอบรหัสผ่าน ยูสเคสการเรียกเก็บภาษี และยูสเคสการแสดงใบเสร็จตามลำดับ ดังรูปที่ 4.1 และถ้าต้องการเปลี่ยนแปลงการทำงานร่วมกันระหว่างยูสเคสหลักและยูสเคสย่อย เช่น การเปลี่ยนแปลงขั้นตอนการทำงานในการเรียกใช้งานยูสเคสย่อย การมียูสเคสย่อยมา

ทำงานร่วมกับยูสเคสหลักเพิ่มขึ้น และการยกเลิกการทำงานร่วมกันระหว่างยูสเคสหลักกับยูสเคสย่อย ด้วยกรณีการเปลี่ยนแปลงการทำงานร่วมกันระหว่างยูสเคสหลักและยูสเคสย่อยต่างๆ ดังกล่าว ทำให้ต้องทำการแก้ไขลำดับเหตุการณ์ภายในคำอธิบายยูสเคสหลัก ซึ่งมีความยุ่งยากในการที่จะต้องเปลี่ยนแปลงลำดับเหตุการณ์ทั้งหมดภายในคำอธิบายยูสเคสหลัก ในระบบซอฟต์แวร์ที่ประกอบด้วยยูสเคสหลักและยูสเคสย่อยที่ทำงานร่วมกันเป็นจำนวนมาก และมีความซับซ้อนในการทำงานร่วมกัน เมื่อต้องการเปลี่ยนแปลงการทำงานร่วมกันระหว่างยูสเคสหลักกับยูสเคสย่อยจะก่อให้เกิดความยุ่งยากในการเปลี่ยนแปลงลำดับเหตุการณ์ของคำอธิบายยูสเคสหลักเป็นอย่างมาก

ชื่อยูสเคส	การโอนเงิน
คำอธิบาย	การโอนเงินทางอินเทอร์เน็ต
เงื่อนไขก่อน	-
จุดต่อขยาย	-
ยูสเคสที่ถูกใช้งาน	1. การตรวจสอบรหัสผ่าน 2. การเรียกเก็บภาษี 3. แสดงใบเสร็จ
ลำดับเหตุการณ์	1. ลูกค้าเลือกเมนูการโอนเงิน 2. ลูกค้ากำหนดจำนวนเงินที่ต้องการโอน, หมายเลขบัญชีปลายทาง และหมายเลขสาขาของธนาคาร *3. ระบบทำการตรวจสอบรหัสผ่าน* *3.1 ใช้งาน "การตรวจสอบรหัสผ่าน"* 4.ระบบตรวจสอบความถูกต้องของข้อมูลการโอนที่ได้กำหนดไปในขั้นตอนที่แล้ว 5.ระบบหักเงินจากบัญชีของลูกค้าและเพิ่มเงินให้กับบัญชีปลายทาง *6. ระบบทำการเรียกเก็บภาษี* *6.1 ใช้งาน "การเรียกเก็บภาษี"* *7. ระบบแสดงใบเสร็จ* *7.1 ใช้งาน "แสดงใบเสร็จ"*
ลำดับเหตุการณ์ทางเลือก	4.1 ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง 4.1.1 ระบบแสดงข้อความเพื่อบอกถึงข้อมูลที่ไม่ถูกต้อง 4.1.2 กลับไปที่ขั้นตอนที่ 2
เงื่อนไขหลัง	-

รูปที่ 4.1 แสดงการระบุจุดเพื่อการเรียกใช้ยูสเคสย่อยของคำอธิบายยูสเคสการโอนเงินในกรณีศึกษาข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายยูสเคส

ชื่อแอสเปคยูสเคส	การตรวจสอบรหัสผ่าน	
คำอธิบายยูสเคส	ตรวจสอบรหัสผ่านของผู้มีสิทธิใช้งาน	
ลำดับเหตุการณ์	1. ระบบแสดงกรอบสนทนา (Dialog Box) สำหรับใส่รหัสผ่าน 2. ลูกค้ากรอกรหัสผ่าน 3. ระบบเปรียบเทียบรหัสผ่านที่ลูกค้ากรอกกับรหัสผ่านของบัญชี	
ลำดับเหตุการณ์ทางเลือก	3.1 ลูกค้าใส่รหัสผ่านไม่ถูกต้อง 3.1.1 ระบบแสดงข้อความเพื่อบอกว่ารหัสผ่านไม่ถูกต้อง 3.1.2 กลับไปยังขั้นตอนที่ 2	
ยูสเคสที่ได้รับผลกระทบ	*เงื่อนไข*	*การเชื่อมต่อ*
การแสดงยอดคงเหลือ	*ระบบแสดงยอดเงินในบัญชี*	*before*
การแสดงรายการธุรกรรม	*ระบบแสดงรายการของการถอนเงิน,การฝากเงินและคำอธิบาย ซึ่งเรียงตามลำดับวันที่*	*before*
การโอนเงิน	*ลูกค้ากำหนดจำนวนเงินที่ต้องการโอน, หมายเลขบัญชีปลายทางและหมายเลขสาขาของธนาคาร*	*after*
การชำระเงินสินค้า	*ลูกค้ากำหนดจำนวนเงินที่ต้องการชำระ, หมายเลขบัญชีสินค้าที่ต้องการชำระ*	*after*

รูปที่ 4.2 แสดงส่วนสำหรับการระบุถึงยูสเคสหลักของคำอธิบายแอสเปคยูสเคส

การเขียนข้อกำหนดความต้องการด้วยคำอธิบายแอสเปคยูสเคส สามารถแก้ไขความยุ่งยากในการเปลี่ยนแปลงการทำงานร่วมกันระหว่างยูสเคสหลักและยูสเคสย่อยได้ โดยภายในคำอธิบายแอสเปคยูสเคสมีส่วน "ยูสเคสที่ได้รับผลกระทบ" สำหรับการระบุถึงยูสเคสหลักที่แอสเปคยูสเคสได้เข้าไปทำงานร่วมด้วย และส่วน "การเชื่อมต่อ" สำหรับระบุถึงขั้นตอนการทำงานของยูสเคสหลักที่การเรียกใช้งานแอสเปคยูสเคส ดังรูปที่ 4.2 ที่แสดงคำอธิบายแอสเปคยูสเคสการตรวจสอบรหัสผ่านซึ่งเป็นยูสเคสย่อยที่ทำงานร่วมกับยูสเคสการโอนเงินซึ่งเป็นยูสเคสหลัก ทำให้ภายในคำอธิบายยูสเคสหลักไม่ต้องมีการระบุการเรียกใช้งานยูสเคสย่อยในขั้นตอนการทำงานของยูสเคสหลักที่อธิบายด้วยคำอธิบายแอสเปคยูสเคส ดังรูปที่ 4.3 ซึ่งแสดงคำอธิบายยูสเคสการโอน

เงินในกรณีศึกษาข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายแอสเปคยูสเคส

ชื่อยูสเคส	การโอนเงิน
คำอธิบาย	การโอนเงินทางอินเทอร์เน็ต
เงื่อนไขก่อน	-
จุดต่อขยาย	-
ยูสเคสที่ถูกใช้งาน	1. การแสดงใบเสร็จ
ลำดับเหตุการณ์	1. ลูกค้าเลือกเมนูการโอนเงิน 2. ลูกค้ากำหนดจำนวนเงินที่ต้องการโอน, หมายเลขบัญชีปลายทาง และหมายเลขสาขาของธนาคาร 3.ระบบตรวจสอบความถูกต้องของข้อมูลการโอนที่ได้กำหนดไปในขั้นตอนที่แล้ว 4.ระบบหักเงินจากบัญชีของลูกค้าและเพิ่มเงินให้กับบัญชีปลายทาง *5. ระบบแสดงใบเสร็จ* *5.1 ใช้งาน "แสดงใบเสร็จ"*
ลำดับเหตุการณ์ทางเลือก	3.1 ข้อมูลที่ลูกค้ากำหนดไม่ถูกต้อง 3.1.1 ระบบแสดงข้อความเพื่อบอกถึงข้อมูลที่ไม่ถูกต้อง 3.1.2 กลับไปที่ขั้นตอนที่ 2
เงื่อนไขหลัง	-

รูปที่ 4.3 แสดงคำอธิบายยูสเคสการโอนเงินในกรณีศึกษาข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายยูสเคส

เมื่อต้องการเปลี่ยนแปลงการทำงานร่วมกันระหว่างยูสเคสหลักกับยูสเคสย่อยทำให้ไม่ต้องทำการแก้ไขลำดับขั้นตอนการทำงานภายในคำอธิบายยูสเคสหลัก เพียงแต่ทำการเปลี่ยนแปลงยูสเคสหลักที่เรียกใช้งานแอสเปคยูสเคสที่กำหนดไว้ในคำอธิบายแอสเปคยูสเคสเท่านั้นเอง ทำให้การเขียนคำอธิบายข้อกำหนดความต้องการ และการเปลี่ยนแปลงการทำงานร่วมกันระหว่างยูสเคสหลักกับยูสเคสย่อยในระบบซอฟต์แวร์ลดความยุ่งยากลงและสามารถทำได้ง่ายขึ้น

2. การประเมินผลการสร้างกรณีทดสอบจากยูสเคสเปรียบเทียบกับการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส

2.1 จากการเปรียบเทียบขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสกับขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคส ขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสได้จัดเตรียมรูปแบบคำอธิบายแอสเปคยูสเคสโดยมีส่วนประกอบสำหรับกำหนดยูสเคสหลัก เงื่อนไขในการเรียกใช้แอสเปคยูสเคสของยูสเคสหลักและรูปแบบการเชื่อมต่อแอสเปคยูสเคสกับยูสเคสหลักไว้ ส่วนประกอบดังกล่าวก็คือ "ยูสเคสที่ได้รับผลกระทบ" "เงื่อนไข" และ "การเชื่อม" ดังตารางที่ 3.1 ในขั้นตอนการสร้างแผนภาพกิจกรรมของขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสได้กำหนดให้แต่ละคำอธิบายยูสเคสหลักและคำอธิบายแอสเปคยูสเคสสร้างแผนภาพกิจกรรมของตัวเองขึ้นมา ซึ่งการนำแผนภาพกิจกรรมแอสเปคยูสเคสเชื่อมต่อเข้ากับแผนภาพกิจกรรมยูสเคสหลักสามารถทำตาม "เงื่อนไข" และ "การเชื่อมต่อ" ของยูสเคสหลักที่ต้องการเชื่อมต่อที่ได้กำหนดไว้ในคำอธิบายแอสเปคยูสเคส และด้วยขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสได้แยกพิจารณาคำอธิบายยูสเคสหลักกับคำอธิบายแอสเปคยูสเคส ทำให้สามารถทำความเข้าใจขั้นตอนการทำงานของยูสเคสหลักและการทำงานของแอสเปคยูสเคสได้ง่าย และส่งผลทำให้สามารถสร้างแผนภาพกิจกรรมของยูสเคสหลักกับแอสเปคยูสเคสได้ง่ายด้วยเช่นกัน แตกต่างจากการสร้างแผนภาพกิจกรรมจากคำอธิบายยูสเคสที่ต้องนำคำอธิบายยูสเคสหลักและคำอธิบายยูสเคสย่อยมาพิจารณาร่วมกันถึงการเรียกใช้งานของยูสเคสหลักที่มีต่อยูสเคสย่อย แล้วจึงสร้างเป็นแผนภาพกิจกรรมหลักที่มีการเชื่อมต่อกับการทำงานของยูสเคสย่อยที่เกี่ยวข้องที่สมบูรณ์ ภายในระบบซอฟต์แวร์ยูสเคสหลักมีความเกี่ยวข้องกับยูสเคสย่อยจำนวนมาก ทำให้การสร้างแผนภาพกิจกรรมจากคำอธิบายยูสเคสหลักนั้นจะต้องนำคำอธิบายยูสเคสย่อยจำนวนมากที่เกี่ยวข้องมาร่วมพิจารณาด้วย ซึ่งทำให้ยากต่อการทำความเข้าใจขั้นตอนการทำงานของยูสเคสหลักกับยูสเคสย่อยจำนวนมาก และทำให้การสร้างแผนภาพกิจกรรมสามารถสร้างได้ยากมากขึ้นเช่นกัน

2.2 จากกรณีทดสอบของยูสเคสการโอนเงินในข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายยูสเคสที่ได้จากการทดสอบเบซิสพาสซึ่งมีความครอบคลุมทุกเส้นทางการทำงานและเพียงพอต่อการทดสอบกับกรณีทดสอบของยูสเคสการโอนเงินในข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายแอสเปคยูสเคสที่ได้จากขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส ได้ผลของกรณีทดสอบที่เหมือนกัน เนื่องจากการสร้างกรณีทดสอบของยูสเคสการโอนเงินในข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบายด้วยคำอธิบายแอสเปคยูสเคสสามารถสร้างแผนภาพกิจกรรมยูสเคสการโอนเงินที่มีขั้นตอนการทำงานที่เหมือนกับแผนภาพกิจกรรมยูสเคสการโอนเงินการสร้างกรณีทดสอบของยูสเคสการโอนเงินในข้อกำหนดความต้องการระบบธนาคารอินเทอร์เน็ตที่อธิบาย

ด้วยคำอธิบายยูสเคสที่ได้จากการทดสอบเบซิสพาส ดังรูปที่ 2.15 และรูปที่ 3.9 และด้วยขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสได้ใช้การทดสอบเบซิสพาทเพื่อหาเส้นทางอิสระเชิงเส้นสำหรับนำไปกำหนดเป็นซีแนริโอเพื่อสร้างเป็นกรณีทดสอบเช่นเดียวกับการสร้างกรณีทดสอบจากยูสเคสด้วยการทดสอบเบซิสพาททำให้ได้กรณีทดสอบที่เหมือนกัน แสดงให้เห็นว่าขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสสามารถสร้างกรณีทดสอบของข้อกำหนดความต้องการที่เกิดการครอสคัตติ้งคอนเซินได้ครอบคลุมทุกเส้นทางการทำงานและเพียงพอต่อการทดสอบเช่นเดียวกับการสร้างกรณีทดสอบจากยูสเคสด้วยการทดสอบเบซิสพาท

4.3 สรุปการประเมินผล

จากการประเมินผลรูปแบบคำอธิบายแอสเปคยูสเคส และขั้นตอนวิธีขั้นตอนการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสข้างต้น สามารถสรุปผลการประเมินผลได้ดังนี้ คือ

4.3.1 รูปแบบคำอธิบายแอสเปคยูสเคสสามารถอธิบายข้อกำหนดความต้องการของแอสเปคตามมาตรฐานของแอสเปคเจได้อย่างครบถ้วน ไม่ว่าจะเป็นจอยพอยท์ พอยท์คัต แอดไวซ์ และแอสเปค

4.3.2 การเขียนข้อกำหนดความต้องการด้วยคำอธิบายแอสเปคยูสเคส ทำให้การเขียนข้อกำหนดความต้องการของยูสเคสหลักกับยูสเคสย่อยที่ทำงานร่วมกันสามารถทำได้ง่ายขึ้น และลดความยุ่งยากเมื่อมีการเปลี่ยนแปลงการทำงานระหว่างกัน เนื่องคำอธิบายแอสเปคยูสเคสได้มีส่วนสำหรับกำหนดยูสเคสหลักและเงื่อนไขการเรียกใช้แอสเปคยูสเคสของยูสเคสหลัก ทำให้ง่ายต่อการแก้ไขเมื่อต้องการเปลี่ยนแปลงการทำงานระหว่างกันของยูสเคสหลักกับยูสเคสย่อย

4.3.3 ขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสมีขั้นตอนการสร้างกรณีทดสอบที่ง่ายกว่าขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคส เนื่องจากทำการแยกพิจารณาระหว่างคำอธิบายยูสเคสหลักและคำอธิบายแอสเปคยูสเคส จึงทำให้สามารถเข้าใจและสร้างแผนภาพกิจกรรมจากคำอธิบายยูสเคสและจากคำอธิบายแอสเปคยูสเคสได้ง่าย และการนำแผนภาพกิจกรรมแอสเปคยูสเคสไปเชื่อมต่อกับแผนภาพกิจกรรมยูสเคสหลัก สามารถทำได้โดยดูเงื่อนไขการเชื่อมต่อได้จากส่วนประกอบ “ยูสเคสที่ได้รับผลกระทบ” “เงื่อนไข” และ “การเชื่อม” ของคำอธิบายแอสเปค ด้วยวิธีการนี้ทำให้การสร้างแผนภาพกิจกรรมสามารถทำได้ง่ายขึ้นกว่าการสร้างแผนภาพกิจกรรมที่ต้องนำคำอธิบายยูสเคสหลักกับคำอธิบายยูสเคสย่อยที่เกี่ยวข้องมาพิจารณาร่วมกันของขั้นตอนวิธีการสร้างกรณีทดสอบจากยูสเคส ซึ่งจะก่อให้เกิดความยุ่งยากในการสร้างแผนภาพกิจกรรม

4.3.4 ขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสที่ได้นำเสนอในงานวิจัยนี้ สามารถสร้างกรณีทดสอบของข้อกำหนดความต้องการที่เกิดการครอสคัตติ้งคอนเซิน

ได้ครอบคลุมทุกเส้นการทำงาน และเพียงพอต่อการทดสอบ เนื่องจากทำการค้นหาเส้นทางของกรณีทดสอบด้วยการทดสอบเบซิสพาท

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุป

การทดสอบซอฟต์แวร์เป็นขั้นตอนหนึ่งในกระบวนการพัฒนาซอฟแวร์ เพื่อช่วยให้ซอฟต์แวร์ที่พัฒนามีความถูกต้องตามข้อกำหนดความต้องการของผู้ใช้งาน จากการศึกษาพบว่าการทดสอบซอฟต์แวร์จะเริ่มต้นที่ขั้นตอนการโปรแกรมเป็นส่วนใหญ่ ซึ่งการเริ่มต้นการทดสอบซอฟต์แวร์ที่ขั้นตอนการโปรแกรมอาจก่อให้เกิดความเสียหายอย่างมากในโครงการพัฒนาซอฟต์แวร์ได้ โดยเฉพาะข้อผิดพลาดที่ค้นพบเกิดขึ้นตั้งแต่ขั้นตอนการวิเคราะห์และออกแบบซอฟต์แวร์ ข้อผิดพลาดดังกล่าวทำให้ระยะเวลาและค่าใช้จ่ายเพิ่มขึ้นจากการที่ต้องเริ่มต้นการพัฒนาซอฟต์แวร์ใหม่ทั้งหมด มีการนำเสนอแนวทางเพื่อช่วยแก้ไขปัญหาข้างต้นโดยการเริ่มต้นการทดสอบซอฟต์แวร์ตั้งแต่ขั้นตอนการวิเคราะห์และออกแบบซอฟต์แวร์ ซึ่งจะช่วยลดความเสี่ยงในการเกิดข้อผิดพลาดในขั้นตอนอื่นๆ ของกระบวนการพัฒนาซอฟต์แวร์ได้ การสร้างกรณีทดสอบจากยูสเคสเป็นวิธีการหนึ่งในการทดสอบซอฟต์แวร์ตั้งแต่ขั้นตอนการวิเคราะห์และออกแบบซอฟต์แวร์ที่ถูกนำเสนอเป็นงานวิจัยและได้รับความสนใจเป็นอย่างมาก โดยเป็นการสร้างกรณีทดสอบจากข้อกำหนดความต้องในรูปแบบของยูสเคส

ในปัจจุบันมีกระบวนการพัฒนาซอฟต์แวร์เชิงแอสเปค ซึ่งเป็นกระบวนการพัฒนาซอฟต์แวร์ที่พัฒนาขึ้นเพื่อแก้ไขปัญหาการเกิดครอสคัตติ้งคอนเซิน โดยทำการแยกส่วนคอนเซินที่มีลักษณะปัญหาดังกล่าวออกเป็นมอดูลใหม่ที่เรียกว่าแอสเปค มีงานวิจัยจำนวนมากที่เกี่ยวกับการทดสอบซอฟต์แวร์เชิงแอสเปค แต่งานวิจัยส่วนใหญ่นำเสนอวิธีการทดสอบซอฟต์แวร์เชิงแอสเปคในขั้นตอนการโปรแกรมและหลังจากขั้นตอนการพัฒนาซอฟต์แวร์เสร็จสิ้น ดังนั้นงานวิจัยนี้จึงนำเสนอแนวทางการทดสอบซอฟต์แวร์เชิงแอสเปคตั้งแต่ขั้นตอนการวิเคราะห์และออกแบบซอฟต์แวร์ โดยออกแบบวิธีการสร้างกรณีทดสอบจากข้อกำหนดความต้องในรูปแบบของคำอธิบายยูสเคสให้มีความครอบคลุมทุกเส้นทางการทำงานของข้อกำหนดความต้องการ และได้ออกแบบรูปแบบคำอธิบายแอสเปคยูสเคสเพื่ออธิบายขั้นตอนการทำงานของแอสเปคตามมาตรฐานของแอสเปคเจ

ผลการประเมินงานวิจัยสามารถแบบออกได้เป็น 2 ส่วน คือ

1. รูปแบบคำอธิบายแอสเปคยูสเคสที่ได้นำเสนอในงานวิจัยนี้สามารถอธิบายขั้นตอนการทำงานและโครงสร้างต่างๆ ของแอสเปคตามมาตรฐานแอสเปคเจได้

2. ขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสสามารถสร้างกรณีทดสอบของข้อกำหนดความต้องการมีการครอสคัตติ้งคอนเซินได้อย่างครอบคลุมทุกเส้นทางการทำงานในข้อกำหนดความต้องการ เพียงพอต่อการทดสอบ และง่ายต่อการสร้างกรณีทดสอบ อีกทั้งการเขียนคำอธิบายข้อกำหนดความต้องการด้วยคำอธิบายแอสเปคยูสเคสช่วยทำให้การเขียนคำอธิบายของยูสเคสหลักกับยูสเคสย่อยที่ทำงานร่วมกันสามารถทำได้ง่ายขึ้น และช่วยลดความยุ่งยากในการเปลี่ยนแปลงการทำงานระหว่างกันของยูสเคสหลักและยูสเคสย่อย

5.2 ข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนอขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคส โดยเน้นที่การออกแบบรูปแบบคำอธิบายแอสเปคยูสเคสเพื่อให้สามารถอธิบายขั้นตอนการทำงานของแอสเปค และออกแบบขั้นตอนวิธีการสร้างกรณีทดสอบจากคำอธิบายแอสเปคยูสเคสเพื่อสร้างกรณีทดสอบที่ครอบคลุมทุกขั้นตอนการทำงานและเพียงพอต่อการทดสอบ ซึ่งสามารถนำไปใช้เป็นงานวิจัยพื้นฐานในการทำงานวิจัยเพิ่มเติมอื่นๆ เช่น การสร้างกรณีทดสอบจากขั้นตอนการวิเคราะห์และออกแบบซอฟต์แวร์ในการพัฒนาซอฟต์แวร์เชิงแอสเปคอย่างอัตโนมัติ

เอกสารอ้างอิง

[1] Awais Rashid. **Aspect-Orented Database Systems.** Berlin : Springer-Verlag. 2004.

[2] BioLogic Software Consulting. **"Use Case Modeling."** [Online]. Available http://www.biologicsoftwareconsulting.com/training/examples/Use%20Case%20Modeling.pdf. 2005.

[3] Carol Britton and Jill Doake. **Object-Oriented Systems Development: a gentle introduction.** New York : McGraw-Hill Higher Education. 2000.

[4] Carol Britton and Jill Doake. **Software System Development : A Gentle Introduction.** New York : McGraw-Hill Higher Education. 2005.

[5] Chen Mingsong, Qiu Xiaokang and Li Xuandong. "Automatic Test Case Generation for UML Activity Diagrams." **Proceedings of the 2006 international workshop on Automation of software test.** 2006. pp. 2-8.

[6] Cockburn A. **Writing Effective Use Case.** MA: Addison-Wesley. 2001.

[7] Cristina Videira Lopes and Trung Chi Ngo. "Unit-Testing Aspectual Behavior." **Proceedings of the 4th International Conference on Aspect-Oriented Software Development.** 2005.

[8] Dave Wood and Jim Reis. "Use Case Derived Test Cases." **STAREAST on Software Quality Engineering Conference.** 1999.

[9] Dianxiang Xu, Weifeng Xu and Kendall Nygard. "A State-Based Approach to Testing Aspect-Oriented Programs." **Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering.** 2005.

[10] Eric J. Braude. **Software Design: from Programming to Architecture. NJ : Wiley. 2003.**

[11] Georgia Sousa, Sergio Soares, Paulo Borba and Jaelson Castro. "Separation of Crosscutting Concerns from Requirements to Design: Adapting an Use Case Driven Approach." **Proceedings Early Aspects Workshop at AOSD.** 2004. pp. 98-107.

[12] Geri Schneider and Jason P. Winters. **Applying Use Cases : A Practical Guide.** Canada : Addison-Wesley Professional. 1998.

[13] Gunnar Overgaard and Karin Palmkvist. **Use Cases : Patterns and Blueprints.** Canada : Addison-Wesley Professional. 2004.

[14] Javier J. Gutierrez, Maria J. Escalona, Manuel Mejias and Jesus Torres. "An Approach to Generate Test Cases from Use Cases." **Proceedings of the 6th international conference on Web engineering.** 2006. pp. 113-114

[15] Jerry Gao, H-S Jacob Tsao and Ye Wu. **Testing and Quality Assurance for Component-Based Software.** Boston : Artech House Publishers. 2003.

[16] Jianjun Zhao. "Data-Flow-Based Unit Testing of Aspect-Oriented Programs." **Proceedings of the 27th Annual International Conference on Computer Software and Applications.** 2003. pp. 188.

[17] Jim Heumann. **"Generating Test Cases From Use Cases."** Rational Software, IBM. 2001.

[18] Kshirasagar Naik and Priyadarshi Tripathy. **Software Testing and Quality Assurance : Theory and Practice.** NJ : Wiley. 2008.

[19] Kwang Ik Seo. and Eun Man Choi. "Comparison of Five Black-box Testing Methods for Object-Oriented Software." **Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications.** 2006. pp. 213-220.

[20] Noraida Ismail, Rosziati Ibrahim and Noraini Ibrahim. "Automatic Generation of Test Cases from Use Case Diagram." **Proceedings of the International Conference on Electrical Engineering and Informatics.** 2007. pp. 699-702.

[21] Paul C. Jorgensen. **Software Testing : A Craftsman's Approach.** FL : CRC Press. 1995.

[22] Perdita Stevens and Rob Pooley. **Using UML: Software Engineering with Objects and Components.** Canada : Addison-Wesley Professional. 1999.

[23] Peter Zielczynski. **"Traceability from Use Cases to Test Cases."** [Online]. Available http://www.ibm.com/developerworks/rational/library/04/r-3217/. 2006.

[24] Ramnivas Laddad. **AspectJ in Action.** Connecticut : Manning Publications. 2003.

[25] Ramnivas Laddad. **"I want my AOP!, Part 1."** [Online]. Available http://www.javaworld.com/javaworld/jw-01-2002/jw-0118-aspect.html. 2002.

[26] Rebecca J. Wirfs-Brock. "Designing Scenarios: Making the Case for a Use Case Framework" **The Smalltalk Report Vol.3, No. 3.** 1993.

[27] Renaud Pawlak and the JAC development team. **"What is AOP ?."** [Online]. Available http://jac.ow2.org/docs/introduction.html. 2002.

[28] Reza Meimandi Parizi and Abdul Azim Ghani. "A Survey on aspect-Oriented Testing

Approaches." **Proceedings of the The 2007 International Conference Computational Science and its Applications.** 2007. pp. 78-85.

[29] Robert E. Filman, Tzilla Elrad, Siobhán Clarke and Mehmet Aksit. **Aspect-Oriented Software Development.** Boston : Addison-Wesley Professional. 2004.

[30] Robert V. Binder. **Testing Object-Oriented Systems: Models, Patterns, and Tools.** Canada : Addison-Wesley Professional. 1999.

[31] Sant'Anna, C., Garcia, A., Chavez, C., Lucena, C., Staa, A. "On the Reuse and Maintenance of Aspect-Oriented Software: An Assessment Framework." **Proceedings of Brazilian Symposium on Software Engineering.** 2003. pp. 19-34.

[32] The Aspect-Oriented Software Architecture Design Portal. **"Aspect-Oriented Software Development."** [Online]. Available http://trese.cs.utwente.nl/taosad/. 2007.

[33] Wikipedia. **"Aspect-Oriented Software Development."** [Online]. Available http://en.wikipedia.org/wiki/Aspect-Oriented_Software_Development. 2009.

[34] William E. Parry. **Effective Methods for Software Testing.** NJ : Wiley. 2000.

[35] วิกิพีเดีย สารานุกรมเสรี. **"Software testing."** [Online]. Available http://th.wikipedia.org/wiki/Software_testing. 2552.

[36] ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. **"White-Box Tesing."** [Online]. Available http://wiki.nectec.or.th/setec/Knowledge/SoftwareTesting. 2008.

[37] สุนทริน วงศ์ศิริกุล และ ชัยวัฒน์ สิทธิกรโอฬารกุล. **การพัฒนาโมเดลสำหรับการเขียนโปรแกรมเชิงวัตถุด้วย UML 2.0.** กรุงเทพมหานคร : บริษัท ซัคเซส มีเดีย จำกัด. 2005.

ประวัติผู้เขียน

ชื่อ-สกุล	นายองอาจ คูศรี
วัน เดือน ปีเกิด	21 มกราคม พ.ศ. 2522
ที่อยู่	211 หมู่ 1 ตำบลปลวกแดง อำเภอปลวกแดง จังหวัดระยอง 21140
ประวัติการศึกษา	2544 วิทยาศาสตรบัณฑิต สาขาฟิสิกส์อุตสาหกรรมและอุปกรณ์การแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ 2546 วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหิดล