

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

ระบบสนับสนุนการทดสอบซอฟต์แวร์

SOFTWARE TESTING SUPPORT SYSTEM

โดย



H005947



ยูวดี จันทรรัตน์

YUWADEE CHANTON

อาจารย์ที่ปรึกษา

ผศ.ดร.พรฤดี เนติโสภาคกุล

ณ

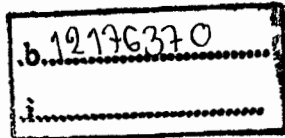
บ 4428  
2551

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

เลขหมู่.....  
05947

คณะเทคโนโลยีสารสนเทศ



เลขทะเบียน.....  
3 ก.พ. 2553

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาคเรียนที่ 2 ปีการศึกษา 2551 อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# SOFTWARE TESTING SUPPORT SYSTEM



**A SYSTEM DEVELOPMENT PROJECT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY  
FACULTY OF INFORMATION TECNOLOGY  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อปี 2/ 2008 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2009**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG** ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|                  |                               |
|------------------|-------------------------------|
| หัวข้อ           | ระบบสนับสนุนการทดสอบซอฟต์แวร์ |
| นักศึกษา         | นางสาวยุวดี จันทร์ตัน         |
| รหัสนักศึกษา     | 49066816                      |
| ปริญญา           | วิทยาศาสตร์มหาบัณฑิต          |
| สาขาวิชา         | เทคโนโลยีสารสนเทศ             |
| แขนงวิชา         | วิทยาการสารสนเทศ              |
| ปีการศึกษา       | 2551                          |
| อาจารย์ที่ปรึกษา | ผศ.ดร.พรฤดี เนติโสภาคกุล      |

### บทคัดย่อ

ปัจจุบันหน่วยงานหรือองค์กรต่างๆ ได้นำระบบคอมพิวเตอร์เข้ามาใช้งานอย่างแพร่หลาย เนื่องจากระบบคอมพิวเตอร์มีส่วนช่วยให้การทำงานในด้านต่างๆ สะดวกสบาย และรวดเร็วขึ้น แต่การลงทุนติดตั้งระบบคอมพิวเตอร์นั้น ยังเป็นการลงทุนที่มีมูลค่าค่อนข้างสูง ทั้งในด้านอุปกรณ์ชิ้นส่วนต่างๆ ตลอดจนซอฟต์แวร์ที่นำมาใช้งาน การลงทุนทางด้านฮาร์ดแวร์นั้นผู้ใช้สามารถเลือกฮาร์ดแวร์ ที่มีประสิทธิภาพเหมาะสมกับการใช้งาน และมีการรับประกันความเสียหายของฮาร์ดแวร์ที่เกิดขึ้นในระยะเวลาที่กำหนด ซึ่งแตกต่างจากซอฟต์แวร์ที่ไม่มีใครสามารถรับประกันได้ว่าซอฟต์แวร์ที่นำมาใช้มีประสิทธิภาพและมีคุณสมบัติตรงกับความต้องการหรือไม่ นอกจากนี้การทดสอบซอฟต์แวร์จนผู้ใช่มั่นใจได้ว่าซอฟต์แวร์ที่จะนำมาใช้งานมีประสิทธิภาพและตรงกับความต้องการของผู้ใช้งาน การทดสอบซอฟต์แวร์เป็นขั้นตอนที่สำคัญในกระบวนการพัฒนาซอฟต์แวร์ เพื่อให้ผู้ใช้งานซอฟต์แวร์มั่นใจได้ว่า ซอฟต์แวร์ที่จะนำมาใช้งานสามารถทำงานได้ถูกต้องตรงตามความต้องการในองค์กรขนาดใหญ่ที่ต้องการความถูกต้องแม่นยำของข้อมูลสูง เช่น ธนาคาร จะให้ความสำคัญกับทุกขั้นตอนในกระบวนการพัฒนาซอฟต์แวร์ จากการศึกษากระบวนการทดสอบซอฟต์แวร์ของธนาคารพาณิชย์แห่งหนึ่งพบว่า ขั้นตอนการทดสอบซอฟต์แวร์จะให้ความสำคัญด้านเอกสาร เนื่องจากใช้เป็นหลักฐานยืนยันการทดสอบและพบว่าผู้ทดสอบแต่ละคนต้องทดสอบซอฟต์แวร์หลายๆระบบ ให้รองรับกับนโยบายของธนาคารที่ต้องการให้ทดสอบซอฟต์แวร์ก่อนที่จะใช้งานจริง ดังนั้นในกระบวนการทดสอบซอฟต์แวร์ควรจะมีซอฟต์แวร์ที่สามารถอำนวยความสะดวกให้กับผู้ทดสอบในระดับหนึ่ง เพื่อให้ผู้ทดสอบได้ทำการทดสอบซอฟต์แวร์ได้อย่างเต็มประสิทธิภาพ

|                      |  |
|----------------------|--|
| <b>Title</b>         | Software Testing Support System        |
| <b>Student</b>       | Miss Yuwadee Chanton                   |
| <b>Student ID.</b>   | 49066816                               |
| <b>Degree</b>        | Master of Science                      |
| <b>Program</b>       | Information Science                    |
| <b>Academic Year</b> | 2008                                   |
| <b>Advisor</b>       | Assoc. Prof. Dr. Ponrudee Netisophakul |

## ABSTRACT

Software testing is an essential process under the software development life cycle. It is conducted in order to confirm whether software can be used properly. In large organizations e.g. banks which require highly concrete data / information, focus will be on every process of software development. According to the survey on software testing process of a commercial bank, the findings showed that importance is given to documentation because it can be used to confirm testing – each user need to test several software systems before implementation in order to be in line with the bank’s policy. So, on software testing process there should be software that can support the users at a certain level to make a complete and efficient testing.

Due to the extensive use of computer within units, IT Support Department is required to do system maintenance, provide assistance and solve problems as requested by users. At present, user request is prepared as paper document signed off by supervisor before submitting to IT Support Department., resulting in waste and operation delay. In addition, there is no efficient process to check and follow up requests. Sometimes problems are the same or recurring but officers need to take time to search for old documents and it’s difficult to check and follow up results. All in all, therefore, a new system has been developed to support different terms of use to be more efficient. Also, the Department’s officers are ready to service to enable users to fully use computers and equipments. Given this, levels of user satisfaction is so important for the evaluation of working operations. The system for accepting user requests can take user satisfaction into consideration as part of the performance evaluation of the Department’s officers.

# กิตติกรรมประกาศ

โครงการเรื่อง “ระบบสนับสนุนการทดสอบซอฟต์แวร์” นี้สำเร็จได้ด้วยความสำเร็จจากอาจารย์ที่ปรึกษา ศศ.ดร.พรฤดี เนติโสภากุล ที่กรุณาให้คำปรึกษาแนะนำ และช่วยตรวจสอบแก้ไขข้อบกพร่องของโครงการนี้ ตลอดจนให้ความรู้และข้อคิดเห็นที่เป็นประโยชน์อย่างยิ่งต่อโครงการ

ขอขอบคุณบุคลากรของคณะเทคโนโลยีสารสนเทศทุกท่าน ที่ให้ความช่วยเหลือเรื่องเอกสารและให้ความอนุเคราะห์ตลอดมา

ขอขอบคุณเพื่อนๆ และพี่ๆ สาขาวิชาเทคโนโลยีสารสนเทศ ที่ได้ให้คำแนะนำช่วยเหลือผู้จัดทำ ในเรื่องแนวทางการเขียนและแก้ไข โปรแกรมให้สามารถทำงานได้ประสบผลสำเร็จ

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่ร่วมเป็นกำลังใจ และให้การสนับสนุนในทุกๆเรื่อง ด้วยดีเสมอมา

สำหรับคุณงามความดีและประโยชน์อันพึงมาจากโครงการฉบับนี้ ข้าพเจ้าขอมอบให้กับ บิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

ยวดี จันทร์ตัน

# สารบัญ

หน้า

|                         |      |
|-------------------------|------|
| บทคัดย่อภาษาไทย .....   | I    |
| บทคัดย่อภาษาอังกฤษ..... | II   |
| กิตติกรรมประกาศ.....    | III  |
| สารบัญ.....             | IV   |
| สารบัญตาราง.....        | VII  |
| สารบัญภาพ.....          | VIII |

|   |    |
|---|----|
| บทที่ 1 บทนำ.....   | 1  |
| 1.1 ความเป็นมาในการพัฒนาระบบ.....                               | 1  |
| 1.2 วัตถุประสงค์ในการพัฒนาระบบ.....                             | 2  |
| 1.3 ขอบเขตของระบบ.....  | 2  |
| 1.4 ขั้นตอนในการวิเคราะห์และออกแบบระบบ.....                     | 3  |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ.....                              | 3  |
| บทที่ 2 ทฤษฎีพื้นฐานที่เกี่ยวข้อง.....                          | 5  |
| 2.1 เทคโนโลยี.....  | 5  |
| .Net.....   |    |
| 2.2 .Net Framework คืออะไร.....                                 | 5  |
| 2.3 ส่วนประกอบของ .Net Framework.....                           |    |
| 2.4 Microsoft SQL Server 2005 .....                             | 7  |
| 2.5 ทฤษฎีการทดสอบซอฟต์แวร์.....                                 | 5  |
| 2.5.1 การออกแบบกรณีทดสอบ (Test Case Design).....                | 7  |
| 2.5.2 การทดสอบแบบกล่องขาว (White Box Testing).....              | 9  |
| 2.5.3 การทดสอบโครงสร้างควบคุม (Control Structure).....          | 13 |
| 2.5.4 การทดสอบแบบกล่องดำ (Black Box Testing).....               | 14 |
| 2.5.5 การทดสอบซอฟต์แวร์เชิงวัตถุ (Object Oriented Testing)..... | 16 |

|  |    |
|--|----|
| บทที่ 3 การวิเคราะห์และออกแบบระบบ..... | 18 |
|--|----|

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

|   | หน้า |
|---|------|
| 3.1 ปัญหาความเป็นมาและแนวทางการแก้ไข..... | 18   |
| 3.2 การวิเคราะห์ความต้องการของระบบ.....   | 19   |
| 3.3 ระบบสนับสนุนการทดสอบซอฟต์แวร์         | 21   |
| 3.3.1 การออกแบบยูสเคสไดอะแกรม.....        | 22   |
| 3.3.2 การออกแบบแอกทิวิตีไดอะแกรม.....     | 30   |
| 3.3.3 การออกแบบคลาสไดอะแกรม.....          | 31   |
| 3.3.4 การออกแบบซีแควนซ์ไดอะแกรม.....      | 34   |
| 3.3.5 การออกแบบอีอาร์ไดอะแกรม.....        | 39   |
| 3.3.6 พจนานุกรมข้อมูล.....                | 40   |
| บทที่ 4 การพัฒนาระบบงาน.....              | 48   |
| 4.1 ส่วนประกอบของระบบ.....                | 48   |
| 4.2 การออกแบบจอภาพ.....                   | 48   |
| 4.2.1 กำหนดรายละเอียดการทดสอบ .....       | 49   |
| 4.2.2 ปรับปรุงระยะเวลาการทดสอบ.....       | 50   |
| 4.2.3 กรณีทดสอบ.....                      | 52   |
| 4.2.4 เพิ่มข้อมูลรายงานการทดสอบ.....      | 53   |
| 4.2.4 ข้อมูลสรุปผลการทดสอบ.....           | 55   |
| 4.2.6 ข้อมูลพนักงาน.....                  | 56   |
| 4.2.7 ข้อมูลผู้ใช้งาน.....                | 56   |
| 4.2.8 รายละเอียดทีม.....                  | 57   |
| 4.3 รายงาน.....                           | 58   |
| 4.3.1 ใบกำหนดระยะเวลาการทดสอบ.....        | 58   |
| 4.3.2 ใบสรุปผลการทดสอบ.....               | 59   |
| 4.4 การอิมพลีเมนต์ระบบ.....               | 60   |
| บทที่ 5 บทสรุปและข้อเสนอแนะ.....          | 62   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

|                      | หน้า |
|----------------------|------|
| 5.1 สรุปโครงการ..... | 62   |
| 5.2 ข้อเสนอแนะ.....  | 62   |
| บรรณานุกรม.....      | 63   |
| ประวัติผู้เขียน..... | 64   |



# สารบัญตาราง

| ตารางที่   | หน้า |
|--|------|
| 2.1 ตัวอย่างการแบ่งส่วนสมมูล.....                          | 16   |
| 3.1 รายละเอียดชุดทดสอบ Create software Detail.....         | 22   |
| 3.2 รายละเอียดชุดทดสอบ Generate Test plan.....             | 25   |
| 3.3 รายละเอียดชุดทดสอบ Generate Test Case.....             | 25   |
| 3.4 รายละเอียดชุดทดสอบ Record Test Result .....            | 26   |
| 3.5 รายละเอียดชุดทดสอบ Generate Summarize Test Report..... | 27   |
| 3.6 รายละเอียดชุดทดสอบ Generate Test Result Report .....   | 28   |
| 3.7 เอนทิตีระบบสนับสนุนการควบคุมคุณภาพซอฟต์แวร์.....       | 40   |
| 3.8 Application.....                                       | 41   |
| 3.9 DevMember.....   | 41   |
| 3.10 Emp_pos.....  | 41   |
| 3.11 Employee.....   | 41   |
| 3.12 ItsqMember.....                                       | 42   |
| 3.13 List_Program.....                                     | 42   |
| 3.14 List_Report.....                                      | 42   |
| 3.15 LogOn.....  | 43   |
| 3.16 Report.....   | 43   |
| 3.17 TestRequest.....                                      | 44   |
| 3.18 summary_Report.....                                   | 45   |
| 3.19 Team.....   | 46   |
| 3.20 Test_case.....  | 47   |
| 3.21 Test_Group.....                                       | 47   |
| 3.22 TypeDescription.....                                  | 47   |

# สารบัญรูป

| ภาพที่  | หน้า |
|---|------|
| 2.1 แสดงส่วนประกอบของ .Net Framework.....                                 | 6    |
| 2.2 การออกแบบกรณีทดสอบ.....   | 8    |
| 2.3 สัญลักษณ์กราฟกระแสนงาน.....   | 10   |
| 2.4 ตัวอย่างของโปรแกรมคำนวณค่าเฉลี่ย.....                                 | 10   |
| 2.4 Flow chart ของโปรแกรมคำนวณค่าเฉลี่ย.....                              | 11   |
| 2.6 ภาพกระแสนงาน.....   | 11   |
| 2.7 เมตริกซ์กราฟ.....   | 12   |
| 2.8 กลุ่มของรูปแต่ละประเภท.....   | 14   |
| 2.9 ลักษณะการทำงานของ การทดสอบแบบกึ่งอัตโนมัติ.....                       | 14   |
| 3.1 แสดงรูปแบบการทำงานของระบบภายใต้แนวคิดไคลเอนท์ เซิร์ฟเวอร์.....        | 21   |
| 3.2 แสดง Use Case Diagram ของระบบ Software Quality Assurance Support..... | 22   |
| 3.3 แอ็กทิวิตี้ไดอะแกรมแสดงการทำงานของระบบ.....                           | 30   |
| 3.4 แสดง Class Diagram .....  | 32   |
| 3.5 แสดง Sequence Diagram: Create software Detail .....                   | 35   |
| 3.6 แสดง Sequence Diagram: Generate Test Plan .....                       | 36   |
| 3.7 แสดง Sequence Diagram: Create Test Case .....                         | 36   |
| 3.8 แสดง Sequence Diagram: Record Test Result.....                        | 37   |
| 3.9 แสดง Sequence Diagram : Generate Summarize Test Report.....           | 38   |
| 3.10 อีอาร์ไดอะแกรมสนับสนุนการควบคุมคุณภาพซอฟต์แวร์.....                  | 39   |
| 4.1 หน้าจอรายละเอียดการทดสอบ.....   | 50   |
| 4.2 หน้าจอแผนการทดสอบ.....  | 51   |
| 4.3 หน้าจอกรณีทดสอบ.....  | 52   |
| 4.4 เพิ่มข้อมูลรายงานการทดสอบ.....  | 53   |
| 4.5 รายงานการตรวจระบบงานเทคโนโลยีสารสนเทศ.....                            | 54   |
| 4.6 ใบสรุปการตรวจระบบงานคุณภาพระบบงานเทคโนโลยีสารสนเทศ.....               | 55   |
| 4.7 ข้อมูลพนักงาน (ผู้เกี่ยวข้อง).....                                    | 56   |
| 4.8 ข้อมูลผู้ใช้งาน.....  | 56   |
| 4.9 รายละเอียดทีม.....  | 57   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและวัตถุประสงค์ในการพัฒนาระบบ

ปัจจุบันการพัฒนาซอฟต์แวร์ทางคอมพิวเตอร์มีความจำเป็นต่อทุกองค์กร เนื่องจากต้องการความรวดเร็วในการทำงาน และความถูกต้องของข้อมูล ซอฟต์แวร์ที่ติดตั้งทำงานได้ตรงตามวัตถุประสงค์ สามารถให้ข้อมูลที่ถูกต้อง และเป็นประโยชน์ต่อผู้ใช้งาน ซอฟต์แวร์ที่น่าเชื่อถือต้องไม่มีความผิดพลาดเกิดขึ้น ดังนั้นในการพัฒนาซอฟต์แวร์จำเป็นต้องมีการทดสอบเพื่อหาข้อผิดพลาดและทดสอบเพื่อให้ได้ข้อมูลที่ถูกต้องตามความต้องการ หน่วยงานควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศ จึงมีบทบาทสำคัญในกระบวนการประกันคุณภาพ เพื่อให้ทีมพัฒนาและผู้ใช้งานมั่นใจได้ว่าซอฟต์แวร์ที่จะนำไปใช้สามารถทำงานได้ถูกต้อง

หน่วยงานควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศเป็นหน่วยงานหนึ่งของกลุ่มงานเทคโนโลยีสารสนเทศ ทำหน้าที่ในการดูแล ติดตาม และทดสอบความถูกต้องของซอฟต์แวร์ที่พัฒนา หรือเปลี่ยนแปลงแก้ไข โดยสายงานเทคโนโลยีสารสนเทศ ก่อนนำออกใช้งานจริง ทบทวนความถูกต้อง ครบถ้วน ของข้อมูลทดสอบ (Test Case) และผลลัพธ์ที่ได้จากการทดสอบของหน่วยงานผู้ใช้ (User Acceptance Test) รับประกันความถูกต้องของซอฟต์แวร์เพื่อทำให้ผู้ใช้งานซอฟต์แวร์สามารถมั่นใจได้ว่า ซอฟต์แวร์ที่จะนำมาใช้งานจริงสามารถทำงานได้ถูกต้องตรงตามความต้องการ ก่อนที่หน่วยงานควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศจะรับประกันความถูกต้องของซอฟต์แวร์ ทางหน่วยงานจำเป็นที่จะต้องทำการทดสอบการทำงานของซอฟต์แวร์ โดยการทดสอบเป็นการทดสอบเสมือนหนึ่งผู้ทดสอบเป็นผู้ใช้งานซอฟต์แวร์ นอกเหนือจากหน้าที่ในการทดสอบแล้ว ผู้ทดสอบยังมีหน้าที่อื่นๆ ประกอบการทดสอบ เช่น การกำหนดระยะเวลาในการทดสอบ การออกแบบกรณีทดสอบ การบันทึกผลจากการทดสอบ การสรุปผลจากการทดสอบ และการจัดทำรายงานการทดสอบ นอกเหนือจากการทดสอบจะสังเกตเห็นได้ว่างานต่างๆ ที่เกี่ยวข้องกับการทดสอบซอฟต์แวร์จะเกี่ยวข้องกับงานทางด้านเอกสาร ทางผู้ทดสอบจึงเสียเวลาให้กับการจัดการด้านเอกสารเสียเป็นส่วนใหญ่ซึ่งไม่ใช่หน้าที่หลักของผู้ทดสอบ ดังนั้นผู้จัดทำจึงเล็งเห็นว่าควรนำเอาเทคโนโลยีปัจจุบันมาช่วยอำนวยความสะดวกให้กับผู้ทดสอบ

## 1.2 วัตถุประสงค์ในการพัฒนาระบบ

การพัฒนาระบบสนับสนุนการทดสอบซอฟต์แวร์ ช่วยให้การวางแผนการทดสอบซอฟต์แวร์ให้เป็นไปอย่างมีประสิทธิภาพและแม่นยำ โดยการสร้างเป็น Gantt Chart และลดระยะเวลาในการจัดทำเอกสารต่างๆ ที่เกี่ยวข้องกับกระบวนการทดสอบซอฟต์แวร์ ซึ่งได้แก่ การกำหนดระยะเวลาในการทดสอบ การบันทึกผลการทดสอบ การสรุปผลการทดสอบ และ รายงานการทดสอบ เพื่อแจ้งให้ผู้ที่เกี่ยวข้องทราบว่าซอฟต์แวร์ได้ผ่านการทดสอบและพร้อมที่จะนำไปใช้งานตลอดจนการจัดเก็บกรณีทดสอบและผลการทดสอบให้เป็นมาตรฐานเดียวกันเพื่อเพิ่มความสะดวกรวดเร็วในการค้นหากรณีทดสอบเพื่อรองรับการทำงานแทนกันในกรณีผู้ทดสอบไม่สามารถทำการทดสอบซอฟต์แวร์ที่ตนรับผิดชอบได้ จากการที่มีการจัดเก็บเอกสารที่เป็นมาตรฐานและสามารถติดตามและสืบค้นได้อย่างสะดวกทำให้ผู้ทดสอบคนอื่นๆ สามารถเรียนรู้การทำงานของซอฟต์แวร์ได้อย่างรวดเร็ว

## 1.3 ขอบเขตของระบบ

กระบวนการทดสอบซอฟต์แวร์ผู้ทดสอบมีขอบเขตของกิจกรรมต่างๆที่เกิดขึ้นระหว่างผู้ทดสอบ (Tester) ผู้พัฒนาซอฟต์แวร์ (Developer) และผู้ใช้งาน (User) ดังนี้

1. การจัดทำตารางการทดสอบซอฟต์แวร์ของผู้ทดสอบเพื่อความสะดวกในการวางแผนการทดสอบ และการรับมอบหมายงานอื่นๆ จากหัวหน้างาน และหัวหน้างานจะได้ทราบถึงผลการทำงานของสมาชิกในทีม โดยสามารถนำผลงานดังกล่าวไปประกอบการประเมินผลการปฏิบัติงานของสมาชิกในทีม
2. การจัดเก็บข้อมูลกรณีทดสอบ ข้อมูลผู้ที่เกี่ยวข้องในกระบวนการทดสอบ รายละเอียดเกี่ยวกับการทดสอบ เอกสารที่ใช้ประกอบการทดสอบ ไว้ในฐานข้อมูลเดียวกันเพื่อลดปัญหาความซ้ำซ้อนและความขัดแย้งของข้อมูล
3. การสร้างรายงานที่เกี่ยวข้องกับกระบวนการทดสอบ โดยการดึงข้อมูลจากฐานข้อมูลเดียวกันทำให้ผู้ใช้ไม่ต้องเสียเวลาในการจัดทำเอกสาร
4. การติดตามสถานะของการทดสอบเพื่อให้ผู้บริหารสามารถวางแผนการมอบหมายงานการทดสอบให้กับผู้ทดสอบแต่ละคน
5. การประเมินประสิทธิภาพการทำงานของผู้ทดสอบแต่ละคน แต่ละทีม จากปริมาณงานที่ได้รับมอบหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 ขั้นตอนในการดำเนินงาน

ศึกษากระบวนการกระบวนการทดสอบซอฟต์แวร์ในธนาคารพาณิชย์แห่งหนึ่ง เพื่อใช้เป็นแนวทางในการพัฒนาออกแบบระบบให้มีประสิทธิภาพและตรงกับความต้องการในการใช้งานมากยิ่งขึ้น

1. ศึกษากระบวนการทดสอบซอฟต์แวร์จากหน่วยงานควบคุมคุณภาพซอฟต์แวร์เพื่อให้ทราบถึงขั้นตอนในการทดสอบซอฟต์แวร์
2. ศึกษาทฤษฎีและเทคนิคในการทดสอบซอฟต์แวร์เพื่อให้เข้าใจหลักการในการออกแบบกรณีทดสอบ
3. ศึกษาเทคโนโลยี .NET เพื่อนำมาใช้ในการพัฒนาระบบ
4. ศึกษาเทคโนโลยี SQL 2005 เพื่อนำมาใช้ในการสร้างฐานข้อมูล
5. นำข้อมูลที่ได้ทำการรวบรวมศึกษามาวิเคราะห์และออกแบบระบบ
6. ศึกษาเทคโนโลยีที่ใช้ในการพัฒนาระบบงาน เพื่อเลือกใช้เทคโนโลยีให้เหมาะสมกับการพัฒนามากที่สุด
7. พัฒนาโปรแกรมให้เป็นไปตามที่ได้ออกแบบไว้

ทดสอบระบบและดำเนินการแก้ไข ปรับปรุงข้อผิดพลาดจากการทดสอบระบบ เพื่อให้ระบบสามารถทำงานได้อย่างถูกต้อง

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยให้การวางแผนการทดสอบซอฟต์แวร์เป็น ไปอย่างมีประสิทธิภาพและแม่นยำ
2. ลดระยะเวลาในการจัดทำเอกสารต่างที่เกี่ยวข้องกับกระบวนการทดสอบซอฟต์แวร์ ซึ่งได้แก่ การกำหนดระยะเวลาในการทดสอบ การบันทึกผลการทดสอบ ใบสรุปผลการทดสอบ และ รายงานการทดสอบ เพื่อแจ้งให้ผู้ที่เกี่ยวข้องทราบว่าซอฟต์แวร์ได้ผ่านการทดสอบและพร้อมที่จะนำไปใช้งานตลอดจนการจัดเก็บกรณีทดสอบและผลการทดสอบให้เป็นมาตรฐานเดียวกัน
3. เพิ่มความสะดวกรวดเร็วในการค้นหากรณีทดสอบ โดยสามารถรองรับการทำงานแทนกันในกรณีผู้ทดสอบไม่สามารถทำการทดสอบซอฟต์แวร์ที่ตนรับผิดชอบได้ จากการที่มีการจัดเก็บเอกสารที่เป็นมาตรฐานและสามารถติดตามและสืบค้นได้อย่างสะดวกทำให้ผู้ทดสอบคนอื่นๆ
4. การทดสอบซอฟต์แวร์มีประสิทธิภาพมากขึ้น ลดข้อบกพร่องในการทดสอบ เนื่องจากผู้ทดสอบมีเวลาในการทดสอบมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. กระบวนการทดสอบซอฟต์แวร์แต่ละApplication สามารถทำได้รวดเร็วขึ้นเนื่องจากมีระบบที่อำนวยความสะดวกในการทดสอบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีพื้นฐานที่เกี่ยวข้อง

### 2.1 เทคโนโลยี .NET

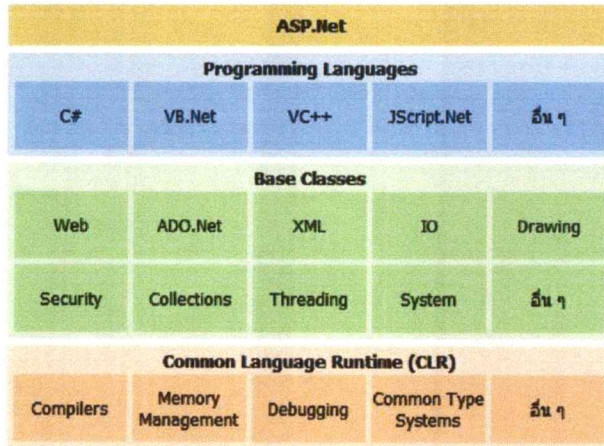
เทคโนโลยี .Net เป็นแนวคิดที่ต้องการนำเอาอุปกรณ์ทุกอย่างบนโลกนี้มาเชื่อมต่อกันเหมือนตาข่าย (net) แต่ปัญหาของการนำเอาทุกอย่างบนโลกนี้มาเชื่อมต่อกันคือ สิ่งเหล่านั้นได้รับการออกแบบมาแตกต่างกัน การที่จะทำให้สิ่งที่แตกต่างกันติดต่อสื่อสารกันได้เป็นเรื่องยาก จึงได้มีการคิดค้นและพัฒนาเทคโนโลยีที่ทำให้อุปกรณ์ทุกๆชนิดทั่วโลกสามารถติดต่อสื่อสารกันได้

### 2.2 .Net Framework คืออะไร

ดอตเน็ตเฟรมเวิร์ก (.NET Framework) คือแพลตฟอร์มสำหรับพัฒนาซอฟต์แวร์ ถูกพัฒนาขึ้นโดยไมโครซอฟท์ สามารถรองรับภาษาดอตเน็ตได้หลายภาษา ซึ่งมีไลบรารีเป็นจำนวนมากสำหรับการเขียนโปรแกรม รวมถึงบริหารตัวดำเนินการของโปรแกรมบนดอตเน็ตเฟรมเวิร์ก โดยไลบรารีรวมถึงส่วนต่อประสานกับผู้ใช้ การเชื่อมต่อฐานข้อมูล วิทยาการเข้ารหัสลับ อัลกอริทึม การเชื่อมต่อเครือข่ายคอมพิวเตอร์ และการพัฒนาเว็บแอปพลิเคชัน โปรแกรมที่เขียนบนดอตเน็ตเฟรมเวิร์ก จะทำงานบนสภาพแวดล้อมที่บริหาร โดย Common Language Runtime (CLR) ซึ่งเป็นส่วนหนึ่งในดอตเน็ตเฟรมเวิร์ก โดย CLR นั้นเตรียมสภาพแวดล้อมเสมือน ทำให้ผู้พัฒนาไม่ต้องคำนึงถึงความสามารถที่แตกต่างระหว่างหน่วยประมวลผลต่างๆ และ CLR ยังให้บริการด้านกลไกระบบความปลอดภัย การบริหารหน่วยความจำ และ Exception handling

### 2.3 ส่วนประกอบของ .Net Framework

1. Common Language Runtime (CLR)
2. ASP.Net
3. Programming Languages
4. Base Classes



ภาพที่ 2.1 แสดงส่วนประกอบของ .Net Framework

**1. Common Language Runtime (CLR)** คือเวอร์ชวลแมชีน ซึ่งเป็นส่วนประกอบของคอตเน็ตเฟรมเวิร์ก และพัฒนาตามมาตรฐานเปิด Common Language Infrastructure ที่ไม่ใคร่ขอพท์ได้พัฒนาขึ้นมา ซึ่งอธิบายถึงสภาพแวดล้อมสำหรับโค้ดที่ทำงานบน CLR โดย CLR จะรันจากไบต์โค้ดที่เรียกว่า Microsoft Intermediate Language (MSIL) ซึ่งพัฒนาตามมาตรฐาน Common Intermediate Language (CIL) ผู้พัฒนาใช้ CLR ด้วยการเขียนโค้ดด้วยภาษาระดับสูงอย่าง C# หรือ VB.NET โดยช่วงเวลาคอมไพล์ คอตเน็ตคอมไพเลอร์จะทำการแปลงโค้ดดังกล่าวไปเป็นโค้ด MSIL (Microsoft Intermediate Language) และเวลาที่รันโค้ด CLR's just-in-time compiler จะทำการแปลงโค้ด MSIL ไปเป็นภาษาเครื่องสำหรับระบบปฏิบัติการเพื่อให้ทำงานได้ หรือหากต้องการโค้ด MSIL สามารถคอมไพล์ไปยังเป็นภาษาเครื่องก่อนที่จจะรัน ก็ทำได้เช่นกัน ซึ่งในกรณีนี้จะทำให้การรันโค้ดตอนแรกเร็วขึ้น เพราะไม่เสียเวลาแปลงโค้ด MSIL ไปยังภาษาเครื่อง

**2. Base Classes** เป็น class library พื้นฐาน ที่โปรแกรมต่าง ๆ ไม่ว่าจะเขียนด้วยภาษาใดบน .Net ก็สามารถใช้ร่วมกันได้ เช่น การติดต่อระบบฐานข้อมูล (ADO.Net), การติดต่อกับ file system ของ server (IO), ฯลฯ

**3. Programming Languages** เป็นกลุ่มของภาษาคอมพิวเตอร์ ที่ออกแบบมาเพื่อการเขียนโปรแกรมบน .Net Framework ซึ่งไม่ว่าจะเขียนโปรแกรมด้วยภาษาใดก็ตาม Compiler ใน CLR ก็ compile โปรแกรมนั้นให้อยู่ในรูปของ Intermediate Language (IL) ซึ่งจะถูกนำไปแปลเป็นภาษาเครื่อง (Native Code) เมื่อตอนที่นำไปใช้

**4. ASP.Net** เป็นภาษา script ที่พัฒนามาจาก ASP เพื่อให้โปรแกรมเมอร์สามารถพัฒนา web application ให้ใช้ .Net ได้สะดวกขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอเอสพีคอตเน็ต (ASP.NET)

เอเอสพีคอตเน็ต (ASP.NET) คือเทคโนโลยีสำหรับพัฒนาเว็บไซต์ เว็บแอปพลิเคชัน และเว็บเซอร์วิส ซึ่งเป็นส่วนหนึ่งของคอตเน็ตเฟรมเวิร์ก พัฒนาโดยไมโครซอฟท์

### ข้อเปรียบเทียบระหว่าง ASP.NET และ ASP

1. โค้ดจะได้รับการ compiled ทำให้การทำงานรวดเร็วขึ้น และช่วยจับข้อผิดพลาดในช่วงการออกแบบได้
2. ระบบการจัดการข้อผิดพลาด (Exception handling) ที่ดีขึ้นกว่าเดิม
3. ใช้วิธีการพัฒนาเว็บแอปพลิเคชันเช่นการใช้ controls หรือ events ซึ่งทำให้พัฒนาง่ายขึ้น
4. มี controls หลากหลายและมีไลบรารีที่เตรียมไว้สำหรับการใช้งานทำให้การพัฒนาสะดวก และรวดเร็วขึ้น
5. สามารถพัฒนาได้หลากหลายภาษา
6. สามารถทำการแคชได้ทั้งหน้า หรือส่วนหนึ่งของหน้าที่ต้องการ
7. สามารถแยกส่วนโค้ดออกมาต่างหากจากหน้ารูปแบบ
8. Session สามารถเลือกที่จะบันทึกในฐานข้อมูลได้ ทำให้ session ไม่สูญหายหากมีการรีเซตเว็บเซิร์ฟเวอร์
9. รองรับมาตรฐานเว็บดีขึ้นกว่าเดิม รวมถึงการทำงานร่วมกับ CSS

## 2.4 Microsoft SQL Server 2005 (ไมโครซอฟท์ซีควอลเซิร์ฟเวอร์ 2005)

SQL Server 2005 เป็นแพลตฟอร์มดาต้าเบสครบวงจร ประกอบด้วยระบบบริหารข้อมูลระดับเอนเตอร์ไพรซ์ และเครื่องมือระบบธุรกิจอัจฉริยะ (business intelligence -BI) มีกลไกดาต้าเบสของ SQL Server 2005 ที่ช่วยให้จัดเก็บข้อมูลรีเลชันแนลและข้อมูลที่มีโครงสร้างได้ปลอดภัยมากขึ้นและมีเสถียรภาพมากขึ้น รวมทั้งช่วยให้สามารถสร้างและบริหารแอปพลิเคชันข้อมูลที่มีประสิทธิภาพสูง

กลไกข้อมูลของ SQL Server 2005 มีความสำคัญต่อโซลูชันบริหารข้อมูลระดับเอนเตอร์ไพรซ์ เนื่องจาก SQL Server 2005 ได้ผสมผสานระบบวิเคราะห์ ระบบทำรายงาน ระบบผสานข้อมูล และระบบแจ้งเตือนเข้าไว้ด้วยกัน

## 2.5 ทฤษฎีการทดสอบซอฟต์แวร์

### 2.5.1 การออกแบบกรณีทดสอบ (Test case design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- วันที่ทำการทดสอบ (Date)
- ผลการทดสอบจากกรณีทดสอบว่าผ่านหรือไม่ผ่านกรณีทดสอบ (Result)
- เวอร์ชัน (Version)
- ผู้ทำการทดสอบ (Run by)

### 2.5.2 การทดสอบแบบกล่องขาว (White-box testing)

การทดสอบแบบกล่องขาว มีชื่อเรียกอีกอย่างว่าการทดสอบแบบกล่องแก้ว (Glass-Box testing) หมายถึง การทดสอบขั้นตอนรายละเอียดต่างๆ ทางตรรกะวิทยาโดยละเอียด ตามโครงสร้างการทำงานของโปรแกรม ได้แก่ การทำงานตามโครงสร้างแบบลำดับ การทำงานตามโครงสร้างแบบเลือกทำ การทำงานตามโครงสร้างวนซ้ำหรือวนรอบ เป็นการทดสอบคำสั่งต่างๆ ในโปรแกรม การออกแบบกรณีทดสอบ (Test case design) จะต้องได้ผลลัพธ์ดังนี้

- 1) รับประกันว่าได้ทดสอบทุกเส้นทางอิสระ (Independent path) ในโมดูล เพื่อทดสอบว่าทุกเส้นทางในกระบวนการจะต้องสามารถทำงานได้ถูกต้อง
- 2) ได้ทดสอบทุกการตัดสินใจทางตรรกะ (Logical decision) ทั้งกรณีจริงและกรณีเท็จ
- 3) ทดสอบการทำงานภายในรูปทุกรูป
- 4) ได้ทดสอบรูปและทุกโครงสร้างข้อมูลภายใน (Internal data) ให้ถูกต้องก่อนที่จะส่งไปประมวลผลต่อ

#### 2.5.2.1 การทดสอบเส้นทางมูลฐาน (Basis path testing)

การทดสอบเส้นทางมูลฐาน (Basis path testing) เป็นเทคนิคการทดสอบแบบกล่องขาวที่เน้นการตรวจสอบการพัฒนาระบบว่าได้ทำอย่างถูกต้องหรือไม่ (Verify) โดยมีวัตถุประสงค์เพื่อทดสอบความซับซ้อนซ้อนทางตรรกะวิทยาโดยจะทดสอบทุกๆ เส้นทาง

ค่าของเส้นทางความซับซ้อนซ้อนของปัญหา  $V(G)$  จะทำให้ได้เส้นทางเพื่อนำมากำหนดในการทดสอบ สัญลักษณ์กระแสวน (Flow Graph Notation)

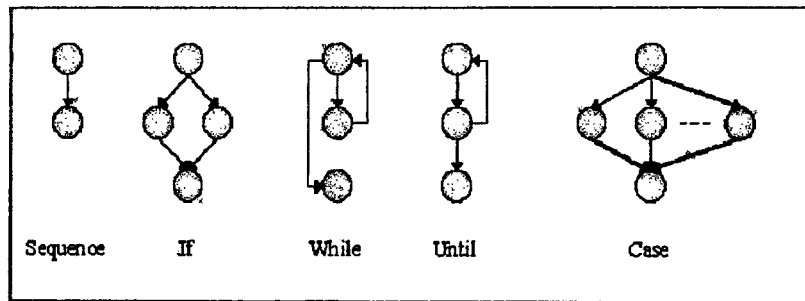
#### กราฟกระแสวน (Flow Graph) หรือกราฟโปรแกรม (Program Graph)

เป็นการสร้างกรณีทดสอบจากโค้ดของโปรแกรม เพื่อกำหนดเส้นทางการทดสอบ และกำหนดชุดข้อมูลเพื่อทำการทดสอบแต่ละเส้นทาง เพื่อให้มั่นใจได้ว่าทุกเส้นทางได้ผ่านการทดสอบแล้วอย่างน้อย 1 ครั้ง

การทดสอบแบบกราฟกระแสวนแสดงให้เห็นการควบคุมทางตรรกะ โดยประกอบด้วย โหนดกระแสวน (Flow Graph node) แทนประโยคคำสั่งและการตัดสินใจ เส้นเชื่อม (Edges) หรือทางเชื่อม (Lines) แทนกระแสการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 สัญลักษณ์กราฟกระแสน

### การสร้างกรณีทดสอบ (Deriving Test Cases)

วิธีการทดสอบเส้นทางมูลฐานสามารถประยุกต์ใช้กับชิ้นงานออกแบบ (Procedure Design) หรือกับรหัสต้นฉบับ (Source Code) ซึ่งมีขั้นตอนดังต่อไปนี้

1. ใช้การแปลงประโยคคำสั่งในโค้ดไปเป็นโหนดในกราฟกระแสน ตัวอย่างเป็นส่วนหนึ่งของโปรแกรมคำนวณค่าเฉลี่ยตัวเลข 1-100

```

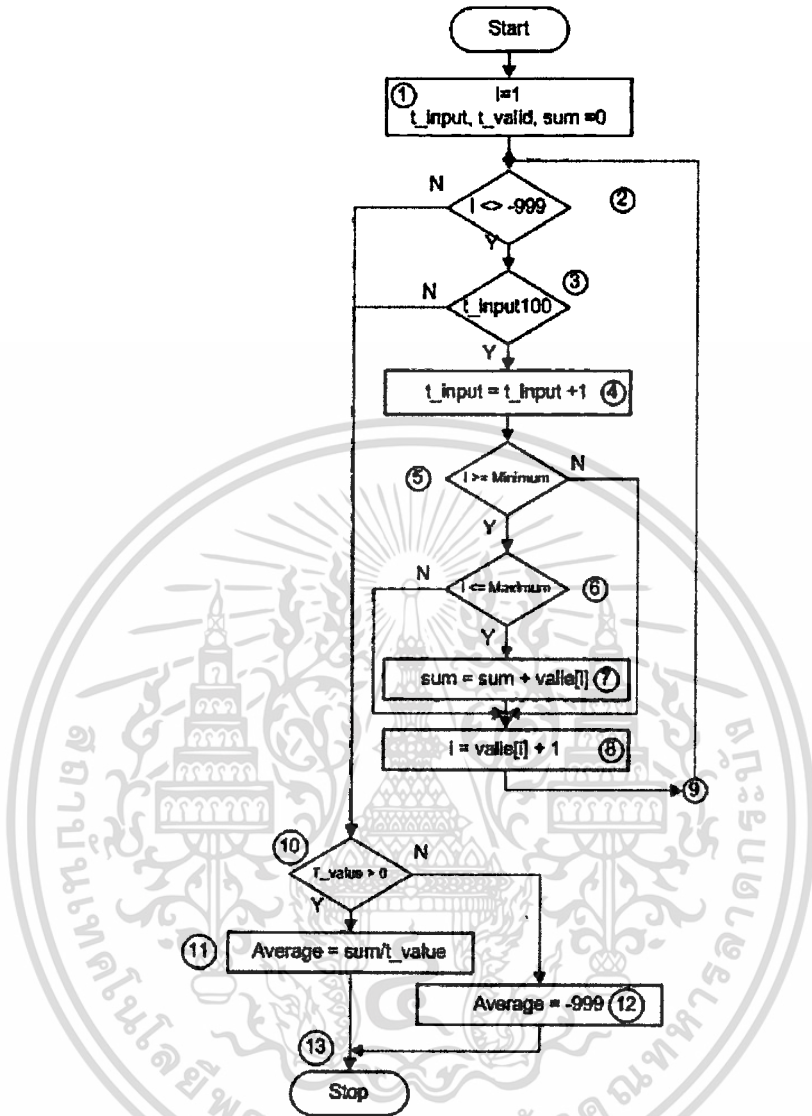
i = 1
t_input = t_value = 0;
sum = 0;
DO WHILE value [i] < -999 and t_input < 100
    increment t_input by 1;
    If value[i] >= minimum and value [i] <= maximum
        THEN increment t_value by 1;
        Sum = sum + value[i];
    ELSE skip
ENDIF
increment i by 1;
ENDDO
IF t_value > 0
    THEN average = sum/t_value;
    ELSE averate = -999;
ENDIF

```

ภาพที่ 2.4 ตัวอย่าง โปรแกรมคำนวณค่าเฉลี่ย

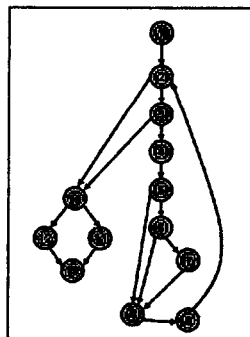
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่าง โปรแกรมสามารถนำมาเขียนเป็นผังงานได้ดังนี้



ภาพที่ 2.5 Flow chart ของ โปรแกรมคำนวณค่าเฉลี่ย

2. คำนวณหาค่าความซับซ้อนโดยการนับจำนวนคำสั่งทดสอบเงื่อนไข โดยการพิจารณาเส้นทางที่แยกเป็นอิสระ (Independent path)



ภาพที่ 2.6 ภาพกระแสน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V(G) = 6 \text{ เขต}$$

$$V(G) = 17 \text{ เส้น} - 9 \text{ โหนด} + 2 = 6$$

$$V(G) = 5 \text{ โหนดทำนาย} + 1 = 6$$

3. หาจำนวนเส้นทางที่แยกเป็นอิสระ ได้จากค่า  $V(G)$  6 เส้นทาง (path) ดังนั้น แต่ละเส้นทางมีรายละเอียดดังนี้

Path1: 1-2-10-11-13

Path2: 1-2-10-12-13

Path3: 1-2-3-10-11-13

Path4: 1-2-3-4-5-8-9-2-...

Path5: 1-2-3-4-5-6-8-9-2-...

Path6: 1-2-3-4-5-6-7-8-9-2-...

ทุกเส้นทางจะต้องทดสอบด้วยค่าที่เป็นจริงและค่าที่เป็นเท็จ แล้วนำผลลัพธ์ที่ได้จากการทดสอบมาเปรียบเทียบกับข้อมูลชุดทดสอบ ซึ่งจะต้องประกอบไปด้วยข้อมูลนำเข้าและผลลัพธ์ที่คาดการณ์ไว้

### เมตริกซ์กราฟ (Graph Matrix)

เมตริกซ์กราฟ เป็นตารางที่มีขนาดเท่ากับจำนวนโหนดของกราฟสายงาน โดยให้แถวและคอลัมน์แทนหมายเลขโหนด เส้นเชื่อมระหว่างโหนดกำหนดเป็นตัวอักษร

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| 1  | A |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 2  |   | R |   |   |   |   |   |   |   |    | I  |    |    |    |    |    |    |
| 3  |   |   | C |   |   |   |   |   |   | J  |    |    |    |    |    |    |    |
| 4  |   |   |   | D |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 5  |   |   |   |   | E | O |   |   |   |    |    |    |    |    |    |    |    |
| 6  |   |   |   |   |   | R | Q | F |   |    |    |    |    |    |    |    |    |
| 7  |   |   |   |   |   |   |   | P |   |    |    |    |    |    |    |    |    |
| 8  |   |   |   |   |   |   |   |   | G |    |    |    |    |    |    |    |    |
| 9  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 10 |   |   |   |   |   |   |   |   |   | N  | K  |    |    |    |    |    |    |
| 11 |   |   |   |   |   |   |   |   |   |    |    |    |    | M  |    |    |    |
| 12 |   |   |   |   |   |   |   |   |   |    |    |    |    |    | L  |    |    |
| 13 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 14 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 15 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 16 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| 17 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |

ภาพที่ 2.7 เมตริกซ์กราฟ

ข้อดีของการทดสอบแบบกล่องขาว สามารถตรวจหาข้อผิดพลาดของโปรแกรมได้อย่างละเอียด เนื่องจากทำการตรวจสอบจากภายในซึ่งก็คือโค้ดของโปรแกรม จึงมั่นใจได้ว่าโปรแกรมเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีความผิดพลาดน้อย แต่ว่าการทดสอบต้องอาศัยเวลาก่อนข้างมากและต้องใช้ต้นทุนสูงพอสมควร

### 2.5.3 การทดสอบโครงสร้างควบคุม (Control Structure Testing)

#### 2.5.3.1 การทดสอบเงื่อนไข (Condition Testing)

การทดสอบเงื่อนไขเป็นการออกแบบกรณีทดสอบเงื่อนไขทางตรรกะได้แก่ ตัวแปรบูลีน หรือนิพจน์เชิงสัมพันธ์

E1 < ตัวกระทำเชิงสัมพันธ์ > E2

เมื่อ E1 และ E2 เป็นนิพจน์คำนวณ ประกอบด้วย ตัวกระทำบูลีน (Boolean Operation) ตัวแปรบูลีน (Boolean Expression) วงเล็บ ตัวกระทำเชิงสัมพันธ์หรือนิพจน์คำนวณ

<, ≤, =, ≠, >, ≥, OR (|), AND (&), NOT (¬)

การทดสอบเป็นการเน้นทดสอบเงื่อนไขในซอฟต์แวร์ เพื่อเป็นการประกันว่าแต่ละเงื่อนไขในซอฟต์แวร์ไม่มีความผิดพลาด

#### 2.5.3.2 การทดสอบกระแสข้อมูล (Data flow testing)

การทดสอบกระแสข้อมูล เป็นวิธีการทดสอบโปรแกรมตามจุดนิยามและจุดที่ใช้ตัวแปรในซอฟต์แวร์โดยการกำหนดให้ในแต่ละคำสั่งในซอฟต์แวร์มีหมายเลขเฉพาะและไม่มีการเปลี่ยนค่าของพารามิเตอร์และตัวแปรส่วนกลาง

DEF(S) = {X | ประโยคคำสั่ง S มีนิยามของ X}

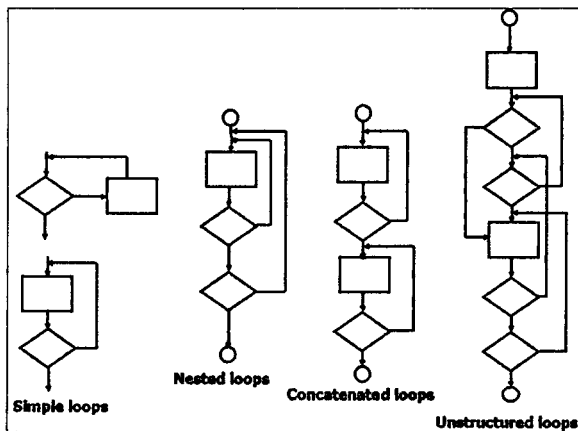
DEF(S) = {X | ประโยคคำสั่ง S มีการใช้งาน X}

#### 2.5.3.2 การทดสอบการวนซ้ำ (Loop Testing)

เป็นเทคนิคในการทดสอบแบบกล่องขาว (White-Box) โดยมุ่งประเด็นไปที่ความถูกต้องของโครงสร้างของการวนซ้ำหรือลูป (loop constructs)

กลุ่มของลูปที่แตกต่างกันได้แก่

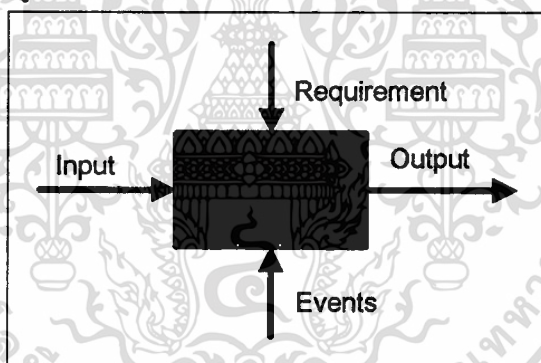
1. Simple Loop เป็นรูปแบบง่ายไม่มีความซับซ้อน
2. Nested loop เป็นลูปที่มีโครงสร้างลักษณะการนำ Simple Loop มาซ้อนกันภายในมีความซับซ้อนมากขึ้น
3. Concatenated loop เป็นลูปที่มีการนำ Simple Loop มาต่อกันเป็นโครงสร้างใหญ่ขึ้น
4. Unstructured loops เป็นลูปที่ไม่มีโครงสร้างของลูปชัดเจนมีความซับซ้อน การทดสอบมีความยุ่งยากมาก



ภาพที่ 2.8 กลุ่มของรูปแต่ละประเภท

#### 2.5.4 การทดสอบแบบกล่องดำ (Black-box testing)

การทดสอบแบบกล่องดำ เป็นการทดสอบเชิงพฤติกรรม (Behavioral Testing) โดยไม่คำนึงถึงคำสั่งภายใน โปรแกรม แต่จะสนใจเฉพาะข้อมูลเข้า (Input) และผลลัพธ์ (Output) ที่ได้จากการทดสอบเท่านั้น โดยพิจารณาจากข้อกำหนด (Specification) ของโปรแกรมตามความต้องการ (Requirements) ของผู้ใช้



ภาพที่ 2.9 ลักษณะการทำงานของการทดสอบแบบกล่องดำ

เนื่องจากการทดสอบผลที่ได้รับจากการทำงานของซอฟต์แวร์ในแต่ละหน้าที่ตามข้อกำหนดเพื่อเป็นการทดสอบว่าซอฟต์แวร์ได้ทำงานตามข้อตกลงหรือไม่ โดยไม่ได้คำนึงถึงคำสั่งภายใน ดังนั้นการทดสอบแบบกล่องดำไม่ได้ถูกนำมาแทนที่การทดสอบแบบกล่องขาวแต่เป็นการทดสอบโดยใช้เทคนิคที่แตกต่างกัน เทคนิคทั้ง 2 วิธีจะเป็นการทดสอบที่สนับสนุนซึ่งกันและกัน ความผิดพลาดที่พบบ่อยในการทดสอบแบบกล่องดำมีดังนี้

- 1) ความผิดพลาดในหน้าที่การทำงาน
- 2) หน้าที่การทำงานบางอย่างขาดหายไป
- 3) ความผิดพลาดในการเชื่อมต่อและประสานงานกับระบบอื่นๆ
- 4) ความผิดพลาดของการตัดสินใจและความผิดพลาดที่ก่อให้เกิดการหยุดการ

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5) ความผิดพลาดจากการประมวลผลข้อมูล

### ข้อควรพิจารณาในการทดสอบ

- 1) การดูขอบเขตของข้อมูลว่าข้อมูลที่ต้องการทดสอบอยู่ในขอบเขตที่ถูกต้องตรงตามเงื่อนไข
- 2) กรณีที่ข้อมูลนำเข้าไม่ถูกต้องหรือไม่ตรงกับเงื่อนไขมีข้อความแจ้งเตือนหรือไม่
- 3) กรณีที่ข้อมูลนำเข้าไม่ถูกต้องหรือไม่ตรงกับเงื่อนไขซอฟต์แวร์จะต่ออย่างไร ประมวลผลหรือหยุดการทำงานต่ออย่างไร

#### 2.5.4.1 วิธีการทดสอบเชิงกราฟ (Graph-base Testing Methods)

วิธีการทดสอบเชิงกราฟเป็นวิธีการทดสอบซอฟต์แวร์ในเรื่องความสัมพันธ์ระหว่างวัตถุ โดยใช้โหนดแทนวัตถุและใช้ลิงค์แทนความสัมพันธ์ระหว่างวัตถุ

#### 2.5.4.2 การแบ่งส่วนสมมูล (Equivalence partitioning)

การแบ่งส่วนสมมูล เป็นการทดสอบแบบกล่องดำที่มีการแบ่งกลุ่มของข้อมูลนำเข้า สร้างกรณีทดสอบโดยการแบ่งชั้นของข้อมูลออกเป็น 2 ชั้นคือ

- 1) ข้อมูลปกติที่สามารถทำงานได้ถูกต้องและให้ผลลัพธ์ตามที่ต้องการ
- 2) ข้อมูลที่ทำให้เกิดข้อผิดพลาดเป็นชั้นของข้อมูลเมื่อผ่านกระบวนการทดสอบแล้วจะต้องได้ข้อมูลที่ไม่ถูกต้อง

#### ข้อแนะนำในการแบ่งกลุ่มของข้อมูลดังต่อไปนี้

- 1) ถ้าเงื่อนไขระบุข้อมูลเป็นช่วงต้องนิยามค่าที่ตรงตามเงื่อนไขหนึ่งกลุ่มและค่าที่อยู่ภายนอกเงื่อนไขสองกลุ่ม
- 2) ถ้าค่าของข้อมูลนำเข้าเป็นค่าเฉพาะให้นิยามค่าสำหรับการทดสอบที่ตรงตามเงื่อนไขของค่าเฉพาะหนึ่งค่าและค่าที่แตกต่างออกไปอีกหนึ่งค่า
- 3) ถ้าเงื่อนไขของข้อมูลนำเข้าเป็นสมาชิกของเซตให้นิยามค่าตามเงื่อนไขหนึ่งกลุ่มและนอกเหนือเงื่อนไขหนึ่งกลุ่ม
- 4) ถ้าเงื่อนไขของข้อมูลนำเข้าเป็นบูลีนให้นิยามค่าที่เป็นจริงหนึ่งค่าและค่าที่เป็นเท็จหนึ่งค่า

ตัวอย่าง เงื่อนไขในการคิดส่วนลดราคาสินค้าโดยเมื่อลูกค้าซื้อของในร้านตามจำนวนเงินที่ทางร้านกำหนดไว้ทางร้านก็จะให้ส่วนลดกับลูกค้า ดังนี้

จำนวนเงินที่ซื้อ  $\leq 999$  ไม่มีส่วนลด

จำนวนเงินที่ซื้อ 1000-4,999 ให้ส่วนลด 10%

จำนวนเงินที่ซื้อ  $\geq 5,000$  ให้ส่วนลด 15%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## วิธีการ

- 1) หาค่าพิสัยต่ำสุดของข้อมูล
- 2) หาค่าสูงสุดของพิสัยข้อมูล
- 3) หาค่าข้อมูลตัวแทนกลุ่มซึ่งควรจะเป็นค่าที่ใกล้เคียงกับค่ากลางของพิสัยข้อมูล
- 4) ค่าข้อมูลเกินพิสัย หรือเกินขอบเขตของเงื่อนไขในแต่ละข้อ

ตารางที่ 2.1 ตัวอย่างการแบ่งส่วนสมมูล

| เงื่อนไข      | ค่าของข้อมูลที่ต้องทำการทดสอบ |
|---------------|-------------------------------|
| 1-999         | 1, 499, 999                   |
| 1,000 – 4,999 | 1000, 2020, 4,999             |
| 5000 Up       | 5000, 5099, 9999              |
| ค่าเกินพิสัย  | ไม่มี                         |

ค่าเกินพิสัยไม่มีเนื่องจากไม่อยู่ในเงื่อนไข แต่ถ้าหากอยากจะทำลองทดสอบก็ให้ลองทดสอบค่าตัวเลขมากๆ เพื่อดูข้อจำกัดของซอฟต์แวร์ว่าสามารถรองรับค่าได้มากที่สุดเท่าไรซึ่งในความเป็นจริงแล้วจะต้องมีเงื่อนไขระบุค่าดังกล่าว

### 2.5.4.3 การวิเคราะห์ค่าขอบเขต (Boundary Value Analysis-BVA)

การวิเคราะห์ค่าขอบเขตเป็นการทดสอบที่หาค่าความผิดพลาดจากส่วนของข้อมูลนำเข้าอีกประเภทหนึ่งที่คล้ายกับวิธีการแบ่งส่วนสมมูล แต่แตกต่างกันที่การวิเคราะห์ค่าขอบเขตมีการทดสอบค่าที่ใกล้เคียงกับเงื่อนไข และส่วนที่น้อยกว่าหรือมากกว่าที่เงื่อนไขกำหนด เช่นการกำหนดเงื่อนไขข้อมูลนำเข้าที่กำหนดว่าต้องเท่ากับ 3 และ 5 การนิยามกรณีทดสอบต้องทำการทดสอบข้อมูลนำเข้าที่มีค่า 3, 5, มากกว่า 3, มากกว่า 5, น้อยกว่า 3 และน้อยกว่า 5

ข้อดีของการทดสอบแบบกล่องดำ ผู้ทดสอบไม่จำเป็นต้องมีความรู้ทางด้านโปรแกรมที่นำมาทดสอบเนื่องจากการทดสอบเน้นที่ความต้องการของผู้ใช้งานเป็นหลัก หลังจากที่มีการกำหนดคุณสมบัติของซอฟต์แวร์ก็สามารถออกแบบกรณีทดสอบได้เลย ในการทดสอบโปรแกรมขนาดใหญ่ซึ่งมีโค้ดหลายๆ บรรทัดก็สามารถทดสอบได้โดยไม่ต้องเสียเวลาในการทดสอบมากเหมือนกับการทดสอบแบบกล่องขาว

การทดสอบแบบกล่องดำมีข้อด้อยในเรื่องของการทดสอบข้อมูลนำเข้าเนื่องจากการยากที่จะทำการทดสอบค่าทุกค่าที่ไหลเข้าสู่ระบบ ทำได้เพียงทดสอบค่าที่คาดว่าจะก่อให้เกิดข้อผิดพลาด หรือเลือกค่าขอบมาทำการทดสอบเท่านั้น

### 2.5.5 การทดสอบซอฟต์แวร์เชิงวัตถุ (Object-Oriented Testing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบซอฟต์แวร์เชิงวัตถุมีเป้าหมายเหมือนกับการทดสอบซอฟต์แวร์ทั่วไป แต่จะมีการทดสอบและนิยามการทดสอบแต่ละระดับแตกต่างกันไปจากการทดสอบซอฟต์แวร์ทั่วไป

### 2.5.5.1 การทดสอบระดับหน่วย (Unit Testing)

การทดสอบระดับหน่วยของซอฟต์แวร์เชิงวัตถุเรียกว่า “การทดสอบคลาส” (Class Testing) ภายในคลาสประกอบด้วย ข้อมูล (Data) และพฤติกรรมหรือส่วนดำเนินการ (Operation) ซึ่งแตกต่างจากซอฟต์แวร์ทั่วไปที่มีการแยกส่วนการทำงาน ถึงแม้ว่าการทดสอบระดับหน่วยสำหรับซอฟต์แวร์เชิงวัตถุหมายถึงถึงคลาสแต่ก็ต้องการทดสอบและพิจารณาในส่วนดำเนินการและพฤติกรรมในแต่ละสถานการณ์ของคลาส เนื่องจากส่วนดำเนินการเป็นส่วนที่ทำให้เกิดกระบวนการทำงานของซอฟต์แวร์เชิงวัตถุ ซึ่งในแต่ละคลาสประกอบด้วยส่วนดำเนินการหรือเมธอด อาจจะมีบาง เมธอดที่เป็นส่วนหนึ่งของคลาสหลายๆ คลาส เช่นคลาสที่สืบทอดคุณสมบัติได้ ทำให้เมธอดดังกล่าวใช้คุณสมบัติของเอทริบิวต์แตกต่างกัน การตอบสนองของเมธอดก็จะแตกต่างกันถึงแม้บางครั้งเมธอดดังกล่าวอาจจะถูกใช้ในคลาสเดียวกันก็ตาม

### 2.5.5.2 การทดสอบระดับรวมหน่วย (Integration Testing)

ซอฟต์แวร์เชิงวัตถุไม่มี โครงสร้างแบบลำดับชั้นที่แน่นอนเหมือนกับซอฟต์แวร์ทั่วไป ผลลัพธ์ที่ได้ย่อมแตกต่างกันออกไปตามสถานการณ์ที่เกิดขึ้น การทดสอบโดยการเพิ่ม โมดูลจากบนลงล่าง (Top-Down Approach) หรือการทดสอบแบบล่างขึ้นบน (Bottom-Up Approach) ไม่สามารถใช้ได้กับการทดสอบซอฟต์แวร์เชิงวัตถุ การทดสอบระดับรวมหน่วยของซอฟต์แวร์เชิงวัตถุได้แก่ Thread-base Testing และ Use based Testing

#### Thread-base Testing

Thread-base Testing เป็นวิธีการทดสอบซอฟต์แวร์เชิงวัตถุแบบรวมหน่วย โดยการจัดกลุ่มคลาสที่ตอบสนองเหตุการณ์เดียวกันไว้ในกลุ่มเดียวกัน เรียกคลาสแต่ละกลุ่มว่า “Thread” ซึ่งแต่ละ Thread จะถูกทดสอบโดยอิสระ

#### Use-base Testing

Use-base Testing เป็นวิธีการทดสอบซอฟต์แวร์เชิงวัตถุแบบรวมหน่วย โดยการทดสอบทีละคลาสและเริ่มทดสอบคลาสที่ไม่มีการเรียกใช้บริการจากคลาสอื่นหรือเรียกใช้บริการน้อยที่สุด โดยคลาสที่ไม่มีการเรียกใช้บริการจากคลาสอื่นเรียกว่า “Independent Class” หรือคลาสอิสระ หลังจากทดสอบคลาสที่ไม่มีการเรียกใช้บริการจากคลาสอื่นหรือเรียกใช้บริการน้อย จึงเริ่มเพิ่มระดับการทดสอบ โดยการทดสอบคลาสที่มีการเรียกใช้บริการจากคลาสอื่นซึ่งเรียกว่า “Dependent Class” และทำการทดสอบจนครบทุกคลาส

## บทที่ 3

# การวิเคราะห์และออกแบบระบบ

### 3.1 ปัญหาความเป็นมาและแนวทางการแก้ไข

สายงานควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศเป็นหน่วยงานหนึ่งของกลุ่มงานเทคโนโลยีสารสนเทศ ทำหน้าที่ในการดูแล ติดตาม และทดสอบความถูกต้องของซอฟต์แวร์ที่พัฒนา หรือเปลี่ยนแปลงแก้ไข โดยสายงานเทคโนโลยีสารสนเทศ ดังนั้นเมื่อซอฟต์แวร์ถูกพัฒนาเสร็จโดยสายงานพัฒนาเทคโนโลยีสารสนเทศจะส่งมอบซอฟต์แวร์ให้ฝ่ายควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศทดสอบความถูกต้องของซอฟต์แวร์ โดยงานที่เกี่ยวข้องโดยตรงกับฝ่ายควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศในกระบวนการทดสอบซอฟต์แวร์ได้แก่

1. การวางแผนกำหนดระยะเวลาการทดสอบซอฟต์แวร์ระหว่างโปรแกรมเมอร์กับผู้ทดสอบ
2. เมื่อโปรแกรมเมอร์พัฒนาระบบเสร็จจะส่งใบกำหนดการทดสอบให้ทางทีมทดสอบเพื่อทำการทดสอบ
3. ผู้ทดสอบยืนยันระยะเวลาการทดสอบ (ส่งใบกำหนดระยะเวลาการทดสอบ)
4. ผู้ทดสอบทำการทดสอบซอฟต์แวร์ โดยในระหว่างหรือก่อนการทดสอบผู้ทดสอบจะต้องจัดทำกรณีทดสอบและบันทึกผลการทดสอบ
5. ผู้ทดสอบจัดทำใบสรุปผลการทดสอบเพื่อแจ้งให้ผู้ที่เกี่ยวข้องทราบ
6. ผู้ทดสอบจัดทำรายงานผลการทดสอบเพื่อแจ้งให้ผู้ที่เกี่ยวข้องทราบ

### ปัญหาที่เกิดขึ้น

1. มีการทำงานบางอย่างที่ซ้ำซ้อนเช่นวางแผนการทดสอบและการกำหนดระยะเวลาในการทดสอบ
2. ผู้ทดสอบเสียเวลาในเรื่องการจัดการด้านเอกสารมากเกินไป
3. การจัดเก็บเอกสารมีการจัดเก็บเป็นรูปแบบไฟล์ สร้างความยุ่งยากในการค้นหา
4. ผู้ทดสอบต้องเสียเวลาในการจัดการด้านเอกสารเป็นจำนวนมาก ทำให้การทดสอบล่าช้า
5. เอกสารที่ใช้ในการทดสอบแยกกันจัดเก็บตามทีมทดสอบเมื่อต้องการความช่วยเหลือด้านการทดสอบจากทีมอื่นๆทำได้ยากเนื่องจากไม่ทราบข้อมูลสำหรับการทดสอบ

### แนวทางการแก้ไข

การนำระบบการสนับสนุนการทำงานของหน่วยงานควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศ โดยเน้นการจัดการเอกสารแบบอัตโนมัติ เพื่อช่วยลดระยะเวลาในการจัดทำเอกสาร เอกสารเป็นเอกสารที่ส่งวนเวียนสำหรับการส่งงานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนักศึกษาเห็นประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้ผู้ทดสอบให้เวลาและความสำคัญกับการทดสอบเพื่อส่งผลให้การทดสอบมีประสิทธิภาพ การวางแผนการทดสอบอย่างมีประสิทธิภาพและแม่นยำ การจัดเก็บกรณีทดสอบเพื่อการสืบค้นที่สะดวกและรวดเร็วขึ้น มีระบบการติดตามสถานะการทดสอบ เพื่อผู้บริหารได้ทราบถึงปัญหาของการทดสอบในปัจจุบันและหาแนวทางในการแก้ไขปัญหาดังกล่าว ซึ่งแนวทางการแก้ปัญหาที่คาดว่าจะได้รับประโยชน์ดังนี้

1. เพื่อช่วยให้การวางแผนการทดสอบมีประสิทธิภาพมากยิ่งขึ้น
2. ลดภาระงานด้านเอกสารของผู้ทดสอบทำให้ผู้ทดสอบสามารถทดสอบซอฟต์แวร์ได้เต็มประสิทธิภาพ
3. ผู้บริหารและผู้ทดสอบสามารถติดตามสถานะของการทดสอบและทราบถึงปัญหา ระหว่างการทดสอบ และช่วยค้นหาแนวทางในการแก้ปัญหานั้น
4. ลดขั้นตอนการติดต่อประสานงาน กับผู้ที่เกี่ยวข้องเนื่องจากการจัดเก็บข้อมูลของผู้ที่เกี่ยวข้องไว้ในฐานข้อมูลเดียวกัน โดยแยกผู้ที่เกี่ยวข้องออกเป็นระบบต่างๆที่ทีมผู้ทดสอบได้ทำการทดสอบ
5. ลดขั้นตอนการทดสอบเนื่องจากการจัดเก็บกรณีทดสอบอย่างมีระบบ ผู้ทดสอบสามารถค้นหกรณีสอบครั้งก่อน ได้สะดวกรวดเร็วขึ้น และสามารถบันทึกกรณีทดสอบ ณ ปัจจุบันได้
6. ผู้ทดสอบมีความกระตือรือร้นในการทดสอบ ทำให้การทดสอบมีประสิทธิภาพมากยิ่งขึ้น เนื่องจากผู้บริหารสามารถมองเห็นประสิทธิภาพ ปริมาณงานของผู้ทดสอบแต่ละคนที่ได้รับมอบหมายได้

ดังนั้นจึงได้นำเทคโนโลยีเข้ามาช่วยในการจัดทำระบบสนับสนุนฝ่ายควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศ โดยได้นำเอาเทคโนโลยีเว็บไซต์ ที่พัฒนาด้วย ASP.Net เพื่อทำการติดต่อประสานงานในหน่วยงานสะดวกและรวดเร็วขึ้นการจัดเก็บข้อมูลใน Database ด้วย SQL Server 2005 ที่มีประสิทธิภาพ เพื่อทำให้มีการเก็บรวบรวมข้อมูลที่จำเป็นไว้ในส่วนกลาง ผู้ใช้ที่เกี่ยวข้องสามารถค้นหาและบันทึกข้อมูลที่จำเป็นและเป็นประโยชน์ต่องานการทดสอบ

### 3.2 การวิเคราะห์ความต้องการของระบบ

ระบบที่พัฒนาจะต้องรองรับแนวทางในการแก้ไขปัญหาทั้งหมดแต่ยังคงไว้ซึ่งระเบียบปฏิบัติ และขั้นตอนการทำงานในหน่วยงาน ดังนั้นระบบที่ถูกพัฒนาขึ้นมาใหม่สามารถรองรับการปฏิบัติงานดังนี้

1. บันทึกข้อมูลผู้ทดสอบ ผู้ที่เกี่ยวข้องในการทดสอบ
2. บันทึกข้อมูลการทดสอบ รายละเอียดของซอฟต์แวร์ที่จะทดสอบ

เอกสารนี้เป็นเอกสารทูลงานวิศวกรรมเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. บันทึก วัน-เวลา ที่ จะ ทำ การ ทดสอบ และ ปรับปรุง แผน การ ทดสอบ
4. บันทึก ข้อมูล ที่ ใช้ สำหรับ พิมพ์ เอกสาร ที่ เกี่ยว ข้อง กับการ ทดสอบ
5. บันทึก สถานะ การ ทดสอบ
6. บันทึก กรณี ทดสอบ และ ผล การ ทดสอบ
7. แสดง หน้าจอ แผน การ ทดสอบ
8. พิมพ์ ไป กำหนด ระยะเวลา การ ทดสอบ
9. พิมพ์ ไป สรุป ผล การ ทดสอบ
10. พิมพ์ รายงาน การ ทดสอบ
11. หน้าจอ ประเมิน ประสิทธิภาพ การ ทดสอบ

### ส่วนข้อมูลผู้ทดสอบ ผู้ที่เกี่ยวข้องในการทดสอบ

เป็นการจัดเก็บข้อมูลผู้ที่เกี่ยวข้องในกระบวนการทดสอบ ซึ่ง ได้แก่ ผู้ทดสอบ (Tester) , ผู้ใช้งานซอฟต์แวร์ (User), ผู้พัฒนาซอฟต์แวร์ (Developer) ,ผู้จัดการทีมทดสอบ (Tester Manager) , ผู้จัดการทีมพัฒนาซอฟต์แวร์ (Developer Manager)

### ส่วนข้อมูลการทดสอบ รายละเอียดของซอฟต์แวร์

เป็นส่วนบันทึกรายละเอียดเกี่ยวกับซอฟต์แวร์ที่ต้องการทดสอบ เช่น ชื่อของซอฟต์แวร์ ทีมพัฒนา ทีมทดสอบ ผู้ใช้งาน เป็นการทดสอบครั้งที่ หมายเลข UR(User Requirement) ครั้งที่ ทดสอบ และรายละเอียดอื่นๆ ที่เกี่ยวข้อง

### ส่วนแผนการทดสอบ

เป็นการกำหนดระยะเวลาในการทดสอบ โดยมีการกำหนดวันที่เริ่มต้นการทดสอบ วันที่ ทดสอบเสร็จและวันที่ติดตั้ง

### ส่วนสถานะการทดสอบ

เป็นการแสดงถึงสถานะที่ทำการทดสอบ เช่น ยังไม่ถึงระยะเวลาทดสอบ กำลัง ดำเนินการทดสอบ ติดปัญหายกเลิกการทดสอบ เลื่อนการทดสอบ ตลอดจนยกเลิกการทดสอบ พร้อมทั้งหากเกิดปัญหาในการทดสอบก็จะระบุสาเหตุของปัญหานั้นด้วย

### ส่วนของกรณีทดสอบและผลการทดสอบ

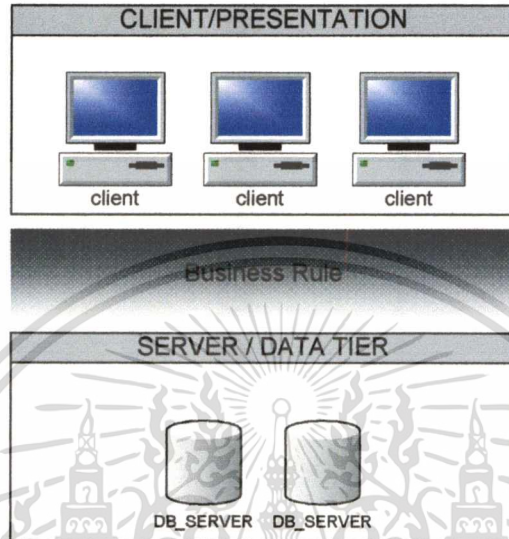
เป็นส่วนที่ใช้บันทึก แสดงผลและแก้ไขกรณีทดสอบ สามารถกำหนดได้ก่อนและ ระหว่างการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ส่วนของรายงาน

เป็นส่วนของการออกรายงานที่เกี่ยวข้องกับการทดสอบ เช่น ใบกําหนดระยะเวลาการทดสอบ , ใบบสรูปผลการทดสอบ, รายงานการทดสอบ และใบบประเมินการทดสอบ

## ส่วนจัดการข้อมูลระบบสนับสนุนการทดสอบซอฟต์แวร์



ภาพที่ 3.1 แสดงรูปแบบการทำงานของระบบภายใต้แนวคิด ไคลเอนท์ เซิร์ฟเวอร์

การจัดการข้อมูลใช้แนวความคิดของ ไคลเอนท์-เซิร์ฟเวอร์ คือ มีการแบ่งหน้าที่การทำงานให้กับเครื่องคอมพิวเตอร์แต่ละเครื่อง โดยประกอบด้วย 2 เทียร์คือ

- 1) คาด้าเทียร์ ทำหน้าที่เป็นคาด้าเบสเซิร์ฟเวอร์
- 2) ฟริเซนเทชั่นเทียร์ หรือ ไคลเอนท์ ทำหน้าที่แสดงผล

ในการแสดงข้อมูลของฟริเซนเทชั่นเทียร์ที่ร้องขอข้อมูลไปยังคาด้าเทียร์กระทำผ่านทางตัวกลางซึ่งมีหน้าที่กำหนดกฎ กติกา หรือเงื่อนไขในการติดต่อกับคาด้าเทียร์ตัวกลางที่เรียกว่า บิสสิเนสเทียร์หรือบิสสิเนสรูลซึ่งเป็นกฎที่คอยกำกับฟริเซนเทชั่นเทียร์ที่ติดต่อยังคาด้าเทียร์ให้สามารถทำงานสอดคล้องกับข้อกำหนดของคาด้าเทียร์ โดยสถาปัตยกรรม 2 เทียร์ นี้เป็นการใช้แอปพลิเคชันของ Visual Studio 2005 ทำหน้าที่เป็นฟริเซนเทชั่นเทียร์และ Microsoft SQL Server 2005 ทำหน้าที่เป็นคาด้าเทียร์

### 3.3 ระบบสนับสนุนการทดสอบซอฟต์แวร์

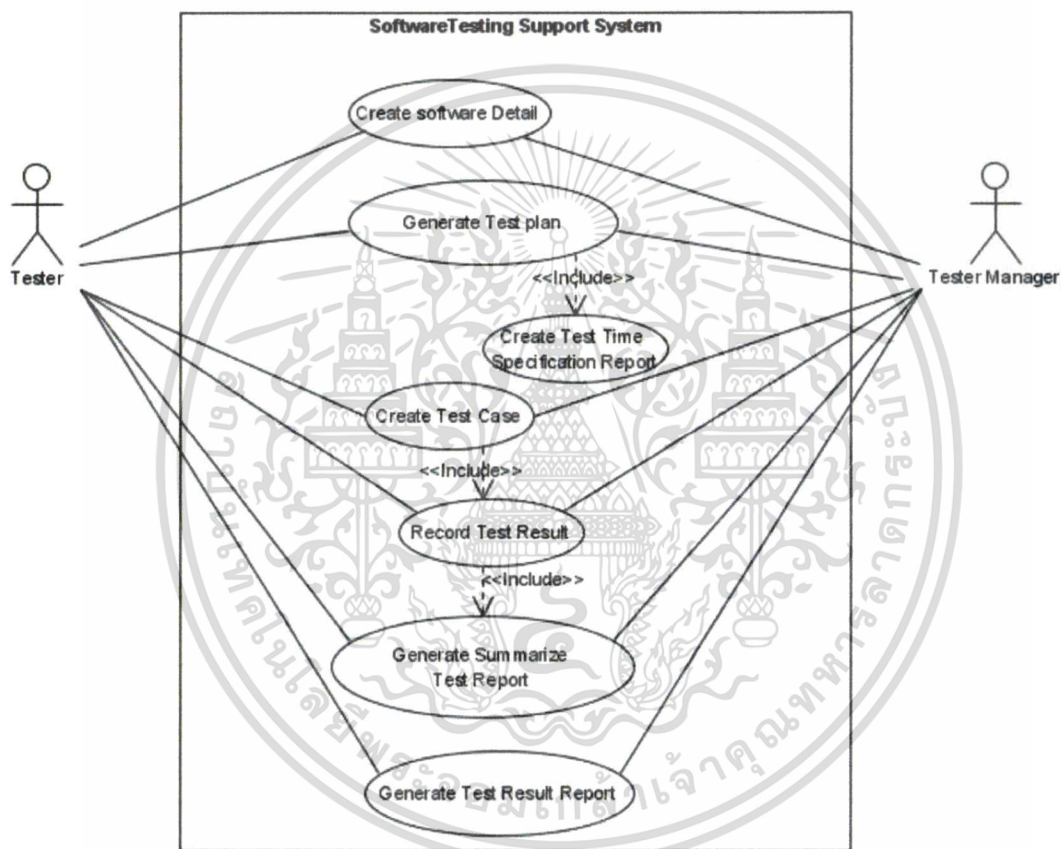
นำเสนอหลักการวิเคราะห์ระบบเชิงวัตถุ โดยวิเคราะห์ปัญหาจากความต้องการของระบบที่เกิดขึ้น โดยจำแนกกระบวนการออกเป็นแต่ละกิจกรรมแสดงเป็นแผนภาพรวมในการพัฒนาระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วยไดอะแกรมแสดงรายละเอียดการทำงานของระบบตามลำดับนี้ Use Case Diagram, Activity Diagram, Sequence Diagram และ Class Diagram

### 3.3.1 Use Case Diagram

**Use Case Diagram** คือ ไดอะแกรมที่อธิบายหน้าที่ที่ระบบต้องทำ โดยการใช้สัญลักษณ์รูปวงรี พร้อมทั้งเขียนชื่อ Use Case กำกับไว้โดยการใช้คำกริยาหรือกริยาวิเศษณ์ และมีผู้ใช้งานระบบหรือ Actor เพื่อแสดงความเกี่ยวข้องในแต่ละ Use Case ซึ่งในระบบนี้สามารถแสดงได้ดังภาพ



ภาพที่ 3.2 แสดง Use Case Diagram ของระบบ Software Quality Assurance Support

จาก Use Case Diagram ข้างต้น ประกอบด้วย Actor ซึ่งเป็นส่วนแสดงบุคคลผู้เกี่ยวข้องในกระบวนการทดสอบซอฟต์แวร์ ซึ่งได้แก่ ผู้ทดสอบ (Tester) ผู้จัดการทีมทดสอบ (Tester Manager)

**ผู้ทดสอบ (Tester)** คือ ผู้ที่ต้องทำการทดสอบซอฟต์แวร์ ทำหน้าที่ในการบันทึกข้อมูลผู้ที่เกี่ยวข้องในกระบวนการทดสอบ บันทึกรายละเอียดการทดสอบ ปรับปรุงระยะเวลาการทดสอบ ออกแบบกรณีทดสอบ บันทึกผลการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ผู้จัดการทีมทดสอบ (Tester Manager)** คือหัวหน้าของผู้ทดสอบ ทำหน้าที่ อนุมัติ ระยะเวลาการทดสอบ อนุมัติผลการทดสอบ และรับรายงานผลการทดสอบ

ส่วนฟังก์ชันการทำงานหลักได้แยกแสดงออกเป็น Use Case ซึ่งประกอบด้วย

**1. Create software Detail (บันทึกรายละเอียดการทดสอบ)** เป็นส่วนของการบันทึก รายละเอียดเกี่ยวกับซอฟต์แวร์ที่จะทำการทดสอบ เช่น ชื่อระบบที่จะทำการทดสอบ หมายเลข User Requirement ทีมพัฒนา ทีมทดสอบ ผู้พัฒนา ผู้ทดสอบ กำหนดระยะเวลาการพัฒนา วันติดตั้ง ครั้งที่ของการทดสอบ

**2. Generate Test plan (สร้างแผนการทดสอบ)** เป็นการสร้างตารางการทดสอบ โดย ระบบจะสร้างตารางการทดสอบจากวันที่เริ่มต้นการทดสอบ วันที่สิ้นสุดการทดสอบ ตามที่ผู้ ทดสอบได้บันทึกข้อมูลไว้ในส่วนของ การบันทึกรายละเอียดการทดสอบ

**3. Generate Test Case (สร้างกรณีทดสอบ)** เป็นขั้นตอนของการบันทึกข้อมูลกรณี ทดสอบที่ผู้ทดสอบ ได้ออกแบบไว้

**4. Record Test Result (บันทึกผลการทดสอบ)** เป็นการบันทึกผลการทดสอบจากการ ทดสอบตามกรณีทดสอบที่ได้มีการออกแบบไว้

**5. Generate Summarize Test Report (สร้างรายงานสรุปผลการทดสอบ)** เป็นการ สร้างรายงานสรุปผลการทดสอบ จากการทดสอบตามกรณีทดสอบที่ได้ออกแบบไว้

**6. Generate Test Result Report (สร้างรายงานการตรวจคุณภาพระบบงานเทคโนโลยี สารสนเทศ)** เป็นการสร้างรายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศ หลังจาก ที่ได้มีการทดสอบซอฟต์แวร์เสร็จและพร้อมติดตั้ง

**ตารางที่ 3.1** รายละเอียดยูสเคส Create software Detail (บันทึกรายละเอียดการทดสอบ)

| ชื่อยูสเคส                   | เคส Create software Detail  |
|------------------------------|---|
| คำอธิบาย                     | การบันทึกรายละเอียดเกี่ยวกับซอฟต์แวร์ที่จะทำการทดสอบ เช่น ชื่อระบบที่ จะทำการทดสอบ หมายเลข User Requirement ทีมพัฒนา ทีมทดสอบ ผู้พัฒนา ผู้ทดสอบ กำหนดระยะเวลาการพัฒนา วันติดตั้ง ครั้งที่ของการ ทดสอบ |
| เหตุการณ์ที่กระตุ้น การทำงาน | เมื่อมีการส่งมอบซอฟต์แวร์ให้ทำการทดสอบ  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 3.1 (ต่อ)

|                     |  |
|---------------------|--|
| คำอธิบายยูสเคส      | จัดเก็บข้อมูลที่เกี่ยวข้องกับการทดสอบซอฟต์แวร์เพื่อนำไปใช้ในกระบวนการทดสอบซอฟต์แวร์ ซึ่งได้แก่ เลขที่ User Request, ระบบที่ทำการทดสอบ, ครั้งที่ทดสอบ, วันเริ่มต้นการพัฒนา, วันสิ้นสุดการพัฒนา จำนวนวันที่ใช้ในการพัฒนา, วันที่เริ่มต้น, ในการทดสอบ, วันที่สิ้นสุดการทดสอบ, จำนวนวันที่ใช้ในการทดสอบ, วันที่ติดตั้ง, หมายเหตุ, สถานะของการทดสอบ, เลขที่หนังสือส่งตรวจ, วันที่ส่งตรวจ  |
| แอกเตอร์            | Tester   |
| ยูสเคสที่เกี่ยวข้อง | -  |
| ผู้เกี่ยวข้องอื่น   | Tester Manager   |
| เงื่อนไขเริ่มต้น    | เมื่อมีการส่งมอบซอฟต์แวร์ให้ทำการทดสอบ   |
| เงื่อนไขเมื่อสำเร็จ | ข้อมูลเกี่ยวกับซอฟต์แวร์ที่ทดสอบ   |
| การทำงาน            | <ol style="list-style-type: none"> <li>1. ผู้ทดสอบบันทึกข้อมูล User Request โดยบันทึกเป็นตัวเลขจำนวน 8 หลัก</li> <li>2. ผู้ทดสอบบันทึกโดยบันทึกเป็น ปีปัจจุบัน และครั้งที่ทดสอบระบบ โดยที่ระบบจะบันทึกเป็น “YYYY/XX”</li> <li>3. ผู้ทดสอบเลือกระบบที่ต้องการทดสอบ โดยเลือกระบบที่ได้มีการเพิ่มข้อมูลการทดสอบไว้แล้ว โดยจะแสดงข้อมูลชื่อระบบเป็นภาษาอังกฤษและบันทึกเป็นรหัสของระบบที่ทำการทดสอบนั้น</li> <li>4. ผู้ทดสอบเลือกทีมพัฒนาและเลือกผู้พัฒนาที่ทำการพัฒนาซอฟต์แวร์ที่ส่งตรวจ โดยสามารถเลือกผู้พัฒนาได้มากกว่า 1 คน แต่ต้องสังกัดอยู่ที่ทีมเดียวกัน</li> <li>5. ผู้ทดสอบระบุวันที่เริ่มต้นและสิ้นสุดในการพัฒนาซอฟต์แวร์</li> <li>6. ผู้ทดสอบกำหนดผู้ทดสอบซึ่งสามารถเพิ่มได้มากกว่า 1 คน แต่ผู้ทดสอบทุกคนต้องอยู่ที่ทีมเดียวกัน</li> <li>7. ผู้ทดสอบระบุวันที่เริ่มต้นและสิ้นสุดในการทดสอบซอฟต์แวร์ โดยวันที่ทดสอบโดยถ้ายังไม่ได้เลือกค่าเริ่มต้นของการทดสอบจะเป็นวันที่ถัดจากวันที่พัฒนาซอฟต์แวร์ 1 วัน วันสิ้นสุดเท่ากับวันเริ่มต้นในการพัฒนา บวกกับ 25% ของวันที่ใช้ในการพัฒนาซอฟต์แวร์</li> <li>8. ผู้ทดสอบระบุวันที่ติดตั้งซอฟต์แวร์</li> <li>9. ผู้ทดสอบระบุสถานะการทดสอบหากไม่ระบุจะแสดงเป็น ยังไม่ได้รับมอบหมาย</li> </ol> |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับทีมใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.1 (ต่อ)

|                  |  |
|------------------|--|
| เงื่อนไขการทำงาน | <ol style="list-style-type: none"> <li>1. ผู้ทดสอบอาจระบุได้มากกว่า 1 คน แต่ผู้ทดสอบต้องสังกัดอยู่ที่ทีมเดียวกัน</li> <li>2. ผู้พัฒนาอาจระบุได้มากกว่า 1 คน แต่ผู้พัฒนาต้องสังกัดอยู่ที่ทีมเดียวกัน</li> <li>3. ข้อมูลวันที่เริ่มต้นและสิ้นสุดสามารถปรับปรุงได้</li> </ol> |
|------------------|--|

### ตารางที่ 3.2 รายละเอียดยูสเคส Generate Test plan (สร้างแผนการทดสอบ)

| ชื่อยูสเคส                  | Generate Test plan   |
|-----------------------------|--|
| คำอธิบาย                    | เป็นการสร้างตารางการทดสอบ โดยระบบจะสร้างตารางการทดสอบจากวันที่เริ่มต้นการทดสอบ วันที่สิ้นสุดการทดสอบ ตามที่ผู้ทดสอบได้บันทึกข้อมูลไว้ในส่วนของ การบันทึกรายละเอียดการทดสอบ |
| เหตุการณ์ที่กระตุ้นการทำงาน | เมื่อมีการบันทึกรายละเอียดของการทดสอบ หรือ ผู้ทดสอบปรับปรุงระยะเวลาการทดสอบ  |
| คำอธิบายยูสเคส              | แสดงเลขที่ User Request ที่ผู้ทดสอบแต่ละคนต้องทำการทดสอบ ในรูปแบบของ Calendar  |
| แอกเตอร์                    | Tester   |
| ยูสเคสที่เกี่ยวข้อง         | Create software Detail   |
| ผู้เกี่ยวข้องอื่น           | Tester Manager   |
| เงื่อนไขเริ่มต้น            | เมื่อมีการเพิ่มรายละเอียดของซอฟต์แวร์ที่ต้องทดสอบ  |
| เงื่อนไขเมื่อสำเร็จ         | แผนการทดสอบซอฟต์แวร์   |
| การทำงาน                    | ผู้ทดสอบสามารถมองเห็นแผนการทดสอบ และหากต้องการดูรายละเอียดของซอฟต์แวร์ที่ต้องทดสอบสามารถเลือกดูรายละเอียดของแต่ละ User Request ได้   |
| เงื่อนไขการทำงาน            | ผู้ทดสอบแต่ละคนสามารถมองเห็นแผนการทำงานของตนเท่านั้น ส่วนผู้จัดการผู้ทดสอบสามารถเลือกดูแผนการทดสอบของแต่ละคนได้  |

### ตารางที่ 3.3 รายละเอียดยูสเคส Generate Test Case (สร้างกรณีทดสอบ)

| ชื่อยูสเคส | Generate Test Case  |
|------------|---|
| คำอธิบาย   | เป็นขั้นตอนของการบันทึกข้อมูลกรณีทดสอบที่ผู้ทดสอบได้ออกแบบไว้ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.3 (ต่อ)

|                             |  |
|-----------------------------|--|
| เหตุการณ์ที่กระตุ้นการทำงาน | เมื่อมีการยืนยันวันที่ทำการทดสอบซอฟต์แวร์แล้ว  |
| คำอธิบายยูสเคส              | เก็บรวบรวมกรณีทดสอบที่ผู้ทดสอบได้มีการออกแบบไว้  |
| แอกเตอร์                    | Tester   |
| ยูสเคสที่เกี่ยวข้อง         | 1. Create software Detail<br>2. Generate Test plan   |
| ผู้เกี่ยวข้องอื่น           | Tester Manager   |
| เงื่อนไขเริ่มต้น            | เมื่อมีการออกแบบกรณีทดสอบ  |
| เงื่อนไขเมื่อสำเร็จ         | ข้อมูลกรณีทดสอบสำหรับนำไปใช้ในกระบวนการทดสอบ   |
| การทำงาน                    | ผู้ทดสอบเลือก User Request และระบบที่ทำการทดสอบ เพื่อบันทึกกรณีทดสอบที่ผู้ทดสอบได้ทำการออกแบบไว้ |
| เงื่อนไขการทำงาน            | การบันทึกกรณีทดสอบ ผู้ทดสอบต้องบันทึกรายละเอียดการทดสอบแล้ว                                      |

### ตารางที่ 3.4 รายละเอียดยูสเคส Record Test Result (บันทึกผลการทดสอบ)

|                             |   |
|-----------------------------|---|
| ชื่อยูสเคส                  | <b>Record Test Result</b>   |
| คำอธิบาย                    | เป็นการบันทึกผลการทดสอบจากการทดสอบตามกรณีทดสอบที่ได้มีการออกแบบไว้          |
| เหตุการณ์ที่กระตุ้นการทำงาน | เมื่อ Tester ทดสอบซอฟต์แวร์   |
| คำอธิบายยูสเคส              | บันทึกผลการทดสอบซอฟต์แวร์ตามกรณีทดสอบที่ได้มีการออกแบบไว้แล้ว               |
| แอกเตอร์                    | Tester  |
| ยูสเคสที่เกี่ยวข้อง         | 1. Create software Detail<br>2. Generate Test plan<br>3. Generate Test Case |
| ผู้เกี่ยวข้องอื่น           | Tester Manager  |
| เงื่อนไขเริ่มต้น            | เมื่อเริ่มต้นการทดสอบซอฟต์แวร์  |
| เงื่อนไขเมื่อสำเร็จ         | ผลลัพธ์จากการทดสอบ (Test Result)  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.4 (ต่อ)

|                  |   |
|------------------|---|
| การทำงาน         | <ol style="list-style-type: none"> <li>1. ผู้ทดสอบเลือก User Request ระบบที่ทำการทดสอบ และกรณีทดสอบ ที่ทำการทดสอบเพื่อบันทึกผลการทดสอบที่ผู้ทดสอบได้ทำการทดสอบ</li> <li>2. หากผู้ทดสอบได้ ทำการทดสอบกรณีทดสอบแล้ว ระบบจะประเมิน เปรียบเทียบผลของการทดสอบ</li> </ol> |
| เงื่อนไขการทำงาน | ผู้ทดสอบสามารถเพิ่มกรณีทดสอบในขณะที่ทำการทดสอบได้   |

### ตารางที่ 3.5 รายละเอียดยูสเคส Generate Summarize Test Report

|                             |  |
|-----------------------------|--|
| ชื่อยูสเคส                  | <b>Generate Summarize Test Report</b>  |
| คำอธิบาย                    | เป็นการสร้างรายงานสรุปผลการทดสอบ   |
| เหตุการณ์ที่กระตุ้นการทำงาน | เมื่อผู้ทดสอบทดสอบซอฟต์แวร์เสร็จ และเรียกดู รายงานสรุปผลการทดสอบ   |
| คำอธิบายยูสเคส              | สร้างรายงานสรุปผลการทดสอบเพื่อนเป็นการแจ้งให้ผู้ที่เกี่ยวข้องซึ่งได้แก่ ผู้พัฒนาและผู้ใช้งานรับทราบว่าซอฟต์แวร์ได้รับการทดสอบแล้ว  |
| แอกเตอร์                    | Tester   |
| ยูสเคสที่เกี่ยวข้อง         | <ol style="list-style-type: none"> <li>1. Create software Detail</li> <li>2. Generate Test plan</li> <li>3. Generate Test Case</li> <li>4. Record Test Result</li> </ol>   |
| ผู้เกี่ยวข้องอื่น           | Tester Manager   |
| เงื่อนไขเริ่มต้น            | เมื่อการทดสอบเสร็จ   |
| เงื่อนไขเมื่อสำเร็จ         | ข้อมูลสรุปผลการทดสอบ   |
| การทำงาน                    | <p>ผู้ทดสอบเลือก User Request ระบบที่ทำการทดสอบ และเพิ่มข้อมูลที่ต้องใช้ในการสร้างรายงานสรุปผลการทดสอบซอฟต์แวร์เพิ่มเติม ได้แก่</p> <ol style="list-style-type: none"> <li>1. ผลการตรวจ</li> <li>2. ความพร้อมในการส่ง โปรแกรมมาให้ทดสอบ</li> <li>3. เอกสารระบบงาน <ul style="list-style-type: none"> <li>● SYSTEM FLOWCHART</li> <li>● PROGRAM ABSTRACT</li> <li>● LIST OF FILES</li> <li>● LIST OF PROGRAM</li> <li>● FILE DESCRIPTION</li> <li>● LIST OF REPORT</li> </ul> </li> </ol> |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.5(ต่อ)

|                  |   |
|------------------|---|
| การทำงาน (ต่อ)   | <ul style="list-style-type: none"> <li>● FILE RECORD LAYOUT</li> <li>● SAMPLE REPORT</li> <li>● RECORD DESCRIPTION</li> <li>● USER MANUAL</li> <li>● CODE DESCRIPTION</li> </ul> <p>4. การควบคุมภายในของระบบงาน</p> |
| เงื่อนไขการทำงาน | ผู้ทดสอบสามารถดูรายงานสรุปผลการทดสอบได้เฉพาะของตัวเองเท่านั้น ส่วนผู้จัดการผู้ทดสอบสามารถดูรายงานสรุปผลการทดสอบ ของทุกคนได้   |

### ตารางที่ 3.6 รายละเอียดยูสเคส Generate Test Result Report

(สร้างรายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศ)

|                             |   |
|-----------------------------|---|
| ชื่อยูสเคส                  | Generate Test Result Report   |
| คำอธิบาย                    | เป็นการสร้างรายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศ หลังจากที่ทดสอบซอฟต์แวร์เสร็จมีความพร้อมในการติดตั้ง  |
| เหตุการณ์ที่กระตุ้นการทำงาน | เมื่อผู้ทดสอบทดสอบซอฟต์แวร์เสร็จ และเรียกดู รายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศ   |
| คำอธิบายยูสเคส              | สร้างรายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศ หลังจากที่ได้สอบซอฟต์แวร์เสร็จมีความพร้อมในการติดตั้ง เพื่อเป็นการแจ้งให้ผู้ที่เกี่ยวข้องซึ่งได้แก่ ผู้พัฒนาและผู้ใช้งานรับทราบว่าซอฟต์แวร์ได้รับการทดสอบแล้วและพร้อมติดตั้ง |
| แอกเตอร์                    | Tester  |
| ยูสเคสที่เกี่ยวข้อง         | <ol style="list-style-type: none"> <li>5. Create software Detail</li> <li>6. Generate Test plan</li> <li>7. Generate Test Case</li> <li>8. Record Test Result</li> </ol>  |
| ผู้เกี่ยวข้องอื่น           | Tester Manager  |
| เงื่อนไขเริ่มต้น            | เมื่อมีการสรุปผลการทดสอบ  |
| เงื่อนไขเมื่อสำเร็จ         | ข้อมูลผลการทดสอบ  |
| การทำงาน                    | <p>ผู้ทดสอบเลือก User Request ระบบที่ทำการทดสอบ และเพิ่มข้อมูลที่ต้องใช้ในการสร้างรายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศเพิ่มเติม ได้แก่</p> <ol style="list-style-type: none"> <li>1. เลขที่รายงาน</li> </ol>           |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

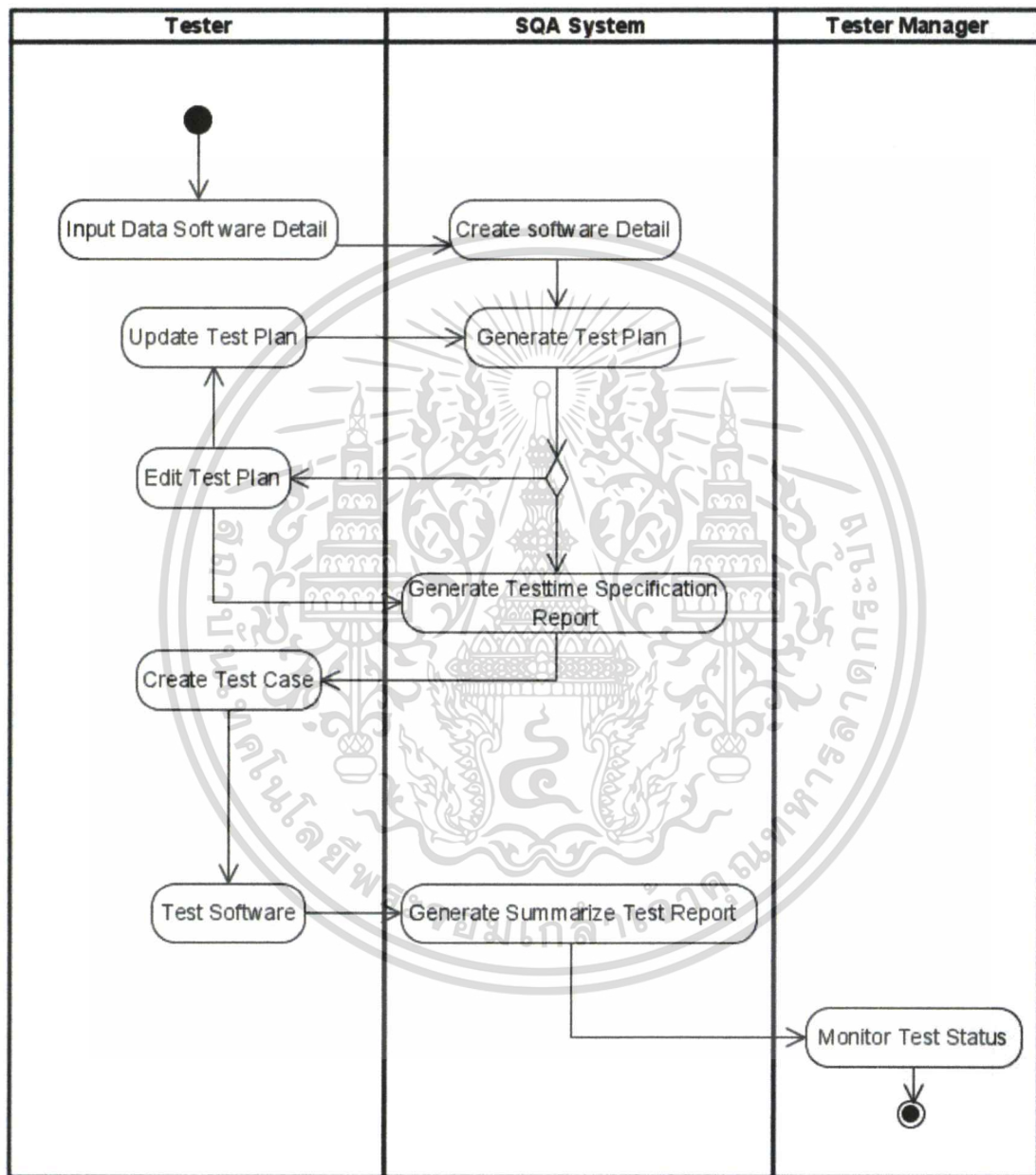
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตารางที่ 3.6 (ต่อ)

|                         |  |
|-------------------------|--|
| <p>การทำงาน (ต่อ)</p>   | <ol style="list-style-type: none"> <li>2. วันที่ออกรายงาน</li> <li>3. ผลการตรวจ <ul style="list-style-type: none"> <li>● ด้านโปรแกรม</li> <li>● ด้านเอกสารระบบงาน</li> <li>● ด้านการควบคุมภายใน</li> </ul> </li> <li>4. วัตถุประสงค์ของการปรับปรุงโปรแกรม</li> <li>5. ขอบเขตการตรวจ</li> <li>6. Test Data Method</li> <li>7. Base Case System Evaluation</li> <li>8. ผลการตรวจในแต่ละด้าน <ul style="list-style-type: none"> <li>● ด้านโปรแกรม</li> <li>● ด้านการทดสอบ</li> <li>● ด้านเอกสารระบบงาน</li> <li>● ด้านการควบคุมภายใน</li> </ul> </li> </ol> |
| <p>เงื่อนไขการทำงาน</p> | <p>ผู้ทดสอบสามารถดูรายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศ ได้เฉพาะของตัวเองเท่านั้น ส่วนผู้จัดการผู้ทดสอบสามารถดูรายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศ ของทุกคนได้</p>  |

### 3.2.2 Activity Diagram

Activity Diagram คือ ไดอะแกรมแสดงให้เห็นลำดับการดำเนินงาน (Activity) จากกิจกรรมหนึ่งไปยังกิจกรรมหนึ่งภายในระบบที่เกิดจากการทำงานของออบเจกต์ โดยลักษณะของไดอะแกรมจะคล้ายกับ Flow Chart



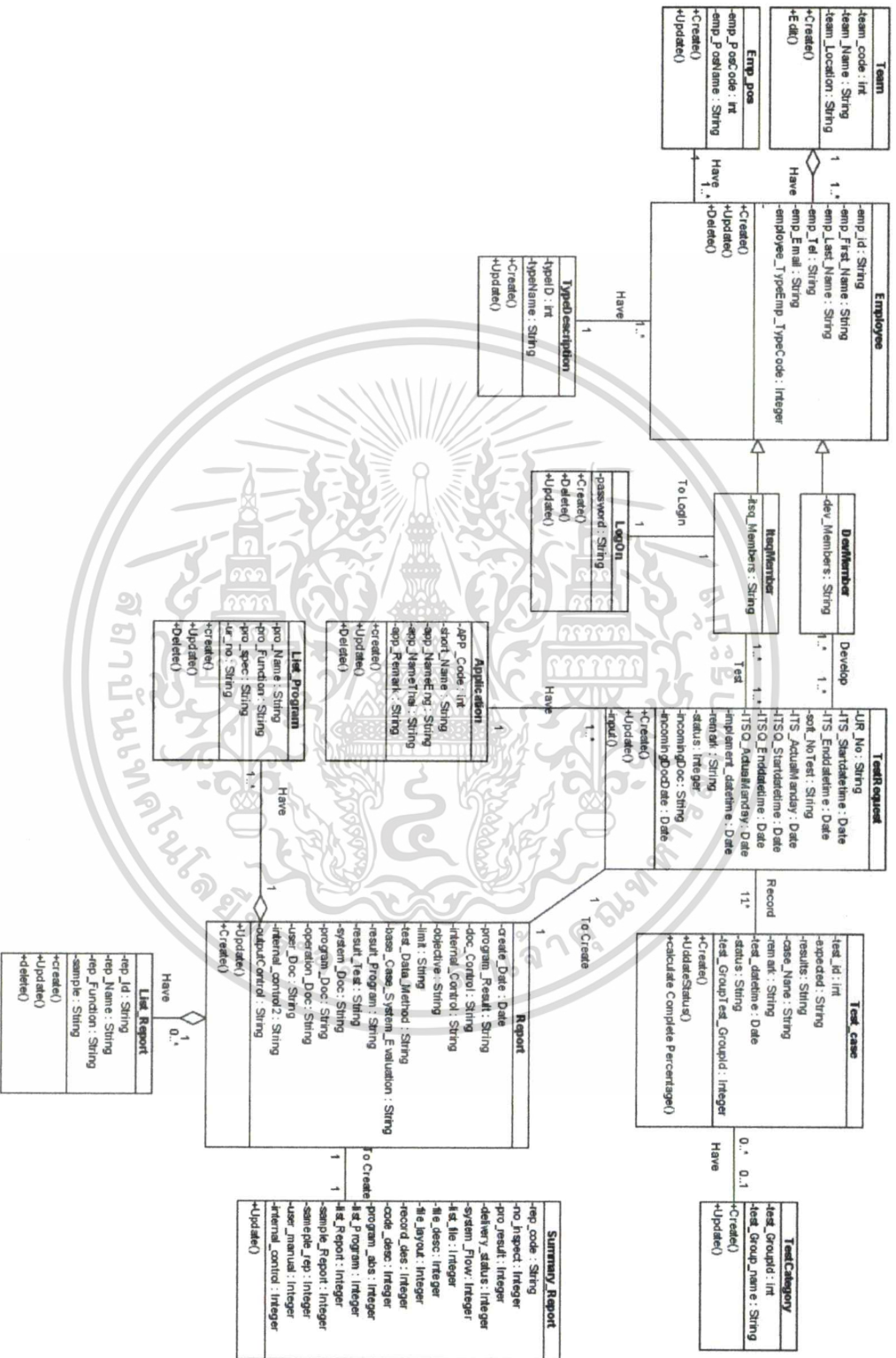
ภาพที่ 3.3 แอกทิวิตีไดอะแกรมแสดงการทำงานของระบบสนับสนุนการทดสอบซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.3 Class Diagram

Class Diagram คือ ไดอะแกรมที่ใช้ในการแสดงกลุ่มของคลาส โครงสร้างของคลาส อินเตอร์เฟซ (Interface) และแสดงความสัมพันธ์ (Relationship) ระหว่างคลาส ซึ่งการพัฒนา ระบบนี้สามารถแสดงกลุ่มของคลาสต่างๆ แสดงได้ดังนี้

- คลาส Application เป็นคลาสของรายละเอียดโปรแกรมที่ทดสอบ
- คลาส DevMember เป็นคลาสของสมาชิกผู้พัฒนาโปรแกรมที่ส่งมาทดสอบ
- คลาส Emp\_pos เป็น คลาสของตำแหน่งของพนักงาน
- คลาส Employee เป็นคลาสของพนักงานรวมถึง ผู้ทดสอบ ผู้พัฒนาและผู้ใช้งาน
- คลาส ItsqMember เป็นคลาสของสมาชิกผู้ทดสอบ
- คลาส List\_Program เป็นคลาสของโปรแกรมย่อยที่ส่งมาทดสอบ
- คลาส List\_Report เป็นคลาสของรายงานที่ส่งมาทดสอบ
- คลาส LogOn เป็นคลาสของข้อมูลการเข้าระบบ
- คลาส Report เป็นคลาสของรายละเอียดในรายงาน
- คลาส TestRequest เป็นคลาสรายละเอียดของซอฟต์แวร์
- คลาส Summary\_Report เป็นคลาสของรายละเอียดใบสรุปผลการทดสอบ
- คลาส Team เป็นคลาสของรายละเอียดทีมที่พนักงานสังกัด
- คลาส Test\_case เป็นคลาสของกรณีทดสอบ
- คลาส Test\_Group เป็นคลาสของกลุ่มกรณีทดสอบ
- คลาส TypeDescription เป็นคลาสของประเภท ได้แก่ พนักงาน และทีม



ภาพที่ 3.4 แสดง Class Diagram ของระบบสนับสนุนการทดสอบซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 32  
 ไม่ว่าจะผิดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ความสัมพันธ์ของคลาส

คลาส TestRequest สัมพันธ์กับคลาส DevMember โดยโปรแกรมที่ส่งมาทดสอบอาจถูกพัฒนาโดยผู้ทดสอบอย่างน้อย 1 คน และ ผู้ทดสอบแต่ละคนสามารถพัฒนาซอฟต์แวร์ได้มากกว่า 1 ซอฟต์แวร์

คลาส TestRequest สัมพันธ์กับคลาส Its Member โดยโปรแกรมที่ส่งมาทดสอบอาจถูกทดสอบโดยผู้ทดสอบมากกว่า 1 คน โดยผู้ทดสอบแต่ละคนสามารถทดสอบโปรแกรมได้หลายโปรแกรม

คลาส TestRequest สัมพันธ์กับคลาส Application โดยโปรแกรมที่ส่งมาทดสอบจะถูกส่งมาทดสอบครั้งละ 1 โปรแกรม และ 1 โปรแกรมสามารถส่งมาทดสอบได้หลายครั้ง

คลาส TestRequest สัมพันธ์กับคลาส Test\_case โดยในการทดสอบแต่ละครั้งประกอบด้วยกรณีทดสอบได้หลายกรณีทดสอบ โดยกรณีทดสอบ 1 กรณีทดสอบอาจจะทดสอบได้มากกว่า 1 ครั้ง

คลาส TestRequest สัมพันธ์กับคลาส Report โดยในการทดสอบแต่ละครั้งจะมีรายละเอียดข้อมูลที่จะนำไปสร้างรายได้ 1 รายงาน และรายงาน 1 รายงานเป็นข้อมูลของการทดสอบครั้งเดียว

คลาส Emp\_pos สัมพันธ์กับคลาส Emp\_pos โดยพนักงาน 1 คน ต้องมีตำแหน่ง 1 ตำแหน่ง และใน 1 ตำแหน่งอาจมีพนักงานที่มีตำแหน่งเดียวกัน

คลาส Emp\_pos สัมพันธ์กับคลาส Team โดยที่พนักงาน 1 คนสังกัดทีมได้ 1 ทีมและในแต่ละทีมอาจมีพนักงานสังกัดอยู่ได้มากกว่า 1 คน และในทีมแต่ละทีมต้องมีพนักงานสังกัดอยู่อย่างน้อย 1 คน

คลาส Emp\_pos สัมพันธ์กับคลาส TypeDescription โดยพนักงาน 1 คนสามารถทำหน้าที่ได้ 1 หน้าที่ และ หน้าที่ 1 หน้าที่อาจมีพนักงานที่มีหน้าที่เหมือนกัน มากกว่า 1 คน

คลาส Emp\_pos สัมพันธ์กับคลาส LogOn โดยพนักงานแต่ละคนจะมีรหัสผู้ใช้ได้รหัสเดียว และรหัสผู้ใช้รหัสเดียวสามารถใช้ได้กับพนักงาน 1 คนเท่านั้น

คลาส Report สัมพันธ์กับคลาส List\_Program โดยการทดสอบแต่ละครั้งอาจมีการทดสอบโปรแกรมย่อยหลายๆ โปรแกรม และ โปรแกรมย่อยแต่ละ โปรแกรมอาจมีการส่งตรวจได้หลายครั้งและในการทดสอบแต่ละครั้งต้องประกอบด้วยโปรแกรมอย่างน้อย 1 โปรแกรม

คลาส Report สัมพันธ์กับคลาส List\_Report โดยในการทดสอบแต่ละครั้งอาจมีการตรวจสอบรายงานด้วย ซึ่งอาจมีการตรวจสอบรายงานมากกว่า 1 รายงาน และแต่ละรายงานอาจมีการส่งตรวจมากกว่า 1 ครั้งได้

คลาส Report สัมพันธ์กับคลาส Summary\_Report โดยในการทดสอบแต่ละครั้งมีการออกรายงานการสรุปครั้งเดียว และรายงานการสรุปแต่ละรายงานใช้กับการทดสอบครั้งเดียวเท่านั้น

คลาส Test\_case สัมพันธ์กับคลาส Test\_Group โดยกรณีทดสอบแต่ละกรณีทดสอบจะอยู่ในกลุ่มของกรณีทดสอบแค่กลุ่มเดียว และในกลุ่มของกรณีทดสอบแต่ละกลุ่มอาจมีกรณีทดสอบได้มากกว่า 1 กรณีทดสอบ

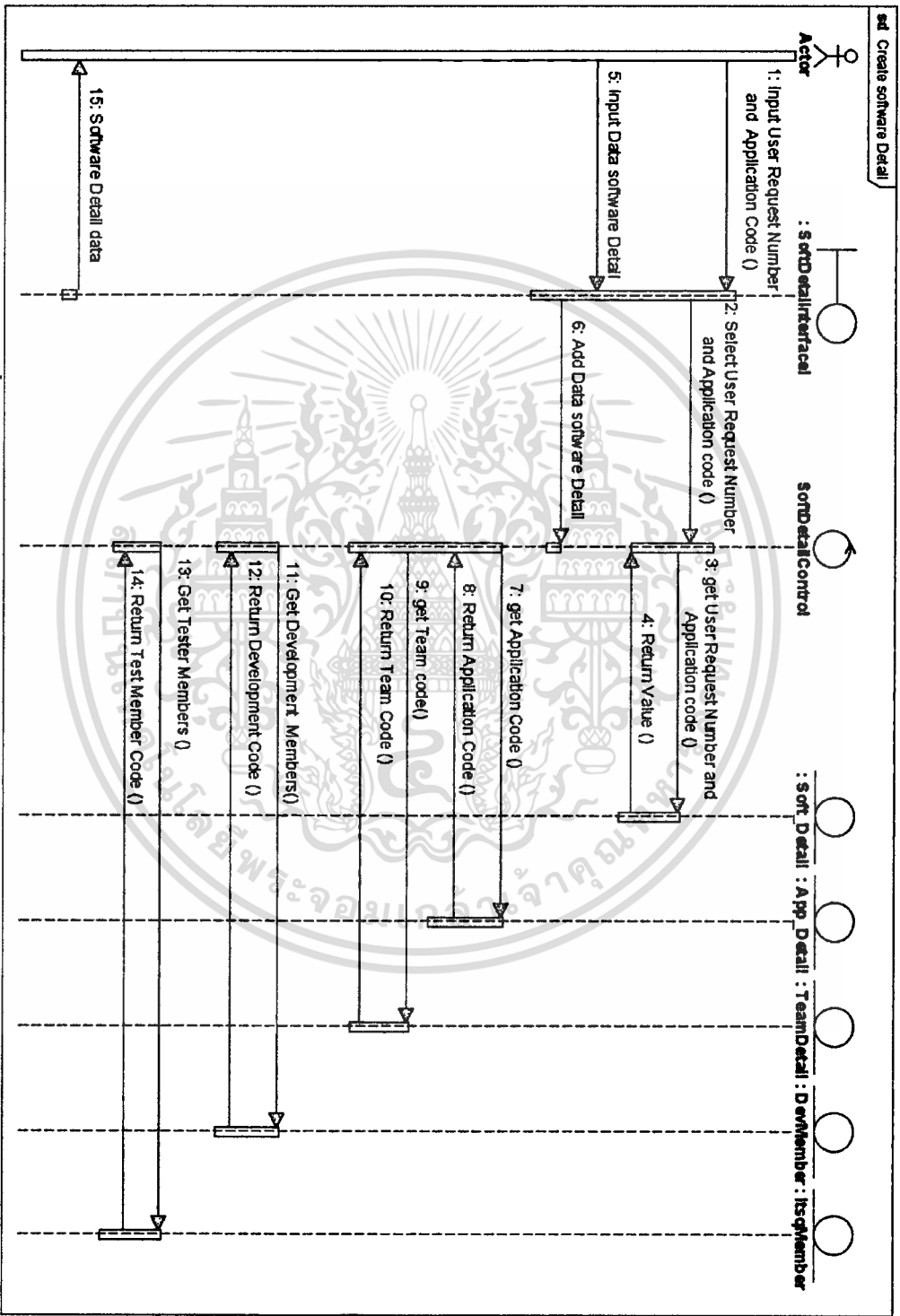
จะสังเกตได้ว่า Class Employee จะประกอบด้วย Sub Class ItsqMember และ Sub Class DevMember ทั้งผู้ทดสอบและผู้พัฒนาต่างก็เป็นพนักงานของบริษัท ส่วน Class Team เป็น Class ที่เก็บทีมของพนักงาน ดังนั้นในแต่ละทีมก็ประกอบด้วยพนักงาน (พนักงานเป็นส่วนหนึ่งของทีม) ส่วน Class TestRequest เป็นคลาสรายละเอียดของซอฟต์แวร์ที่ส่งมาทดสอบแต่ละครั้ง ดังนั้นในการทดสอบแต่ละครั้งจะประกอบด้วยโปรแกรมย่อยต่างๆ ที่ส่งมาทดสอบ และรายงานที่ได้จากโปรแกรมที่ทดสอบ จึงสรุปได้ว่าในการทดสอบซอฟต์แวร์แต่ละครั้งประกอบด้วยโปรแกรมย่อยที่ส่งมาทดสอบและรายงานที่ได้จากโปรแกรมที่ทดสอบ

### 3.3.4 Sequence Diagram

Sequence Diagram คือ ไดอะแกรมแสดงให้เห็นถึงการมีปฏิสัมพันธ์ (Interaction) ระหว่างออบเจกต์ของคลาสโดยเฉพาะการส่ง Message ระหว่างออบเจกต์ตามลำดับของเวลา (Sequence) ที่เกิดขึ้น แสดงด้วยสัญลักษณ์แสดงให้เห็นลำดับการส่งอย่างชัดเจน ประกอบด้วยออบเจกต์และคลาสทั้งสามส่วน ได้แก่ Boundary Class, Control Class และ Entity Class

#### 3.3.4.1 Sequence Diagram: Create software Detail

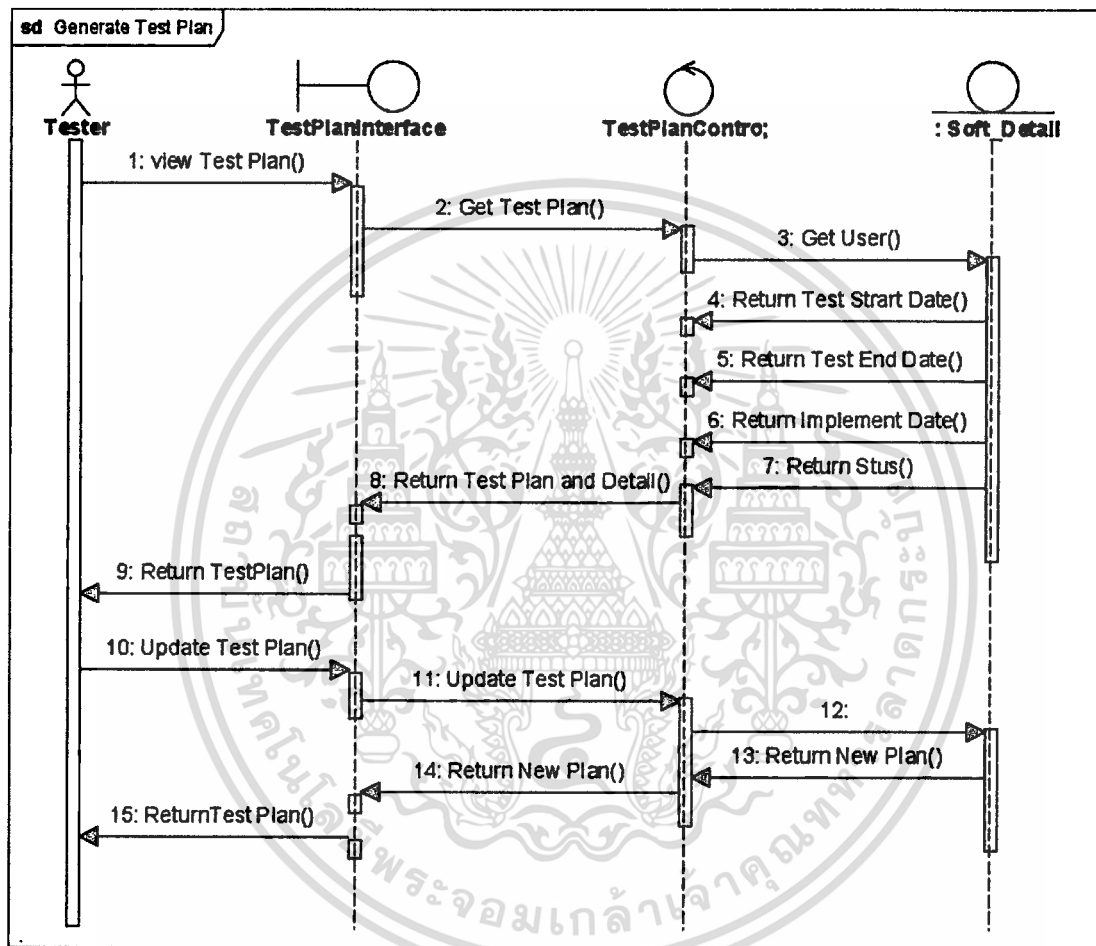
เป็นขั้นตอนของการบันทึกรายละเอียดเกี่ยวกับซอฟต์แวร์ที่จะทำการทดสอบ โดยผู้ทดสอบระบุ User Request Number ผ่านทางหน้าจอ Soft Detail Interface ส่วนของการ Interface ส่งข้อมูลให้ส่วน Soft Detail Control เพื่อทำการตรวจสอบว่า User Request ที่ระบุมีอยู่ในระบบหรือไม่ หากยังไม่มีก็อนุญาตให้เพิ่มข้อมูลของซอฟต์แวร์ที่ต้องการทดสอบได้ แต่หากว่ามี User Request และ ระบบที่ต้องการทดสอบอยู่ในระบบแล้ว โปรแกรมจะไม่อนุญาตให้เพิ่มข้อมูลซ้ำ หากระบบอนุญาตให้เพิ่มข้อมูลผู้ทดสอบ ผู้ทดสอบต้องเพิ่มข้อมูล โดยการเลือก โปรแกรมที่ต้องการทดสอบ โดยดึงค่าจาก Entity Application ข้อมูลทีมของผู้ทดสอบและผู้พัฒนาจาก Entity Team ข้อมูลผู้พัฒนาจาก Entity DevMember และข้อมูลผู้ทดสอบ จาก Entity ItsqMember จากนั้นข้อมูลรายละเอียดของซอฟต์แวร์จะถูกบันทึกใน Entity Soft\_Detail



ภาพที่ 3.5 แสดง Sequence Diagram: Create software Detail

### 3.3.4.2 Sequence Diagram: Generate Test Plan

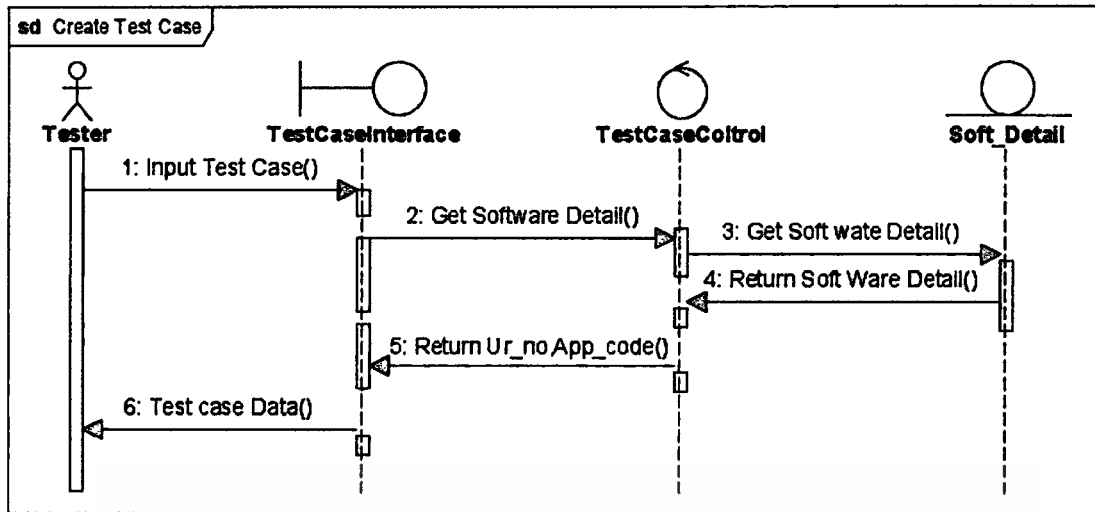
เป็นขั้นตอนของสร้างแผนการทดสอบ ผ่านทางหน้าจอ TestPlanInterface และ Control โดยเมื่อผู้ทดสอบ ต้องการดูแผนการทดสอบ ระบบ จะ ดึงข้อมูล User Request Number วันที่เริ่มการทดสอบ วันที่สิ้นสุดการทดสอบ จาก Entity TestRequest มาแสดงให้เห็นเฉพาะระบบที่ผู้ทดสอบทำการทดสอบเท่านั้น



ภาพที่ 3.6 แสดง Sequence Diagram: Generate Test Plan

### 3.3.4.3 Sequence Diagram: Create Test Case

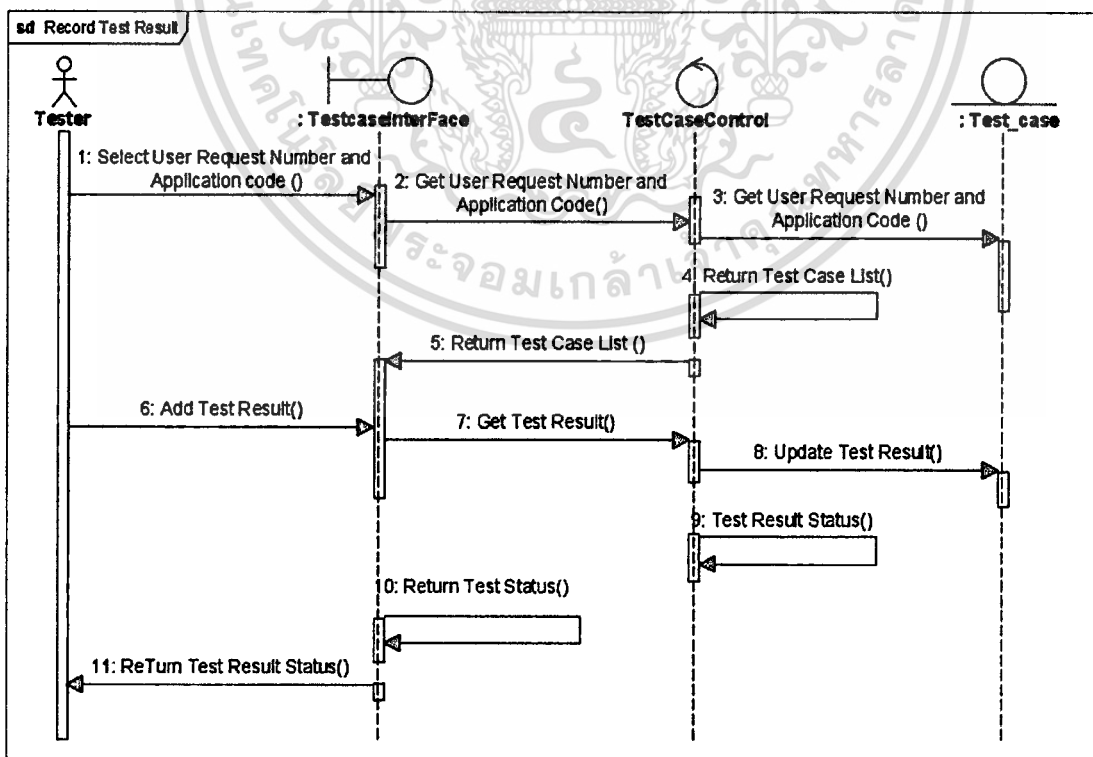
เป็นขั้นตอนของสร้างออกแบบกรณีทดสอบเพื่อจัดเก็บไว้ในระบบ โดยผู้ทดสอบกำหนดว่าจะบันทึกแผนการทดสอบของ User Request Number ไหน และระบบอะไร โดยการเรียกดูข้อมูลจาก Entity Soft\_Detail



ภาพที่ 3.7 แสดง Sequence Diagram: Create Test Case

### 3.3.4.4 Diagram: Record Test Result

เป็นขั้นตอนของบันทึกผลการทดสอบโดยทำการทดสอบตามกรณีทดสอบที่ได้มีการออกแบบไว้ก่อนหน้านี้ ในขั้นตอนนี้ผู้ทดสอบสามารถเพิ่มกรณีทดสอบเพื่อทำการทดสอบเพิ่มได้ หากพบว่ากรณีทดสอบที่ได้ออกแบบไว้แล้ว ไม่ครอบคลุมทุกกรณีที่จะเกิดขึ้นได้จากการทำงานของซอฟต์แวร์ที่ทดสอบ

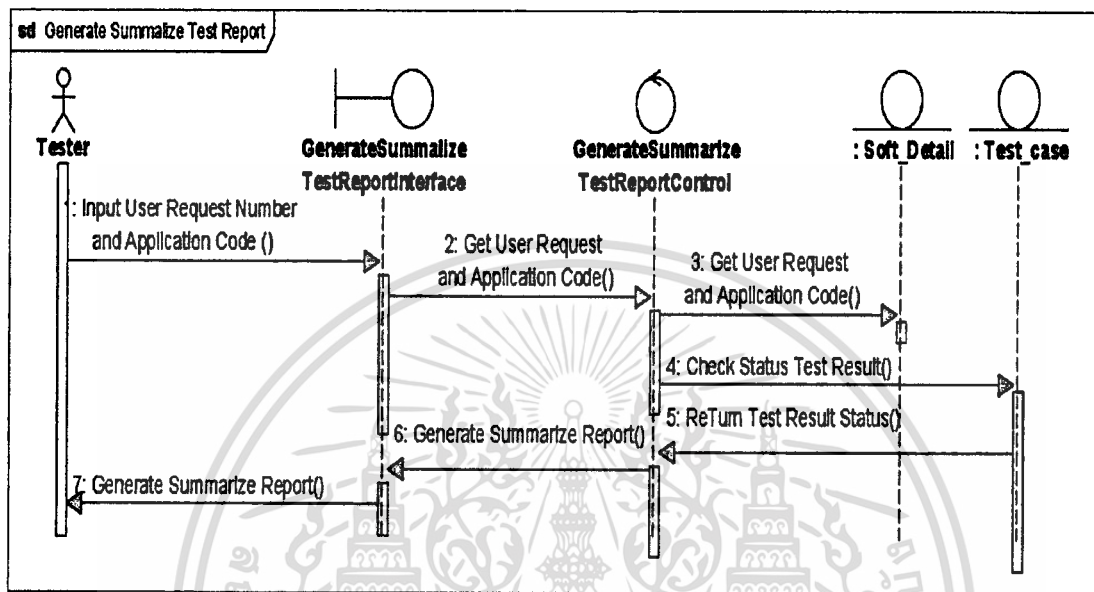


ภาพที่ 3.8 แสดง Sequence Diagram: Record Test Result

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.4.5 Sequence Diagram : Generate Summarize Test Report

เป็นขั้นตอนสร้างรายงานสรุปผลการทดสอบ เมื่อมีการทดสอบซอฟต์แวร์เสร็จ โดยมีผู้ทดสอบสามารถเรียกใช้หน้าจอได้โดยการใช้ User Request Number เป็นคีย์ในการตรวจสอบว่ามีข้อมูลของซอฟต์แวร์ที่ทดสอบแล้วและต้องการออกรายงานสรุปผลการทดสอบหรือไม่



ภาพที่ 3.9 แสดง Sequence Diagram : Generate Summarize Test Report



ตารางที่ 3.7 เอนทิตีระบบสนับสนุนการทดสอบซอฟต์แวร์

| ลำดับที่ | เอนทิตี         | คำอธิบายเอนทิตี                                    |
|----------|-----------------|--|
| 1        | Application     | เอนทิตีรายละเอียด โปรแกรมที่ทดสอบ                  |
| 2        | DevMember       | เอนทิตีสมาชิกผู้พัฒนาโปรแกรมที่ส่งมาทดสอบ          |
| 3        | Emp_pos         | เอนทิตีตำแหน่งของพนักงาน                           |
| 4        | Employee        | เอนทิตีพนักงานรวมถึง ผู้ทดสอบ ผู้พัฒนาและผู้ใช้งาน |
| 5        | ItsqMember      | เอนทิตีสมาชิกผู้ทดสอบ                              |
| 6        | List_Program    | เอนทิตีโปรแกรมย่อยที่ส่งมาทดสอบ                    |
| 7        | List_Report     | เอนทิตีรายงานที่ส่งมาทดสอบ                         |
| 8        | LogOn           | เอนทิตีเก็บข้อมูลการเข้าระบบ                       |
| 9        | Report          | เอนทิตีเกี่ยวกับรายละเอียดของรายงาน                |
| 10       | TestRequest     | เอนทิตีเกี่ยวกับรายละเอียดของซอฟต์แวร์             |
| 11       | summary_Report  | เอนทิตีเกี่ยวกับรายละเอียดของใบสรุปผลการทดสอบ      |
| 12       | Team            | เอนทิตีเกี่ยวกับรายละเอียดทีมที่พนักงานสังกัด      |
| 13       | Test_case       | เอนทิตีกรณีทดสอบ                                   |
| 14       | Test_Group      | เอนทิตีกลุ่มกรณีทดสอบ                              |
| 15       | TypeDescription | เอนทิตีเก็บประเภท ได้แก่ พนักงาน และทีม            |

จากเอนทิตีของระบบสนับสนุนของฝ่ายควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศ ที่ได้แสดงไว้ตามตารางข้างต้นนั้น เอนทิตีเหล่านี้มีความสัมพันธ์กัน โดยจะแสดงความสัมพันธ์ของเอนทิตีเหล่านี้ผ่านอีอาร์ไออะแกรม ซึ่งมีรายละเอียด ตามภาพที่ 4.1

### 3.6 พจนานุกรมข้อมูล

จากอีอาร์ไออะแกรมที่ได้นั้นสามารถแปลงเอนทิตีให้เป็นฐานข้อมูลเชิงสัมพันธ์ได้ โดยนำเสนอผ่านพจนานุกรมข้อมูลที่จะแสดงให้เห็นรายละเอียดของข้อมูลต่างๆที่เกี่ยวข้องกับการทำงานของระบบ ซึ่งได้เป็นตารางที่มีความสัมพันธ์กันทั้งหมด 15 ตาราง ดังรายการตามตารางที่ 3.7

ตารางที่ 3.8 รายละเอียดของเอนทิตี Application

| ชื่อแอททริบิวต์ | คำอธิบาย                   | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|----------------------------|------------|---------|------|--------------|
| APP_Code        | รหัสระบบที่ทำการทดสอบ      | int        |         | PK   |              |
| Short_Name      | ชื่อย่อระบบที่ทดสอบ        | nvarchar   | 300     |      |              |
| App_NameEng     | ชื่อภาษาอังกฤษระบบที่ทดสอบ | nvarchar   | 300     |      |              |
| App_NameThai    | ชื่อภาษาไทยระบบที่ทดสอบ    | nvarchar   | 300     |      |              |
| App_Remark      | หมายเหตุ                   | nvarchar   | 300     |      |              |

ตารางที่ 3.9 รายละเอียดของเอนทิตี DevMember

| ชื่อแอททริบิวต์ | คำอธิบาย                        | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|---------------------------------|------------|---------|------|--------------|
| UR_No           | User Request Number             | nvarchar   | 8       | PK   |              |
| AppCode         | รหัสระบบที่ทำการทดสอบ           | int        |         | FK   | Application  |
| Dev_Members     | รหัสพนักงานผู้พัฒนาระบบที่ทดสอบ | nvarchar   | 5       | FK   | Employee     |

ตารางที่ 3.10 รายละเอียดของเอนทิตี Emp\_pos

| ชื่อแอททริบิวต์ | คำอธิบาย           | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|--------------------|------------|---------|------|--------------|
| Emp_PosCode     | รหัสตำแหน่งพนักงาน | int        |         | PK   |              |
| Emp_PosName     | ชื่อตำแหน่งพนักงาน | nvarchar   | 200     |      |              |

ตารางที่ 3.11 รายละเอียดของเอนทิตี Employee

| ชื่อแอททริบิวต์ | คำอธิบาย           | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|--------------------|------------|---------|------|--------------|
| Emp_id          | รหัสพนักงาน        | nvarchar   | 5       | PK   |              |
| Emp_First_Name  | ชื่อพนักงาน        | nvarchar   | 100     |      |              |
| Emp_Last_Name   | นามสกุลพนักงาน     | nvarchar   | 100     |      |              |
| Emp_pos_code    | รหัสตำแหน่งพนักงาน | int        |         | FK   | Emp_pos      |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 (ต่อ)

| ชื่อแอททริบิวต์ | คำอธิบาย                | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง    |
|-----------------|-------------------------|------------|---------|------|-----------------|
| Emp_Tel         | เบอร์โทรศัพท์พนักงาน    | nvarchar   | 10      |      |                 |
| Emp_Email       | E-Mail พนักงาน          | nvarchar   | 100     |      |                 |
| Emp_Type        | รหัสประเภทพนักงาน       | int        |         | FK   | TypeDescription |
| Emp_TeamCode    | รหัสทีมที่พนักงานสังกัด | int        |         |      |                 |

ตารางที่ 3.12 รายละเอียดของเอนทิตี ItsqMember

| ชื่อแอททริบิวต์ | คำอธิบาย              | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|-----------------------|------------|---------|------|--------------|
| UR_No           | User Request Number   | nvarchar   | 8       | PK   |              |
| App_Code        | รหัสระบบที่ทำการทดสอบ | int        |         |      |              |
| Itsq_Members    | รหัสพนักงานผู้ทดสอบ   | nvarchar   | 5       | PK   |              |

ตารางที่ 3.13 รายละเอียดของเอนทิตี List\_Program

| ชื่อแอททริบิวต์ | คำอธิบาย                         | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|----------------------------------|------------|---------|------|--------------|
| Rep_code        | เลขที่หนังสือออก                 | nvarchar   | 10      | PK   |              |
| Pro_Name        | ชื่อ โปรแกรมที่ตรวจ              | nvarchar   | 100     |      |              |
| Pro_Function    | หน้าที่การทำงานของโปรแกรม        | nvarchar   | 100     |      |              |
| Pro_spec        | Programspec<br>0: ไม่มี<br>1: มี | nvarchar   | 3       |      |              |
| Ur_no           | User Request Number              | nvarchar   | 8       | PK   |              |

ตารางที่ 3.14 รายละเอียดของเอนทิตี List\_Report

| ชื่อแอททริบิวต์ | คำอธิบาย            | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|---------------------|------------|---------|------|--------------|
| Rep_code        | เลขที่หนังสือออก    | nvarchar   | 10      | PK   |              |
| Rep_Id          | เลขที่รายงานที่ตรวจ | nvarchar   | 50      |      |              |
| Rep_Name        | ชื่อรายงานที่ตรวจ   | nvarchar   | 100     |      |              |

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์ ไม่อนุญาตให้ทำไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.14 (ต่อ)

| ชื่อแอททริบิวต์ | คำอธิบาย                           | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|------------------------------------|------------|---------|------|--------------|
| Rep_Function    | หน้าที่การทำงานของรายงาน           | nvarchar   | 100     |      |              |
| sample          | sample Report<br>0: ไม่มี<br>1: มี | nvarchar   | 3       |      |              |

ตารางที่ 3.15 รายละเอียดของเอนทิตี LogOn

| ชื่อแอททริบิวต์ | คำอธิบาย   | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|------------|------------|---------|------|--------------|
| User_Id         | รหัสผู้ใช้ | nvarchar   | 5       | PK   |              |
| Password        | รหัสผ่าน   | nvarchar   | 10      |      |              |

ตารางที่ 3.16 รายละเอียดของเอนทิตี Reportl

| ชื่อแอททริบิวต์             | คำอธิบาย                                   | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------------------|--|------------|---------|------|--------------|
| Rep_code                    | เลขที่หนังสือออก                           | nvarchar   | 10      | PK   |              |
| Create_Date                 | วันที่ออกรายงาน                            | datetime   |         |      |              |
| Program_Result              | ผลการตรวจค้นโปรแกรม                        | nvarchar   | 100     |      |              |
| Doc_Control                 | ผลการตรวจค้นเอกสารระบบงาน                  | nvarchar   | 100     |      |              |
| Internal_Control            | ผลการตรวจค้นการควบคุมภายใน                 | nvarchar   | 50      |      |              |
| Objective                   | วัตถุประสงค์ของการปรับปรุงโปรแกรม          | nvarchar   | MAX     |      |              |
| Limit                       | ขอบเขตการตรวจ                              | nvarchar   | MAX     |      |              |
| Test_Data_Method            | วิธีการตรวจ<br>Test Data Method            | nvarchar   | MAX     |      |              |
| Base_Case_System_Evaluation | วิธีการตรวจ<br>Base Case System Evaluation | nvarchar   | MAX     |      |              |
| Result_Program              | ผลการตรวจ ค้นโปรแกรม                       | nvarchar   | MAX     |      |              |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนูญได้เห็นไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.16 (ต่อ)

| ชื่อแอททริบิวต์   | คำอธิบาย  | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-------------------|---|------------|---------|------|--------------|
| Result_Test       | ผลการตรวจ ด้านการทดสอบ                          | nvarchar   | MAX     |      |              |
| System_Doc        | ผลการตรวจด้านเอกสารระบบงาน<br>SYSTEM DOCUMENT   | nvarchar   | 10      |      |              |
| Program_Doc       | ผลการตรวจ ด้านเอกสารระบบงาน<br>PROGRAM DOCUMENT | nvarchar   | 10      |      |              |
| Operation_Doc     | ผลการตรวจ ด้านเอกสาร<br>OPERATION DOCUMENT      | nvarchar   | 10      |      |              |
| User_Doc          | ผลการตรวจ ด้านเอกสารระบบงาน                     | nvarchar   | 10      |      |              |
| Internal_control2 | ผลการตรวจ ด้านการควบคุมภายใน                    | nvarchar   | MAX     |      |              |

ตารางที่ 3.17 รายละเอียดของเอนทิตี TestRequest

| ชื่อแอททริบิวต์    | คำอธิบาย                  | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|--------------------|---------------------------|------------|---------|------|--------------|
| UR_No              | User Request Number       | nvarchar   | 8       | PK   |              |
| App_Code           | รหัสระบบที่ทำการทดสอบ     | int        |         | FK   | Application  |
| Soft_NoTest        | ครั้งที่ทดสอบ             | nvarchar   | 10      |      |              |
| ITS_Startdatetime  | วันที่เริ่มดำเนินการพัฒนา | datetime   |         |      |              |
| ITS_Enddatetime    | วันที่สิ้นสุดการพัฒนา     | datetime   |         |      |              |
| ITS_ActualManday   | จำนวนวันที่ใช้ในการพัฒนา  | datetime   |         |      |              |
| ITSQ_Startdatetime | วันที่เริ่มดำเนินการทดสอบ | datetime   |         |      |              |
| ITSQ_Enddatetime   | วันที่สิ้นสุดการทดสอบ     | datetime   |         |      |              |
| ITSQ_ActualManday  | จำนวนวันที่ใช้ในการทดสอบ  | datetime   |         |      |              |
| Implement_datetime | วันที่ติดตั้ง             | datetime   |         |      |              |
| Remark             | หมายเหตุ                  | nvarchar   | 300     |      |              |
| Status             | สถานะการทดสอบ             | int        |         |      |              |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.18 รายละเอียดของเอนทิตี summary

| ชื่อแอททริบิวต์ | คำอธิบาย  | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|---|------------|---------|------|--------------|
| Rep_code        | เลขที่หนังสือออก  | nvarchar   | 10      | PK   |              |
| No_inspect      | จำนวนครั้งที่ทดสอบ<br>เช่น 1 ครั้งเมื่อทดสอบแล้ว<br>ไม่พบข้อผิดพลาด | int        |         |      |              |
| Pro_result      | ผลการตรวจค้นโปรแกรม   | int        |         |      |              |
| Delivery_status | ผลการตรวจความพร้อม<br>ในการส่งโปรแกรมมาให้<br>ทดสอบ                 | int        |         |      |              |
| System_Flow     | SYSTEM FLOWCHART  | int        |         |      |              |
| List_file       | LIST OF FILES<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ                | int        |         |      |              |
| File_desc       | FILE DESCRIPTION<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ             | int        |         |      |              |
| File_layout     | FILE RECORD LAYOUT<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ           | int        |         |      |              |
| Record_des      | RECORD DESCRIPTION<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ           | int        |         |      |              |
| Code_desc       | CODE DESCRIPTION<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ             | int        |         |      |              |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.18 (ต่อ)

| ชื่อแอททริบิวต์  | คำอธิบาย  | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|------------------|---|------------|---------|------|--------------|
| Program_abs      | PROGRAM ABSTRACT<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ | int        |         |      |              |
| List_Program     | LIST OF PROGRAM<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ  | int        |         |      |              |
| List_Report      | LIST OF REPORT<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ   | int        |         |      |              |
| Sample_Report    | SAMPLE REPORT<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ    | int        |         |      |              |
| User_manual      | USER MANUAL<br>0 : มี<br>1 : ไม่มี<br>2 : ไม่กระทบ      | int        |         |      |              |
| Internal_control | การควบคุมภายในของระบบงาน                                | int        |         |      |              |

ตารางที่ 3.19 รายละเอียดของเอนทิตี Team

| ชื่อแอททริบิวต์ | คำอธิบาย                | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|-------------------------|------------|---------|------|--------------|
| Team_code       | รหัสทีมที่พนักงานสังกัด | int        |         | PK   |              |
| Team_Name       | ชื่อทีมที่พนักงานสังกัด | nvarchar   | 200     |      |              |
| Team_Location   | ที่ตั้งของทีม           | nvarchar   | 200     |      |              |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.20 รายละเอียดของเอนทิตี Test case

| ชื่อแอททริบิวต์ | คำอธิบาย                 | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|--------------------------|------------|---------|------|--------------|
| Test_id         | รหัสกรณีทดสอบ            | int        |         | PK   |              |
| Test_Group      | รหัสกลุ่มกรณีทดสอบ       | int        |         | FK   | Test_Groupid |
| caseName        | ชื่อกรณีทดสอบ            | nvarchar   | 300     |      |              |
| Expected        | ผลทดสอบที่คาดว่าจะได้รับ | nvarchar   | 300     |      |              |
| Results         | ผลการทดสอบ               | nvarchar   | 300     |      |              |
| Remark          | หมายเหตุ                 | nvarchar   | 300     |      |              |
| Test_datetime   | วันที่ทดสอบ              | datetime   |         |      |              |
| Status          | สถานะการทดสอบ            | nvarchar   | 300     |      |              |

ตารางที่ 3.21 รายละเอียดของเอนทิตี Test Group

| ชื่อแอททริบิวต์ | คำอธิบาย           | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|--------------------|------------|---------|------|--------------|
| Test_GroupId    | รหัสกลุ่มกรณีทดสอบ | int        |         | PK   |              |
| Test_Group_name | ชื่อกลุ่มกรณีทดสอบ | nvarchar   | MAX     |      |              |

ตารางที่ 3.22 รายละเอียดของเอนทิตี TypeDescription

| ชื่อแอททริบิวต์ | คำอธิบาย   | ชนิดข้อมูล | ความยาว | คีย์ | ตารางอ้างอิง |
|-----------------|------------|------------|---------|------|--------------|
| TypeID          | ประเภท     | int        |         | PK   |              |
| TypeName        | ชื่อประเภท | nvarchar   | 10      |      |              |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การพัฒนาระบบงาน

การออกแบบส่วนต่อประสานงานกับผู้ใช้ (User Interface Design) เป็นการออกแบบส่วนการใช้งานระหว่างผู้ใช้งานกับระบบ จะมุ่งเน้นเรื่องการปฏิสัมพันธ์ระหว่างผู้ใช้งานกับคอมพิวเตอร์ ดังนั้นการออกแบบ ส่วนต่อประสานงานกับผู้ใช้ต้องได้รับการออกแบบให้เหมาะสม คือ การเลือกใช้คำสั่งต่างๆ บนจอภาพสามารถสื่อสารเข้าใจง่าย เมื่อผู้ใช้งานมีการโต้ตอบกับระบบระบบควรตอบสนองความต้องการแก่ผู้ใช้งานอย่างมีประสิทธิภาพ การออกแบบส่วนต่อประสาน ระหว่างผู้ใช้งานกับคอมพิวเตอร์ โดยมีวัตถุประสงค์หลักคือ สามารถใช้งานได้ง่าย และเรียนรู้ได้ง่าย ในการพัฒนาระบบสารสนเทศนั้น การออกแบบส่วนติดต่อกับผู้ใช้งานถือว่าเป็นขั้นตอนที่มีความสำคัญขั้นต้นหนึ่ง ไม่ว่าจะเป็นการออกแบบจอภาพเพื่อติดต่อกับผู้ใช้งานหรือการออกแบบรายงานซึ่งถือเป็นส่วนที่ช่วยแสดงผลการทำงานผ่านทางระบบที่พัฒนาขึ้นมาได้เป็นอย่างดี การที่ผู้ใช้ระบบงานนั้นจะสามารถใช้งานระบบสารสนเทศได้ง่ายและสะดวกขึ้นอยู่กับ การออกแบบจอภาพและรายงานที่คิดของผู้ออกแบบ ทั้งนี้ระบบสนับสนุนการทดสอบซอฟต์แวร์เป็นระบบงานแบบแอปพลิเคชันที่พัฒนาระบบงานด้วยโปรแกรม Visual Studio.NET 2005 โดยติดต่อกับฐานข้อมูล Microsoft SQL Server 2005 การออกแบบเน้นการสร้างส่วนติดต่อกับผู้ใช้งานช่วยให้ผู้ใช้ระบบทำงานได้สะดวกและเข้าใจการทำงานของระบบได้อย่างรวดเร็วขึ้น โดยมี หน้าจอหลักของระบบ ดังแสดงในภาพที่ 4.1 เป็นหน้าจอหลักของผู้ทดสอบแต่ละคนซึ่งสามารถมองเห็นงานที่ตนเองรับผิดชอบเท่านั้น

#### 4.1 ส่วนประกอบของระบบ

หลังจากผ่านการวิเคราะห์ระบบการบริหารจัดการคลังโดยผู้ขายแล้ว พบว่าระบบควรจะมีการออกแบบส่วนติดต่อกับผู้ใช้งานตามลำดับขั้นตอนการทำงานของระบบ โดยจะแสดงเป็นเมนูการทำงานของระบบเพื่อเข้าสู่การใช้งานในส่วนต่างๆดังต่อไปนี้ ในส่วนหน้าจอหลักประกอบด้วยเมนูหลักๆ 3 เมนู และแต่ละเมนูหลักก็จะมีเมนูย่อยๆเพื่อแสดงการเข้าถึงโปรแกรมแต่ละตัว

## เมนูแสดงการทำงานของระบบสนับสนุนการทดสอบซอฟต์แวร์

### 1. ผู้ทดสอบ

- กำหนดรายละเอียดการทดสอบ
- เพิ่มข้อมูลรายงานการทดสอบ
- ปรับปรุงระยะเวลาการทดสอบ
- ข้อมูลสรุปผลการทดสอบ
- กรณีทดสอบ

### 2. ผู้ดูแลระบบ

- ข้อมูลพนักงาน
- รายละเอียดซอฟต์แวร์
- ข้อมูลผู้ใช้งาน
- รายละเอียดทีม

### 3. รายงาน

- ใบกำหนดระยะเวลาการทดสอบ
- รายงานการตรวจคุณภาพระบบงาน
- ใบสรุปผลการทดสอบ
- เทคโนโลยีสารสนเทศ

## 4.2 การออกแบบจอภาพ

### 4.2.1 กำหนดรายละเอียดการทดสอบ – รายละเอียดซอฟต์แวร์

เป็นหน้าจอส่วนการกำหนดรายละเอียดของซอฟต์แวร์ที่จะทำการทดสอบ โดยผู้ทดสอบระบบ

- User Request Number
- ระบบที่ทำการทดสอบ
- ครั้งที่ทดสอบ
- รายละเอียดด้านการพัฒนา ผู้พัฒนาสามารถระบุได้มากกว่า 1 คน แต่ผู้พัฒนาต้องสังกัดอยู่ที่ทีมเดียวกัน วันเริ่มต้นและสิ้นสุดของการพัฒนา จำนวนวันที่ใช้ในการพัฒนา ทีมพัฒนา
- รายละเอียดทีมทดสอบ ผู้ทดสอบซึ่งสามารถระบุได้มากกว่า 1 คน โดยผู้ทดสอบสังกัดอยู่ที่ทีมเดียวกัน วันเริ่มต้นสิ้นสุดในการทดสอบ จำนวนวันที่ใช้ในการทดสอบ ทีมทดสอบ
- วันที่ติดตั้ง
- สถานะของการทดสอบ
- หมายเหตุ

32382 บอร์ด จันทรัตน์

รายละเอียดของโฟลเดอร์

User Request: 51700022

รหัสโครงการ: PRIMARY MARKET SYSTEM

ครั้งที่ทดสอบ: 2009 / 3

เรื่อง: - การปรับปรุงโปรแกรมเกี่ยวกับกรทำงานในข้อมูลตามการจำหน่ายและปิดยอดสภานับครุฑพระเทพนิมิต 0 พ.ศ. 2551 ครึ่งที่ 1, 2, 3 และ 4, 5 และ 6, 7 ปีที่ส่งไปยังกรมการขนส่งทางบก

| ข้อมูลการพัฒนา (Development) | ข้อมูลการทดสอบ (Soft Quality Assurance) |
|------------------------------|---|
| วันเริ่มต้น: 20/03/2009      | วันเริ่มต้น: 03/04/2009                 |
| วันสิ้นสุด: 29/03/2009       | วันสิ้นสุด: 04/04/2009                  |
| จำนวนวัน: 9.00 วัน           | จำนวนวัน: 2.00 วัน                      |

| เลขที่หนังสือส่งตรวจ | เลขที่หนังสือออก    |
|----------------------|---------------------|
| ที่ บพส. 1-02-51004  | ที่ บพส. 2-4-510007 |
| ลงวันที่: 07/04/2009 |                     |

ผู้ใช้งาน: สมศักดิ์ นนทิกานันท์

วันที่ติดตั้ง: 07/04/2009 00:00:00

สถานะ:

- ยังไม่ได้รับมอบหมาย
- กำลังดำเนินการ
- ทดสอบเสร็จพร้อมนำออกใช้งานได้
- ติดปัญหาส่งกลับเพื่อทำการแก้ไขและตรวจสอบใหม่

บันทึกการแก้ไขบนระบบและบันทึกลงโฟลเดอร์

ข้อมูลสรุปผลการทดสอบ

ภาพที่ 4.1 หน้าจอรายละเอียดการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2.2 ปรับปรุงระยะเวลาการทดสอบ

ผู้ทดสอบสามารถปรับปรุงระยะเวลาการทดสอบได้จากหน้าจอหลัก คือ หน้าจอแผนการทดสอบ

32382 ฤกษ์ จันทรรัตน์

การวางแผนการทดสอบของโครงการคอมพิวเตอร์ จักรวรรดิ

กำหนดวันที่ต้องการดูรายงานการทดสอบ

วันที่เริ่มนับ: 01/01/2009 MM/DD/YYYY

วันที่สิ้นสุด: 03/30/2009 MM/DD/YYYY

OK

แผนการทดสอบของโครงการ

| ระบบ  | ครั้งที่ทดสอบ | User Request | วันที่เริ่มต้นการทดสอบ | วันที่สิ้นสุดการทดสอบ | วันที่ต้อง |
|-------|---------------|--------------|------------------------|-----------------------|------------|
| PMS   | 2009/3        | 51020022     | Apr 3 200              | Apr 4 200             | Apr 7 200  |
| BCB I | 2009/3        | 51040056     | Mar 10 200             | Mar 12 200            | Mar 6 200  |
| AIS   | 2009/100      | 22223337     | Mar 11 200             | Mar 18 200            | Mar 20 200 |
| RGT   | 2009/1        | 35102007     | Mar 14 200             | Mar 14 200            | Mar 15 200 |
| PMS   | 2009/1        | 51010034     | Mar 16 200             | Mar 20 200            | Mar 21 200 |
| AIS   | 2009/22       | 45445549     | Mar 18 200             | Mar 25 200            | Mar 7 200  |

รายการทดสอบทั้งหมด

ภาพที่ 4.2 หน้าจอแผนการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.2.3 กรณีทดสอบ

เป็นหน้าจอการบันทึกกรณีทดสอบที่ผู้ทดสอบได้ออกแบบไว้ โดยผู้ทดสอบระบุ

- User Request Number ต้องผ่านการบันทึกรายละเอียดของการทดสอบมาก่อน
- ระบบที่ทำการทดสอบ
- ครั้งที่ทดสอบ
- รหัสกรณีทดสอบ
- ชื่อกรณีทดสอบ
- ผลที่คาดว่าจะได้รับ
- ผลการทดสอบ
- วันที่ทำการทดสอบ
- หมายเหตุ
- สถานะการทดสอบ

32382 บาท จำนวน

กรณีทดสอบ (Test Case)

User Request: 22223337  
 รหัสโปรแกรม: PRIMARY MARKET SYSTEM  
 รหัสสารคดีทดสอบ: PMS-167  
 Test Group:   
 ชื่อกรณีทดสอบ: ทดสอบจำนวน Amount เป็นเลขทศนิยม 299.49666  
 ผลที่คาดว่าจะได้รับ: บันทึกเป็น Amount เป็นเลขทศนิยม 299.50  
 ผลการทดสอบ:   
 หมายเหตุ:   
 สถานะการทดสอบ: กำลังทดสอบ

| Test Id | Test Group                 | Case  | Expect Result   | Results   | Status   |
|---------|----------------------------|---|---|---|--|
| PMS-    |                            | ทดสอบการตั้งข้อมูลอัตราดอกเบี้ยจากแหล่งข้อมูลใหม่ | ดึงข้อมูลจาก Table TMS_rate   |   | กำลังทดสอบ <input type="button" value="Select"/> |
| PMS-3   |                            | ทดสอบการแสดงผลข้อมูลจุดตัดที่มี 5 ตำแหน่ง         | สามารถแสดงผลจุดตัดที่มี 5 ตำแหน่งโดยไม่มีการปิดเศษ  |   | กำลังทดสอบ <input type="button" value="Select"/> |
| PMS-56  | ตรวจสอบรูปแบบของแฟ้มข้อมูล | กดปุ่ม ยังไม่ทำการโอนเงินทุกรายการ                | แสดง Msg "เปลี่ยนสถานะเป็นยังไม่ทำการโอนเงินทุกรายการเรียบร้อยแล้ว" และแก้ไขสถานะรายการใน Table CEF_TRANS | แสดง Msg "เปลี่ยนสถานะเป็นยังไม่ทำการโอนเงินทุกรายการเรียบร้อยแล้ว" และแก้ไขสถานะรายการใน Table CEF_TRANS | กำลังทดสอบ <input type="button" value="Select"/> |

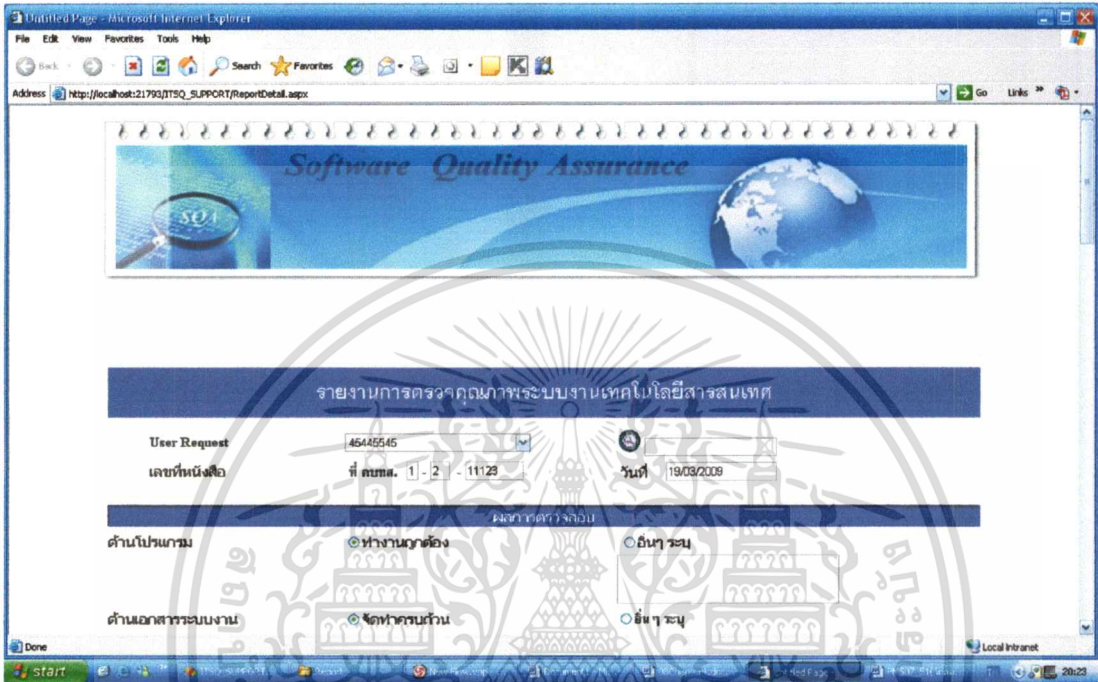
ภาพที่ 4.3 หน้าจอกรณีทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.4 เพิ่มข้อมูลรายงานการทดสอบ

เป็นหน้าจอการเพิ่มรายละเอียดของข้อมูลที่จะต้องนำไปใช้ในการสร้างรายงานต่างๆ ในระบบ ผู้ทดสอบต้องเลือก User Request Number ดังนั้นการที่จะออกรายงานได้ ผู้ทดสอบต้องบันทึกรายละเอียดของซอฟต์แวร์ก่อน



ภาพที่ 4.3 หน้าจอเพิ่มข้อมูลรายงานการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รายงานการตรวจคุณภาพระบบงานเทคโนโลยีสารสนเทศ**

User Request: 45445645

เลขที่หนังสือ: ที่ คสมท. 1 - 2 - 11123 วันที่: 19/03/2009

**ผลการตรวจสอบ**

ด้านโปรแกรม:  ทำงานถูกต้อง  อื่นๆ ระบุ

ด้านเอกสารระบบงาน:  จัดทำครบถ้วน  อื่นๆ ระบุ

ด้านการควบคุมภายใน:  รัดกุม  อื่นๆ ระบุ

**รายละเอียดการตรวจสอบระบบงานเทคโนโลยีสารสนเทศ**

**1. ข้อประสงค์ของการพัฒนาโปรแกรม**  
 วัตถุประสงค์เชิงปริมาณกับผลสัมฤทธิ์ของระบบสารสนเทศในประเทศไทย

**2. ขอบเขตการตรวจ**  
 ตรวจสอบการพิมพ์ที่เอกสาร

**3. วิธีการตรวจ**  
 ในการตรวจไปใช้ต่างๆ ดังนี้

3.1 Test Data Method: โดยทำการสร้างข้อมูลทดสอบขึ้นมาเพื่อทดสอบการทำงาน

3.2 Base Case System Evaluation: ตรวจสอบ Test Case จากภาพของแผนผังโครงข่าย ผู้ใช้ระบบงานและผู้ใช้ (USER) ในการตรวจหาข้อบกพร่อง

**4. ผลการตรวจ**  
 จากการตรวจโปรแกรมปรากฏดังต่อไปนี้ คือ

4.1 ด้านโปรแกรม:  โปรแกรมทำงานถูกต้อง  อื่นๆ ระบุ

4.2 ด้านการทดสอบ:  ไม่พบข้อผิดพลาดในการทดสอบ  อื่นๆ ระบุ

4.3 ด้านเอกสารระบบงาน

4.3.1 SYSTEM DOCUMENT:  จัดทำครบถ้วน  อื่นๆ ระบุ

4.3.2 PROGRAM DOCUMENT:  จัดทำครบถ้วน  อื่นๆ ระบุ

4.3.3 OPERATION DOCUMENT:  ไม่กระหน  อื่นๆ ระบุ

4.3.4 USER DOCUMENT:  ไม่กระหน  อื่นๆ ระบุ

4.4 ด้านการควบคุมภายใน

1. INPUT CONTROLS: การควบคุมภายในของระบบยังไม่เปลี่ยนแปลง

2. OUTPUT CONTROLS:

**ภาพที่ 4.5** หน้าจอรายงานการตรวจระบบงานเทคโนโลยีสารสนเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2.5 ข้อมูลสรุปผลการทดสอบ

เป็นข้อมูลสรุปผลการทดสอบเพื่อนำไปสร้างรายงานสรุปผลการทดสอบ

ในสรุปผลการตรวจระบบงานคุณภาพระบบงานเทคโนโลยีสารสนเทศ

ระบบ

เลขที่

รายงาน

จำนวนครั้ง

ที่ตรวจ

1. ผลการตรวจ

1.1 โปรแกรม  ทำงานถูกต้อง  มีข้อผิดพลาด ดังนี้

1.2 ความพร้อมในการส่งโปรแกรมมาให้ทดสอบ  พร้อม  ไม่พร้อม

2. เอกสารที่แนบ

|                    | มี                       | ไม่มี                    | ไม่ระบุ                  |
|--------------------|--------------------------|--------------------------|--------------------------|
| SYSTEM FLOWCHART   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| LIST OF FILES      | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| FILE DESCRIPTION   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| FILE RECORD LAYOUT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| RECORD DESCRIPTION | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| CODE DESCRIPTION   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| PROGRAM ABSTRACT   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| LIST OF PROGRAM    | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| LIST OF REPORT     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SAMPLE REPORT      | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| USER MANUAL        | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

3. ความถูกต้องภายในของระบบงาน  ครอบคลุม  ไม่ครอบคลุม  มีจุดบกพร่องดังต่อไปนี้

ภาพที่ 4.6 ใบสรุปการตรวจระบบงานคุณภาพระบบงานเทคโนโลยีสารสนเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2.5 ข้อมูลพนักงาน

Software Quality Assurance

ข้อมูลผู้ที่เกี่ยวข้อง

|               |                         |                   |                                |
|---------------|-------------------------|-------------------|--------------------------------|
| รหัสพนักงาน   | 32382                   | นามสกุล           | พิมพ์ดีด                       |
| ชื่อ          | บุษดี                   | สังกัด            | งานควบคุมการประเมินมาตรฐานไอที |
| ประเภทงาน     | ITSA                    |                   |                                |
| ตำแหน่ง       | เจ้าหน้าที่พัฒนาระบบงาน | เพิ่มข้อมูลสังกัด |                                |
| เบอร์โทรศัพท์ | 025441938               | E-Mail            | yuwadee.chantorn@scb.co.th     |

บันทึกการเปลี่ยนแปลง    ค้นหาข้อมูลพนักงาน    เพิ่มข้อมูลพนักงาน    ยกเลิก

ภาพที่ 4.7 ข้อมูลพนักงาน (ผู้ที่เกี่ยวข้อง)

## 4.2.6 ข้อมูลผู้ใช้งาน

Software Quality Assurance

รหัสผู้ใช้งาน

|                  |  |
|------------------|--|
| Username         |  |
| Password         |  |
| Confirm Password |  |

เพิ่มรหัสผู้ใช้งาน    ยกเลิก

ภาพที่ 4.8 ข้อมูลผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2.7 รายละเอียดทีม

The screenshot shows a web browser window displaying a page titled 'Software Quality Assurance'. The page features a header with a magnifying glass over 'SQ' and a globe. Below the header is a search bar with the text 'ข้อมูลทีม' (Team Information). The search bar contains the text 'ชื่อทีม' (Team Name) and 'ทีมใด' (Which team). There are two buttons: 'เป็นขโมยคืน' (Return as stolen) and 'ยกเลิก' (Cancel). Below the search bar is a link: 'คลิกไปเพื่อดูเป็นข้อมูลหน่วยงาน' (Click to view unit information).

The main content is a table with the following columns: 'รหัสทีม' (Team ID), 'ชื่อทีม' (Team Name), and 'ที่ตั้ง' (Location). The table lists 18 teams, each with an 'Edit' link.

| รหัสทีม | ชื่อทีม                                      | ที่ตั้ง  |      |
|---------|--|----------|------|
| 1       | ควบคุมคุณภาพระบบงานเทคโนโลยี1                | 20C      | Edit |
| 2       | ควบคุมคุณภาพระบบงานเทคโนโลยี2                | 20C      | Edit |
| 3       | ควบคุมคุณภาพระบบงานเทคโนโลยี3                | 20C      | Edit |
| 4       | ควบคุมคุณภาพระบบงานเทคโนโลยี4                | 20C      | Edit |
| 5       | ควบคุมคุณภาพระบบงานเทคโนโลยี5                | 20C      | Edit |
| 6       | ควบคุมคุณภาพระบบงานเทคโนโลยี6                | 20C      | Edit |
| 7       | ควบคุมคุณภาพระบบงานเทคโนโลยี7                | 20C      | Edit |
| 8       | ควบคุมคุณภาพระบบงานเทคโนโลยี8                | 20C      | Edit |
| 9       | ควบคุมคุณภาพระบบงานเทคโนโลยี9                | 20C      | Edit |
| 10      | ควบคุมคุณภาพระบบงานเทคโนโลยี10               | 20C      | Edit |
| 11      | ทีมพัฒนาระบบงาน Treasury                     | 14B      | Edit |
| 12      | ทีมพัฒนาระบบงาน Investment Product           | 14B      | Edit |
| 13      | ควบคุมคุณภาพระบบงานเทคโนโลยี12               | 13A      | Edit |
| 14      | ควบคุมคุณภาพระบบงานเทคโนโลยี13               |          | Edit |
| 15      | ผู้จัดการบัญชีสินทรัพย์บริหารเงิน            | 10A      | Edit |
| 16      | เลขานุการผู้จัดการบริหารการเงินและหลักทรัพย์ | ชิดลม    | Edit |
| 17      | Admin  |          | Edit |
| 18      | พนักงานด้านธนาคาร                            | BBI สีลม | Edit |

ภาพที่ 4.9 รายละเอียดทีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.3 รายงาน

### 4.3.1 ใบกำหนดระยะเวลาการทดสอบ

#### แบบฟอร์มสำหรับตอบรับแรก

เรียน <<ผู้จัดการทีมพัฒนาระบบงาน Treasury >>

ตามที่<<พัฒนาระบบงานเทคโนโลยีสารสนเทศ 1>> ได้ส่งหนังสือให้ตรวจรับโปรแกรมของระบบ AIS ที่ บพต. <<X-XX-XXXXXX>> ลงวันที่ <<1 เมษายน 2551>> เรื่อง <<การปรับปรุงโปรแกรมเพื่อรองรับเงื่อนไขใหม่ >>(UR : <<RXXXXXXXX>>) รายละเอียด ดังนี้

<<

1. ปรับปรุงโปรแกรมการแสดงผลข้อมูลการ Settlement ของ Product FI ที่ระบบ AIS
2. ปรับปรุงการ Interface ข้อมูลการ Settlement ของ Product FI จากระบบ AIS ไปยังระบบ ECB
3. ยกเลิกการ Interface ข้อมูลการ Settlement ของ Product FI จากระบบ SIERRA ไปยังระบบ ECB

>>

<<ทีมคุณภาพระบบงานเทคโนโลยีสารสนเทศ >> ได้พิจารณารายละเอียดของงานแล้ว ได้กำหนดระยะเวลาและผู้ทำการตรวจรับดังนี้

1. ระยะเวลาการตรวจรับ <<9/6/08 - 12/6/08>>

และจะสามารถสรุปผลการตรวจรับครั้งแรกในวันที่ <<12/6/08>>

2. ผู้ตรวจรับระบบงาน

<<นางสาวยุวดี จันทร์ตัน>> โทร <<1938>>

e-mail <<yuwadee.chanton@scb.co.th>>

ทั้งนี้ถ้าทีมสามารถทำการตรวจรับเสร็จก่อนกำหนด จะทำการสรุปผลให้ทราบทันที

<< \_\_\_\_\_ >>

<<ผู้จัดการทีมคุณภาพระบบงานเทคโนโลยีสารสนเทศ X>>

### 4.3.2 ใบสรุปผลการทดสอบ

ใบสรุปผลการตรวจระบบงาน  
คุณภาพระบบงานเทคโนโลยีสารสนเทศ

ระบบ ระบบ AUTOMATE INVESTMENT SYSTEM - AIS ครั้งที่ 2/2551 งาน การปรับปรุงโปรแกรมระบบงาน AUTOMATE INVESTMENT SYSTEM - AIS ให้สามารถรองรับการปฏิบัติงานของบริการ Custodian ดังนี้ รายการ Net Clearing, Pre-match, Generate PTI format (ฝาก/ถอน), Issuing corporate cheque, Assign ID and Declare สิทธิประโยชน์, Auto generate TR for GPF, Generate รายการ Add Temp-ENC และบันทึกบัญชี GL ทิมทีโอนนำส่งเช็คถูกทำ, BOT Report (UR No. : RXXXXXXXXX)

วันที่เริ่มตรวจ 21 ตุลาคม 2551 วันที่ตรวจเสร็จ 14 พฤศจิกายน 2551

จำนวนครั้งที่ตรวจ 2 ครั้ง

#### 1. ผลการตรวจ

- 1.1 โปรแกรม  /  ทำงานถูกต้อง  
 /  มีข้อผิดพลาด ดังนี้

- 1.2 ความพร้อมในการส่งโปรแกรมมาให้ทดสอบ  
 /  พร้อม  
 /  ไม่พร้อม เนื่องจาก

#### 2. เอกสารระบบงาน

|                    | มี  | ไม่มี   | ไม่ครบ  | มี               | ไม่มี   | ไม่ครบ  |
|--------------------|---|---|---|------------------|---|---|
| SYSTEM FLOWCHART   | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | PROGRAM ABSTRACT | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> |
| LIST OF FILES      | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | LIST OF PROGRAM  | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> |
| FILE DESCRIPTION   | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | LIST OF REPORT   | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> |
| FILE RECORD LAYOUT | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | SAMPLE REPORT    | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> |
| RECORD DESCRIPTION | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | USER MANUAL      | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> |
| CODE DESCRIPTION   | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> | <input type="checkbox"/> / <input type="checkbox"/> |                  |   |   |

#### 3. การควบคุมภายในของระบบงาน

- /  รัดกุม  /  ไม่เปลี่ยนแปลง  
 /  มีจุดบกพร่องดังต่อไปนี้

คุณภาพระบบงานเทคโนโลยีสารสนเทศ  
ผู้จัดการทีมคุณภาพระบบงานเทคโนโลยีสารสนเทศ x

(นายสุรศักดิ์ สมประสงค์)  
\_\_\_\_\_/\_\_\_\_\_/\_\_\_\_/

พัฒนาระบบงานเทคโนโลยีสารสนเทศ 1  
ผู้จัดการทีมพัฒนาระบบงาน

(นายสุรชัย ปรีชาการ)  
\_\_\_\_\_/\_\_\_\_\_/\_\_\_\_/

ผู้ใช้ระบบงาน

(นาย วิทัก จิระประเสริฐกุล)  
\_\_\_\_\_/\_\_\_\_\_/\_\_\_\_/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.4 การอิมพลีเมนต์ระบบ

### 4.4.1 สถาปัตยกรรมของระบบ

สถาปัตยกรรมของระบบสนับสนุนการทดสอบซอฟต์แวร์นั้น มีแนวทางการการอิมพลีเมนต์ระบบ โดยในรูปแบบไคลเอนท์ เซิร์ฟเวอร์

การเชื่อมต่อเครือข่าย

การออกแบบสถาปัตยกรรมเครือข่ายของระบบสนับสนุนการทดสอบซอฟต์แวร์นั้น เป็นการทำงานโดยเชื่อมต่อระหว่างเครื่องของผู้ใช้ และเซิร์ฟเวอร์สำหรับเก็บข้อมูลในระบบแลน

รายละเอียดฮาร์ดแวร์ ซอฟต์แวร์

สำหรับความต้องการด้านองค์ประกอบแต่ละส่วนทางฮาร์ดแวร์ และซอฟต์แวร์นั้น มีรายละเอียดความต้องการตามตารางที่ 6.1

ตารางที่ 6.1 ความต้องการด้านฮาร์ดแวร์และซอฟต์แวร์

| เครื่องคอมพิวเตอร์แม่ข่าย (เครื่องเซิร์ฟเวอร์)   |   |
|--|---|
| หน่วยประมวลผล                                    | มีโปรเซสเซอร์ Intel Pentium 4 หรือสูงกว่า |
| หน่วยความจำ                                      | DDR RAM ขนาดไม่ต่ำกว่า 1.0 GB             |
| ฮาร์ดดิสก์                                       | หน่วยความจำไม่ต่ำกว่า 250 GB              |
| Network Interface                                | 10/100 Ethernet Interface                 |
| เครื่องคอมพิวเตอร์สำหรับผู้ใช้ (เครื่องไคลเอนท์) |   |
| หน่วยประมวลผล                                    | มีโปรเซสเซอร์ Intel Pentium 4 หรือสูงกว่า |
| หน่วยความจำ                                      | DDR RAM ขนาดไม่ต่ำกว่า 512 MB             |
| ฮาร์ดดิสก์                                       | หน่วยความจำไม่ต่ำกว่า 80 GB               |

### 4.4.2 เครื่องมือที่ใช้พัฒนาระบบ

สำหรับการพัฒนาระบบสนับสนุนการทดสอบซอฟต์แวร์นั้น ใช้โปรแกรมในการพัฒนา ดังรายละเอียดต่อไปนี้

**Visual Studio .NET 2005**

โดยใช้เทคโนโลยีดอตเน็ตเฟรมเวิร์ค (.Net Framework) และภาษา VB ในการเขียนโปรแกรม ระบบรับคำร้องของผู้ใช้ สำหรับฝ่ายสนับสนุนสารสนเทศเป็นเว็บแอปพลิเคชันทำหน้าที่เป็นอินเทอร์เฟซกับผู้ใช้ ทำให้ผู้ใช้สามารถใช้งานได้อย่างง่ายดาย และมีประสิทธิภาพ

**ประโยชน์จากดอตเน็ตเฟรมเวิร์ค**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถใช้ภาษาโปรแกรมใดก็ได้ และใช้กับแอปพลิเคชันที่เขียนขึ้นด้วยภาษาโปรแกรมใดๆ เพื่อรวมเข้าด้วยกันอย่างสมบูรณ์ นักพัฒนาจึงสามารถใช้ความชำนาญที่มีอยู่ทำงานต่อได้โดยไม่ต้องฝึกอบรมใหม่
- เขียนโปรแกรมน้อยลง คอตเน็ตเฟรมเวิร์คเน้นเน้นการออกแบบโดยใช้คอมโพเนนต์อย่างมาก โปรแกรมเมอร์จึงให้ความสนใจกับการเขียนบิสสิเนสลอจิกได้อย่างเต็มที่
- คอตเน็ตเฟรมเวิร์คได้รวมเทคโนโลยีเพื่อให้แอปพลิเคชันมีความเชื่อถือได้มากยิ่งขึ้น ตัวอย่างเช่น หน่วยความจำ, เซรด และ โปรเซส จะถูกจัดการโดยคอตเน็ตเฟรมเวิร์ค

### Microsoft SQL Server 2005 Server

SQL Server 2005 Server เป็นโปรแกรมระบบจัดการฐานข้อมูลที่ใช้จัดการระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) ที่มีประสิทธิภาพสูงและเป็นที่ยอมรับอย่างแพร่หลายตัวหนึ่งซึ่งพัฒนามาจาก SQL Server 2000 แต่มีการนำมาใช้งานร่วมกับการพัฒนาโปรแกรมด้วย Visual Studio .NET 2005 ทำให้ได้รับความนิยมและออกแบบมาให้สามารถรองรับการทำงานในระบบงานใหญ่ๆ ที่มีผู้ใช้ฐานข้อมูลได้หลายคน

## บทที่ 5

# บทสรุปและข้อเสนอแนะ

### 5.1 สรุปโครงการ

การพัฒนาระบบสนับสนุนการทดสอบซอฟต์แวร์ได้นำเสนอรูปแบบของการติดต่อระหว่างผู้ใช้และเจ้าหน้าที่ฝ่ายสนับสนุนสารสนเทศ โดยมีผู้ใช้อื่นที่เกี่ยวข้องได้แก่ ผู้บังคับบัญชาของผู้ใช้ รวมถึงผู้จัดการฝ่ายสารสนเทศ ซึ่งเป็นการทำงานผ่านเว็บแอปพลิเคชัน ดังนั้นผู้ใช้จึงสามารถใช้งานระบบสนับสนุนการทดสอบซอฟต์แวร์ได้ ไม่ว่าจะทำงานอยู่ส่วนใดขององค์กร หากมีการเชื่อมต่อผ่านระบบแลนขององค์กรแล้ว ก็จะสามารถใช้งานได้

การใช้งานของผู้ใช้แต่ละคนใช้ รหัสผู้ใช้งาน โดยเป็นรหัสพนักงานของผู้ใช้ จึงมีการกำหนดสิทธิ์ว่าผู้ทดสอบแต่ละคนสามารถจัดการกับการทดสอบซอฟต์แวร์ที่ตนเองรับผิดชอบเท่านั้น ส่วนผู้ทดสอบระดับหัวหน้างาน สามารถติดตามการทดสอบของผู้ทดสอบทุกคนได้

แนวความคิดพัฒนาระบบสารสนเทศของโครงการนี้ ได้นำเสนอการศึกษาและวิเคราะห์การทำงานของระบบเพื่อให้เกิดประโยชน์สูงสุดในการติดต่อระหว่างผู้ใช้ และการทำงานของเจ้าหน้าที่ฝ่ายควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศ โดยระบบได้สนับสนุนให้ผู้ทดสอบสามารถทำการทดสอบซอฟต์แวร์ได้อย่างเต็มประสิทธิภาพ เกิดคุณภาพและความน่าเชื่อถือในการทดสอบซอฟต์แวร์ และทำให้ผู้ใช้งานได้ซอฟต์แวร์ที่ตรงกับความต้องการ และเป็นตัวขับเคลื่อนให้ธุรกิจดำเนินกิจกรรมทางธุรกิจอย่างเต็มประสิทธิภาพ

### 5.2 ข้อเสนอแนะ

โครงการพัฒนาระบบสนับสนุนการทดสอบซอฟต์แวร์ สำหรับฝ่ายควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศนั้น มีจุดมุ่งหมายเพื่อให้สามารถนำไปใช้ในการปฏิบัติงานได้จริง อย่างไรก็ตามเมื่อมีการนำระบบนี้ไปใช้งานแล้ว หากในอนาคตหน่วยงานมีการขยายตัว หรือมีการเปลี่ยนแปลงขั้นตอนการทำงาน อาจต้องมีการปรับเปลี่ยนการทำงานของโปรแกรมเพิ่มเติมให้เข้ากับลักษณะการทำงาน เพื่อให้เกิดประโยชน์สูงสุดกับองค์กร

## บรรณานุกรม

- กิตติ ภัคศิวัฒน์กุล. 2548. **คัมภีร์ การพัฒนาระบบเชิงวัตถุด้วย UML และ Java**. กรุงเทพฯ :  
เคทีพี คอมพ์ แอนด์ คอนซัลท์.
- เกษม กมลรัชยพิสิฐ. 2550. **รอบรู้ประยุกต์ใช้ SQL Server 2005**. กรุงเทพฯ: สำนักพิมพ์แห่ง  
จุฬาลงกรณ์มหาวิทยาลัย.
- รวิชัย สุริยะทองธรรม. 2548. **พัฒนาเว็บแอปพลิเคชัน ASP.NET ฉบับสมบูรณ์**. กรุงเทพฯ:  
บริษัท ชัคเซส มีเดีย จำกัด.
- พรฤดี เนติโสภากุล. 2549. **“วิศวกรรมซอฟต์แวร์”**. สำนักพิมพ์ท็อป.
- โอภาส เขียมศิริวงศ์. 2546. **การวิเคราะห์และออกแบบระบบ**. กรุงเทพฯ: บริษัท ซีเอ็ดยูเคชั่น  
จำกัด.
- John Satzinger, Robert Jackson and Stephen Burd. 2007. **System Analysis and Design in a  
Changing World, Fourth Edition**. Thomson Course Technology.
- Peter Rob and Carlos Coronel. 2007. **Database Systems Design, Implement, and  
Management, Seventh Edition**. Thomson Course Technology.
- T. S. Pressman. 1997. **“Software engineering”**. McGraw- Hill International.

## ประวัติผู้เขียน

|                    |   |
|--------------------|---|
| ชื่อผู้เขียน       | นางสาวยุวดี จันทร์ตัน   |
| สถานที่เกิด        | จังหวัดสงขลา  |
| การศึกษา           | ระดับปริญญาตรี<br>บธ.บ. (บริหารธุรกิจบัณฑิต)<br>สาขาวิชาคอมพิวเตอร์ธุรกิจ<br>มหาวิทยาลัยกรุงเทพ |
| ประสบการณ์การทำงาน | เจ้าหน้าที่ควบคุมคุณภาพระบบงานเทคโนโลยีสารสนเทศ<br>ธนาคารไทยพาณิชย์ จำกัด (มหาชน)               |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้